

PostGIS, 3.x

Para analistas SIG, geografía y ciencias terrestres...



PostgreSQL



7/2/2025

PRITS

Iván Santiago

isantiago@prits.pr.gov



Tabla de contenido

1. Introducción:	6
Sobre PostgreSQL	6
PostGIS	7
Objetivos de este libro	8
Teoría de bases de datos relacionales	9
1A. Bases de datos relacionales	9
1B: Lenguaje Structured Query Language (SQL):	10
1C: SQL Espacial	13
Integración con QGIS	14
1D: Tipos de datos ¿Cómo se codifican y guardan los datos?	15
Consultas básicas en SQL	16
SQL básico	16
Queries básicos en DBeaver	19
Select Distinct	22
Realizar cálculos	23
Uso de Alias para identificar campos/columnas	23
Los alias pueden definirse sin usar la palabra AS.	24
Cláusula WHERE	24
Operadores lógicos en la cláusula WHERE	25
Operadores AND, OR	26
Operadores IN, BETWEEN y LIKE	27
ORDER BY, LIMIT	30
Offset	31
Orden de precedencia al evaluar condicionalmente	31
GROUP BY: queries con datos agregados	32
Queries simples con geodatos	33
Usar Alias para identificar tablas temporalmente	34
Orden de operaciones SQL	36
Metadatos de las tablas con geometrías	36
Hacer un listado de columnas de una tabla	37
Modificar datos: DROP, CREATE, INSERT, ALTER, UPDATE	38



DROP.....	38
CREATE.....	38
INSERT	39
ALTER.....	39
UPDATE.....	40
Prácticas con SQL Básico.....	40
SQL espacial para geometrías vectoriales.....	47
Operaciones espaciales: Adyacencia, continencia, intersección, distancia.....	47
El modelo <i>Simple Features</i> del Open Geospatial Consortium	48
Dimensión de las geometrías.....	48
Interior, contorno y exterior de las geometrías	49
Predicados para las relaciones topológicas	50
Desglose de relaciones topológicas por geometría	51
Usar QGIS para ejemplos de relaciones topológicas	52
ST_Disjoint.....	54
ST_Touches.....	56
ST_Crosses.....	59
ST_Within/ST_Contains.....	60
ST_Overlaps	62
ST_Intersects.....	64
ST_Covers & ST_Coveredby.....	66
ST_DWithin.....	67
Queries con geometrías:.....	68
ST_Touches.....	68
Visualizar las geometrías en PostGIS.....	72
Mostrar área, perímetro y tipo de geometría	73
ST_Length.....	73
Round()	74
ST_Distance.....	76
ST_XMin, ST_YMin.....	77
ST_Transform	79
Group by (ejemplo trivía)	80
Spatial Joins.....	80



Left Join.....	82
Subqueries:.....	83
ST_Disjoint y Subqueries.....	83
Tablas de distancias: Distancias entre plazas de recreo.....	86
Cross Join Lateral con ST_Distance y el operador de distancias <->	87
Prácticas:.....	90
Consideraciones importantes antes del análisis geográfico y geoprocesos.....	97
Índices espaciales.....	97
Predicados/Funciones espaciales que usan índice geoespacial por defecto.....	98
Creación de tablas a partir de un query	98
Homogenizar geometrías devueltas al sobreponer geodatos.....	101
Utilización de función extra: STX_Extract	106
Tolerancia en el análisis espacial.....	108
Análisis geoespacial.....	114
Intersección.....	114
Erase	116
Qué es dependencia funcional.....	118
Calcular áreas potencialmente agrícolas sustituidas por uso urbano	118
Unión de layers	121
Identity.....	128
Update.....	138
Clip.....	139
Buffer.....	141
Selección por distancia.....	144
Append.....	144
Dissolve.....	144
Interpolación areal.....	146
Prácticas.....	150
Geodatos ráster en PostGIS.....	157
Descarga de datos para el ejercicio:.....	161
Apéndice A-1: Inventario de relaciones topológicas	162
Apéndice A-2: Relaciones topológicas clasificadas por predicado	183
Patrón de 9 caracteres para matriz DE-9IM.....	183



ST_Equals.....	184
ST_Disjoint.....	185
ST_Touches.....	187
ST_Crosses.....	193
ST_Within.....	199
ST_Contains	203
ST_Overlaps.....	205
ST_Intersects.....	208
Operador &&.....	208
ST_Covers	208
ST_CoveredBy.....	212
Referencias.....	216



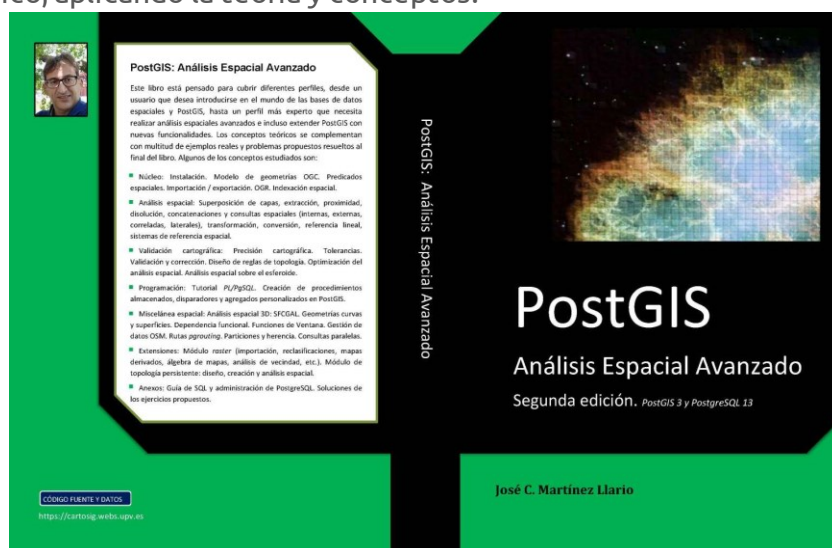
1. Introducción:

Audiencia: Este tutorial está dirigido a personas con experiencia en el uso de programas de procesamiento de datos geográficos (conocidos por Sistemas de Información Geográfica) en su modalidad de interfaces gráficas.

Este tutorial le dará la oportunidad de aprender el uso del lenguaje SQL espacial mediante PostGIS, dentro del gestor de bases de datos (RDBMS) PostgreSQL. Éste, al igual que otros RDBMS, maneja tipos de datos numéricos, de texto y otros tipos de datos especiales para resguardar datos geográficos.

Durante el transcurso de este tutorial, podrá notar las capacidades para resumir y analizar información geográfica de una manera diferente a cualquier otro SIG con interfaz gráfica.

Los ejemplos de geoprocесamiento y teoría cartográfica (conceptos de escala, tolerancias en el geoprocесamiento, y otros) están basados en el libro [PostGIS Análisis Espacial Avanzado 2ª edición, 2020](#) del profesor **José C. Martínez Llario**, de la **Universidad Politécnica de Valencia**, España. Usamos los ejemplos de código de este libro y los adaptamos usando datos de Puerto Rico, aplicando la teoría y conceptos.



Si desean profundizar sobre PostGIS y PostgreSQL, el libro del Prof. Martínez le dará información detallada sobre otros temas de importancia.

Sobre PostgreSQL

PostgreSQL comenzó a desarrollarse a mediados de los años 80 en la Universidad de California en Berkeley. El origen del nombre proviene de su predecesor, el gestor de bases de datos relacionales “Ingres”, el cual fue desarrollándose desde 1977 al 1985. Se decidió ponerle el nombre post-Ingres, quedando finalmente como Postgres en 1986.

Postgres experimentó desarrollos durante 8 años. Luego fue comercializado con el nombre de **Illustra**. Éste fue adquirido por Informix, que a su vez fue comprado por IBM en 2001.



A partir de un trabajo de investigación académica de 1995, Postgres salió de la academia y empezó una nueva vida en la comunidad open source. Luego de ocho años más, Postgres fue transformado radicalmente por la comunidad open source, integrando consistencia y uniformidad al código de base, entre otras cosas.

Postgres fue ganando reputación por su estabilidad. Fue durante este periodo que su nombre cambió a PostgreSQL (*póstgre-skiué*). Este manejador de bases de datos relacionales fue publicado desde la versión 6.0. Hasta hoy, este programa ha continuado su desarrollo por parte de la comunidad open source.

PostGIS

PostGIS surgió a partir de la necesidad de manejar datos geográficos que estaban en archivos separados pero que debían ser modificados frecuentemente. Esta necesidad propició la idea de manejar estos datos geográficos *dentro* de una base de datos de manera *versionada*. **Las bases de datos están diseñadas para ambientes compartidos, donde hay dos o más usuarios accediendo a, y modificando los mismos datos, pero en versiones diferentes.**

La elección de usar el manejador de bases de datos (RDBMS) PostgreSQL fue un golpe de suerte en ese momento (2001), según indican sus desarrolladores iniciales (refractions.net). Este RDBMS tiene la capacidad de crear '*custom data types*', lo cual dejó el camino libre para desarrollar un modelo de almacenamiento de datos geométricos para fines de almacenamiento, manejo y análisis. El documento '[Simple Features for SQL](#)' del [OpenGIS Consortium](#) dio las bases para la creación del primer intento de crear un tipo de dato geométrico para GIS.

Durante el transcurso del desarrollo tuvieron que descartar el primer modelo por su lentitud y decidieron crear su propio '*geometry*' data type, el cual resultó ser mucho más rápido y eficiente que el primer modelo. Finalmente, se aprovechó la existencia de un tipo de [índice R-Tree](#) dentro de PostgreSQL, el cual serviría para acelerar las búsquedas en términos de milisegundos en tablas espaciales con millones de récords.

Para fines de mayo de 2001, se publicó la primera versión de PostGIS 0.1. Desde esa fecha, este producto se ha mantenido dentro de la comunidad open source y al presente va por la versión 3.x, integrando tanto los modelos vectoriales y matriciales de almacenamiento de datos geográficos. Además, desde la versión 2.0 se retomó el asunto de guardar datos vectoriales usando un modelo de topología persistente, diferente al modelo Simple Features del OGC. Además, en la versión 2 se integró el modelo ráster dentro del conjunto de funciones de PostGIS. Esto permite usar funciones analíticas vectoriales y ráster dentro de PostGIS. La versión 3.0 fue publicada a fines de 2019 y se ha estado concentrando en mejoras al desempeño de distintas funciones. Las versiones 3.x integran nuevas funciones para ráster y adoptan las bibliotecas de programación [GDAL](#).



Fuentes:

<http://www.refractive.net/products/postgis/history/> recuperado 01 jul, 2025

<https://www.postgresql.org/about/> recuperado 01 jul, 2025

<https://carto.com/blog/postgis-3-release> recuperado 01 jul, 2025

<https://www.crunchydata.com/blog/postgis-3.2-new-and-improved> recuperado 01 jul, 2025

Objetivos de este libro

Este tutorial se enfocará en dar algunos ejemplos básicos de consultas **Structured Query Language (SQL) para datos geográficos**. Dedicaremos una parte a mostrarles funciones básicas de SQL no espacial y luego pasaremos a mostrarles cómo aplicar funciones de **SQL espacial**.

- No se tratarán aspectos de administración de bases de datos, resguardo, creación de cuentas, creación de '*schemas*', estadísticas, y otras tareas administrativas. Estas tareas extenderían demasiado este taller.
- El objetivo es dar nociones básicas del lenguaje SQL. Usaremos la interfaz gráfica de administración de bases de datos [DBeaver](#) para redactar y ejecutar las sentencias SQL (queries).
- Trabajaremos funciones SQL para el manejo y análisis de datos geográficos vectoriales en PostGIS.
- En caso de ser necesario, integraremos el uso del programa QGIS para visualización de estos datos.



Teoría de bases de datos relacionales

1A. BASES DE DATOS RELACIONALES

Pregunta funcional: ¿Cuándo una tabla o serie de tablas debe convertirse en una base de datos?

Podemos usar hojas de cálculo e incluso programas de bancos de datos como MS Access y SQLite para manejar datos de cierta complejidad y volumen. Sin embargo, estos programas mencionados no son aptos para ambientes compartidos (en una empresa, agencia, en la Internet), donde dos o más usuarios están accediendo y modificando datos simultáneamente.

Una de las contestaciones a esta pregunta tiene que ver con el propósito de uso de los datos, el volumen, complejidad del manejo de los datos. Otro aspecto clave es determinar si varias personas o usuarios deben acceder a los datos simultáneamente (ambientes compartidos). Si los datos a usarse tienen estas características mencionadas, debe considerar el uso de un programa de manejo de bases de datos (RDBMS en inglés).

Los datos en tablas de hojas de cálculo y en una base de datos, se manejan de forma diferente. En una hoja de cálculo puede haber una o varias tablas, pero estas no necesariamente están relacionadas entre filas o columnas.

En programas de manejo de bases de datos *relacionales*, los datos están separados por temas en distintas tablas. En términos de **teoría de conjuntos y lógica de primer orden, una tabla es una relación** (Viescas, 2018). Cada tabla contiene datos sobre un tema. Esto se hace para evitar redundancia o repetición además de darle organización a los datos. Sin embargo, ya que las tablas están separadas, será necesario un mecanismo para poder relacionarlas.. Esto se hace mediante el mecanismo de identificadores comunes, presentes en las tablas. Estos son los *primary keys* y *foreign keys*.

Las personas que han usado programas SIG ya están acostumbrados a enlazar al menos dos tablas usando la función **Join** en ArcGIS o en QGIS, entre otros programas. Esta función **Join** generalmente enlaza una fila (récord) con otra fila en otra tabla. En otros tipos de **Join espacial**, puede relacionarse un identificador con múltiples elementos. En ese caso se producen múltiples instancias del mismo objeto, una para cada fila que contiene información diferente. En otros casos estos programas proveen el uso de relaciones (*Relate*), donde la correspondencia (cardinalidad) entre identificadores ya no es 1 a 1 sino 1 a múltiples.



1B: LENGUAJE STRUCTURED QUERY LANGUAGE (SQL):

Qué es SQL

SQL (Structured Query Language) es un lenguaje de **dominio específico**, esto es, un lenguaje de programación **dedicado a la creación-destrucción, manipulación-actualización y búsqueda de datos** en bases de datos y objetos relacionales.

Orígenes:

El lenguaje SQL está basado en el álgebra de conjuntos. Esta rama de las matemáticas fue desarrollada por [Georg Cantor](#) a lo largo de la última mitad del siglo XIX. En 1968, [el matemático holandés D. L. Childs](#) publicó en la Universidad de Michigan la tesis "[Feasibility of a set-theoretical data structure](#)". En esta tesis, Childs afirma que se puede explicar todo tipo de preguntas solo con **tres funciones básicas: selección, relación y reagrupamiento**. Esta invención dio lugar al lenguaje SQL.

Para la misma época y por distintos caminos, [Edgar Frank Codd](#) inventaba un sistema de almacenamiento de datos más efectivo que el de tipo secuencial indexado y el de tipo jerárquico. El nuevo tipo de base de datos utilizaría las relaciones entre conjuntos para contestar preguntas sobre los datos.

Elementos principales del lenguaje SQL:

El lenguaje SQL está dividido tres sub-lenguajes.

- **Lenguaje de manipulación de datos (Data Manipulation Language o DML)**
Comandos para **devolver, modificar o insertar datos** en las tablas, una vez estos objetos sean creados.
- **Lenguaje para definición de datos (Data Definition Language o DDL)** contiene instrucciones y comandos para **generar o borrar bases de datos, crear/borrar tablas y objetos**.
- **Lenguaje de control de datos (Data Control Language o DCL)**. Estos comandos se usan para el **manejo de cuentas, control de acceso mediante permisos y control de las transacciones**. Ejemplos de estos comandos son Create Role, Grant/Revoke, Commit, Rollback.

Ejemplos de Data Manipulation Language:

SELECT. Este es el más conocido y se usa para extraer datos de una o múltiples tablas.

SELECT * from localidades;

Esta instrucción, sentencia o consulta **devuelve todos los registros (filas) y campos (columnas) de la tabla localidades**.

Podemos reducir la cantidad de datos a devolver estableciendo condiciones, trayendo solo los que nos interesan. Por ejemplo,

SELECT nombre FROM localidades -- trae filas del campo nombre de la tabla localidades



WHERE nombre LIKE 'San%'; -- WHERE se usa para establecer condiciones

Esta consulta nos devuelve solamente los nombres de localidades que comiencen con la secuencia de caracteres "San", seguido de cualquier otra secuencia de caracteres. Esto puede incluir San Juan, Santa Isabel, etc....

Otro ejemplo puede ser:

SELECT * FROM localidades -- trae todas las filas y campos de la tabla *localidades*

WHERE población > 1000; -- donde la población sea menor de 1,000

En esta consulta, se devolverán los datos de la tabla *localidades* con todas las columnas donde la población supere los mil habitantes.

INSERT

Este comando se usa para añadir registros a las tablas ya existentes. Por ejemplo:

INSERT INTO localidades -- inserta filas en la tabla existente *localidades*

(barrio, población) -- en los campos *barrio* y *población*

VALUES ('Santolaya', 1500); -- los valores 'Santolaya' y 1,000

Al insertar datos dejando columnas sin datos, estos adoptarán el valor "NULL". Esto quiere decir que esta celda no tiene valor, no tiene dato. Esto tendrá consecuencias más adelante al tratar de hacer operaciones matemáticas, por ejemplo. No tendrán valor "NULL" aquellos campos con tipo de dato numérico *serial*, que se llenan automáticamente al insertar una fila.

UPDATE

Este comando se usa para **modificar** el **contenido** de las **tablas**.

UPDATE localidades -- actualiza la tabla existente *localidades*

SET nombre = 'Santa Olaya' -- escribe 'Santa Olaya' en el campo *nombre*

WHERE nombre = 'Santolaya'; -- en aquellas filas donde el *nombre* sea 'Santolaya'

En este ejemplo, estamos cambiando el nombre 'Santolaya' por el nombre 'Santa Olaya'.

Además, se puede **recalcular** sobre registros existentes

UPDATE localidades -- actualiza la tabla existente *localidades*

SET población = población + 300 -- sumar 300 a la población existente

WHERE nombre = 'Santa Olaya'; -- donde el nombre sea 'Santa Olaya'

Aquí se está sumando 300 a la población existente solo para la localidad 'Santa Olaya'.

DELETE

Se usa para **borrar registros** de una tabla.

DELETE FROM menú -- borra filas de la tabla existente *localidades*

WHERE nombre_plato = 'Arroz con guinea' -- donde el nombre del plato sea 'Arroz con guinea'

Aquí en esta sentencia, borraremos el registro o registros de la tabla *menú* que contengan la cadena de caracteres 'Arroz con guinea' en la columna *nombre_plato*. Note que **las cadenas de caracteres van encerradas en comillas simples**.



Ejemplos de Data Definition Language:

```
CREATE DATABASE base_de_datos; -- crear una base de datos
```

```
CREATE TABLE localidades; -- crear tabla localidades
```

```
(gid serial primary key, nombre varchar, poblacion integer); -- con los campos:
```

```
/* (gid: tipo de dato serial, con identificador primario, nombre: tipo de dato varchar o de texto y  
de longitud variable, poblacion: tipo de dato entero) las columnas van encerradas entre  
paréntesis ().*/
```

```
-- Los símbolos /* y */ son para hacer comentarios multilínea en PostgreSQL.
```

Crear la tabla llamada 'localidades' con una serie de columnas o atributos con diferentes tipos de dato numéricos (integer) o de texto (varchar).

```
ALTER TABLE localidades ADD COLUMN geom geometry (point, 6566);
```

Añadir la columna 'geom', en la tabla existente *localidades*, la cual guardará las coordenadas de los lugares registrados. Estas se registrarán usando **geometría puntual**, en el **sistema de referencia espacial** con **código 6566** (State Plane Coordinate System, Puerto Rico, NAD83(2011)).

```
USE poblacion_2010; -- Este se usa para indicar cuál es la base de datos a consultar o  
hacer modificaciones.
```

```
DROP TABLE barrios_exclusivos; -- Este comando sirve para borrar tablas y objetos de  
la base de datos.
```

```
DROP DATABASE base_de_datos;
```

También puede usarse para **borrar la base de datos**.

Ejemplos de Data Control Language (DCL)

Crear una cuenta o usuario nuevo en una base de datos dentro de un grupo o rol ya establecido:

```
CREATE USER estudiante1 WITH PASSWORD 'estudiante1';
```

Dar acceso a conectarse a una base de datos:

```
GRANT CONNECT ON DATABASE gisdb TO estudiante1;
```

Dar privilegio de entrada/uso dentro de un 'schema'

```
GRANT USAGE IN SCHEMA pr_geodata TO estudiante1;
```

Dar privilegio de lectura a un usuario, y a una tabla:

```
GRANT SELECT ON g01_biota_bosques_auxiliares;
```

Revocar todos los privilegios en todas las tablas dentro del schema pr_geodata al rol llamado estudiantes.

```
REVOKE ALL PRIVELEGES ON ALL TABLES IN SCHEMA pr_geodata FROM estudiantes;
```




Más adelante estaremos explicando otras funciones importantes de SQL tales como las funciones de agregado de datos, unión de tablas, y algunas de las funciones geoespaciales.

Referencias:

Historia

Breve historia (orígenes-fundamentos matemáticos) del lenguaje SQL:

http://www.decideo.fr/Petite-histoire-des-bases-relationnelles-et-du-langage-SQL_a6377.html recuperado en 01 jul, 2025.

Breve historia: Hernández, M., Viescas J.L, SQL Queries for mere mortals, 2nd Ed, Sep. 2007, pp. 53-65.

Genealogía de los RDBMS: <http://fadace.developpez.com/sqbdcmp/story/> recuperado el 01 jul, 2025.

1C: SQL ESPACIAL

Orígenes:

PostGIS fue publicado por la empresa consultora informática canadiense [Refractions Research](#) en el año 2001. En ese momento solo contaba con varias funciones de manejo de datos geométricos (geometry data type). La publicación de un [estándar abierto de este tipo de datos geométricos](#) OGC SFS, (1999), hizo que el trabajo de mejoras y actualizaciones de PostGIS pudiera continuar hasta estos días.

Referencias:

Martínez Llario, José C., PostGIS 2 Análisis Espacial Avanzado, 2012-13. <http://cartosig.upv.es>. Universidad Politécnica de Valencia, España.

<http://www.refractions.net/>, recuperado en 01 jul, 2025



PostGIS, 3.x

INTEGRACIÓN CON QGIS

QuantumGIS (QGIS) será el programado que utilizaremos para ver algunos de los resultados de las consultas SQL que vamos a realizar en PostGIS. De hecho, QGIS surgió por la necesidad de tener un visualizador de los datos geospaciales guardados en PostGIS. Si no tiene el programa QGIS instalado, descárguelo gratis desde la dirección <http://qgis.org/en/site/forusers/download.html>. Descargue e instale, según su sistema operativo: Windows 32bit o 64bit, Mac, o las distintas variedades de Linux.

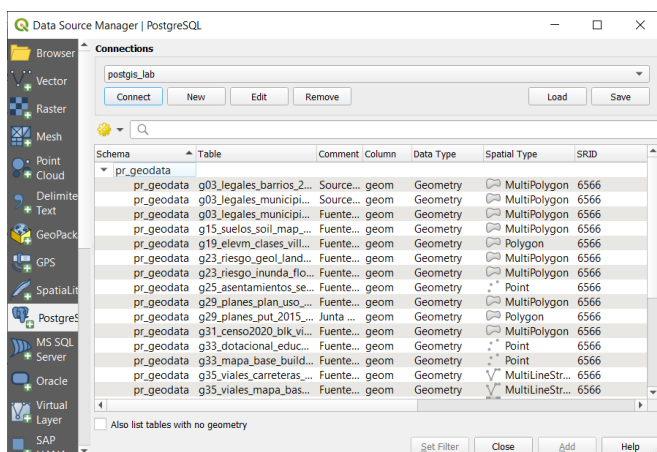
Offline (Standalone) installers:

Long Term Version for Windows (3.40 LTR)

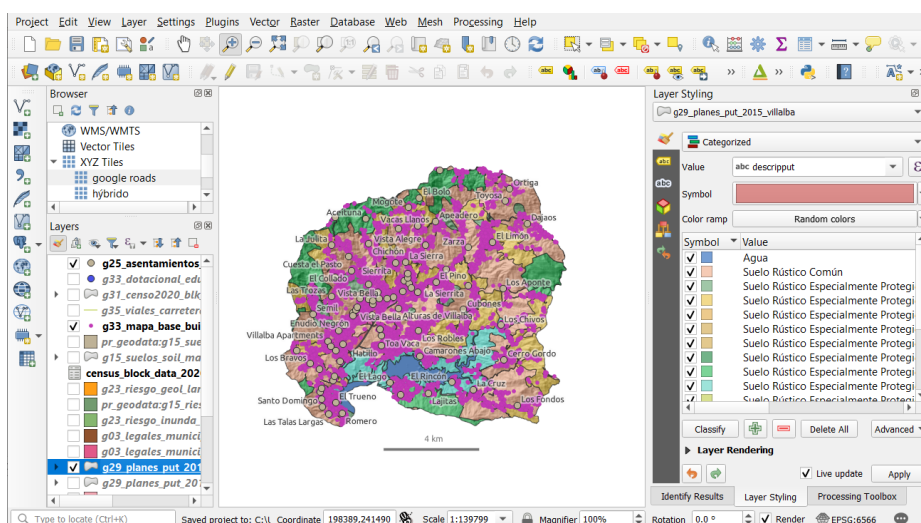
Latest Version for Windows (3.44)

Latest Version for Windows (3.44) with Qt6 (experimental)

Long Term Version for Windows (3.40) with Qt6 (experimental)



Ventana para añadir tablas geospaciales (layers) al visor de geodatos de QGIS



Visualización en QGIS: Algunos de los layers guardados en PostgreSQL/PostGIS.



1D: TIPOS DE DATOS ¿CÓMO SE CODIFICAN Y GUARDAN LOS DATOS?

La tabla de atributos se compone principalmente de tres [tipos de datos](#), dos de ellos son los más comunes:

- **Texto** (*character, string*): letras, palabras, frases, oraciones, códigos **alfanuméricos**, identificadores.
No se usan para operaciones matemáticas. Generalmente se manipulan con funciones de texto como concatenaciones, extracción, etc. Puede usarse ordenamiento (*sorting*).
- **Cifras, números** enteros, decimales, binarios, fechas.
En estos es común el ordenamiento y operaciones matemáticas.
- **Objetos**, (datos en formato que solo puede interpretar la computadora mediante instrucciones)
Ciertas bases de datos pueden guardar las coordenadas de un punto, línea, área, celda(s) en un campo de una tabla. Usualmente se usa el tipo de dato numérico "**binario**" para guardarlos.

Sea prudente al momento de escoger un tipo de representación numérica.

- Evite usar números con decimales cuando sepa de antemano que todos los números del conjunto de datos son enteros.
- Use el menor espacio posible para los atributos de texto. Si va a guardar un código que no pasa de tres espacios, no use el espacio por defecto de algunos programas (50 espacios)

Al hacer esto, se economiza espacio en disco y el rendimiento del programa se mantendrá óptimo.



Consultas básicas en SQL

SQL BÁSICO

En esta parte vamos a practicar sentencias SQL simples. Un enunciado SQL básico de selección de datos está compuesto de:

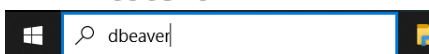
SELECT *columna1, columna2, columna3,... columna_n, o un asterisco * para todas las columnas y récords.*

FROM *para indicar la tabla o tablas que vamos a usar*

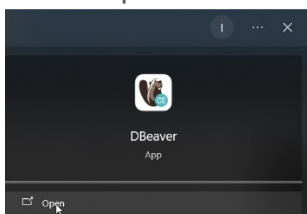
; *punto y coma para cerrar la sentencia SQL*

Ejemplo para practicar:

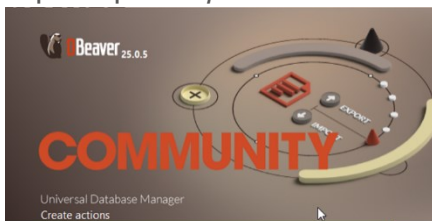
- ☐ Abra una sesión del programa **DBeaver**, vaya a la caja de texto **Search** de Windows y escriba **dbeaver**



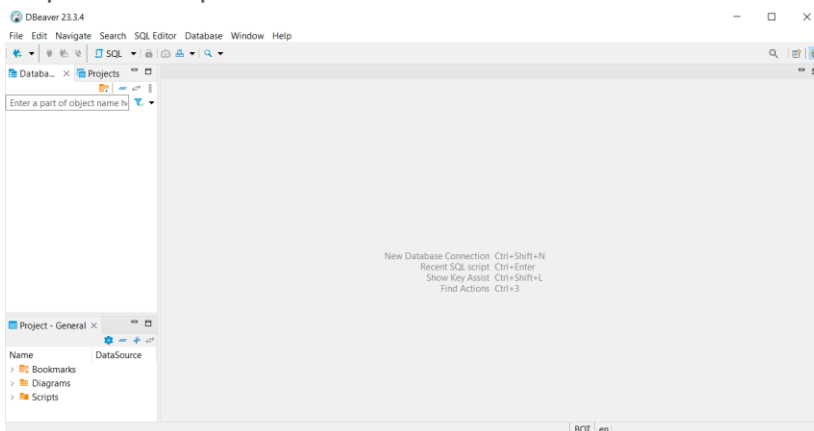
- ☐ Abra la aplicación haciendo **click** en la opción **Open** de **DBeaver**.



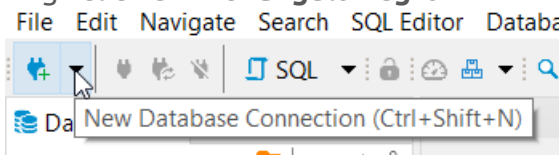
- ☐ Espere que el *splash screen* termine de hacer la carga y aparezca la aplicación.



Así debe aparecer la aplicación DBeaver



- ☐ Haga **click** en el **triángulo negro** del botón **New Database Connection (Ctrl+Shift+N)**





PostGIS, 3.x

- ❑ Escoja (click en) la opción **PostgreSQL**
- ❑ En la forma **Connect to a database**, siga la secuencia con los siguientes parámetros:

1: Escriba la dirección donde se encuentra la base de datos. Si está tomando el adiestramiento de forma presencial, se le dirá cuál es la dirección. Si está siguiendo el libro desde su domicilio o trabajo, la dirección debe ser **localhost**.

2: La base de datos es **gisdb_lab**.

3y4: Escriba el nombre del usuario de la base de datos. Si está tomando el adiestramiento de forma presencial, se le indicará el nombre de usuario y contraseña. Si usted está siguiendo el libro desde su domicilio o trabajo, la dirección (**host**) debe ser **localhost**

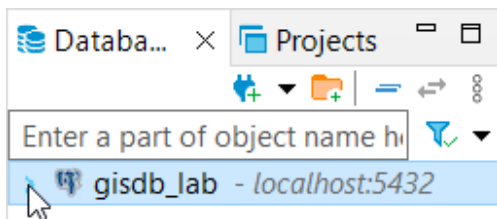
5: Haga **click** en el botón **Test Connection** para asegurarse que se puede conectar a la base de datos. Si todo está bien, aparecerá la forma **Connection test** y solo necesitará hacer **click** en el botón OK. De lo contrario, revise los parámetros antes mencionados.

6: Haga **click** en el botón **Finish** en la forma **Connections Settings** para cerrarla.

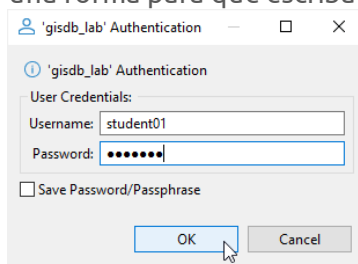
- ❑ Para los que no estén de forma presencial en el adiestramiento, es su decisión guardar o no el password en la interfaz de conexión de DBeaver.
- ❑ Si usted instaló en su computadora, deberá estar seguro de cuál es el password del usuario postgres al momento que instaló el PostgreSQL y PostGIS.
- ❑ En la lista arborescente del panel **Database Navigator**, haga **click** en el **nodo** al lado izquierdo del nombre de la base de datos **gisdb_lab**.



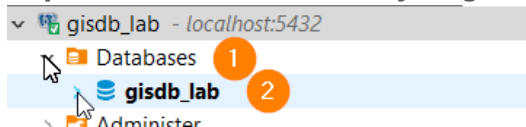
PostGIS, 3.x



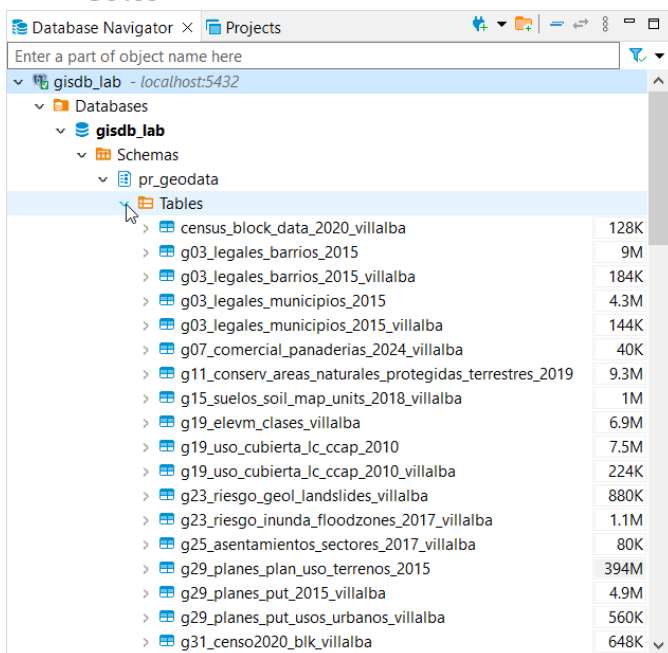
Si no guardó el password de usuario, aparecerá una forma para que escriba el password.



- **Expanda** el nodo **Databases** y luego **expanda** el nodo **gisdb_lab**



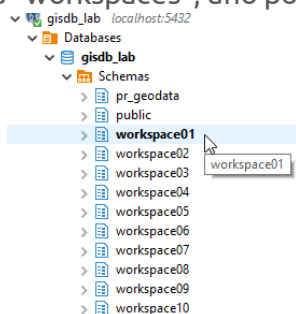
- Continuando, **expanda** el nodo **Schemas**, luego el nodo **pr_geodata** y finalmente el nodo **Tables**.



Al lado derecho de cada **tabla o "relación"** como se les conocen también a las tablas, verá el espacio en disco que ocupa cada tabla.

Las tablas que vamos a usar están bajo el nodo **Schemas > pr_geodata > Tables**

Las tablas que vamos a generar en la sección de Análisis geográfico van a guardarse en distintos "workspaces", uno por cada participante

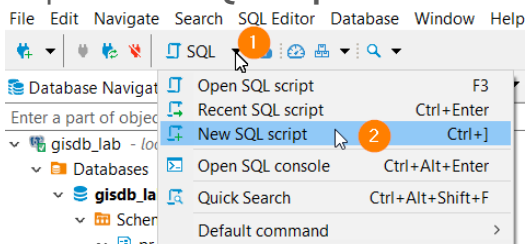


- Cada participante tendrá un espacio **workspaceXX** asignado para evitar conflictos con las tablas de otros participantes.
- Por ejemplo, **student01** tiene asignado **workspace01**, y así sucesivamente.



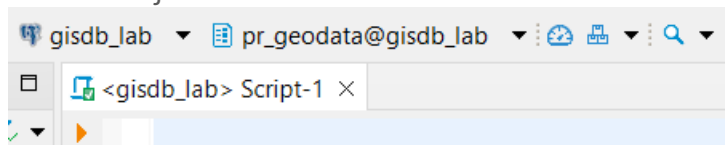
QUERIES BÁSICOS EN DBEAVER

- Una vez esté disponible la base de datos para estas prácticas, en la interfaz gráfica de **DBeaver** deberá hacer **click en el triángulo negro** del botón **SQL** y luego **click** en la opción **New SQL script**



De igual efecto será ir al **menú principal**, hacer **click en SQL Editor** y escoger la opción **New SQL Script**.

- Fíjese que en la parte superior de la ventana **<gisdb_lab> Script-1** aparece el nombre de la base de datos activa (**gisdb_lab**), así como el nombre de su *Schema* activo (**workspaceXX**). **Eso es importante**. De lo contrario, estará ubicado en otro *Schema* y le dará mensajes de error eventualmente.



Vamos a realizar la primera **búsqueda simple** (query) para comenzar a interactuar con una de las tablas de esta base de datos. Primero, vamos a mostrar el diccionario de datos de la tabla **census_block_data_2020_villalba**.

Esta es la **estructura y descripción** de las columnas de esta tabla (no es geodato).

Columna	Tipo de dato	Descripción
gid	integer	Identificador (ID) interno
logrecno	integer	número lógico de récord (ID censal)
sumlev	character varying	Summary level. Nivel geográfico (bloque, grupo de bloque, tract, barrio, subbarrio, municipio, PR, USA)
geoids	character varying	ID compuesto para bloque censal 2020
cousub	character varying	ID para barrios
barrio	character varying	Nombre del barrio
tract	character varying	ID Tract, sector censal
blkgrp	character varying	ID Block group, grupo de bloques censales
block	character varying	ID Bloque censal
name	character varying	Nombre del bloque censal
vtd	character varying	Voting District: Unidad electoral, de la Comisión Estatal de Elecciones. Los límites no necesariamente concuerdan con el mapa 2020 de la CEE.
pop100	integer	Conteo 100% de población, 2020
hu100	integer	Conteo 100% de viviendas, 2020
h0010002	integer	Viviendas ocupadas, 2020
h0010003	integer	Viviendas vacantes, 2020

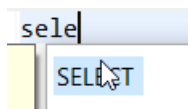


PostGIS, 3.x

DBeaver facilita la escritura de queries, ya que al escribir se presentan opciones para completar la cláusula, función, tabla, etc.

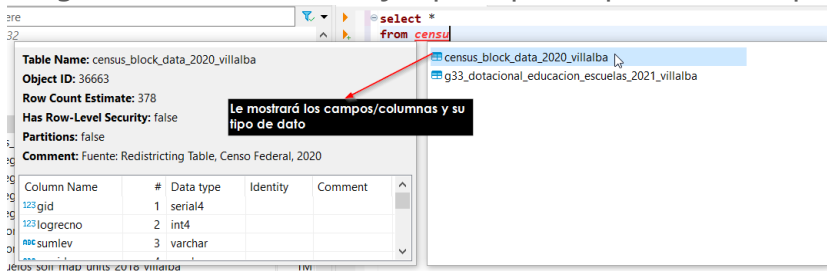
- En la pestaña <gisdb_lab Script-1, escriba la siguiente sentencia o query:

sele

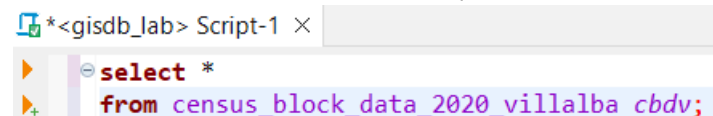


use la tecla tab para que le ayude a completar la palabra **SELECT**

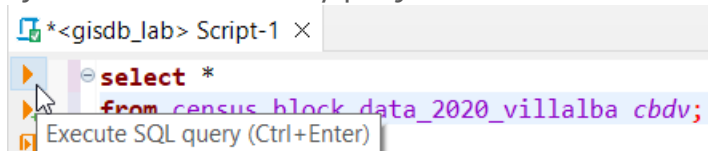
- Escriba * y luego presione la tecla **enter**
- Luego escriba **FROM censu** y espere que le aparezcan las opciones de tablas



- Use la tecla tab para **escoger** la tabla **census_block_data_2020_villalba**. Fíjese que el DBeaver le otorgó un **alias** a esta tabla "**cbdvd**" para facilitar otras operaciones. Termine la sentencia escribiendo ; al final.



- Haga **click** en el botón **Execute SQL Query** o presione las teclas ctrl+enter para ejecutar esta sentencia/query



Podrá ver los resultados de esta forma en la interfaz gráfica de DBeaver:



PostGIS, 3.x

The screenshot shows the DBeaver 23.3.4 interface. The 'Database Navigator' on the left shows the 'gisdb_lab' database with a schema 'pr_geodata'. The 'SQL Editor' on the right contains the query: `select * from census_block_data_2020_villalba cbdv;`. The 'Results' pane on the right displays a table with columns: gid, logrecno, sumlev, and geoids. The table contains 200 rows of data. The status bar at the bottom indicates '200 rows fetched - 0.008s (0.004s fetch), on 2024-02-12 at 12:36:35'.

- ☐ Bajo la pestaña **census_block_data_2020_villalba 1**, use las barras de navegación hacia los lados y de arriba y abajo para explorar este resultado

Este query que acabamos de escribir y ejecutar quiere decir que pedimos a **DBeaver** que nos dé **todas** las columnas y **todos** records/filas de la tabla **census_block_data_2020_villalba**. Esta tabla contiene datos básicos de población y vivienda para el Censo de 2020, además de una serie de campos de niveles geográficos.

- ☐ Haga **click** en el botón **Calculate total row count** inmediatamente debajo de los resultados, y verá que **la tabla contiene 378 records** (Total rows).

The screenshot shows the bottom of the DBeaver interface. The 'Results' pane is visible, and the 'Calculate total row count' button is highlighted. The status bar at the bottom indicates '200 rows fetched - 0.008s (0.004s fetch), on 2024-02-12 at 12:36:35'.

Cuidado al usar SELECT *

Un detalle importante es que **el uso de SELECT *, no es recomendado**. Esta **instrucción trae todos los datos a través de la red, y si el contenido de una o más tablas es grande (miles o millones de records), esto va a afectar el desempeño de la búsqueda**. Imagínese, traer esa cantidad de datos, multiplicado por el número de usuarios de una base de datos.

Ya que se mostró el contenido de la tabla **census_block_data_2020_villalba**, haremos una serie de búsquedas usando esta tabla.



SELECT DISTINCT

Como pudimos ver en los resultados de la búsqueda pasada, hay columnas con valores repetidos. Por ejemplo, los ID y nombres de barrios están repetidos porque la tabla tiene **datos a nivel de bloque censal** y los barrios están constituidos por múltiples bloques censales. Por otro lado, **los ID (columna: geoids) de bloques no deben estar repetidos**. Podemos usar la opción “**DISTINCT**” para que nos devuelva una lista única de barrios y sus identificadores que están en la columna “*cousub*”. Use la tabla de diccionario de datos de la tabla **census_block_data_2020_villalba** en la página 19 como referencia para contestar las siguientes preguntas.

¿Cuál es el nombre de la columna de nombres de barrios? _____

¿Cuál es el nombre de la columna con los códigos ID de barrios? _____

- ☐ Vaya a la caja de texto del **Query Script** y borre el contenido del query escrito anteriormente.
- ☐ Ya que vamos a traer más de una columna, debemos **separar cada columna usando una coma**. Escriba lo siguiente:

```
select distinct cousub, --trae datos sin repetir de cousub
      barrio --y de nombres de barrios
from --de la tabla census block data 2020 villalba
pr_geodata.census_block_data_2020_villalba;
```

-- Los comentarios en las líneas de código son para su beneficio. No los tiene que copiar.

- ☐ Presione **ctrl+enter** o haga **click** en el botón **Execute**.

The screenshot shows the QGIS Query Builder window. The SQL query is entered in the top text area: `select distinct cousub, barrio from pr_geodata.census_block_data_2020_villalba;`. Below the query, the table structure is displayed with columns `cousub` and `barrio`. The bottom section shows the first 8 rows of the query results.

	cousub	barrio
1	33898	Hato Puerco Arriba
2	86917	Villalba Abajo
3	33855	Hato Puerco Abajo
4	86874	Villalba CUT
5	13430	Caonillas Abajo
6	86960	Villalba Arriba
7	13516	Caonillas Arriba
8	84595	Vacas

At the bottom of the window, it indicates: 8 row(s) fetched - 0.001s, on 2025-07-07 at 14:32:41

Aquí podemos ver que hay **ocho barrios** en el **Municipio de Villalba**. Los nombres de barrios responden a **referencias históricas** del uso de territorios, asentamientos, flora, fauna, incluso hatos de animales de crianza.

Los barrios **Hato Puerco** y **Vacas** fueron lugares de crianza de **ganado porcino** y de **ganado vacuno** en siglos pasados. En el caso del nombre **Caonillas**, este nombre tiene origen taíno y está relacionado a la búsqueda de **oro**, ya que ese nombre está relacionado a lugares donde se buscaba oro en cuerpos de agua. **Villalba CUT** es el **centro urbano tradicional**. La **fundación** de este **municipio** es relativamente reciente, a principios del siglo XX, el **12 de abril de 1917**.



REALIZAR CÁLCULOS

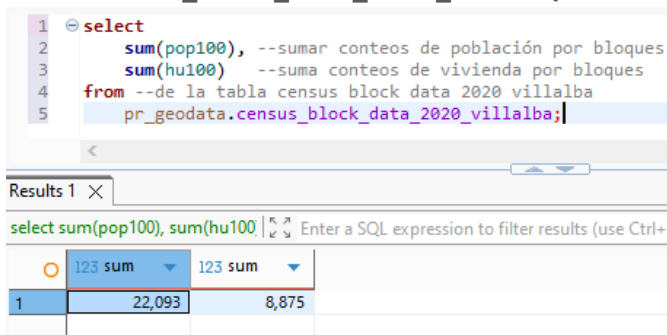
El lenguaje SQL tiene múltiples funciones estándar para operaciones matemáticas básicas como suma, resta, multiplicación, división, exponenciación, y funciones estadísticas básicas, como promedio y otras. Para más detalles sobre las funciones y requisitos para su uso, vaya al siguiente enlace: <https://www.postgresql.org/docs/current/functions-math.html>. Más adelante discutiremos algunas funciones matemáticas agregadas.

Vamos a realizar una **sumatoria de población y vivienda** para todos los bloques censales del Municipio de Villalba.

- ☐ En la **caja de texto de Query Script**, borre el query anterior y **escriba** lo siguiente:

```
select
  sum(pop100), --sumar conteos de población por bloques
  sum(hu100)   --suma conteos de vivienda por bloques
from --de la tabla census block data 2020 villalba
  pr_geodata.census_block_data_2020_villalba;

SELECT SUM(pop100), SUM(hu100)
FROM census_block_data_2020_villalba;
```

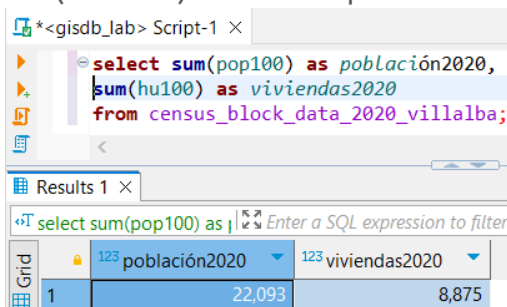


Observe que **solo se devuelve una fila**. Además notará que los nombres de las columnas tienen el mismo nombre. Esto no es recomendado, así que en el próximo ejercicio vamos a darles un nombre a las columnas calculadas.

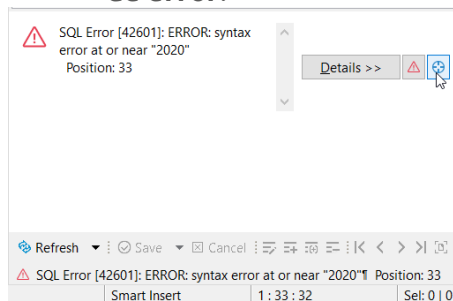
USO DE ALIAS PARA IDENTIFICAR CAMPOS/COLUMNAS

Como se mencionó anteriormente, las columnas del query anterior tiene el mismo nombre. Otras veces, queremos presentar nombres más descriptivos, según sea el caso.

- ☐ Modifique el query anterior. **Añada** las palabras **AS población2020** luego de SUM(POP100). **Añada** las palabras **AS viviendas2020** luego de SUM(HU100)



⚠ No debe dejar **e s p a c i o s** entre **población 2020** ni entre **vivienda 2020** porque **DBeaver devolverá un mensaje de error**.



Esto es como definir otro alias sin referencia a columna alguna.



Si quiere dejar un espacio o quiere usar mayúscula inicial, deberá encerrar entre comillas el alias. “Población 2020”. Lo mismo aplicará a “Vivienda 2020”, como aparece aquí a la derecha. Puede modificar el query anterior para que practique.

Esta práctica no es recomendable para dar nombre a las columnas de una tabla permanente. De lo contrario, tendrá que utilizar comillas “Columna etc.” cada vez que vaya a referirse a una columna con espacios y mayúsculas.

```
*gisdb_lab> Script-1 ×
select sum(pop100) as "población 2020",
sum(hu100) as "viviendas 2020"
from census_block_data_2020_villalba;
```

Results 1 ×

```
select sum(pop100) as "pobla" Enter a SQL expression to filter results
```

	123 población 2020	123 viviendas 2020
1	22,093	8,875

LOS ALIAS PUEDEN DEFINIRSE SIN USAR LA PALABRA AS.

- Practique con el mismo query, esta vez borre las palabras AS:

```
SELECT SUM(POP100) población2020,
SUM(HU100) vivienda2020
FROM census_block_data_2020_villalba;
```

```
*gisdb_lab> Script-1 ×
select sum(pop100) población2020,
sum(hu100) vivienda2020
from census_block_data_2020_villalba;
```

Results 1 ×

```
select sum(pop100) Enter a SQL expression to filter results
```

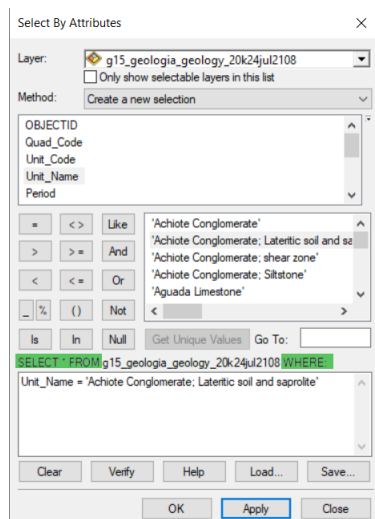
	123 población2020	123 vivienda2020
1	22,093	8,875

Note también que **la columna del query final NO requiere escribir la coma.** Tampoco afecta el query si luego de cada coma hay un “enter” o CR/LF (carriage return/line feed).

CLÁUSULA WHERE

Mediante la cláusula **WHERE** es que comenzaremos a hacer **test cierto** o **falso** para escoger **subconjuntos** de datos en una o más tablas.

WHERE es la cláusula tradicionalmente “visible” en los programas SIG con interfaz gráfica.



Este es un ejemplo de la función **Select by Attribute** de ArcMap (Esri). Más abajo en esa forma, resalto en color verde el query y la sección **WHERE**, la cual es la que se ejecuta con esta función. Para unir tablas, será hacer un *join* aparte en otra función.



Hasta ahora, hemos visto ***SELECT*** *columna1, columna2, col3... FROM tabla*. A esto vamos a añadir la cláusula **WHERE** para establecer condiciones para extraer subconjuntos de datos.

- Vaya a la pestaña **SQL Editor <gisdb_lab> Script-1** y modifique el query de la siguiente manera:

```
*<gisdb_lab> Script-1 ×
select sum(pop100) población2020,
sum(hu100) vivienda2020
from census_block_data_2020_villalba
where barrio = 'Villalba CUT';
```

Results 1 ×

```
select sum(pop100) | Enter a SQL expression to filter res
```

	123 población2020	123 vivienda2020
1	734	301

Este query **ejecuta** una **suma** de **todos los bloques censales que componen el barrio pueblo o la zona del centro urbano tradicional/histórico** identificados como 'Villalba CUT'. El resultado es **734** personas y **301** viviendas.

- **La última columna: vivienda2020 no debe tener coma final.** De lo contrario devolverá un error.

OPERADORES LÓGICOS EN LA CLÁUSULA WHERE

A continuación presentamos una lista de operadores para usar dentro de la cláusula WHERE.

Operador	Descripción
=	Igualdad
>	Mayor que
<	Menor que
>=	Mayor o igual
<=	Menor o igual
<> o !=	Desigualdad
AND	Operador lógico AND (excluyente)
OR	Operador lógico OR (incluyente)
<u>IN</u>	Devuelve TRUE si el valor pareo con cualquier valor en una lista.
<u>BETWEEN</u>	Devuelve TRUE si el valor está entre un conjunto o entre una gama de valores
<u>LIKE</u>	Devuelve TRUE si un valor pareo con un patrón, usualmente de caracteres
<u>IS NULL</u>	Devuelve TRUE si el valor es NULL (ausencia de datos)
NOT	Negar el resultado de otras operaciones

Practiquemos un par de ejemplos con WHERE.



Pregunta: ¿Cuáles son los bloques censales que no tienen población? Añada los barrios para dar una idea somera de dónde están.

- ☐ Vaya a la pestaña **SQL Editor <gisdb_lab> Script-1** y escriba lo siguiente:

```
--select name as bloque, barrio, pop100 -- trae columnas name alias bloque, barrio y pop100
--from census_block_data_2020_villalba -- de esta tabla
--where pop100 = 0; -- condición: filas con valor de población (pop100)=0
```

	bloque	barrio	pop100
1	Block 1007	Vacas	0
2	Block 1013	Vacas	0
3	Block 3006	Caonillas Abajo	0
4	Block 3007	Caonillas Abajo	0
5	Block 3010	Caonillas Arriba	0
6	Block 3012	Caonillas Arriba	0
7	Block 3013	Caonillas Arriba	0
8	Block 3014	Caonillas Arriba	0
9	Block 3016	Caonillas Abajo	0

Este query devuelve **46 filas** que corresponden a bloques censales **que no tenían población para el Censo 2020**, además de los **barrios** donde estos bloques están ubicados.

OPERADORES AND, OR

Este operador puede confundirse con nuestra manera de entender “y” si pensamos literalmente en su traducción. No es adición. **AND** significa buscar **cuáles elementos cumplen simultáneamente dos o más condiciones**. Por ejemplo, hay un solo día en un año que es 12 y a la vez está en el mes de diciembre. Se puede añadir el año y *restringimos* aún más nuestra búsqueda.

Pregunta: ¿Cuáles son los bloques censales que tienen población >= 0 y que están en el barrio Caonillas Arriba?

- ☐ Vaya a la pestaña **SQL Editor <gisdb_lab> Script-1** y escriba lo siguiente:

```
1 select
2   name as bloque, barrio, pop100
3   --trae columnas name alias bloque,
4   --barrio y población2020
5
6   from
7     pr_geodata.census_block_data_2020_villalba
8     --de la tabla de bloques censales 2020
9     --del Municipio de Villalba
10
11  where
12    pop100 >= 0 --donde la población >= 0
13  and  barrio = 'Caonillas Arriba';
14    --y que también el barrio sea Caonillas Arriba
```

	bloque	barrio	pop100
1	Block 2000	Caonillas Arriba	61
2	Block 2001	Caonillas Arriba	26
3	Block 2002	Caonillas Arriba	43
4	Block 2004	Caonillas Arriba	141

Este query nos **devuelve los bloques censales que tienen población cero o más y que a la vez pertenecen al barrio Caonillas Arriba**. En total son 30 filas.

Estos resultados pueden ser mostrados en orden ascendente o descendente usando la cláusula **ORDER BY**, ya sea usando la modalidad ascendente **ASC** o descendente **DESC**. Esto lo veremos más adelante.



Pregunta: ¿Cuál sería el resultado del query anterior si cambiamos el operador AND por el operador OR?

Averigüemos:

- En la pestaña **SQL Editor <gisdb_lab> Script-1**, MODIFIQUE el query anterior

```
1 select
2   name as bloque, barrio, pop100
3   -- trae columnas name alias bloque,
4   -- barrio y población2020
5 from
6   pr_geodata.census_block_data_2020_villalba
7   -- de la tabla de bloques censales 2020
8   -- del Municipio de Villalba
9 where
10  pop100 >= 0 -- donde la población >= 0
11 OR barrio = 'Caonillas Arriba';
12 --o el barrio sea Caonillas Arriba
```

	A-Z bloque	A-Z barrio	123 pop100
1	Block 1000	Hato Puerco Arrib	34
2	Block 1001	Hato Puerco Arrib	101
3	Block 1002	Hato Puerco Arrib	12
4	Block 1003	Hato Puerco Arrib	97

200 378 200 row(s) fetched - 0.005s (0.003s fetch)

BOT en Write Calculate total row count Insert

¿Por qué este query nos devuelve todas las filas de la tabla (378)? Esto pasa porque las condiciones:

Pop100 >= 0 va a dar TRUE en todas las filas.
Barrio = 'Caonillas Arriba' dará TRUE en algunos casos.

Ya que **todos los récords tienen algún valor desde cero**, el query **va a devolver todas las filas**. La condición **OR barrio='Caonillas Arriba'** **no surte efecto** alguno porque la condición de población >= 0 hace TRUE a todo el conjunto de datos de la tabla.

Podemos ponerle el nombre de cualquier barrio, incluso otro nombre que no está y devolverá todas las filas.

OPERADORES IN, BETWEEN Y LIKE

El operador **IN** se usa para buscar filas que tengan valores que pareen dentro de una lista definida en el query.

Hagamos un ejemplo:

- En la pestaña **SQL Editor <gisdb_lab> Script-1**, MODIFIQUE el query anterior

```
1 select name as bloque, barrio, pop100
2 from census_block_data_2020_villalba
3 where barrio in
4 ('Caonillas Abajo', 'Caonillas Arriba', 'Villalba CUT');
```

	bloque	barrio	pop100
1	Block 2000	Caonillas Arriba	61
2	Block 2001	Caonillas Arriba	26
3	Block 2002	Caonillas Arriba	43
4	Block 2004	Caonillas Arriba	141
5	Block 2005	Caonillas Arriba	16
6	Block 2006	Caonillas Arriba	25
7	Block 2007	Caonillas Arriba	102
8	Block 2008	Caonillas Arriba	25
9	Block 2009	Caonillas Arriba	121
10	Block 2010	Caonillas Arriba	82
11	Block 2011	Caonillas Arriba	65
12	Block 3000	Caonillas Arriba	47

Export data 200 78 Smart Insert 4...9

El operador **IN** está separado por un CRLF (enter) pero esto no es necesario. Lo escribí de esta manera para que se pueda ver en este espacio.

El resultado de este query nos devuelve **78 filas que corresponden a todos los bloques que cumplen la condiciones de pertenecer a los barrios Caonillas Arriba o Caonillas Abajo, o a Villalba CUT**. Es un atajo para evitar escribir `barrio = 'Caonillas Arriba' OR barrio = 'Caonillas Abajo' OR barrio = 'Villalba CUT'`.



PostGIS, 3.x

El operador **BETWEEN** se usa para devolver filas que cumplan estar dentro de una gama de valores **numéricos, fechas o de texto**.

Tarea: Provea una lista de bloques censales con población entre 100 a 200 habitantes para el censo 2020 en el Municipio de Villalba.

- En la pestaña **SQL Editor <gisdb_lab> Script-1**, modifique y ejecute el query anterior

```
select name as bloque, barrio, pop100
from census_block_data_2020_villalba
where pop100 between 100 and 220;
```

	bloque	barrio	pop100
1	Block 1001	Hato Puerto Arriba	101
2	Block 1004	Vacas	183
3	Block 1010	Hato Puerto Arriba	109
4	Block 2004	Caonillas Arriba	141
5	Block 2007	Caonillas Arriba	102
6	Block 2009	Caonillas Arriba	121
7	Block 3003	Caonillas Arriba	112
8	Block 3023	Hato Puerto Abajo	133
9	Block 4000	Caonillas Abajo	130
10	Block 4001	Caonillas Abajo	170
11	Block 4008	Caonillas Abajo	182
12	Block 4010	Hato Puerto Abajo	217
13	Block 4020	Caonillas Abajo	122

Note que **no usamos OR** dentro del operador BETWEEN.

En este query el resultado devuelto son los bloques censales que tienen población dentro de las cantidades 100 a 220 habitantes. Se devolverán 53 filas.

Incluye el valor 100 pero ningún bloque tenía valor 100 para este Censo decenal 2020.

El operador **BETWEEN** puede usarse con datos alfanuméricos, por ejemplo:

Tarea: Hacer una lista de barrios cuyos nombres comiencen con la letra C hasta la H.

- En la pestaña **SQL Editor <gisdb_lab> Script-1**, modifique y ejecute el query anterior

<pre>select name as bloque --nombre del bloque censal barrio, --nombre del barrio pop100 --población, conteo de 100% from --tabla de datos por bloque censal, 2020 pr_geodata.census_block_data_2020_villalba where --condición: que los nombres del barrios --comiencen con la letra C hasta la letra H barrio between 'C' and 'Hz';</pre>	Este query nos devuelve las filas que tengan nombres de barrios que comiencen con la letra C hasta los barrios cuyos nombres comiencen con H.																								
<table border="1"><thead><tr><th></th><th>bloque</th><th>barrio</th><th>pop100</th></tr></thead><tbody><tr><td>1</td><td>Block 1000</td><td>Hato Puerto Arriba</td><td>34</td></tr><tr><td>2</td><td>Block 1001</td><td>Hato Puerto Arriba</td><td>101</td></tr><tr><td>3</td><td>Block 1002</td><td>Hato Puerto Arriba</td><td>12</td></tr><tr><td>4</td><td>Block 1003</td><td>Hato Puerto Arriba</td><td>97</td></tr><tr><td>5</td><td>Block 1008</td><td>Hato Puerto Arriba</td><td>88</td></tr></tbody></table>		bloque	barrio	pop100	1	Block 1000	Hato Puerto Arriba	34	2	Block 1001	Hato Puerto Arriba	101	3	Block 1002	Hato Puerto Arriba	12	4	Block 1003	Hato Puerto Arriba	97	5	Block 1008	Hato Puerto Arriba	88	¿Por qué H? Esto hará que se incluyan todos los barrios con nombres comiencen con la letra H, seguido del resto de letras, hasta la combinación Hz (que no está presente, pero es para asegurar incluir las demás letras después de la H).
	bloque	barrio	pop100																						
1	Block 1000	Hato Puerto Arriba	34																						
2	Block 1001	Hato Puerto Arriba	101																						
3	Block 1002	Hato Puerto Arriba	12																						
4	Block 1003	Hato Puerto Arriba	97																						
5	Block 1008	Hato Puerto Arriba	88																						



El operador **LIKE** se usa para buscar un patrón específico de caracteres.

Por ejemplo, hacer una lista de barrios cuyo nombre comience con V. Use la cláusula **DISTINCT** para traer solamente los nombres de los barrios.

- En la pestaña **SQL Editor <gisdb_lab> Script-1, modifique y ejecute** el query anterior

```
select distinct barrio --nombre del barrio sin repetir
from --tabla de datos por bloque censal, 2020
pr_geodata.census_block_data_2020_villalba
where --condición: que los nombres del barrios
--comiencen con la letra V
barrio like 'V%';
```

	A-Z barrio
1	Villalba CUT
2	Villalba Abajo
3	Villalba Arriba
4	Vacas

Refresh Save Cancel

200 4 4 row(s) fetched - 0.001s, on

Este query nos devuelve una **lista de barrios cuyo nombre comienza con la letra V**.

Podemos modificar el patrón y escribir LIKE 'Villa%' para seleccionar solamente los barrios con nombres que comiencen con el patrón 'Villa%' en su nombre.

Es necesario incluir el signo de porcentaje % para incluir el resto de los caracteres después de 'Villa...' de lo contrario, el query no traerá filas porque ningún barrio tiene como nombre V.

Note el uso de la cláusula **DISTINCT** para traer solo una instancia del nombre del barrio. Esta tabla es de bloques censales, pero contiene barrios también.

ILIKE: El operador **ILIKE** hace lo mismo que LIKE, pero a diferencia de éste, **ILIKE no es case sensitive**.



ORDER BY, LIMIT

La cláusula **ORDER BY** tiene como propósito **ordenar las filas de forma ascendente o descendente en una o más columnas**. Veamos un ejemplo.

- En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query:

```
select name as bloque, barrio, pop100 -- trae los nombres de bloque, barrio y población
from census_block_data_2020_villalba -- de la tabla census_block_data_2020_villalba
where pop100 between 100 and 200 -- donde la población sea entre 100 y 200
order by pop100 asc, barrio asc; -- ordena primero por pop100 asc, luego por barrio asc
```

	asc bloque	asc barrio	123 pop100
1	Block 3004	Villalba Abajo	100
2	Block 1001	Hato Puerco Arriba	101
3	Block 1001	Hato Puerco Arriba	101
4	Block 2007	Caonillas Arriba	102
5	Block 1010	Villalba Arriba	102
6	Block 1001	Villalba Arriba	103
7	Block 3020	Villalba Abajo	104
8	Block 2002	Vacas	105
9	Block 2004	Hato Puerco Arriba	106

Se puede especificar el uso de **ASC** para que esté claro que **el orden es ascendente**. Sin embargo, la opción por defecto es **ASC** y no es necesario escribirla. Por el contrario, si deseamos que el orden sea descendente, debemos escribir **DESC**. Además, podemos usar más de una columna para ordenar. En este caso las dos columnas que estamos usando se ordenarán de manera ascendente.

Por su parte, la cláusula **LIMIT** se usa para limitar el número de filas que el query va a devolver. Podemos usarla para ordenar y presentar los primeros o últimos casos de una serie de datos. Veamos un ejemplo:

Tarea: Haga una **lista de bloques censales con mayor población; incluya el nombre del bloque, barrio y población**. Ordene la lista usando la columna **pop100** de manera **descendente**. Limite las filas solo a **5**.

- En la pestaña **SQL Editor <gisdb_lab> Script-1** modifique y ejecute este query de la siguiente manera:

```
1 select name as bloque, barrio, pop100
2 from pr_geodata.census_block_data_2020_villalba
3 order by pop100 desc, barrio asc
4 limit 5;
```

	asc bloque	asc barrio	123 pop100
1	Block 1005	Hato Puerco Arriba	833
2	Block 3005	Villalba Abajo	535
3	Block 1004	Villalba Arriba	471
4	Block 2005	Villalba Arriba	422
5	Block 4019	Hato Puerco Abajo	361

Este query nos trae solo los primeros cinco bloques censales con mayor población en el Municipio de Villalba para el Censo de 2020.



OFFSET

Offset sirve para obviar filas y traer otras. Usualmente se usa en combinación con LIMIT. Sirve para **mostrar la cantidad de filas siguientes, especificadas en OFFSET** en una **lista ordenada** ascendente o descendente. Por ejemplo:

Pregunta: **¿Cuáles son los siguientes bloques censales con mayor población, luego de los primeros 10?**

- ☐ En la pestaña **SQL Editor <gisdb_lab> Script-1** modifique y ejecute este query de la siguiente manera:

```
1 select name as bloque, barrio, pop100 --trae los nombres de bloques, barrio, y población
2 from pr_geodata.census_block_data_2020_villalba -- de la tabla ensus_block_data_2020_villalba
3 order by pop100 desc, barrio asc --ordena primero por población desc, luego por barrio asc
4 limit 10 --devuelve solo 10 filas
5 offset 10; -- muestra las siguientes 10 filas después de las primeras 10
```

Este query nos devuelve los siguientes 10 bloques censales con mayor población después de los primeros 10. Es como tener la lista de los primeros 20 pero solo mostrar los últimos 10.

ORDEN DE PRECEDENCIA AL EVALUAR CONDICIONALMENTE

Por defecto, los programas RDBMS evalúan las condiciones (expuestas en la cláusula WHERE) de **izquierda a derecha**. Cuando una condición de búsqueda contiene varios tipos de condiciones simples el RDBMS los evalúa en orden a base del operador utilizado en cada condición (Viescas, 2018).

Orden de evaluación	Tipo de operador
1ro	() evaluar lo que esté dentro del paréntesis
2do	Signo positivo (+), signo negativo (-)
3ro	Multiplicación (*), división (/)
4to	Suma (+), resta (-)
5to	=, <>, <, >, <=, >=, BETWEEN, IN, LIKE, IS NULL
6to	NOT
7mo	AND
8vo	OR

Fuente: Viescas, 2018. SQL For Mere Mortals.



GROUP BY: QUERIES CON DATOS AGREGADOS

Ya que hemos visto distintas maneras de filtrar filas y columnas en una tabla, ahora pasaremos a discutir cómo hacemos queries para **hacer cálculos agregados por categorías (conteos, sumas, promedios, y otros)**.

Por ejemplo, tenemos esta tabla de bloques censales, la cual tiene varios niveles geográficos (*summary levels*). Podemos **agregar o agrupar** datos en niveles geográficos más altos. Podemos **sumar la población de todos los bloques que componen cada barrio o cada unidad electoral** o una selección de barrios o unidades electorales. Para esto se usa **GROUP BY**.

Pregunta/Tarea: Use la tabla **census_block_data_2020_villalba** para calcular la **población por barrio**. La lista debe tener el **nombre del barrio** y la **población**. **Ordene** la lista **por población** de manera **descendente (mayor a menor)**.

- ☐ En la pestaña **SQL Editor <gisdb_lab> Script-1** modifique y ejecute este query de la siguiente manera:

```
1 select barrio, sum(pop100) población --trae los nombres de barrios, y suma de población
2 from pr_geodata.census_block_data_2020_villalba -- de la tabla ensus_block_data_2020_villalba
3 group by barrio --agrupa las sumas de población por barrio
4 order by población desc; --ordena por población desc
```

Este query nos devuelve una lista con los barrios y su población para el Censo 2020. La cláusula **GROUP BY** sirve para **agregar** (en este caso, **sumar**) la **población de bloques que componen cada barrio**. Al lado de **GROUP BY** se escribe cuál es (son) la(s) **columna(s)** que tiene(n) la **dimensión** o **nivel** en el cual **queremos agregar los datos**. En este caso, agregar población solo por barrio. Note que **SUM(pop100)** **no se añade al GROUP BY**. **SUM(pop100)** **no es un nivel de agregación** sino un grupo de **valores que vamos a computar**. No necesariamente tiene que ser de tipo de dato numérico. Puede ser conteos.

barrio	población
Hato Puerco Arriba	6,304
Villalba Arriba	5,062
Vacas	3,473
Villalba Abajo	2,524
Hato Puerco Abajo	1,459
Caonillas Arriba	1,288
Caonillas Abajo	1,249
Villalba CUT	734

8 row(s) fetched - 0.003s, on 2025-05-30 at 16:27:28

En términos de SIG Desktop, GROUP BY es análogo a usar Dissolve basado en la columna barrio, con la operación de sumatoria de población.



PostGIS, 3.x

Pregunta: ¿Cuántos bloques censales hay por cada barrio?

No usaremos sumas, sino **conteos de bloques** censales para cada barrio. La función que usaremos será **COUNT(nombre_campo)** o **COUNT(*)**. Esto nos devolverá el número de filas con una o más características que hemos de escoger. No hay bloques censales entre barrios.

- ☐ En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query:

```
1 select barrio, --trae los nombres de barrios,
2 count(geoids)as cantidad_bloques --conteo de bloques alias "cantidad_bloques"
3 from pr_geodata.census_block_data_2020_villalba -- de la tabla census_block_data_2020_villalba
4 group by barrio --agrupa las sumas de población por barrio
5 order by 2 desc; --ordena por la columna #2 "cantidad_bloques" desc
```

En este query estamos usando la función **COUNT()** usando la columna **geoids** para que **nos devuelva el número de bloques censales que componen cada barrio en el Municipio de Villalba**. El campo **geoids** es el campo que tiene los **identificadores de cada bloque censal**. Además necesitamos la cláusula **GROUP BY** para agrupar conteos por barrio y **ORDER BY** para ordenar por cantidad de bloques de manera descendente.

barrio	cantidad_bloques
Hato Puerto Arriba	87
Villalba Arriba	85
Vacas	49
Villalba Abajo	49
Hato Puerto Abajo	30
Caonillas Arriba	30
Caonillas Abajo	27
Villalba CUT	21

QUERIES SIMPLES CON GEODATOS

Las preguntas a continuación se contestan en diferentes queries que utilizarán la **función ST_Area(geometry)** en PostGIS.

Pregunta 1: ¿Cuáles son los 10 municipios de mayor área?

- ☐ En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query:

```
*<gisdb_lab> Script-1 ×
1 select municipio, -- trae los nombres de municipios
2 st_area(geom)/1000000 as sqkm -- trae el área municipal/1,000,000 conv en kms cuadrados
3 from g03_legales_municipios_2015 -- de la tabla g03_legales_municipios_2015
4 order by sqkm desc -- ordena por sqkm descendente
5 limit 10; -- reduce la cantidad de filas devueltas a 10
```

ST_Area(geom). Esta función devuelve el área superficial de una geometría en las unidades del sistema de coordenadas del geodato (**metros**, en este caso).

El resultado de **ST_Area(geom)** está dividido por el factor de conversión **1000000** (un millón), para convertir **metros cuadrados** a **kilómetros cuadrados**.

El geodato **g03_legales_municipios_2015** tiene una **columna de geometría (geom)**. Estas columnas son las que usaremos para hacer las tareas de geoprocetamiento más adelante.

Estamos usando **ORDER BY** para ordenar por área y **LIMIT 10** para reducir la lista.

municipio	sqkm
Arecibo	328.5324788907
Ponce	302.7296063593
Utado	297.7907433289
Coamo	202.0604296195
Mayagüez	201.6116075629
Cabo Rojo	186.7987971071
San Sebastián	184.5929428018
Salinas	181.3948914822

Reto: Con las cláusulas ya aprendidas hasta ahora:

¿Cómo podemos modificar este query y mostrar los 10 municipios de menor superficie? Hint: use *order by ...*



Pregunta 2: ¿Cuáles son las áreas ocupadas por cada zona de riesgo por deslizamiento en Villalba?

- ☐ En la pestaña <gisdb_lab> Script 1, escriba y ejecute el siguiente query:

```
select description as riesgo_deslizamiento, --zonas de riesgo
       sum(st_area(geom))/1000000 as sq_kms --área en kms cuadrados
from
  pr_geodata.g15_riesgo_geol_landslides_1979_villalba
group by
  description --agrupar datos por esta columna
order by
  sq_kms desc;
```

	A-Z riesgo_deslizamiento	123 sq_kms
1	AREA OF MODERATE SUSCEPTIBILITY TO LANDSLIDING	86.7016165547
2	AREA OF HIGHEST SUSCEPTIBILITY TO LANDSLIDING	5.7400487812
3	AREA OF LOW SUSCEPTIBILITY TO LANDSLIDING	3.4344950469

Refresh Save Cancel

200 3 3 row(s) fetched - 0.003s (0.001s fetch), on 2025-07-

Reto:

El área total del municipio es de **95.876** kms cuadrados.

¿Cómo definirías otra columna que calcule el porcentaje de ocupación para cada zona de riesgo?

	A-Z riesgo_deslizamiento	123 sq_kms	123 pct
1	AREA OF MODERATE SUSCEPTIBILITY TO LANDSLIDING	86.7016165547	90.4309906073
2	AREA OF HIGHEST SUSCEPTIBILITY TO LANDSLIDING	5.7400487812	5.986950625
3	AREA OF LOW SUSCEPTIBILITY TO LANDSLIDING	3.4344950469	3.5822260492

Refresh Save Cancel

200 3 3 row(s) fetched - 0.001s, on 2025-07-01 at 15:07:00

USAR ALIAS PARA IDENTIFICAR TABLAS TEMPORALMENTE

Como habrá notado, puede ser tedioso nombrar tablas cada vez se va a invocar su uso.

Cuando se añaden más tablas entonces puede ser necesario escribir el nombre de la columna y el nombre de la tabla, seguido de un punto. Por ejemplo:

gid.census_block_data2020_villalba, sum(pop100.census_block_data_2020_villalba).

Como vio en la sección de uso de alias para columnas, además de columnas también se puede usar alias para identificar **temporalmente** tablas. Esto es útil cuando utilizamos más de una tabla en un query.

Pregunta 3: ¿Cuántas edificaciones hay en cada barrio del Municipio de Villalba?

Esta pregunta se debe contestar **combinando dos geodatos**: el de **barrios** y el geodato de **centroides de edificios**.



PostGIS, 3.x

- En la pestaña **gisdb_lab** Script 1, escriba y ejecute el siguiente query:

```
1 select b.barrio, count(e.gid) as num_edifs --barrios y conteo edifs
2 from pr_geodata.g03_legales_barrios_2015_villalba as b, --barrios de Villalba
3 pr_geodata.g33_mapa_base_building_centroids_1998_villalba e --centroides edifs
4 where st_intersects(b.geom,e.geom) --predicado topológico intersección
5 group by b.barrio --agrupa por barrio
6 order by 2 desc; --ordenar descendente por la columna #2
```

Alias en tablas:
Note los alias "b" y "e" usado para nombrar estos geodatos. Al igual que los campos, **podemos usar alias** para **nombrar temporalmente las tablas**. De esa manera nombramos la tabla de **barrios** como **b** y la tabla de **edificios** como **e**.
Predicado Topológico: st_intersects(b.geom, e.geom) puesto en cláusula **WHERE** como **condición: intersección geométrica** entre el geodato/tabla de barrios y la de centroides de edificios. El predicado **st_intersects()** es **conmutativa**, así que el orden de las tablas no importa.

barrio	num_edifs
Hato Puerco Arriba	3,705
Villalba Arriba	2,598
Vacas	1,748
Villalba Abajo	1,510
Caonillas Arriba	1,055
Caonillas Abajo	833
Hato Puerco Abajo	786
Barrio Pueblo	472

8 row(s) fetched - 0.050s, on 2025-06-02 at 10:39:03

- ** Para **contar** los **edificios** usaremos la función **count()** y el campo para hacer el conteo es **gid** de la tabla/geodato **g33_mapa_base_building_centroids**.
** Si escribe **count(e.gid)** o **count(*)**, dan el mismo resultado. Inténtelo. **Count(*)** hace **conteo de filas** y eso es lo que buscamos.

¿Por qué no usamos la función **SUM()**? Vea también que **estamos agregando conteos por barrio**, así que usamos la cláusula **GROUP BY** y el campo **barrio**.

¿Por qué no añadimos el campo **num_edifs** en la cláusula **GROUP BY**?



ORDEN DE OPERACIONES SQL

Para un mejor desempeño en la búsqueda de datos, es importante conocer sobre el **orden de las operaciones en SQL**. Lo primero que se evalúa es la fuente de datos. Por lo tanto, **lo primero que se evalúa** en una sentencia (query) SQL es la **cláusula FROM** y **JOIN** porque son las que **se usan para hacer referencia** a las **tablas/objetos** de la base de datos. El orden se recogerá en esta tabla.

ORDEN	CLÁUSULA	FUNCIÓN
1	FROM/JOIN	Escoger y/o unir tablas para obtener los datos.
2	WHERE	Filtros que se van a aplicar a los datos que están en las tablas de interés.
3	GROUP BY	Agregar los datos si fuere necesario.
4	HAVING	Filtro adicional a los datos agregados por GROUP BY.
5	SELECT	Devuelve los datos finales.
6	ORDER BY	Ordena los datos finales.
7	LIMIT	Limita o reduce el número de filas al número determinado en esta cláusula.

PostgreSQL tiene un mecanismo planificador, el cual lee el query y lo analiza. El analizador optimiza la ejecución y podría alterar el orden de algunas instrucciones.

METADATOS DE LAS TABLAS CON GEOMETRÍAS

Información importante (metadatos) sobre los geodatos en tablas y las columnas (campos) PostgreSQL guarda un tipo de **tabla virtual “View”**, la cual es actualizada con frecuencia. Esta **tabla virtual** llamada **“geometry_columns”** contiene información sobre los geodatos.

Para ver esta **tabla virtual** (view):

- ☐ En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query:

Az f_table_catalog	Az f_table_schema	Az f_table_name	Az f_geometry_column	123 coord_dimension	123 srid	Az type
gisdb_lab	pr_geodata	g03_legales_municipios_2015	geom	2	6,566	MULTIPOLYGON
gisdb_lab	pr_geodata	g03_legales_barrios_2015	geom	2	6,566	MULTIPOLYGON
gisdb_lab	pr_geodata	g29_planes_plan_uso_terrenos_201	geom	2	6,566	MULTIPOLYGON
gisdb_lab	pr_geodata	g31_censo2020_blk_villalba	geom	2	6,566	MULTIPOLYGON
gisdb_lab	pr_geodata	o29_planes_out_2015_villalba	geom	2	6,566	POLYGON

Estas son las tablas que tienen una columna que contenga algún tipo de geometría.

Columnas:

Columna/Campo	Descripción
type	tipos de geometría
srid	código numérico del sistema de referencia espacial . En esta base de datos, los geodatos están registrados en el sistema de coordenadas planas estatales con proyección cartográfica Cónica Conforme de Lambert , Zona 5200 , unidades en metros y datum NAD83 versión/realización de 2011 . Ese sistema tiene el código EPSG:6566 .
f_geometry_column	nombre de la columna que guarda las geometrías. Se le asignó el nombre arbitrario “geom” a cada una de ellas.
f_table_schema	es nombre del schema (especie de folder) donde están los geodatos. Su creación es arbitraria. Nombre: pr_geodata .
f_table_catalog	contiene el nombre de la base de datos .



HACER UN LISTADO DE COLUMNAS DE UNA TABLA

En ocasiones, es conveniente saber cuáles son las columnas que componen una tabla:

- En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query:

```
select ordinal_position
       as colnumber, --posición columna alias colnumber
       column_name,  --nombre de columna
       data_type      --tipo de dato
from   --del view llamado information schema.columns
       information_schema.columns
where  --test: valor del campo table_schema = 'pr_geodata'
       table_schema = 'pr_geodata'
and    --y además que la tabla tenga este nombre
       table_name = 'census_block_data_2020_villalba';
```

Grid	123 colnumber	A-Z column_name	A-Z data_type
1	1	gid	integer
2	2	logrecno	integer
3	3	sumlev	character varying
4	4	geoids	character varying
5	5	cousub	character varying
6	6	barrio	character varying
7	7	tract	character varying
8	8	blkgrp	character varying
9	9	block	character varying
10	10	name	character varying
11	11	vtd	character varying
12	12	pop100	integer
13	13	hu100	integer
14	14	h0010002	integer
15	15	h0010003	integer

Refresh Save Cancel

15 row(s) fetched - 0.004s, on 2025-07-01 at 15:56:57

En este query solamente escogimos las columnas **ordinal_position**, **column_name** y **data_type** del **view** **information_schema.columns**. Un **view** es un tipo de tabla virtual.

La **tabla de interés** es **census_block_data_2020_villalba** y el **schema** es **pr_geodata**.



Modificar datos: DROP, CREATE, INSERT, ALTER, UPDATE

En esta parte veremos otros comandos SQL para borrar tablas, crear tablas, insertar filas en una tabla, cambiar nombres de columnas y actualizar filas de una tabla.

DROP

Se utiliza para borrar bases de datos, tablas, views, columnas de tablas y otros objetos en una base de datos.

Ejemplos:

DROP DATABASE nombre_base_de_datos;	Elimina una base de datos. Solo es posible por medio de usuarios con privilegios administrativos.
DROP TABLE nombre_tabla;	Borrar tabla.
DROP VIEW nombre_view ;	Borrar el view o tabla virtual.
ALTER TABLE tabla DROP COLUMN columna;	Escoger la tabla primero, luego DROP COLUMN columna para eliminarla.

CREATE

Comandos para generar bases de datos, tablas, tablas copiando otra tabla, índices, views, procedures (segmento o script SQL reusable).

Ejemplos:

CREATE DATABASE nombre_base_de_datos;	Crear una base de datos. Solo es posible por medio de usuarios con privilegios administrativos.
CREATE TABLE nombre_tabla (gid SERIAL, columna_1, columna_2); *	Generar una tabla con un campo "gid" numérico entero, secuencial y dos columnas.
CREATE TABLE tabla_nueva AS TABLE tabla_existente;	Crear una tabla idéntica a otra tabla existente.
CREATE TABLE tabla_copia AS SELECT columna1, columna2 FROM tabla_fuente;	Crear una tabla de dos columnas sin filtro usando una tabla existente.

* <https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-serial/>



INSERT

INSERT se usa para insertar filas en una tabla, siguiendo el orden de las columnas. Se usa con la palabra **INTO**. Ejemplo

INSERT INTO tabla(columna1, columna2, etc.) VALUES (valor1, valor2, etc.)	Insertar una fila poniendo valor1 en la columna1 y valor2 en la columna2. Las columnas que no estén mencionadas tendrán valor NULL (vacío).
Procedimiento recomendado para insertar filas desde un query. Fuente: Martínez Llario, 2020.*	
CREATE TABLE pr_geodata.g29_planes_put_usos_urbanos_villalba (gid serial primary key, geom geometry ('MULTIPOLYGON', 6566), tipo character varying, municipio character varying, clasiput character varying, descripput character varying, r_put character varying, v_put character varying); INSERT INTO pr_geodata.g29_planes_put_usos_urbanos_villalba (geom, tipo, municipio, clasiput, descripput, r_put, v_put) SELECT geom, tipo, municipio, clasiput, descripput, r_put, v_put FROM g29_planes_plan_uso_terrenos_2015 WHERE municipio = 'Villalba' and clasiput like 'SU%';	Crear una tabla con el nombre g29_planes_put_usos_urbanos_villalba con toda su definición de columnas y columna geométrica con sistema de referencia geoespacial SRID=6566 (PR, NAD83-2011) Insertar filas en la tabla anteriormente creada. Las filas se generarán a partir de un query donde el municipio = Villalba y tenga usos urbanos.

* Martínez Llario, 2020, p. 114.

ALTER

Se usa para hacer cambios en una tabla existente: cambiar el nombre de la tabla, añadir o modificar columnas (cambiar nombre, cambiar el tipo de datos).

ALTER TABLE tabla RENAME TO nuevo_nombre;	Cambiar el nombre de una tabla.
ALTER TABLE tabla RENAME COLUMN nombre_viejo TO nombre_nuevo ;	Cambiar el nombre de una columna de una tabla.
ALTER TABLE tabla ADD nueva_columna tipo_de_dato;	Añadir una columna y su definición de tipo de dato en una tabla existente.
ALTER TABLE tabla DROP COLUMN columna;	Quitar una columna de la tabla.
ALTER TABLE tabla ALTER COLUMN col1 TYPE varchar(80) ALTER COLUMN col2 TYPE varchar(100);	Cambiar el tipo de datos de dos columnas de una tabla en una sola sentencia SQL.



UPDATE

El enunciado UPDATE permite hacer cambios en los datos de una o varias columnas en una tabla.

UPDATE tabla SET columna1 = valor1, SET columna2 = valor2, SET columna3 = valor3 WHERE condición o filtro;	Modificar los valores una o más columnas. Aplicar filtro WHERE para cambiar solamente un subconjunto de datos. De lo contrario, asignará el mismo valor a todas las filas de las columnas nombradas.
---	--

Ya que contamos con las herramientas para conocer el contenido de la base de datos, y de qué columnas se componen las tablas, podemos pasar a la sección de preguntas de práctica para este módulo de SQL Básico. Además, incluiremos algunas preguntas que conlleven usar geodatos.

Prácticas con SQL Básico

En esta sección proponemos una serie de preguntas para contestar mediante queries a la base de datos.

- Dé una lista de **TODAS** las filas y **TODAS** las columnas de la tabla **census_block_data_2020_villalba**. La cantidad de filas devuelta debe ser 378. Recuerde que por defecto, DBeaver solo da las primeras 200 filas. Use el botón  **Calculate Total Row Count** para obtener el conteo de filas.
- Use la cláusula **DISTINCT** para hacer una **lista de los barrios** que aparecen en la tabla **census_block_data_2020_villalba**. Solo incluya la columna **barrio**. En la lista debe haber **solo 8 barrios**.
- Use la cláusula **DISTINCT** para hacer una **lista de los sectores censales** (tract) que aparecen en la tabla **census_block_data_2020_villalba**. Solo incluya la columna **tract**. En la lista **debe haber 6 filas**.
- ¿Cuántas unidades electorales hay en Villalba (según el Censo de 2020)? Haga primero una lista de las unidades electorales.** Las unidades electorales están en la columna **vtd** de la tabla **census_block_data_2020_villalba**. Debe usar la cláusula **DISTINCT** seguido del campo **vtd**: `Count(Distinct(vtd))`. ¿Cuántas filas devuelve si no usamos **DISTINCT**? ¿Por qué?
- ¿Cuál fue la población total del municipio de Villalba para el censo 2020?** Use la función `sum()` en la columna **pop100**. Debe devolver solo una fila.
- ¿Cuántos habitantes, viviendas, viviendas ocupadas y viviendas vacantes había en Villalba para el Censo de 2020?** Use la función **SUM(column_name)** para sumar cada campo: **pop100**, **hu100**, **h0010002**, **h0010003**. Provea un alias **AS** a cada columna para este query: **pop100 as población2020**, **hu100 as viviendas2020**, **h0010002 as ocupadas**, **h0010003 as vacantes**. Use la tabla **census_block_data_2020_villalba**. Debe producir **una sola fila** con las sumas para cada columna.

Resultado:

	123 población2020	123 vivienda2020	123 ocupadas	123 vacantes
1	22,093	8,875	7,989	886



Queries que usan GROUP BY

- Haga una **lista de barrios con la suma de población y viviendas, ocupadas y desocupadas**. Solo modifique el query anterior. Añada primero el barrio y luego las demás columnas. Debe usar la cláusula **GROUP BY** y luego añadir el campo barrio. Use la tabla **census_block_data_2020_villalba**. Recuerde que **GROUP BY** se escribe debajo o **luego del nombre de la tabla** en la cláusula **FROM**. Debe producir **8 filas**.

	rec barrio	123 población2020	123 vivienda2020	123 ocupadas	123 vacantes
1	Villalba Abajo	2,524	1,062	954	108
2	Villalba Arriba	5,062	2,009	1,839	170
3	Caonillas Abajo	1,249	552	462	90
4	Caonillas Arriba	1,288	527	457	70
5	Hato Puerto Arriba	6,304	2,525	2,298	227
6	Villalba CUT	734	301	257	44
7	Hato Puerto Abajo	1,459	578	506	72
8	Vacas	3,473	1,321	1,216	105

Resultado:

- Haga una **lista de unidades electorales (vtd)** y añada la **población, viviendas, ocupadas y vacantes**. Solo modifique el query anterior cambiando la columna barrio por la columna vtd. Use la cláusula **GROUP BY** y añada la columna **vtd**. Ordene los resultados con **ORDER BY** usando la columna **población2020** de manera **descendente DESC**. Use la tabla **census_block_data_2020_villalba**. ¿Cuál es la unidad electoral con menor población? El número de filas debe ser 17.

	rec vtd	123 población2020	123 vivienda2020	123 ocupadas	123 vacantes
1	065.17	227	67	52	15
2	0065.2	265	128	124	4
3	0065.8	521	216	181	35
4	0065.5	554	233	205	28
5	065.10	577	273	218	55
6	0065.9	603	220	204	16
7	0065.6	877	338	301	37
8	0065.4	1,025	415	367	48
9	065.16	1,128	456	402	54
10	065.14	1,141	499	450	49
11	065.11	1,333	570	506	64
12	0065.1	1,447	619	538	81
13	065.13	1,774	727	649	78
14	0065.7	1,936	836	762	74
15	065.15	2,422	906	865	41
16	0065.3	3,088	1,188	1,098	90
17	065.12	3,175	1,184	1,067	117

Resultados:

- Desglose por unidades electorales y barrios:** Use el mismo query anterior. En la cláusula **SELECT** añada la columna **barrio** después de la columna **vtd**. Incluya la coma, ejemplo: **vtd, barrio**, y deje las demás columnas nombradas. En la cláusula **GROUP BY** inserte la columna **vtd** antes de **barrio**, añada una coma. Por ejemplo: **GROUP BY vtd, barrio**. En la cláusula **ORDER BY** inserte primero el campo **vtd ASC**, añada coma y **barrio asc**. Mantenga **población2020 DESC**. ¿Qué pasó con las sumas de población



y viviendas por unidad electoral (vtd) y por qué? El número de filas debe ser 34.

	asc vtd	asc barrio	12a población2020	12a vivienda2020	12a ocupadas	12a vacantes
1	0065.1	Villalba Arriba	1,447	619	538	81
2	0065.2	Villalba Arriba	265	128	124	4
3	0065.3	Hato Puerco Arriba	1,283	538	468	70
4	0065.3	Vacas	1,796	649	629	20
5	0065.3	Villalba CUT	9	1	1	0
6	0065.4	Vacas	1,025	415	367	48
7	0065.5	Vacas	554	233	205	28
8	0065.6	Hato Puerco Abajo	262	119	90	29
9	0065.6	Hato Puerco Arriba	615	219	211	8
10	0065.7	Caonillas Arriba	17	2	2	0
11	0065.7	Hato Puerco Arriba	1,858	820	755	65
12	0065.7	Vacas	61	14	5	9
13	0065.8	Caonillas Arriba	433	178	159	19
14	0065.8	Hato Puerco Arriba	88	38	22	16
15	0065.9	Caonillas Arriba	603	220	204	16
16	0065.10	Caonillas Abajo	419	174	148	26

Resultados:

10. Haga un **conteo de municipios por letra inicial**. Por ejemplo, ¿cuántos municipios empiezan con la letra A, cuántos con la letra B, hasta con la Y? Use la función **count(municipio)** para hacer los conteos.
 ** Para este query debe usar la función **left(municipio,1)** o la función **substr(municipio,1,1)** para obtener la letra inicial. Para hacer la lista de municipios hacia la derecha, debe usar la función **string_agg(municipio, ',')**. Recuerde dejar un espacio después de la coma. Haga GROUP BY por el campo de letra inicial. Haga orden alfabético ascendente por municipio y de forma descendente por conteo.

Geodato a usar: **g03_legales_municipios_2015**. Por ejemplo:

SELECT left(municipio,1) as letra_inicial,
count(municipio) as cuántos,
string_agg(municipio, ',') as municipios -- no escribir coma aquí.
FROM ESCRIBA EL NOMBRE DEL GEODATO MUNICIPAL
GROUP BY ¿Cuál sería el campo por usar para agrupar? (puede usar número de columna)
ORDER BY _____; -- ¿cuál es el campo para ordenar? ¿de cuál manera?
 ¿Cuáles son las **letras iniciales que no se repiten** en los nombres de municipios?
 _____ (Ver final del listado o cambiar order by ____ asc)

	asc letra_inicial	12a cuántos	asc cuáles
1	C	14	Cabo Rojo, Caguas, Camuy, Canóvanas, Carolina, Cataño, Cayey, Ceiba, C
2	A	8	Adjuntas, Aguada, Aguadilla, Aguas Buenas, Aibonito, Arecibo, Arroyo, A
3	S	7	Sabana Grande, Salinas, San Germán, San Juan, San Lorenzo, San Sebasti
4	M	6	Maratí, Maricao, Manabo, Manzanillo, Moca, Morovis

Resultados parciales:

11. **Crear una copia de una tabla existente:** Use el comando **CREATE TABLE** tabla_nueva **AS** (SELECT * FROM Tabla_Existente) para hacer una copia de una tabla partir de la tabla **census_block_data_2020_villalba**. El nombre debe ser **block_data2020_estudianteXX**. Sustituya el número de la tabla con el número que le corresponde en el taller.

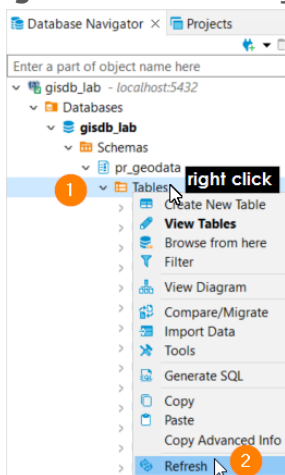
Name	Value
Updated Rows	378
Query	create table block_data2020_estudiante1 as (select * from census_block_data_2020_villalba)
Start time	Thu Feb 15 11:54:05 BOT 2024
Finish time	Thu Feb 15 11:54:05 BOT 2024

Resultado: Click en tab **Statistics 2**



PostGIS, 3.x

Para ver la tabla nueva, vaya al **panel Database Navigator** y haga **right click** en **Tables** y escoja **Refresh** o **F5**



Verifique que la tabla tiene el contenido deseado:

Escriba y ejecute el siguiente query:

SELECT * FROM block_data2020_student01;

12. Supongamos que usted entiende que el nombre del barrio **"Villalba CUT"** está incorrecto o le indicaron que debe **cambiarlo** a **"Villalba Pueblo"**.

Use el comando **UPDATE** para cambiar solamente el nombre de este barrio. **Tiene que filtrar los datos usando WHERE**. Si no lo hace, le pondrá el nombre **Villalba Pueblo** a todas las filas de la columna barrio en la tabla que usted creó. Por ejemplo:

UPDATE block_data2020_studentXX

SET barrio = 'Villalba Barrio-Pueblo'

WHERE barrio = _____;

Resultado:

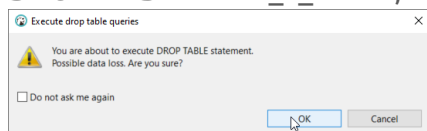
Name	Value
Updated Rows	21
Query	update block_data2020_estudiante1 bde set barrio = 'Villalba Pueblo' where barrio = 'Villalba CUT'
Start time	Thu Feb 15 12:11:54 BOT 2024
Finish time	Thu Feb 15 12:11:54 BOT 2024

Para **verificar**: escriba el query :

select DISTINCT ¿cuál campo? from
block_data2020_student01;

13. **Elimine la tabla block_data2020_student01** usando el comando **DROP**.

DROP TABLE tabla_a_borrar;



Click **OK** para confirmar.

Name	Value
Updated Rows	0
Query	drop table block_data2020_estudiante1
Start time	Thu Feb 15 12:22:26 BOT 2024
Finish time	Thu Feb 15 12:22:26 BOT 2024

Resultado:



Queries geográficas:

14. ¿Cuáles son los 20 municipios de mayor área superficial?

Use el geodato **g03_legales_municipios_2015**. Incluya el nombre del municipio, utilice la función **ST_Area(geom)**. Compute el área en **millas** y **kilómetros cuadrados**. Por ejemplo **ST_Area(geom)/2589988.1103** para **millas cuadradas AS sqmi** y **ST_Area(geom)/1000000** para **kilómetros cuadrados AS sqkm**. Use la cláusula **ORDER BY DESC** para ordenar el área de manera descendente por **sqkm** o **sqmi**. Use **LIMIT** para reducir la lista a solo 20 filas.

Grid	municipio	sqmi	sqkms
	Arecibo	126.84709925275642	328.53247889068314
	Ponce	116.88455447166889	302.72960635933515
	Utuado	114.97764879483009	297.79074332885904
	Coamo	78.01596803317535	202.060429619469
	Mayagüez	77.84267686833975	201.61160756292477
	Cabo Rojo	72.12341877717205	186.79879710706336
	San Sebastián	71.27173366845338	184.59294280176243
	Salinas	70.03695915082955	181.3948914822153
	Yauco	68.28242950458271	176.85068055926715
	Adjuntas	67.11879257749183	173.8368747533957
	Ciales	66.7350693378012	172.84303612495123
	Guayama	65.61200417029657	169.93431069402212
	Orocovis	63.76471098869199	165.14984331742798
	Lares	61.643874858119574	159.65690295535077
	Río Grande	60.80566858452703	157.48595867276723
	Juana Díaz	60.757060550029045	157.3600644413524
	Lajas	60.52811537892443	156.76709917028086
	Caguas	59.024424034101834	152.87255646562932
	Isabela	55.381066548245165	143.436303895688
	Yabucoa	55.23850684868433	143.06707596881753

Resultado:

15. ¿Cuáles es la longitud total de carreteras y calles en el municipio de Villalba? Utilice el geodato **g35_viales_mapa_base_tiger_rds_2006se_villalba**. Debe utilizar las funciones **SUM()** y **ST_Length(geom)** para sumar las longitudes. Estas deben escribirse luego de SELECT. La sintaxis debe ser: **SUM(ST_Length(geom)/1000) AS kms**

SUM(ST_Length(geom)/1609.344) AS mi. Use el geodato **g35_viales_mapa_base_tiger_rds_2006se_villalba**. Recuerde que el resultado solo va a tener una fila.

Grid	kms	mi
	349.4341702467361	217.1283269746775

Resultado:

16. La **Comisión Estatal de Elecciones** ha ubicado asentamientos existentes alrededor de toda la isla. Estos asentamientos pueden ser **sectores, urbanizaciones, condominios, centros de cuido** y están representados por un punto o dos en el caso de estar entre dos o más límites legales.

Dé una lista que muestre cada asentamiento y en cuál unidad electoral está ubicado. En la parte FROM use un alias para cada geodato:

g31_censo2020_blk_villalba AS b y el geodato **g25_asentamientos_sectores_2017 AS s**. Separe las tablas con una coma. En la parte **SELECT** añada: **s.tipo, s.nombre, b.vtd** as **unidad_electoral**. En la cláusula **WHERE** debe ubicar la función



ST_Intersects(a.geom, e.geom). Haga **ORDER BY** por **s.nombre** de forma **ascendente**. El resultado debe tener **81 filas**.

tipo	nombre	unidad_electoral
Sector	Aceituna	065.17
Urbanización	Alturas de Villalba	0065.3
Urbanización	Alturas del Alba	065.15
Sector	Apeadero	0065.4
Barriada	Borinquen	065.15
Sector	Camarones Abajo	065.12
Sector	Cerro Gordo	065.10
Sector	Chichón	0065.2
Barriada	Cooperativa	065.16
Sector	Cooperativa	065.12

Resultado parcial:

17. Realice un **conteo de edificaciones ubicadas en cada tipo de zona inundable**.

En la parte **FROM** los geodatos a combinar son:

g23_riesgo_inunda_floodzones_2017_villalba AS f,

g33_mapa_base_building_centroids_1998_villalba AS b.

– Borre los alias para las tablas que DBEaver asigna por defecto. –

En la parte **SELECT**, use las columnas **fld_zone**, **zone_subty** y añada la función **count(b.gid)** as **num_edifs** para contar filas.

En la cláusula **WHERE** debe ubicar la función **ST_Intersects(f.geom, b.geom)**.

En la cláusula **GROUP BY** use las columnas **fld_zone**, **zone_subty**.

En la cláusula **ORDER BY** ordene el conteo de puntos **num_edifs** de manera **descendente**.

	fld_zone	zone_subty	num_edifs
X	0.2% Annual Chance Flood		241
A			227
A	FLOODWAY		47

Resultados:

Definición de **zona inundable "A"** según FEMA:

Areas with a 1% annual chance of flooding and a 26% chance of flooding over the life of a 30-year mortgage. Because detailed analyses are not performed for such areas; no depths or base flood elevations are shown within these zones. Last updated April 20, 2023.

18. Modifique el query anterior y haga otro **conteo de edificios** pero esta vez use el geodato **g29_planes_put_2015_villalba**.

En la cláusula **FROM** los geodatos son:

g33_mapa_base_building_centroids_1998_villalba AS b

g29_planes_put_2015_villalba AS p

En la cláusula **SELECT** añada **p.clasiput** as **código_de_uso** y **p.descripput** as **descripción**, **count(b.gid)** as **num_edifs**.

En la cláusula **WHERE** debe tener **st_intersects(b.geom,p.geom)** o **(p.geom,b.geom)**

En la cláusula **GROUP BY** añada las columnas **p.clasiput** y **p.descripput**.

	código_de_uso	descripción	num_edifs
1	SU	Suelo Urbano	5,746
2	SRC	Suelo Rústico Común	5,362
3	SREP-A	Suelo Rústico Especialmente Protegido Agrícola	561
4	SREP	Suelo Rústico Especialmente Protegido	443
5	SREP-H	Suelo Rústico Especialmente Protegido Hídrico	386
6	SREP-E	Suelo Rústico Especialmente Protegido Ecológico	144
7	VIAL	Vial	44
8	SURP	Suelo Urbanizable Programado	9
9	SREP-AH	Suelo Rústico Especialmente Protegido Agrícola e Hídrico	8
10	SURNP	Suelo Urbanizable No Programado	2
11	AGUA	Agua	2

Resultado:



19. ¿Cuáles son los **5 barrios con mayor área territorial** y a qué **municipio pertenecen**?
FROM: g03_legales_barrios_2015. Expanda los nodos para ver la lista de campos.
SELECT: municipio, barrio, SUM(ST_Area(geom)/1000000) AS sqkms para que le devuelva los valores **en kilómetros cuadrados**.
ORDER BY en la nueva columna **sqkms desc**.
 Use **LIMIT** para devolver las primeras 5 filas

municipio	barrio	sqkm
Salinas	Lapa	64.15836650564829
Arecibo	Sabana Hoyos	59.604931017627514
Mayagüez	Isla de Mona e Islote Monito	56.85899241097673
Salinas	Aguirre	52.05658306768688
Cabo Rojo	Llanos Costa	51.978746695017534

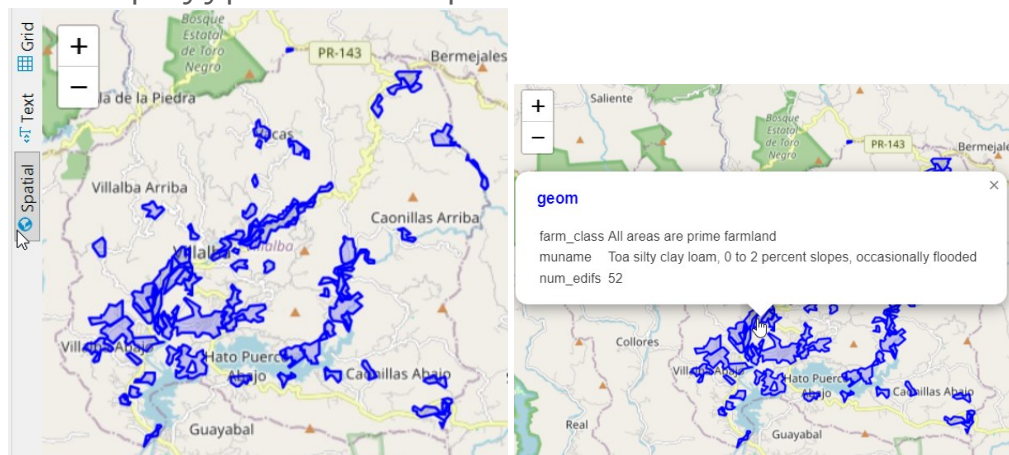
Resultado:

20. Realice un conteo de edificios ubicados en suelos potencialmente agrícolas.
SELECT: s.farm_class, s.muname, count(b.gid) as num_edifs
FROM: g33_mapa_base_building_centroids_1998_villalba AS b,
g15_suelos_soil_map_units_2018_villalba AS s
WHERE: st_intersects(¿geometrías?) AND s.farm_class <> 'Not prime farmland'
GROUP BY: s.farm_class, s.muname
ORDER BY: s.farm_class, num_edifs desc

farm_class	muname	num_edifs
All areas are prime farmland	Montegrande clay, 2 to 12 percent slopes	1,343
All areas are prime farmland	Lares clay, 5 to 12 percent slopes	498
All areas are prime farmland	Toa silty clay loam, 0 to 2 percent slopes, occasionally flooded	204
All areas are prime farmland	Dagüey clay, 12 to 20 percent slopes	41
Farmland of statewide importance	Quebrada silty clay loam, 12 to 20 percent slopes, eroded	1,447
Farmland of statewide importance	Los Güineos clay, 20 to 40 percent slopes	95
Farmland of statewide importance	Humatas clay, 20 to 40 percent slopes	76
Farmland of statewide importance	Alonso clay, 20 to 40 percent slopes	37
Farmland of statewide importance	Callabo silty clay loam, 12 to 20 percent slopes	32
Farmland of statewide importance	Múcara silty clay, 12 to 20 percent slopes, eroded	25
Farmland of statewide importance	Jácana clay, 5 to 12 percent slopes	7
Prime farmland if irrigated	Llanos clay, 2 to 5 percent slopes	7

Resultado:

21. Si deseamos **ver este resultado en el contexto de un mapa**, solo debemos **añadir la geometría del geodato** de suelos en la cláusula SELECT después del count:
count(e.gid) num_edifs, s.geom
 y en la cláusula GROUP BY después de s.muname: **s.muname, s.geom**
 Corra el query y presione el tab Spatial





SQL espacial para geometrías vectoriales

Operaciones espaciales: Adyacencia, continencia, intersección, distancia

En esta parte vamos a practicar algunas funciones espaciales para búsqueda/selección de datos geográficos. En la sección anterior ya habíamos practicado algunas búsquedas geográficas, tanto en las prácticas de demostración, como en las prácticas asignadas.

La mayoría de los queries geográficos practicados han utilizado la función o “predicado” `ST_Intersects(geomA,geomB)` para seleccionar geometrías que tengan algún punto en común. Sin embargo, hay variedad de estos predicados que pueden ser utilizados, según el tipo de búsqueda que interesamos, así como también los tipos de geometría que estaremos comparando. Los apéndices A-1 y A-2 explican en detalle estos predicados. El apéndice A-1 hace un inventario de las posibles combinaciones entre relaciones topológicas por tipo de geometría (dimensión=0 para punto, 1 para líneas, 2 para polígonos). El apéndice A-2 hace otro inventario de los predicados espaciales que vimos en la sección anterior, tales como `Intersects`, `Disjoint`, `Touches`, `Overlaps`, etc. Todos estos se comparan con las distintas dimensiones geométricas. Todas las descripciones en los apéndices [A-1](#), y [A-2](#) provienen de **JC Proteau, Matrices de Clementini et Prédicas Spatiaux de l'OGC, 2011***. En el apéndice [A-2](#) se añade la fuente de la documentación de PostGIS <https://postgis.net/docs/> para la versión “dev” del 23 de enero de 2024.

Comenzaremos por mostrar el modelo de elementos geográficos simples del [Open Geospatial Consortium](#).

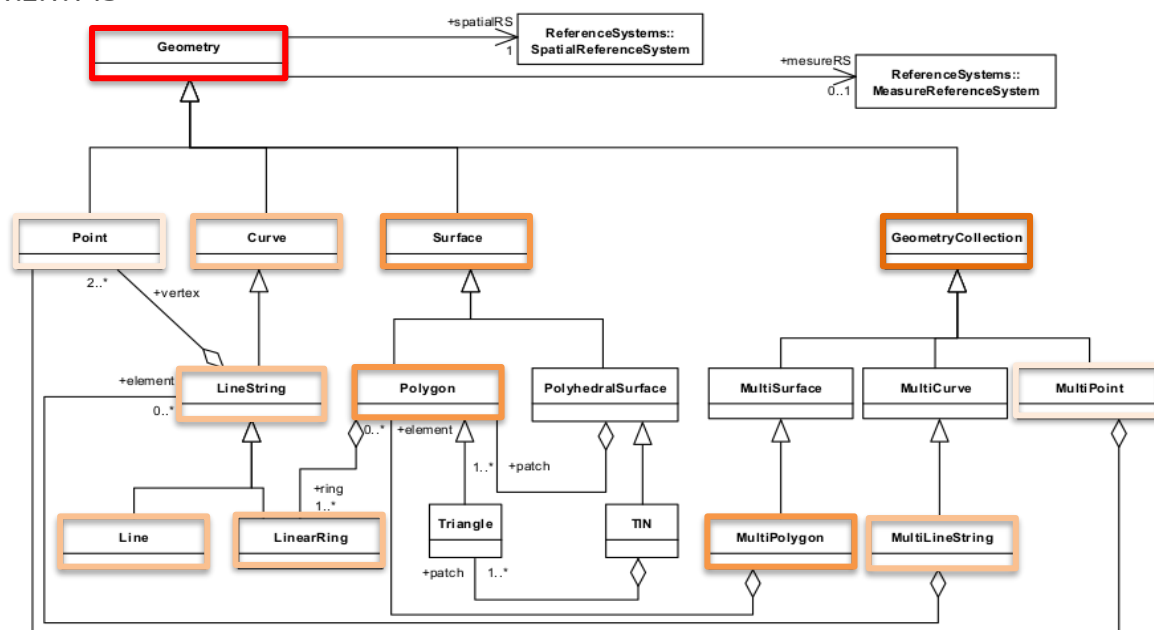
* Nota: el texto citado ya no está disponible (a junio 2, 2025). El enlace http://www.geoinformations.developpement-durable.gouv.fr/fichier/pdf/Predicats_OGC_V3_cle119417.pdf?arg=177828548&cle=baf65cf1b040772ae1d97f164cf9ca67f6dbb698&file=pdf%2FPredicats_OGC_V3_cle119417.pdf devuelve error “Forbidden”. Puede descargar copia del documento desde: https://gis.pr.gov/seccioneducativa/Documents/Tutoriales/PostGIS/Predicats_OGC_V3_cle119417.pdf



El modelo *Simple Features* del Open Geospatial Consortium

Muchos programados *open source* de manejo de datos geográficos utilizan este estándar para codificar y registrar las geometrías que se usarán para representar elementos geográficos y eventos localizables. QGIS hace uso de este estándar y es buena idea describir algunos aspectos de importancia. De esta manera, podremos entender mejor el comportamiento de éste y otros programas que adoptan este estándar.

El siguiente diagrama fue extraído del documento [OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture](#), versión 1.2.1.145



En este diagrama podemos notar las jerarquías de las geometrías. Las de arriba son las geometrías abstractas, *Geometry*, *Point*, *Curve*, *Surface*, *GeometryCollection*, de las cuales se derivan *LineString*, *Polygon*, *MultiPolygon*, *MultiLineString* y otras. QGIS y otros programas tipo *Desktop*, no leen directamente el tipo *GeometryCollection*, pero sí *MultiPolygon*, *MultiPoint* y *MultiLineString*.

Dimensión de las geometrías

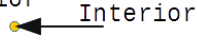
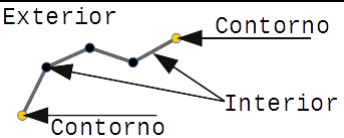
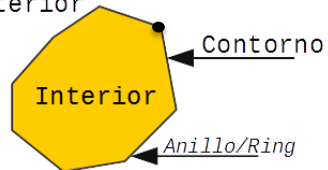
Como se ha mencionado antes, estas son las dimensiones numéricas otorgadas a los tipos básicos de geometría, según los estándares del OGC. **Estas dimensiones no están relacionadas con el número de ejes x,y,z,m que pueden tener algunas geometrías.**

Geometría	Dimensión
Punto, MultiPoint	0
Entidad lineal	1
Entidad superficial	2



Interior, contorno y exterior de las geometrías

Estas características de las geometrías nos ayudan a entender las relaciones espaciales y el uso de los operadores y predicados espaciales:

Point/MultiPoint		Interior	El mismo punto o puntos.
		Contorno/Límite	Vacío/No existe.
Line/MultiLine		Interior	Puntos que no estén en los puntos del contorno.
		Contorno/Límite	Puntos de inicio y final . MultiLine: puntos de contorno que estén en líneas componentes que sean impares .
Polygon/MultiPolygon		Interior	Puntos del interior de los anillos: MultiPolygon: Puntos del interior de los anillos.
		Contorno/Límite	Conjunto de anillos exteriores e interiores.

El **exterior** de estas geometrías estará compuesto por los puntos que no estén ni en el interior ni el contorno. Referencia: https://portal.ogc.org/files/?artifact_id=13227.

Geometría y dimensión	Interior	Límite/Contorno/Boundary	Exterior (área gris)
Punto (Dim=0)	Dim=0	Vacío / No existe	
Poli-línea (Dim=1)	Dim=1	Dim=0	
Polígono/área (Dim=2)	Dim=2	Dim=1	Dim=2

Referencia: [Predicados OGC: 2.2: Conceptos de interior, límite y exterior de objetos](#), pág. 6. Ministerio de Ecología, desarrollo sustentable, transportación y vivienda, Gobierno de Francia. Disponible en: https://gis.pr.gov/seccioneducativa/Documents/Tutoriales/PostGIS/Predicats_OGC_V3_cle119417.pdf



Predicados para las relaciones topológicas

A continuación, una breve explicación de los predicados disponibles en PostGIS. Tabla 3.

Predicado	Geometría: P punto, L polilínea, S polígono	Condiciones
ST_Equals	Todas	geom(A) es igual a geom(B) si: * la <i>relación topológica</i> entre estos es idéntica ... aunque ... * el número de vértices y la dirección de las líneas pueden ser diferentes
ST_Disjoint	Todas	geom(A) es desjunto o separado de geom(B) si: * los objetos no tienen ningún punto en común (interior o límite). (Este es el inverso de <i>Intersects</i>)
ST_Touches	S/S, L/S, L/L, P/S, P/L	geom(A) toca a geom(B) si: * los límites (contornos) de los objetos tienen al menos un punto en común... y ... * si los interiores de ambos no tienen algún punto en común
ST_Crosses	P/S, P/L, L/S, L/L	geom(A) cruza a geom(B) si: * los interiores de los objetos tienen al menos un punto en común... pero ... * no todos en común... y ... * si la dimensión de la intersección de los interiores es <i>inferior</i> a la dimensión máxima de los objetos geom(A) y geom(B) (no aplica a P/P, S/S)
ST_Within	Todas	geom(A) está dentro de geom(B) si: * todo punto de geom(A) está dentro de geom(B) ... y ... * si los interiores tienen al menos algún punto en común ... * ningún punto de geom(A) está en el exterior de geom(B) * (inverso de <i>Contains</i>)
ST_Contains	Todas	geom(A) contiene a geom(B) si: * todo punto de geom(B) es un punto de geom(A) ... y ... * si los interiores tienen al menos algún punto en común * ningún punto de geom(B) está en el exterior de A * (inverso de <i>Within</i>)
ST_Overlaps	S/S, L/L, P/P	geom(A) solapa a geom(B) si a la vez: * geom(A) y geom(B) tienen la misma dimensión (no aplica a P/L, P/S, L/S) ... * geom(A) y geom(B) tienen puntos en común... pero no todos ... * La <i>intersección</i> de los interiores de geom(A) y geom(B) tiene la misma <i>dimensión</i> que geom(A) y geom(B)
ST_Intersects	Todas	geom(A) interseca a geom(B) si: * geom(A) y geom(B) tienen al menos un punto en común (interior o límite) * (Inverso de <i>Disjoint</i>)
ST_Covers	Todas	geom(A) cubre a geom(B) si: * ningún punto de geom(B) está en el exterior de A * todo punto de geom(B) es un punto de geom(A) Compárese con <i>Contains</i>
ST_CoveredBy	Todas	geom(A) está cubierto por geom(B) si: * ningún punto de geom(A) está en el exterior de geom(B) * todo punto de geom(A) es un punto de geom(B) Compárese con <i>Within</i>
ST_Relate (AB, DE-9IM Pattern Matrix)	Todas	* Explica la relación espacial entre geom(A) y geom(B) mediante la aplicación del modelo DE9IM. * Permite la generalización de los predicados espaciales para 98 relaciones topológicas. De las 98 relaciones, 92% son entre LL, LS y SS

***ST_Relate** solo se puede usar desde consultas SQL espaciales a la base de datos o a través del DB Manager de QGIS.



Referencia: [Predicados OGC: 2.2: Conceptos de interior, límite y exterior de objetos](#), pág. 6. Ministerio de Ecología, desarrollo sustentable, transportación y vivienda, Gobierno de Francia. Disponible en:

https://gis.pr.gov/seccioneducativa/Documents/Tutoriales/PostGIS/Predicats_OGC_V3_cle119417.pdf

La función **ST_DWithin(geom_a, geom_b, distancia)** para **análisis de proximidad** es una implementación de **PostGIS**. En cuanto sea posible, **sustituya el uso de ST_Intersects con ST_DWithin**, ya que el parámetro de distancia en ST_DWithin sirve además como umbral de tolerancia mediante una distancia no significativa. (Martínez Llario, 2018, pp. 143-46.) Hsu y Obe (2015, p249-50) se refieren a esta función como un "ST_Intersects con tolerancia".

DESGLOSE DE RELACIONES TOPOLÓGICAS POR GEOMETRÍA

En la tabla anterior, en la sección ST_Relate se mencionó que existen **98 tipos diferentes de relaciones topológicas** para geometrías puntuales, lineales y de área.

Estas se dividen en:

Desglose de Relaciones Topológicas por tipo de geometría					
Puntos		Líneas		Áreas(S)	
Relaciones	Cantidad	Relaciones	Cantidad	Relaciones	Cantidad
Punto/Punto	2	Línea/Línea	47	Área/Área(S)	12
Punto/Línea	3	Línea/Área(S)	31		
Punto/Área(S)	3				

Ver apéndice [A-1](#). Referencia: [Predicados OGC: 2.2: Conceptos de interior, límite y exterior de objetos](#), pág. 6. Ministerio de Ecología, desarrollo sustentable, transportación y vivienda, Gobierno de Francia.

Disponible en:

https://gis.pr.gov/seccioneducativa/Documents/Tutoriales/PostGIS/Predicats_OGC_V3_cle119417.pdf

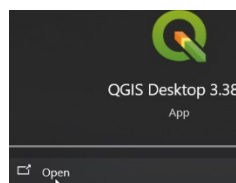


PostGIS, 3.x

USAR QGIS PARA EJEMPLOS DE RELACIONES TOPOLÓGICAS

En esta parte vamos a practicar algunas estas funciones de relación, de manera que las podamos visualizar y poder tener una idea más clara de cómo funcionan. En lugar de escribir los predicados, usaremos la herramienta **Select by Location** de QGIS. Esta contiene la mayoría de los predicados, excepto ST_Relate, el cual es un poco más complejo de construir. **La ventaja de QGIS es que podemos ver todos los geodatos**, a diferencia de PostGIS, donde se ve uno a la vez.

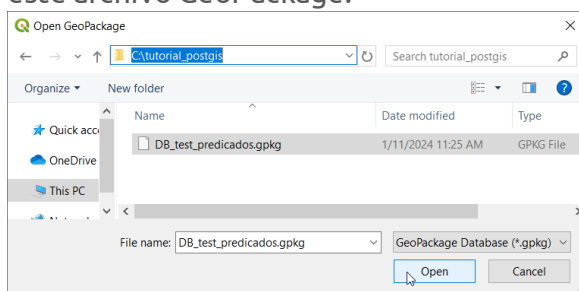
- ☐ Descargue el archivo geopackage destinado para esta práctica en este enlace:
https://gis.pr.gov/seccioneducativa/Documents/Tutoriales/PostGIS/tutorial_postgis.zip
- ☐ Una vez descargado, descomprima este archivo en el folder C:\tutorial_gis\
- ☐ **Abra una sesión de QGIS.**



- ☐ Vaya al panel **Browser** y haga **right click** en **Geopackage** y escoja la opción **New Connection...**



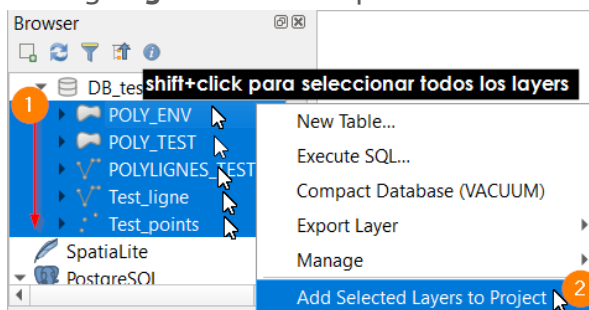
- ☐ Localice y escoja (**Open**) el archivo **DB_test_predicados.gpkg** en el directorio **C:\tutorial_postgis**. Esta localización puede variar, según donde se haya guardado este archivo GeoPackage.



- ☐ Dentro de este GeoPackage, **escoja (shift+click) los 5 layers existentes**.
 - ☐ **Expanda el nodo** del archivo GeoPackage **DB_test_predicados.gpkg**
 - ☐ Haga **shift+click** en **cada geodato**. Manténgalos **seleccionados**.



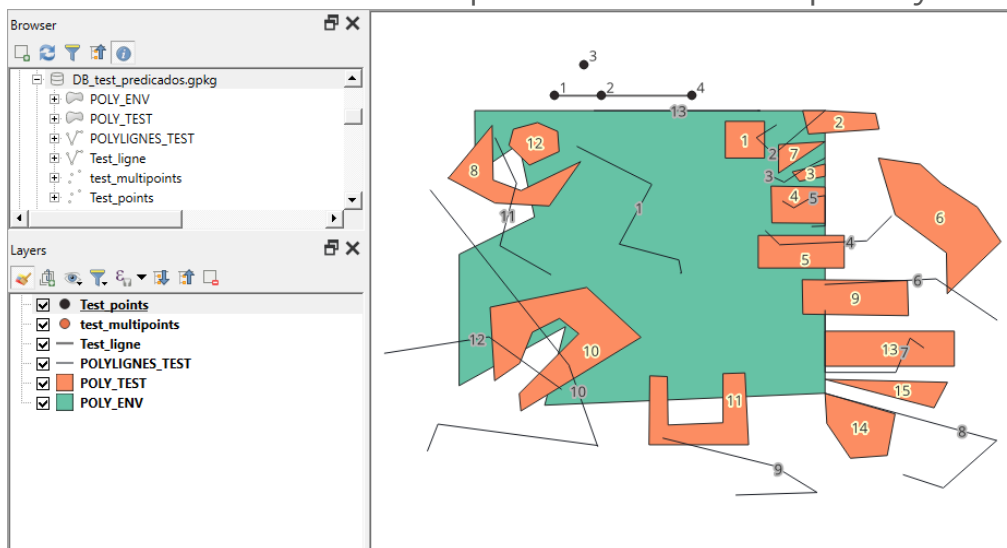
- ☐ Haga **right click** en la opción **Add Selected Layers to Project**



Este archivo *GeoPackage* fue copiado desde el [espacio interministerial de la información geográfica, del gobierno francés](#) (2011). El archivo original estaba en formato Spatialite y fue exportado a GeoPackage, incluyendo una traducción de los textos originales de algunas filas en las tablas. La exportación y traducción fue hecha en nuestra agencia. Ninguna de sus coordenadas fue modificada ni el sistema de referencia geoespacial, ya que esto podría alterar las funciones que vamos a experimentar.

Los geodatos tienen coordenadas reales en el sistema Lambert-93 ([EPSG:2154](#)) y están localizadas en la [Universidad CY Cergy-Paris](#). Por otro lado, los elementos geográficos de este banco de datos son ficticios.

- ☐ En su visor de QGIS deberá ver los geodatos experimentales para las pruebas que vamos a realizar. Se añadieron etiquetas de identificadores para mayor claridad.



- ☐ Vaya a la **barra de botones** en el **lado superior de la interfaz gráfica de QGIS** y haga **click** en el botón **Select by Location**:



- ☐ Una vez aparezca la herramienta, comencemos a hacer las evaluaciones:

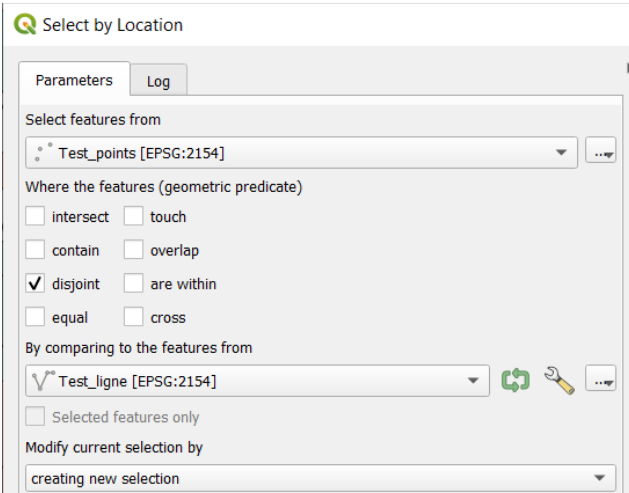


ST_Disjoint

Predicado	Geometría: P punto, L polilínea, S polígono	Condiciones
ST_Disjoint	Todas	geom(A) es desajunto o separado de geom(B) si: * los objetos no tienen ningún punto en común (interior o límite). (Este es el inverso de <i>Intersects</i>)

Disjoint: Puntos con líneas

Disjoint encontrará elementos que no tengan ningún punto en común entre las geometrías A y B, entre los elementos del geodato de puntos y la línea de prueba:



- ☐ Select features from: **Test_points...**
- ☐ Where the features (Geometric predicate) **disjoint** Que estén fuera de las geometrías de...
- ☐ By comparing to the features from: **Test_ligne**
- ☐ Siempre vamos a hacer una **nueva selección**.
- ☐ Haga **click** en el botón **Run** para hacer esta prueba.

Run

Resultado (selección en amarillo):



- ☐ Seleccionará el punto/elemento que está **fuera de la línea de prueba**. Hay que recordar que las líneas están compuestas de inicio y final (límite/borde) y su interior (todos los puntos entre los límites)

- ☐ **Disjoint: Polígonos/Polígonos**. ¿Cuáles polígonos rojos (poly_test) están fuera del polígono verde (poly_env)? Hagamos la prueba.



Select by Location

Parameters Log

Select features from
POLY_TEST [EPSG:2154]

Where the features (geometric predicate)
☐ intersect ☐ touch
☐ contain ☐ overlap
☒ disjoint ☐ are within
☐ equal ☐ cross

By comparing to the features from
POLY_ENV [EPSG:2154]

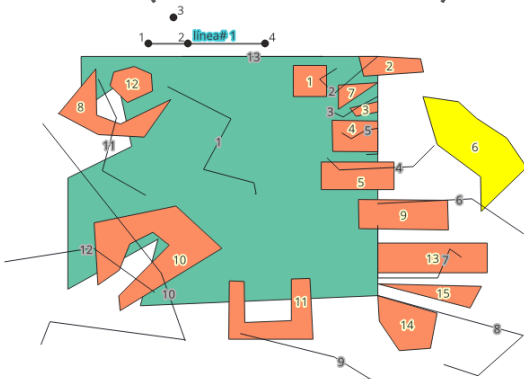
☐ Selected features only

Modify current selection by
creating new selection

- ☐ Seleccionar elementos del geodato de puntos **POLY_TEST**...
- ☐ Geometric predicate: Que estén fuera (**disjoint**) de la geometría de...
- ☐ **By comparing to the features from:** **POLY_ENV**
- ☐ Siempre vamos a hacer una **nueva selección**.
- ☐ Haga **click** en el botón **Run** para hacer esta prueba.

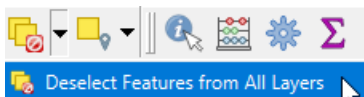
Run

Resultado (selección en amarillo):



- ☐ Seleccionará (en amarillo) el área/elemento #6 que está fuera del polígono (verde) de prueba.

- ☐ **Descarte la selección anterior:**



- ☐ Realice la misma prueba con **disjoint** comparando el geodato de líneas **POLYLIGNES_TEST** con el geodato de área **POLY_ENV**.

Select by Location

Select features from
POLYLIGNES_TEST [EPSG:2154]

Where the features (geometric predicate)
☐ intersect ☐ touch
☐ contain ☐ overlap
☒ disjoint ☐ are within
☐ equal ☐ cross

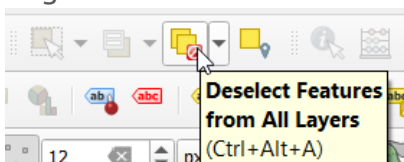
By comparing to the features from
POLY_ENV [EPSG:2154]

☐ Selected features only

Modify current selection by
creating new selection

¿Cuántas geometrías de línea están disjuntas? _____

- ☐ Haga **click** en el botón **Deselect Features** para descartar esta selección.





ST_Touches

Predicado	Geometría: P punto, L polilínea, S polígono	Condiciones
ST_Touches	S/S, L/S, L/L, P/S, P/L	geom(A) toca a geom(B) si: * los límites (contornos) de los objetos tienen al menos un punto en común... y ... * si los interiores de ambos no tienen algún punto en común

Como lee en la tabla,

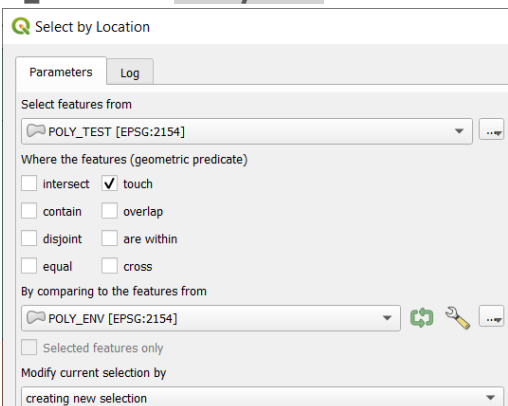
- 1) los **contornos** de las geometrías **deben tener al menos un punto o vértice en común**;
- 2) los **interiores** de las geometrías **no pueden tener ningún punto o vértice en común**.

ST_Touches aplica para comparar/relacionar:

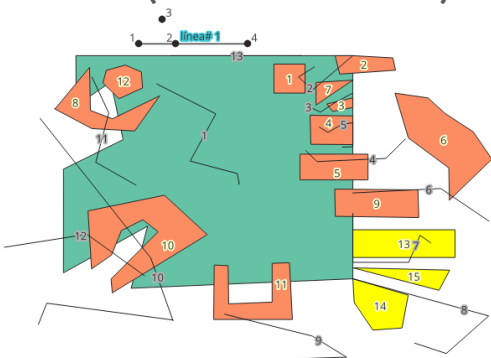
Área/Área, Línea/Área, Línea/Línea, Punto/Área, Punto/Línea

Pasemos a usar **Select by Location** para hacer las pruebas

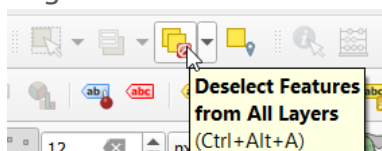
ST_Touches: Área/Área



Resultado (selección en amarillo):



- ☐ Haga **click** en el botón **Deselect Features** para borrar esta selección.



- ☐ Seleccionar elementos del geodato de puntos **POLY_TEST...**
- ☐ Geometric predicate: **touch**
- ☐ **By comparing to the features from: POLY_ENV**
- ☐ Siempre vamos a hacer una **nueva selección**.
- ☐ Haga **click** en el botón **Run** para hacer esta prueba.

1. Según las reglas, los contornos (bordes) de las áreas deben tocarse o al menos tener un vértice en común y además los interiores no pueden tener puntos en común. Seleccionados: 13, 14 y 15.
2. El resto de las áreas/polígonos están completamente dentro, o están parcialmente dentro y hay uno completamente fuera del área de poly_env.
3. Este predicado es útil para encontrar parcelas, municipios adyacentes por ejemplo.



ST_Touches: Líneas/Áreas

Select by Location

Parameters Log

Select features from
POLYLINES_TEST [EPSG:2154]

Where the features (geometric predicate)
☐ intersect ☒ touch
☐ contain ☐ overlap
☐ disjoint ☐ are within
☐ equal ☐ cross

By comparing to the features from
POLY_ENV [EPSG:2154]

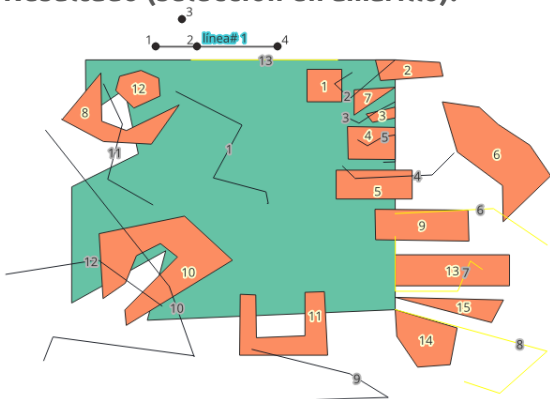
☐ Selected features only

Modify current selection by
creating new selection

- ☐ Seleccionar elementos del geodato de puntos **POLYLINES_TEST** ...
- ☐ Geometric predicate: **touch**
- ☐ By comparing to the features from: **POL_ENV**
- ☐ Siempre vamos a hacer una **nueva selección**.
- ☐ Haga **click** en el botón **Run** para hacer esta prueba.

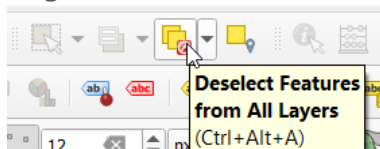
Run

Resultado (selección en amarillo):



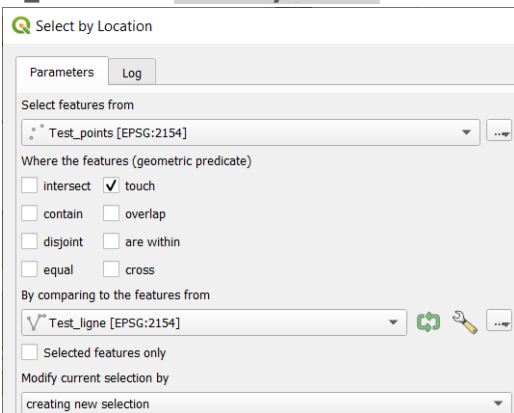
1. **Seleccionó 4 líneas** (6, 7, 8 y 13). Según las reglas, **los contornos (bordes) de las líneas deben tocar o al menos tener un vértice en común con poly_env**. Además **los interiores de las líneas no pueden tener puntos en común con poly_env**.
2. **Ninguna de las líneas seleccionadas tiene vértices en el interior ni están completamente fuera de poly_env**.
3. **Comparten uno más puntos en el borde de poly_env**.
4. **Útil para cotejar conexión entre objetos de infraestructuras. Líneas con líneas, con puntos, con áreas.**

- ☐ Haga **click** en el botón **Deselect Features** para borrar esta selección.

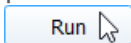




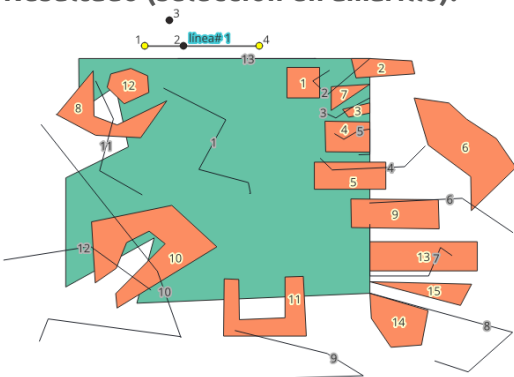
ST_Touches: Puntos/Líneas



- ☐ Seleccionar elementos del geodato de puntos **Test_points...**
- ☐ Geometric predicate: **touch**
- ☐ **By comparing to the features from: Test_ligne**
- ☐ Siempre vamos a hacer una **nueva selección**.
- ☐ Haga **click** en el botón **Run** para hacer esta prueba.

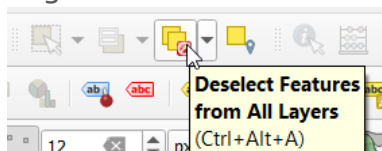


Resultado (selección en amarillo):

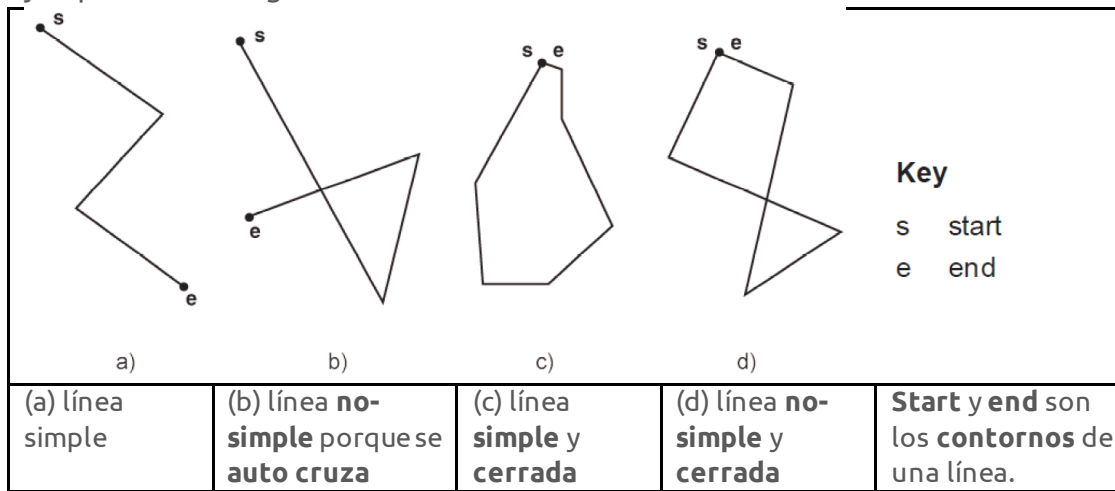


1. Seleccionó 2 puntos: 1 y 4. El punto 2 está en el interior de la línea. El punto 3 está fuera de la línea.
2. El predicado **Touch devolverá cierto si los puntos están en el contorno de la línea**. Los contornos de la línea son los puntos de inicio y final. En el caso de una MultiLine los puntos de contorno están en líneas componentes que sean impares.
3. Útil para cotejar conexión entre objetos de infraestructuras. Líneas con líneas, con puntos, con áreas.

- ☐ Haga **click** en el botón **Deselect Features** para borrar esta selección.



Ejemplos: Linestrings:

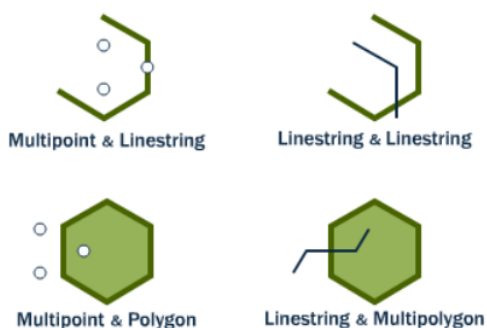




ST_CROSSES

Predicado	Geometría: MP multipoint, L polilínea, S polígono	Condiciones
ST_Crosses	MP/S, MP/L, L/S, L/L	geom(A) cruza a geom(B) si: * los interiores de los objetos tienen al menos un punto en común... pero ... * no todos en común... y ... * si la dimensión de la intersección de los interiores es inferior a la dimensión máxima de los objetos geom(A) y geom(B) (no aplica a P/P, S/S, pero aplica a MultiPoint)

Cross

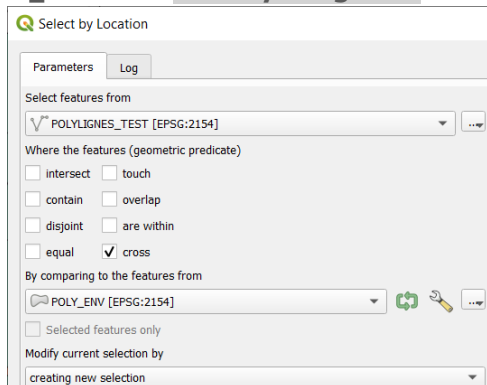


En todos estos casos, hay cruce entre las geometrías comparadas.

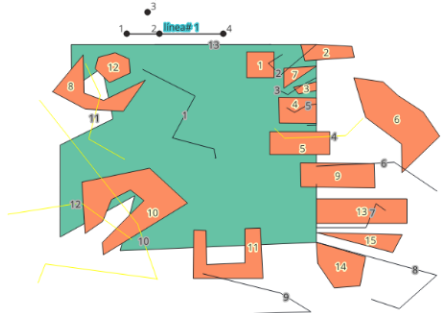
- Multipoint/Line
- Line/Line
- Multipoint/Polígono
- Línea/Multipolígono

Fuente: https://postgis.net/workshops/postgis-intro/spatial_relationships.html

ST_Crosses: Líneas/Polígonos



Resultado (selección en amarillo):



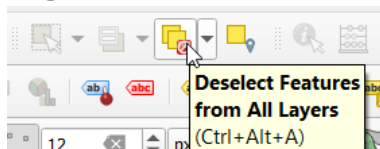
- ☐ Seleccionar elementos del geodato de puntos **POLYLINES_TEST...**
- ☐ Geometric predicate: **cross**
- ☐ **By comparing to the features from: POLY_ENV**
- ☐ Siempre vamos a hacer una **nueva selección**.
- ☐ Haga **click** en el botón **Run** para hacer esta prueba.

Run

1. Las líneas seleccionadas (en amarillo) son la 4, 10, 11 y la 12.
2. El predicado **cross** devolverá cierto si los **interiores de las líneas** tienen uno o más puntos (pero no todos) en el interior del polígono.



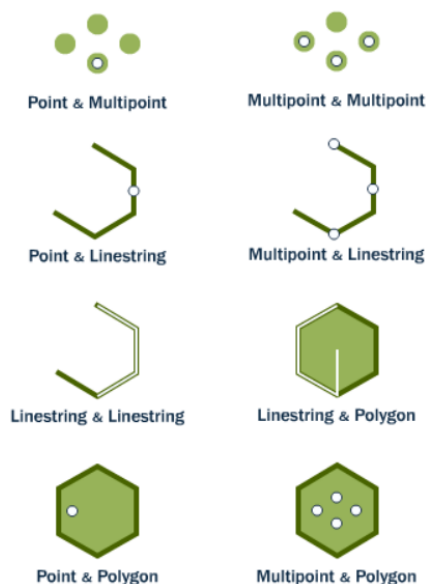
- ☐ Haga **click** en el botón **Deselect Features** para borrar esta selección.



ST_Within/ST_CONTAINS

Predicado	Geometría: MP multipoint, L polilínea, S polígono	Condiciones
ST_Within	Todas	geom(A) está dentro de geom(B) si: * todo punto de geom(A) está dentro de geom(B) ... y ... * si los interiores tienen al menos algún punto en común ... * ningún punto de geom(A) está en el exterior de geom(B) * (inverso de Contains)
ST_Contains	Todas	geom(A) contiene a geom(B) si: * todo punto de geom(B) es un punto de geom(A) ... y ... * si los interiores tienen al menos algún punto en común * ningún punto de geom(B) está en el exterior de A * (inverso de Within)

Within/Contains



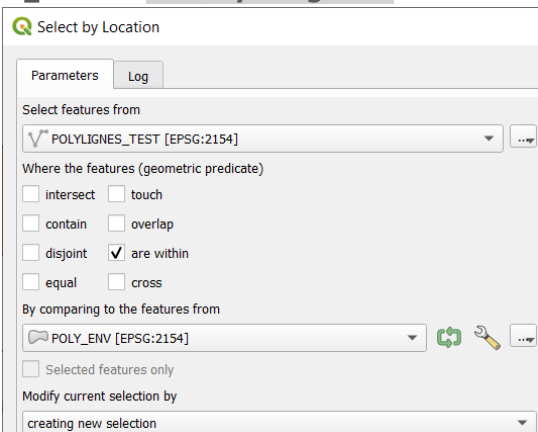
En todos estos casos, hay **continencia** entre las geometrías comparadas.

- Punto/Multipunto
- Multipunto/Multipunto
- Punto/Línea
- Multipunto/Línea
- Línea/Línea
- Punto/Área
- Multipunto/Área

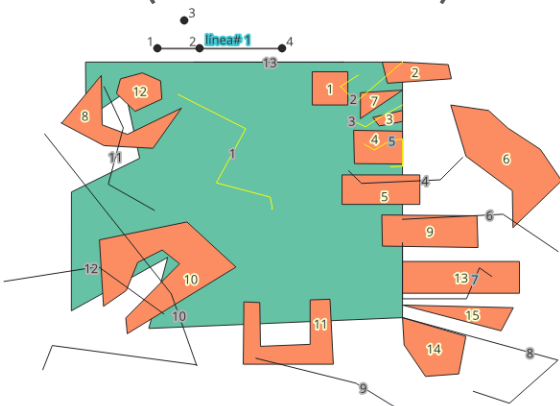
Fuente: https://postgis.net/workshops/postgis-intro/spatial_relationships.html



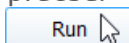
ST_Within: Líneas/Polígonos



Resultado (selección en amarillo):

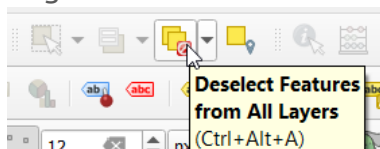


- ☐ Seleccionar elementos del geodato de puntos **POLYLINES_TEST...**
- ☐ Geometric predicate: **are within**
- ☐ **By comparing to the features from:** **POLY_ENV**
- ☐ Siempre vamos a hacer una **nueva selección.**
- ☐ Haga **click** en el botón **Run** para hacer esta prueba.



1. Las líneas seleccionadas (en amarillo) son la 1, 2, 3 y la 5.
2. El predicado **are within** devolverá cierto y seleccionará cuando todos los puntos que componen los elementos están dentro de la otra geometría.

- ☐ Haga **click** en el botón **Deselect Features** para borrar esta selección.





ST_Within: Polígonos/Polígonos

Select by Location

Parameters Log

Select features from

POLY_TEST [EPSG:2154]

Where the features (geometric predicate)

☐ intersect ☐ touch

☐ contain ☐ overlap

☐ disjoint ☒ are within

☐ equal ☐ cross

By comparing to the features from

POLY_ENV [EPSG:2154]

☐ Selected features only

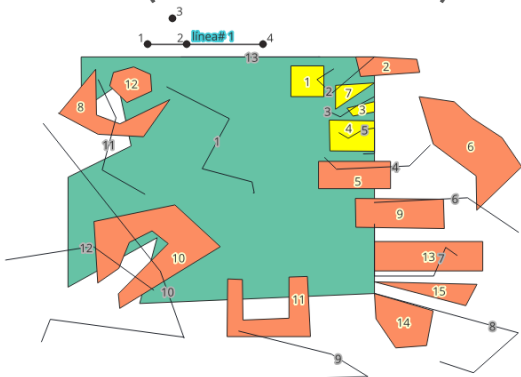
Modify current selection by

creating new selection

- ☐ Seleccionar elementos del geodato de puntos **POLY_TEST...**
- ☐ Geometric predicate: **are within**
- ☐ By comparing to the features from: **POLY_ENV**
- ☐ Siempre vamos a hacer una **nueva selección**.
- ☐ Haga **click** en el botón **Run** para hacer esta prueba.

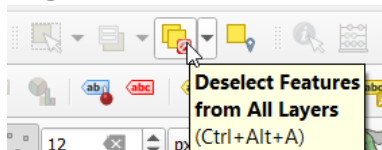
Run

Resultado (selección en amarillo):



- Los polígonos seleccionados (en amarillo) son el 1, 3, 4 y 7.
- El predicado **are within** devolverá cierto y seleccionará cuando todos los puntos que componen los elementos están dentro de la otra geometría (incluyendo el borde).

- ☐ Haga **click** en el botón **Deselect Features** para borrar esta selección.



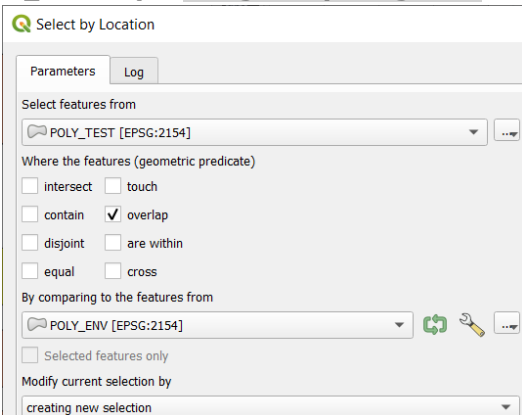
ST_OVERLAPS

Predicado	Geometría: MP multipoint, L polilínea, S polígono	Condiciones
ST_Overlaps	S/S,L/L,P/P	geom(A) solapa a geom(B) si a la vez: * geom(A) y geom(B) tienen la misma dimensión (no aplica a P/L, P/S, L/S) ... * geom(A) y geom(B) tienen puntos en común... pero no todos ... * La <u>intersección</u> de los interiores de geom(A) y geom(B) tiene la misma <u>dimensión</u> que geom(A) y geom(B)





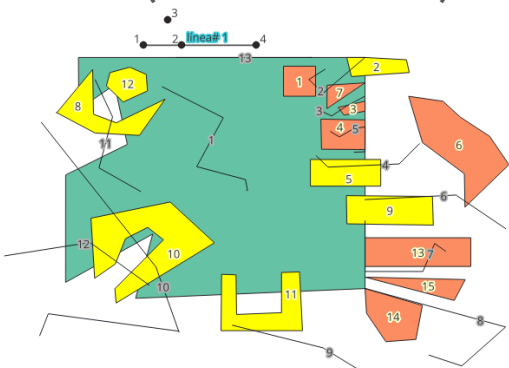
ST_Overlaps: Polígonos/Polígonos



- ☐ Seleccionar elementos del geodato de puntos **POLY_TEST...**
- ☐ Geometric predicate: **overlap**
- ☐ **By comparing to the features from:** **POLY_ENV**
- ☐ Siempre vamos a hacer una **nueva selección**.
- ☐ Haga **click** en el botón **Run** para hacer esta prueba.

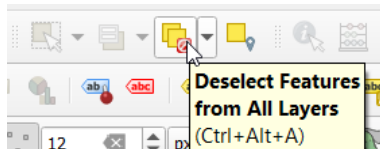
Run

Resultado (selección en amarillo):



1. Los polígonos seleccionados (en amarillo) son el 2, 5, 8, 9, 10, 11, y 12.
2. El predicado **overlap** devolverá cierto y seleccionará cuando los interiores de ambas geometrías tengan puntos en común pero que a la vez no estén completamente dentro de poly_env.

- ☐ Haga **click** en el botón **Deselect Features** para borrar esta selección.

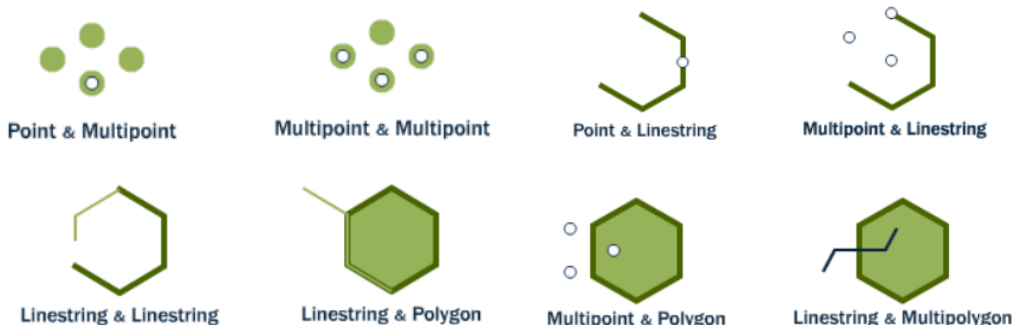




ST_INTERSECTS

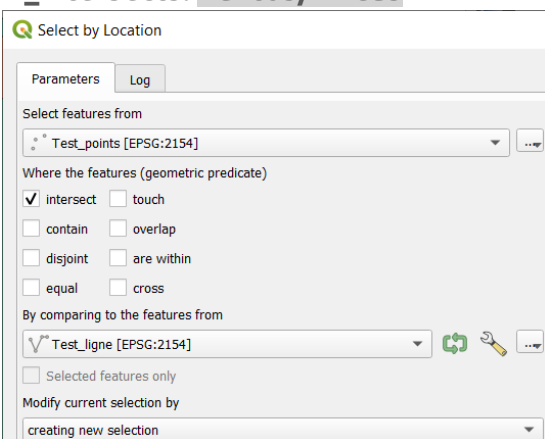
Predicado	Geometría: MP multipoint, L polilínea, S polígono	Condiciones
ST_Intersects	Todas	geom(A) interseca a geom(B) si: * geom(A) y geom(B) tienen al menos un punto en común en el interior o en el límite. * (Inverso de <i>Disjoint</i>)

ST_Intersects:



Fuente: https://postgis.net/workshops/postgis-intro/spatial_relationships.html

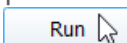
ST_Intersects: Puntos/Líneas



Resultado (selección en amarillo):

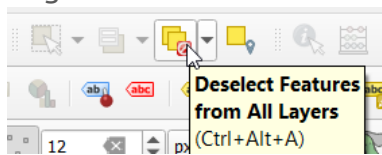


- ☐ Seleccionar elementos del geodato de puntos **Test_points...**
- ☐ Geometric predicate: **intersect**
- ☐ **By comparing to the features from: Test_ligne**
- ☐ Siempre vamos a hacer una **nueva selección.**
- ☐ Haga **click** en el botón **Run** para hacer esta prueba.



- Los puntos seleccionados (en amarillo) son el 1, 2, y 4. El punto 3 está fuera de la línea.
- El predicado **intersect devolverá cierto y seleccionará cuando las geometrías de ambos geodatos tengan al menos un punto en común tanto en el interior, como en el límite o borde.**

- ☐ Haga **click** en el botón **Deselect Features** para borrar esta selección.





ST_Intersects: Líneas/Polígonos

Select by Location

Parameters Log

Select features from

POLYLINES_TEST [EPSG:2154]

Where the features (geometric predicate)

☒ intersect ☐ touch

☐ contain ☐ overlap

☐ disjoint ☐ are within

☐ equal ☐ cross

By comparing to the features from

POLY_ENV [EPSG:2154]

☐ Selected features only

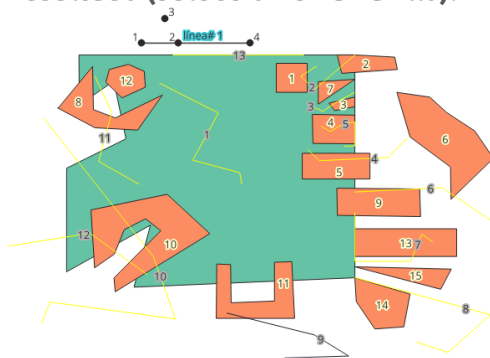
Modify current selection by

creating new selection

- ☐ Seleccionar elementos del geodato de puntos **POLYLINES_TEST...**
- ☐ Geometric predicate: **intersect**
- ☐ **By comparing to the features from: POLY_ENV**
- ☐ Siempre vamos a hacer una **nueva selección.**
- ☐ Haga **click** en el botón **Run** para hacer esta prueba.

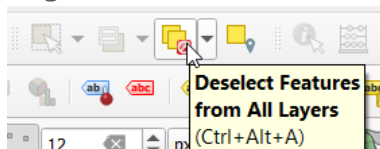
Run

Resultado (selección en amarillo):



1. Las líneas seleccionadas (en amarillo) son 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12 y 13. La línea 9 está fuera de poly_env.
2. El predicado **intersect devolverá cierto y seleccionará** cuando las geometrías de ambos geodatos tengan al menos un punto en común tanto en el interior, como en el límite o borde.

- ☐ Haga **click** en el botón **Deselect Features** para borrar esta selección.



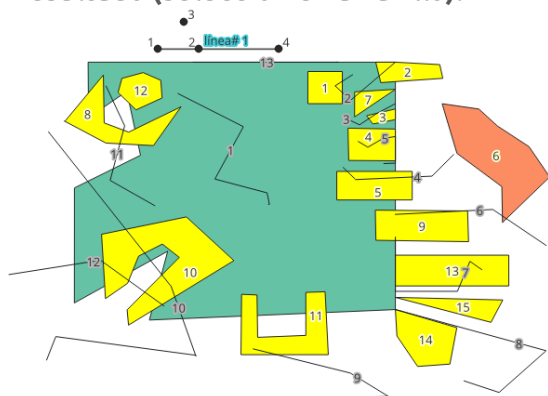


ST_Intersects: Polígonos/Polígonos

- ☐ Seleccionar elementos del geodato de puntos **POLY_TEST...**
- ☐ Geometric predicate: **intersect**
- ☐ **By comparing to the features from: POLY_ENV**
- ☐ Siempre vamos a hacer una **nueva selección.**
- ☐ Haga **click** en el botón **Run** para hacer esta prueba.

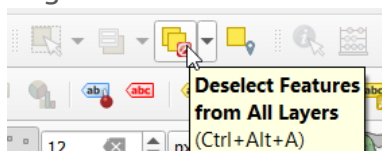
Run

Resultado (selección en amarillo):



1. **Todos** los polígonos del geodato poly_test fueron seleccionados **excepto el número 6** porque **está fuera** de poly_env.
2. El predicado **intersect** devolverá cierto y seleccionará cuando las geometrías de ambos geodatos tengan al menos un punto/vértice en común tanto en el interior, como en el límite o borde.

- ☐ Haga **click** en el botón **Deselect Features** para borrar esta selección.



- ☐ **Cierre** esta sesión de QGIS. No lo usaremos por ahora.

ST_COVERS & ST_COVEREDBY

Estas son dos **funciones propias de PostGIS** entre otras como **ST_DWithin**.

ST_Covers	Todas	geom(A) cubre a geom(B) si: * ningún punto de geom(B) está en el exterior de A * todo punto de geom(B) es un punto de geom(A) Compárese con <i>Contains</i>
ST_CoveredBy	Todas	geom(A) está cubierto por geom(B) si: * ningún punto de geom(A) está en el exterior de geom(B) * todo punto de geom(A) es un punto de geom(B) Compárese con <i>Within</i>

Según la documentación de PostGIS, ellos recomiendan usar **ST_Covers** en lugar de ST_Contains. Su definición es más simple y no tiene la particularidad de que las geometrías no contienen a sus límites. **Estas funciones** al igual que ST_DWithin **no están disponibles en QGIS**.



PostGIS, 3.x

Los predicados geométricos se explican en el [apartado A](#) de este tutorial. Como se mencionó, son 98 situaciones con los tres tipos básicos de geometrías.

Fuente: https://www.geoinformations.developpement-durable.gouv.fr/fichier/pdf/Predicats_OGC_V3_cle119417.pdf?arg=177828548&cle=baf65cf1b040772ae1d97f164cf9ca67f6dbb698&file=pdf%2FPredicats_OGC_V3_cle119417.pdf

ST_DWITHIN

Esta función/predicado seleccionará elementos geométricos que compartan puntos/vértices en común entre geometrías basado en una distancia de búsqueda. Como se mencionó antes, **funciona como ST_Intersects, pero con un umbral de tolerancia** (distancia).



Point & Point (True)



Point & Point (False)



Polygon & Point (True)



Polygon & Point (False)

Fuente: https://postgis.net/workshops/postgis-intro/spatial_relationships.html

ST_DWithin no es un predicado del OGC. Es una implementación de PostGIS.

Ejemplo:

ST_DWithin para mostrar distancias entre escuelas y carreteras estatales.

Regrese a la interfaz de DBeaver.

- En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query para seleccionar las **escuelas** en Villalba que estén ubicadas hasta **300 metros** de distancia de vías del **sistema estatal de carreteras**. **Incluya la distancia** mediante la función **ST_Distance**.

```
1 select e.escuela, c.route_id, --traer escuela, número de ruta
2 st_distance(e.geom,c.geom) as distancia --añade la distancia
3 -- entre escuela y carretera
4 from pr_geodata.g33_dotacional_educacion_escuelas_2021_villalba e,
5 pr_geodata.g35_viales_carreteras_estatales_segmentadas_agosto_2021 c
6 where st_dwithin(c.geom,e.geom,300) -- función st_DWithin a 300 metros
7 order by c.route_id, distancia; -- ordenar por núm carretera y distancia
```

g33_dotacional_educacion_escuelas_2021_villalba(+) 1 X

select e.escuela, c.route_id, st_distance | Enter a SQL expression to filter results (use Ctrl+Space)

	escuela	route_id	distancia
1	DANIEL SERRANO RIVERA	PR-149	82.7400391886
2	LYSANDER BORRERO TERRY	PR-149	103.2738510828
3	CRISTINA (AMADA) MARTINEZ MARTINEZ	PR-149	257.3467495876
4	SU HATILLO SILVIA TORRES TORRES	PR-150	101.274528967
5	CROEV: ESPECIALIZADA EN CIENCIAS Y MATEMATICAS	PR-150	229.2012484341
6	LYSANDER BORRERO TERRY	PR-150	251.6656291848

Refresh Save Cancel Export data 200 19

19 row(s) fetched - 0.017s (0.001s fetch), on 2025-06-03 at 13:55:36

Le devolverá 19 filas. Note las funciones **ST_Distance(geom, geom)** y **ST_DWithin(geom, geom)**. Las escuelas se pueden repetir.



QUERIES CON GEOMETRÍAS:

ST_Touches

Realice una sentencia SQL espacial en la cual se **haga una lista de municipios y cuántos municipios adyacentes tiene cada uno**. El predicado topológico a usar es [ST_Touches](#), el cual está explicado en el [Apartado A-2](#). **ST_Touches** se debe usar, ya que seleccionará las **geometrías que comparten bordes, pero no tienen puntos en el interior de otra geometría**. Agregue los datos (**GROUP BY**) por el campo de nombres de municipios. Ordene los datos, **ORDER BY** usando la columna de conteos (posición #2) en el orden de columnas del query.

- En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query:

```
1 select m1.municipio, --trae columna municipio del geodato m1
2 count(*) as cuántos_adyacentes --conteo de filas
3 from pr_geodata.g03_legales_municipios_2015 m1, --tabla munics as m1
4 pr_geodata.g03_legales_municipios_2015 m2 --tabla munics as m2
5 where st_touches(m1.geom,m2.geom) --predicado solo comparten bordes
6 group by m1.municipio -- agrupa conteo adyacentes por m1.municipio
7 order by 2 desc; --ordena por col#2 (cuántos_adyacentes)
```

- Para encontrar **municipios adyacentes copiamos el mismo geodato y le asignamos otro alias (m2)**.
- **ST_Touches** es el predicado que necesitamos: compartir solo bordes.
- **ORDER BY 2** indica que queremos hacer el **ordenamiento por la columna #2** y de forma descendente para que nos devuelva primero los municipios con **mayor cantidad de municipios adyacentes**.

El conteo de filas es 76. **¿Por qué no aparecen los municipios de Vieques ni de Culebra?**

¿Y si queremos saber cuáles son los nombres de cada municipio adyacente a cada municipio?

La función **string_agg(columna, 'patrón')** se usará para acomodar una lista de ítems de una columna o expresión dentro de una sola fila.

- Por ejemplo, si queremos acomodar una lista de municipios, separados por una coma: En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query:

```
select m1.municipio, --trae columna municipio del geodato m1
count(*) as adyacentes, --contar municipios
string_agg(m2.municipio, --crear lista horizontal
', ' --separada por coma
order by m2.municipio) --ordena lista horizontal ascendente
as municipios_adyacentes --alias municipios_adyacentes
from --tablas de municipios alias m1 y copia, alias m2
pr_geodata.g03_legales_municipios_2015 m1,
pr_geodata.g03_legales_municipios_2015 m2
where --test: que las geometrías toquen solo bordes
ST_Touches(m1.geom,m2.geom)
group by --agrega conteos por m1.municipio
m1.municipio
order by 2 desc; --ordena por la col #2
```




PostGIS, 3.x

```
1 select m1.municipio, --trae columna municipio del geodato m1
2 count(*) as adyacentes, --contar municipios
3 string_agg(m2.municipio, --crear lista horizontal
4           ', ' --separada por coma
5           order by m2.municipio) --ordena lista horizontal ascendente
6 as municipios_adyacentes --alias municipios_adyacentes
7 from --tabla de municipios alias m1 y copia, alias m2
8 pr_geodata.g03_legales_municipios_2015 m1,
9 pr_geodata.g03_legales_municipios_2015 m2
10 where --test: que las geometrías toquen solo bordes
11 st_touches(m1.geom,m2.geom)
12 group by --agrega conteos por m1.municipio
13 m1.municipio
14 order by 2 desc; --ordena por la col #2
```

g03_legales_municipios_2015 1 X

select m1.municipio, count(*) as a | Enter a SQL expression to filter results (use Ctrl+Space)

Grid	AZ municipio	123 adyacentes	AZ municipios_adyacentes
1	Orocovis	8	Barranquitas, Ciales, Coamo, Corozal, Jayuya, Juana Díaz, Morovis, Villalba
2	Lares	8	Adjuntas, Camuy, Hatillo, Las Marías, Maricao, San Sebastián, Utuado, Yauco
3	Cayey	7	Aibonito, Caguas, Cidra, Guayama, Patillas, Salinas, San Lorenzo
4	Barranquitas	7	Aibonito, Cidra, Coamo, Comerío, Corozal, Naranjito, Orocovis
5	Ciales	7	Arecibo, Florida, Jayuya, Manatí, Morovis, Orocovis, Utuado
6	Caguas	7	Aguas Buenas, Cayey, Cidra, Gurabo, San Juan, San Lorenzo, Trujillo Alto
7	Coamo	7	Aibonito, Barranquitas, Juana Díaz, Orocovis, Salinas, Santa Isabel, Villalba
8	Bayamón	7	Aguas Buenas, Cataño, Comerío, Guaynabo, Naranjito, Toa Alta, Toa Baja

Refresh Save Cancel Export data 200

76 row(s) fetched - 3.536s, on 2025-07-07 at 15:33:06

La **diferencia** es **insertar** la función `string_agg(m2.municipio, ', ')` luego de añadir una coma después de la nueva columna cuántos_adyacentes.

Al igual que el query anterior, aparecerán solo 76 filas.



PostGIS, 3.x

¿Cuáles son los barrios con mayor número de barrios adyacentes?

Usar la función [ST_Touches\(geom,geom\)](#) y el mismo geodato de barrios con dos alias **b1** y **b2** para compararlos.

- En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query:

Esté **atento a las comillas, espacios** y la función `concat()`.

```
select concat('Bo. ',b1.barrio,', ', b1.municipio) as barrio_munic,
count(b2.gid) as adyacentes, --contar barrios
string_agg( --función para hacer lista horizontal
    concat( --función concat() para concatenar...
        'Bo. ', --Bo.
        b2.barrio, --barrio, copia geodato barrios
        ', ', --', ', separador entre barrio y municipio
        b2.municipio), --municipio, copia geodato barrios
        '; ' --separador de lista horizontal es '; '
    order by --ordena lista horizontal
        b2.barrio, --barrio ascendente
        b2.municipio --municipio ascendente
    ) --termina función string_agg
as barrios_municipios_adyacentes --alias municipios_adyacentes
from --tabla de municipios alias m1 y copia, alias m2
pr_geodata.g03_legales_barrios_2015 b1,
pr_geodata.g03_legales_barrios_2015 b2
where --test: que las geometrías toquen solo bordes
ST_Touches(b1.geom,b2.geom)
group by --agrega conteos por m1.municipio
barrio_munic
order by 2 desc; --ordena por la col #2 "adyacentes"
```

select concat('Bo. ',b1.barrio,', ', b1.municipio) as barrio_munic, count(b2.gid) as adyacentes, string_agg(concat('Bo. ', b2.barrio, ', ', b2.municipio), '; ' order by b2.barrio, b2.municipio) as barrios_municipios_adyacentes			
Grid	AZ barrio_munic	123 adyacentes	AZ barrios_municipios_adyacentes
1	Bo. Maricao Afuera, Maricao	13	Bo. Barrio Pueblo, Maricao; Bo. Bucarabones, Las Marías; Bo. Bucarabon
2	Bo. Yaurel, Arroyo	11	Bo. Algarrobo, Guayama; Bo. Ancones, Arroyo; Bo. Cacao Alto, Patillas; E
3	Bo. Frailes, Guaynabo	11	Bo. Barrio Pueblo, Guaynabo; Bo. Caimito, San Juan; Bo. Camarones, Gu
4	Bo. Río Blanco, Naguabo	11	Bo. Antón Ruiz, Humacao; Bo. Duque, Naguabo; Bo. El Río, Las Piedras;
5	Bo. Toro Negro, Ciales	11	Bo. Ala de la Piedra, Orocovis; Bo. Cacaos, Orocovis; Bo. Cialitos, Ciales;
6	Bo. Sumidero, Aguas Buenas	10	Bo. Bairoa, Aguas Buenas; Bo. Barrio Pueblo, Aguas Buenas; Bo. Bayamó
7	Bo. Anones, Las Marías	10	Bo. Alto Sano, Las Marías; Bo. Alto Sano, San Sebastián; Bo. Casey Arrib

Refresh Save Cancel

 Export data 200

901 row(s) fetched - 6.693s (0.004s fetch), on 2025-07-07 at 16:32:27

- En la función `string_agg()` insertamos la función `concat()` con una cadena de caracteres, `'Bo. '` y concatenamos el campo `b2.barrio`, luego insertamos una coma `,` y luego concatenamos el campo `b2.municipio`, y finalmente añadimos punto y coma y espacio `;` para separar los valores.



Pregunta: Si **el total de barrios es 902**, ¿por qué devuelve **901**? ¿Cuál barrio no tiene **barrio(s) adyacentes**? Sabemos que Isla de Mona e islote Monito componen un barrio de Mayagüez, pero ¿cómo hacer el query para que devuelva este barrio?

- En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query:

```
select concat('Bo. ',b1.barrio,', ',b1.municipio) as barrio_munic
--traer y concatenar columnas municipio y barrio
--del geodato b1 as barrio_munic

from
pr_geodata.g03_legales_barrios_2015 as b1
--tabla barrios
where not exists( --devuelve fila(s) que quede(n) fuera
--de esta selección
select
1 --replicar primer campo
from
pr_geodata.g03_legales_barrios_2015 as b2
--copia tabla barrios
where ST_Touches(b1.geom,b2.geom)
--test: las geometrías comparten bordes?
and b1.gid<>b2.gid
--descarta comparar con el mismo barrio
); --cerrar cláusula where not exists
```

Grid		A-Z barrio_munic
	1	Bo. Isla de Mona e Islote Monito, Mayagüez

Este query:

- Usa un **sub-select o query anidado** para que devuelva la fila que no cumple con el predicado ST_Touches: **where not exists**(select expresión from tabla, where **ST_Touches**(b1.geom,b2.geom).
- Luego añadir la comparación **b1.geom <> b2.geom** para descartar primero .
- Usó la función **CONCAT()** para unir cadenas de caracteres

Esta es la fila que cumple con la condición: barrio Isla de Mona e Islote Monito.



VISUALIZAR LAS GEOMETRÍAS EN POSTGIS

Usemos un query parecido al de búsqueda de barrios adyacentes. **Ya vimos que el barrio Maricao Afuera del Municipio de Maricao** comparte límites con **13** barrios. Para ver estos vecinos sin salir de PostGIS, haremos un query que incluya la geometría.

- En la pestaña **gisdb_lab> Script 1**, escriba y ejecute el siguiente query.

```
select
    concat('Bo. ', b2.barrio,', ', b2.municipio)as barrio_de_referencia,
    concat('Bo. ', b1.barrio,', ', b1.municipio)as barrios_adyacentes,
    b1.geom --geometría barrios, versión 1

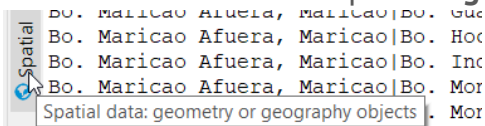
from
    pr_geodata.g03_legales_barrios_2015 b1,
    pr_geodata.g03_legales_barrios_2015 b2 --copia

where
    b2.barrio='Maricao Afuera' --el barrio que nos interesa...
    and ST_Touches(b1.geom,b2.geom)

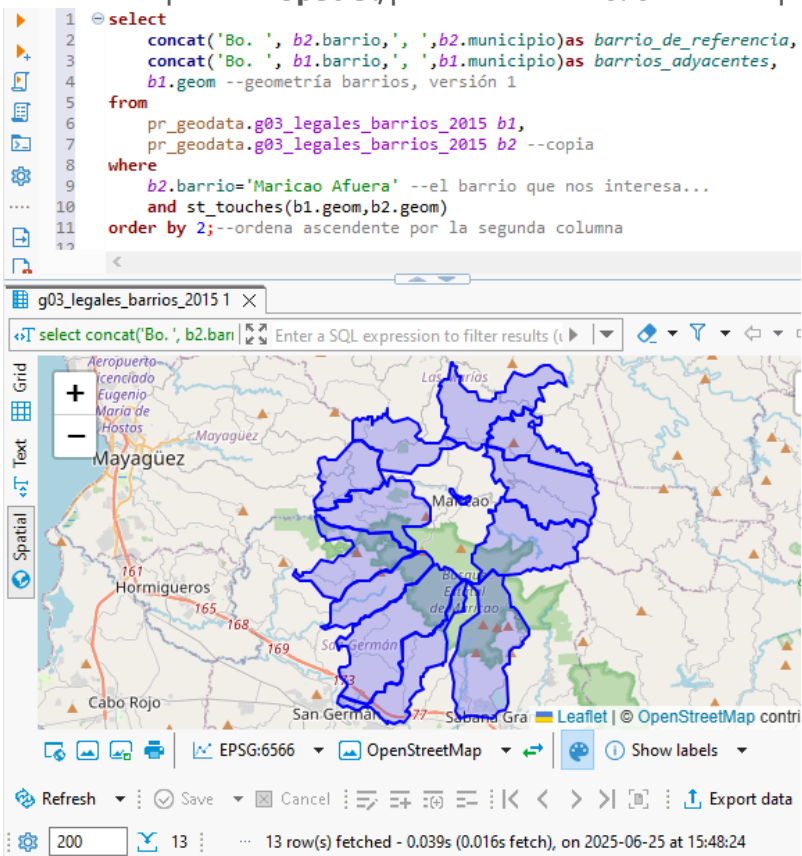
order by 2;--ordena ascendente por la segunda columna
```

Al traer la columna **b1.geom**, convertimos el resultado del query en un geodato temporal que podremos visualizar si usamos la pestaña **Spatial**.

- Luego que devuelva las 13 filas, haga **click** en la pestaña **Spatial** al margen izquierdo del listado devuelto en la pestaña **g03_legales_barrios_2015 1**



- Al activar la pestaña **Spatial**, podemos hacer **click** en cualquier elemento geométrico.





MOSTRAR ÁREA, PERÍMETRO Y TIPO DE GEOMETRÍA

Las propiedades de las geometrías son guardadas internamente. No son campos calculados como aparecería en los shapefiles, por ejemplo.

Muestre una lista con los **nombres de barrios** del Municipio de Villalba, el **área**, así como sus **perímetros** y **tipo de geometría**. Ordene los resultados por área de manera descendente. En la cláusula **ORDER BY** puede **usar un número de referencia de la columna** en vez del nombre de la columna.

- ☐ En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query.

The screenshot shows the QGIS interface. The top window is titled "Script-1" and contains the following SQL query:

```
select barrio, --traer barrio
st_area(b.geom)/1000000 as area_sqkm, --traer área y convertir a sq_kms
st_perimeter(b.geom)/1000 as perim_km, -- traer perímetro y conv a kms
st_geometrytype(b.geom) as tipo_geometria -- trae el tipo de geometria
from g03_legales_barrios_2015_villalba as b --de la tabla barrios_2015_villalba
order by 2 desc; -- ordena usando la columna #2 forma descendente
```

The bottom window is titled "g03_legales_barrios_2015_villalba 1" and shows the results of the query in a table. The table has 5 columns: "barrio", "area_sqkm", "perim_km", and "tipo_geometria". The results are ordered by area_sqkm in descending order.

	barrio	area_sqkm	perim_km	tipo_geometria
1	Caonillas Arriba	20.9428427248	26.12686903	ST_MultiPolygon
2	Villalba Arriba	17.6469141407	19.6906081145	ST_MultiPolygon
3	Caonillas Abajo	16.4577126574	18.673760813	ST_MultiPolygon
4	Hato Puerco Arriba	13.3730023937	31.5976338459	ST_MultiPolygon
5	Vacas	12.1964044741	16.405680526	ST_MultiPolygon
6	Hato Puerco Abajo	7.8200882754	15.28077986	ST_MultiPolygon
7	Villalba Abajo	7.0668390792	12.7572214562	ST_MultiPolygon
8	Barrio Pueblo	0.3723566376	3.6007782863	ST_MultiPolygon

¿Puede haber distintos tipos de geometrías en una misma tabla geoespacial en PostGIS?

La respuesta es sí. Un campo tipo geometría o geografía puede estar compuesto por todas las variantes de POINT/POLYLINE/POLYGON y además **GeometryCollection**. Al momento, los SIG de interfaz gráfica tipo desktop, no dan soporte a las GeometryCollection y solo trabajan con un tipo de geometría a la vez.

Más adelante haremos overlay de geodatos. **¿Es posible que el cruce/overlay de dos geodatos nos devuelva distintos tipos de geometrías (GeometryCollection)?**

La respuesta es sí. Eso puede ser un problema que atenderemos más adelante.

ST_LENGTH

¿Cuál es el kilometraje total de las **calles municipales** del Municipio de Villalba?

Antes, vamos a hacer una búsqueda de **cuáles son los códigos que usaremos como criterio de selección** de lo que son las **vías municipales**, según los **códigos censales** para vías (2006). Este archivo de viales se originó en 2006, pero se añadieron vías adicionales usando la imagen satelital de Google Maps en diciembre de 2023.

Debemos saber primero los códigos CFCC (Census Feature Class Codes, 2016).

- ☐ En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query:



```
*gisdb_lab> Script-1 X
select distinct cfcc --traer valores únicos del campo cfcc
from g35_viales_mapa_base_tiger_rds_2006se_villalba --de esta tabla
order by 1; --ordena ascendente por la primera columna/campo
```

```
g35_viales_mapa_base_tiger_rds_2006se_villalba 1 X
select distinct cfcc from g35_viales_mapa_base_tiger_rds_2006se_villalba
```

cfcc
A31
A35
A41
A51
A63

Refresh Save Cancel Export data

200 5 5 row(s) fetched - 0.002s, on 2024-02-21 at 08:52:13

SELECT DISTINCT para devolver los códigos cfcc sin repeticiones.

Debe darnos 5 filas. [Según uno de los artículos de knowledge base de Esri](#), los códigos censales de vías que necesitamos son los tipo **A41** y **A51**. Los demás **A31**, **A35** y **A63** están asociados con carreteras del sistema estatal.

**** Estamos simplificando este asunto****
Puede haber vías municipales que son mantenidas por la Autoridad de Carreteras y Transportación (ACT). En el caso de Villalba, en la versión de carreteras estatales 2021, no hay vías municipales mantenidas por la ACT.

Con esa información podemos redactar la sentencia (query) para contestar la pregunta: **Dar un estimado de kilometraje en vías municipales, mostrando el código censal cfcc y la suma de kilómetros para cada clase cfcc, para el Municipio de Villalba.** En este caso, vamos a sumar la longitud de las geometrías con la función **sum(ST_Length(geom))/1000** para que lo convierta en **kilómetros**, ya que la unidad de medida del geodato es el **metro**. Solo vamos a sumar las vías municipales, así que **en la cláusula WHERE** usaremos **cfcc IN ('A41','A51')**. En la cláusula **GROUP BY** vamos a **agrupar/sumar por categoría cfcc**.

☐ En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query:

```
*gisdb_lab> Script-1 X
select cfcc, --traer valores del campo cfcc
sum(st_length(geom)/1000) as kms --calcula sumatoria de longitudes geométricas en kms
from g35_viales_mapa_base_tiger_rds_2006se_villalba --de esta tabla
where cfcc in ('A41','A51') --condición: solo valores cfcc=A41,A51
group by cfcc --agrupar sumas de valores usando el campo cfcc
order by 1; --ordena ascendente por la primera columna/campo
```

```
g35_viales_mapa_base_tiger_rds_2006se_villalba 1 X
select cfcc, sum(st_length(geom)/1000) as kms from g35_viales_mapa_base_tiger_rds_2006se_villalba
```

cfcc	kms
A41	236.4427943113927
A51	16.965397693137174

Refresh Save Cancel Export data

200 2 2 row(s) fetched - 0.003s, on 2024-02-21 at 09:05:04

En este query utilizamos:

- **GROUP BY cfcc** para sumar por categoría.
- **WHERE cfcc IN ('A41', 'A51')** para seleccionar solamente las **vías municipales**.

Las vías **cfcc=A51** son por lo regular caminos privados sin asfaltar. Pueden variar en longitud.

En fin, hay alrededor de **236 kilómetros en vías municipales (A41)** en este municipio.

ROUND()

Es una función de uso general para redondear números con decimales. Usaremos el query anterior para aplicar la función **round()** al cómputo de sumatoria de longitudes de las geometrías de vías en Villalba. El valor **ST_Length** devuelto es del [tipo de dato numérico double precision](#), dato numérico *floating point*, no exacto.

Según la [documentación de funciones matemáticas en PostgreSQL](#), la función **round()**:

round (numeric) → numeric

round (double precision) → double precision

Puede recibir un dato tipo *numeric* y **devuelve el valor entero más próximo**.



PostGIS, 3.x

Rounds to the nearest integer. For numeric, ties are broken by rounding away from zero. For double precision, the tie-breaking behavior is platform dependent, but "round to nearest even" is the most common rule. <code>round(42.4) → 42</code>	Puede recibir un tipo de dato <i>double precision</i> y devuelve el valor entero más próximo. En esta variante de <code>round()</code> no se usa el argumento <i>s integer</i> para añadir lugares decimales.
round (v numeric, s integer) → numeric Rounds v to s decimal places. Ties are broken by rounding away from zero. <code>round(42.4382, 2) → 42.44</code> <code>round(1234.56, -1) → 1230</code>	Si usamos <code>round(v numeric, s integer)</code> el valor de entrada debe ser tipo numérico para poder aplicar el número entero que dará la cantidad de lugares decimales. Esta variante de round() debe recibir como entrada valores del tipo de dato <i>numeric</i>.

Por lo tanto, para aplicar `round` a la suma de longitudes extraídas de las geometrías, debemos **cambiar el tipo de dato *double precision* a *numeric***. Esto se hace con la función **CAST** **campo_expresión AS NUMERIC** o usar su equivalente más corto **"::"**.

Usemos el query anterior para aplicar la función **round()** con dos lugares decimales.

- ☐ En la pestaña **<gisdb_lab> Script 1, modifique y ejecute** el pasado query:

The screenshot shows a script editor window titled "<gisdb_lab> Script-1". The SQL query is as follows:

```
select cfcc, --traer valores del campo cfcc
round(sum(st_length(geom)/1000)::numeric,2) as kms --conv a numeric la sum de kms,usa round a 2 lug decimales
from g35_viales_mapa_base_tiger_rds_2006se_villalba --de esta tabla
where cfcc in ('A41','A51') --condición: solo valores cfcc=A41,A51
group by cfcc --agrupar sumas de valores usando el campo cfcc
order by 1; --ordena ascendente por la primera columna/campo
```

Below the script editor, a table viewer shows the results of the query:

Grid	cfcc	kms
1	A41	236.44
2	A51	16.97

En la segunda línea de código, según el orden por paréntesis de adentro hacia afuera:

- 1: obtener las longitudes del campo `geom`,
- 2: luego sumarlas,
- 3: dividir entre 1,000 las sumas para obtener `kms`,
- 4: cambiar las sumas en tipo *double precision* `:: numeric`,
- 5: redondear el dato numérico a 2 lugares decimales.



ST_DISTANCE

Mostrar las distancias entre cada plaza pública municipal. El geodato incluye otras plazas públicas. Este query va a devolver miles de filas, ya que estará comparando distancias de cada plaza pública “de recreo” con todas las demás plazas. **La función ST_Distance(geom, geom) puede abreviarse (a.geom<->b.geom).** En este caso, devolveremos distancias en kilómetros.

- En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query:

```
select --concatenar nombre, coma y espacio y municipio
      concat(p1.nombre,', ', p1.municipio) as plaza_origen,
      concat(p2.nombre,', ', p2.municipio) as plaza_destino,
      round( --redondear...
            ST_Distance(p1.geom,p2.geom) --distancias entre geoms
            ::numeric --convertir resultado ST_Distance a numérico
            /1000, --dividir por 1,000 para conv en kms
            3) --... a tres lugares decimales
      as dist_kms
from --tabla plazas públicas p1 y copia p2
     pr_geodata.g33_dotacional_plazas_publicas_2010 p1,
     pr_geodata.g33_dotacional_plazas_publicas_2010 p2
where --condición: que no sea la misma plaza
      p1.gid<>p2.gid
order by --ordena por distancias usando caja MBR
        p1.geom<->p2.geom;
```

Grid	AZ plaza_origen	AZ plaza_destino	123 dist_kms
1	Plaza de Recreo, Camuy	Plaza de Recreo, Hatillo	2.003
2	Plaza de Recreo, Hatillo	Plaza de Recreo, Camuy	2.003
3	Plaza de Recreo, Dorado	Plaza de Recreo, Toa Baja	2.036
4	Plaza de Recreo, Toa Baja	Plaza de Recreo, Dorado	2.036
5	Plaza de Recreo, Guánica	Plaza Ensenada, Guánica	2.59
6	Plaza Ensenada, Guánica	Plaza de Recreo, Guánica	2.59
7	Plaza Roosevelt, San Juan	Plaza de Convalecencia, San Juan	2.624
8	Plaza de Convalecencia, San Juan	Plaza Roosevelt, San Juan	2.624
9	Plaza de Recreo, Cataño	Plaza de Armas (Viejo San Juan), Sa	2.705
10	Plaza de Armas (Viejo San Juan), Sa	Plaza de Recreo, Cataño	2.705
11	Plaza de Recreo, San Germán	Plaza de Recreo, Lajas	3.931

Refresh Save Cancel

6972 row(s) fetched - 0.041s (0.007s fetch), on 2025-07-07 at 17:09:12

El query hará **6,972** comparaciones de distancia en cada una y el resto de las plazas. Si queremos limitar la comparación de distancias entre plazas, digamos, las primeras más cercanas, debemos hacer un subquery o query anidado, donde se limite la comparación al término que uno desee. Esto lo veremos más adelante en la sección de [Subqueries](#). Usamos el atajo <-> en ORDER BY para ordenar por distancias. Este operador <-> no es tan preciso como ST_Distance porque es un comparador de cajas **Minimum Bounding Boxes (MBR)**, por lo tanto, lo ubicamos en la cláusula ORDER BY.



ST_XMin, ST_YMin

Haga una lista de municipios y barrios que están al extremo sur de PR. Repita el procedimiento para encontrar los municipios con su barrio más al oeste de PR.

Las funciones que vamos a usar son **st_ymin(geom)**, **st_xmin(geom)**. Recuerde que estamos en un plano cartesiano donde las coordenadas Y aumentan de sur hacia el norte y que las coordenadas X aumentan de oeste hacia el este. **Ordene las filas** usando **order by st_ymin(geom)** y luego por **st_xmin(geom)**. **Limite** los resultados solo a **10 filas**.

Municipios y barrios al extremo sur:

- ☐ En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query:

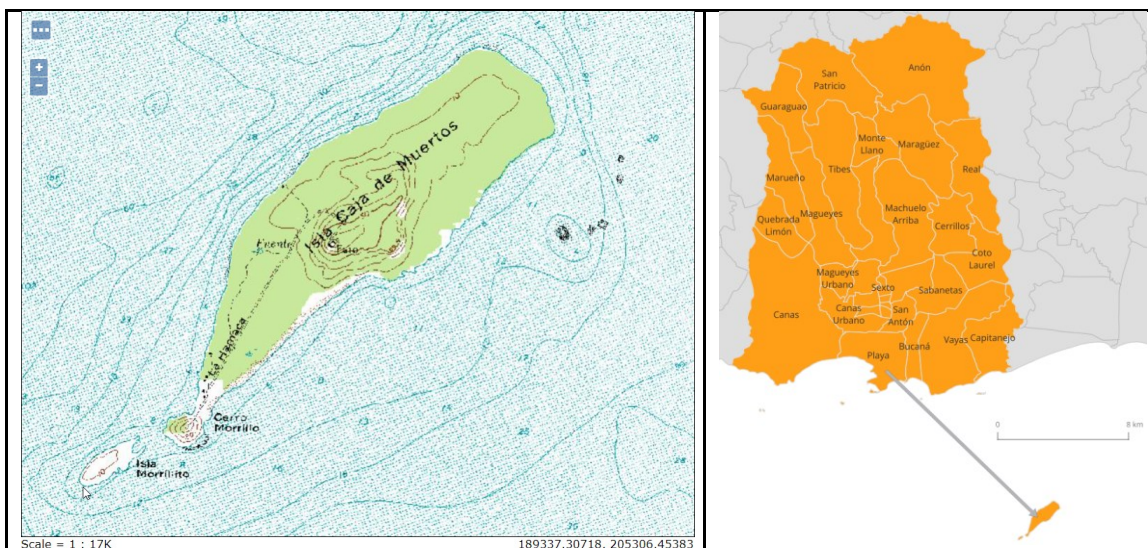
```
/* ¿Cuál es el barrio que está
 * más al sur de PR?
 * Usar función ST_YMin(geom)
 */
select
    barrio,          --traer barrio
    municipio,       --traer municipio
    ST_YMin(geom)    --coord. Y mínima (+al sur)
                    --de cada barrio
from --de la tabla de barrios
    pr_geodata.g03_legales_barrios_2015
order by --ordena por columna #3 (st_ymin)
    3 asc --orden ascendente
limit --devolver solo 10 filas.
    10;
```

	AZ barrio	AZ municipio	123 st_ymin
1	Playa	Ponce	205,305.3163
2	Aguirre	Salinas	208,831.7663
3	Felicia I	Santa Isabel	209,515.0259
4	Jobos	Guayama	209,679.5433
5	Río Cañas Abajo	Juana Díaz	209,962.391
6	Jauca I	Santa Isabel	210,194.54
7	Montalva	Guánica	210,426.4219
8	Llanos Costa	Cabo Rojo	210,989.2501
9	Boquerón	Cabo Rojo	211,138.6875
10	Machete	Guayama	211,603.6226

Refresh Save Cancel

10 row(s) fetched - 0.020s, on 2025-07-08 at 09:57:24

El barrio más al extremo sur es el barrio Playa, en Ponce. Este barrio incluye la isla Caja de Muertos y la isla Morrillito



El punto más al sur, sin salir de la costa debe ser Punta Brea en el barrio Montalva en Guánica.

Si desea obtener los municipios más **al extremo norte**, sustituya la función por **st_ymax(geom)** con orden descendente.

Municipios y barrios al extremo oeste:

- ☐ En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query:

```
/* ¿Cuál es el barrio que está
 * más al oeste de PR?
 * Usar función ST_XMin(geom)
 */
select
    barrio,          --traer barrio
    municipio,       --traer municipio
    ST_XMin(geom)    --coord. X mínima (+al oeste)
                    --de cada barrio

from --de la tabla de barrios
    pr_geodata.g03_legales_barrios_2015
order by --ordena por columna #3 (st_xmin)
    3 asc --orden ascendente
limit --devolver solo 10 filas.
    10;
```

	AZ barrio	AZ municipio	123 st_xmin
1	Isla de Mona e Islote Monito	Mayagüez	39,380.1016
2	Sabanetas	Mayagüez	88,270.1459
3	Puntas	Rincón	111,423.5469
4	Ensenada	Rincón	112,165.3107
5	Pueblo	Rincón	113,270.6046
6	Barrio Pueblo	Rincón	113,299.0906
7	Calvache	Rincón	113,997.2824
8	Río Grande	Rincón	114,010.6636
9	Barrero	Rincón	114,640.7176
10	Río Grande	Aguada	114,805.8168

10 row(s) fetched - 0.021s (0.001s fetch), on 2025-07-08 at 10:07:23

Observe que se trata del mismo query con la diferencia de sustituir por **st_xmin(geom)**. En este caso el orden es ascendente (por defecto, está implícito)

El municipio más al extremo oeste es Mayagüez, por el **barrio Isla de Mona e islote Monito**, seguido por el **barrio Sabanetas**, el cual **incluye la isla Desecheo**.

El punto más al oeste sin salir de la costa debe estar en Punta Higüero en el barrio Puntas del Municipio de Rincón.



ST_TRANSFORM

Transformar coordenadas:

¿Cómo expresamos las coordenadas planas del sistema que estamos usando ([EPSG:6566](#)) a coordenadas geográficas en sistema decimal con datum WGS84 ([EPSG:4326](#))? Para esto podemos usar la función `ST_Transform(geom, epsg_code)` para hacer la transformación de coordenadas sin tener que generar otro archivo geográfico. Por ejemplo:

- ☐ En la pestaña **gisdb_lab** Script 1, escriba y ejecute el siguiente query:

```
select
    barrio,          --traer barrio
    municipio,       --traer municipio
    ST_YMax( --coord. Y máxima (+al norte)
        ST_Transform(geom,4326) --transformar
                                --coordenadas desde
                                --EPSG:6566 a EPSG:4326
                                --metros a lat-long
    ) --de cada barrio
    as ymax --alias ymax
from --de la tabla de barrios
    pr_geodata.g03_legales_barrrios_2015
order by --ordena por columna #3 (ymax)
    ymax desc --orden descendente
limit --devolver solo 10 filas.
    10;
```

	AZ barrio	AZ municipio	123 ymax
1	Bajura	Isabela	18.515968727
2	Maleza Alta	Aguadilla	18.5133521592
3	Aguacate	Aguadilla	18.5131330081
4	Montaña	Aguadilla	18.5127122018
5	Guayabos	Isabela	18.5116769203
6	Maleza Baja	Aguadilla	18.5105786139
7	Jobos	Isabela	18.5103230276
8	Bejucos	Isabela	18.5095211211
9	Barrio Pueblo	Isabela	18.5089080294
10	Borinquen	Aguadilla	18.5000981873

Refresh

Save

Cancel

10 row(s) fetched - 0.566s, on 2025-07-08 at 10:20:47

El barrio más norteño es Bajura en el Municipio de Isabela.



GROUP BY (EJEMPLO TRIVIA)

Aunque este query no utiliza ninguna función geoespacial, podemos usar **GROUP BY** con las funciones **string_agg()** y **left()** para devolver una **lista de municipios por letra inicial**.

¿Cuántos municipios comienzan con la letra A, cuáles con B, etcétera?

- ☐ En la pestaña **<gisdb_lab> Script 1**, escriba y ejecute el siguiente query:

```
*<gisdb_lab> Script-1 x
select left(municipio,1) as letra_inicial, --trae primera letra del municipio
count(municipio) as cuántos, --conteo de municipios
string_agg(municipio, ',' ) as municipios --trae lista municipios en solo una fila
from g03_legales_municipios_2015 --de la tabla municipios_2015
group by letra_inicial --agrupa por campo letra_inicial
order by 2 desc; --ordena por el campo #2, descendente
```

Results 1 x

select left(municipio,1) as letra_inicial, c

letra_inicial	cuántos	municipios
C	14	Cabo Rojo, Caguas, Camuy, Canóvanas, Carolina, Cataño,
A	8	Adjuntas, Aguada, Aguadilla, Aguas Buenas, Aibonito, Ar
S	7	Sabana Grande, Salinas, San Germán, San Juan, San Loren
M	6	Manatí, Maricao, Maunabo, Mayagüez, Moca, Morovis
L	6	Lajas, Lares, Las Marías, Las Piedras, Loíza, Luquillo
G	5	Guayama, Guayanilla, Guaynabo, Gurabo, Guánica
V	4	Vega Alta, Vega Baja, Vieques, Villalba
B	3	Barceloneta, Barranquitas, Bayamón
J	3	Jayuya, Juana Díaz, Juncos
P	3	Patillas, Peñuelas, Ponce

Refresh Save Cancel Export data 200

21 row(s) fetched - 0.001s, on 2024-02-22 at 09:34:20

Esta estructura del query puede adaptarse al uso de códigos identificadores y hacer resúmenes basados en partes de un código-ID.

SPATIAL JOINS

Este tipo de enlace (join) entre tablas geoespaciales es bastante común y no es necesario que haya un identificador común para unir las dos tablas. En muchos casos, podemos intercambiar el contenido de la cláusula WHERE e insertarlo en la parte de JOIN tabla ON *condición*.

Por ejemplo, devolver los sectores que están en cada barrio. El geodato de sectores tiene el barrio, pero podemos usar el predicado [ST_DWithin\(geom, geom, distancia\)](#) para mostrar cuáles son los sectores dentro de cada barrio del Municipio de Villalba. La distancia establecida en ST_DWithin no es significativa, solo para asignar cierto umbral de tolerancia. Igualmente se puede usar [ST_Intersects](#), [ST_Covers](#) o [ST_CoveredBy](#) en la consulta.

Tarea: Hacer una lista de sectores por barrio en Villalba.

- ☐ En la pestaña **Script-1**, escriba y ejecute las siguientes consultas separadamente.
Query usando WHERE:



```
*<gisdb_lab> Script-1 X
select b.barrio, s.nombre as sectores --trae barrio-tablaBarrios, nombre de sectores
from g25_asentamientos_sectores_2017_villalba s, --de la tabla sectores, alias s
g03_legales_barrios_2015_villalba b --y de la tabla barrios_2015, alias b
where st_dwithin(s.geom,b.geom,0.001) --predicado st_dwithin() ambas geometrías deben
--intersecarse dentro de una distancia de 0.001 metros
order by b.barrio,s.nombre; -- ordena por barrio asc y por nombre de sector asc
```

```
g03_legales_barrios_2015_villalba(+) 1 X
select b.barrio, s.nombre as sectores from g03_legales_barrios_2015_villalba b, g03_legales_barrios_2015_villalba s
where st_dwithin(s.geom,b.geom,0.001)
order by b.barrio,s.nombre;
```

barrio	sectores
Barrio Pueblo	Borinquen
Barrio Pueblo	Cooperativa
Barrio Pueblo	Efraín Suárez
Barrio Pueblo	San Cristóbal

Query usando JOIN *tabla ON condición:*

```
*<gisdb_lab> Script-1 X
select b.barrio, s.nombre as sectores --trae barrio-tablaBarrios, nombre de sectores
from g25_asentamientos_sectores_2017_villalba s --de la tabla sectores, alias s, -no usar coma-
join g03_legales_barrios_2015_villalba b --JOIN de la tabla barrios_2015, alias b
on st_dwithin(s.geom,b.geom,0.001) --ON:usando el predicado st_dwithin() ambas geometrías deben
--intersecarse dentro de una distancia de 0.001 metros
order by b.barrio,s.nombre; --ordena por barrio asc y por nombre de sector asc
```

```
g03_legales_barrios_2015_villalba(+) 1 X
select b.barrio, s.nombre as sectores from g03_legales_barrios_2015_villalba b, g03_legales_barrios_2015_villalba s
where st_dwithin(s.geom,b.geom,0.001)
order by b.barrio,s.nombre;
```

barrio	sectores
Barrio Pueblo	Borinquen
Barrio Pueblo	Cooperativa
Barrio Pueblo	Efraín Suárez
Barrio Pueblo	San Cristóbal

En lugar de usar identificadores entre tablas, el JOIN se basa en la **coincidencia espacial** (si ambas geometrías intersecan dentro de 0.001 metros `ST_DWithin(s.geom,b.geom,0.001)`). El orden de geometrías no hace diferencia porque trabaja como `st_intersects` y es conmutativo. En ambas queries se devuelve el mismo número de filas y el tiempo de respuesta es prácticamente el mismo.



LEFT JOIN

Usar **JOIN** puede tener otra utilidad. El tipo de join por defecto es **INNER JOIN** y está implícito (no hay que escribirlo). Este tipo de JOIN devuelve la **intersección** de los conjuntos de filas que parean entre ambas tablas. Las que no pareen en la tabla izquierda no aparecerán en el resultado del query. Por otro lado, si usamos **LEFT JOIN**, preservamos todas las filas de la tabla izquierda (geodato de barrios).

Ejemplo usando **INNER JOIN** y **LEFT JOIN**: **Enumerar barrios que tienen escuelas públicas.**

- Escriba estos queries separadamente, en la pestaña **Script-1** de **DBeaver**:

EJEMPLO INNER JOIN:

```
*gisdb_lab> Script-1 X
select b.barrio, e.escuela --trae barrio-tablaBarrios, nombre de escuelas
from g03_legales_barrios_2015_villalba b --de la tabla barrios, alias b, -no usar coma-
join g33_dotacional_educacion_escuelas_2021_villalba e --JOIN de la tabla escuelas, alias e
on st_dwithin(e.geom,b.geom,0.001) --ON:usando el predicado st_dwithin() ambas geometrias
--deben intersectarse dentro de una distancia de 0.001 metros
order by b.barrio,e.escuela; --ordena por barrio asc y por nombre de sector asc
```

g03_legales_barrios_2015_villalba(+) 1 X

```
select b.barrio, e.escuela from g03_legales_barrios_2015_villalba b
join g33_dotacional_educacion_escuelas_2021_villalba e
on st_dwithin(e.geom,b.geom,0.001)
order by b.barrio,e.escuela;
```

barrio	escuela
Barrio Pueblo	FRANCISCO ZAYAS SANTANA
Hato Puerco Arriba	CRISTINA (AMADA) MARTINEZ MARTINEZ
Hato Puerco Arriba	CROEV: ESPECIALIZADA EN CIENCIAS Y MATEMATICAS
Hato Puerco Arriba	ISABEL ALVARADO ALVARADO
Hato Puerco Arriba	NORMA I TORRES COLON
Hato Puerco Arriba	SU HATILLO SILVIA TORRES TORRES
Vacas	RAMON LOPEZ BERRIOS
Villalba Abajo	DANIEL SERRANO RIVERA
Villalba Arriba	LYSANDER BORRERO TERRY

Devuelve 9 filas, ya que solo hay 9 escuelas en Villalba (datos de 2021). No aparecen los barrios Hato Puerco Abajo, Caonillas Arriba, Caonillas Abajo.

EJEMPLO LEFT JOIN:

```
*gisdb_lab> Script-1 X
select b.barrio, e.escuela --trae barrio-tablaBarrios, nombre de escuelas
from g03_legales_barrios_2015_villalba b --de la tabla barrios, alias b, -no usar coma-
left join g33_dotacional_educacion_escuelas_2021_villalba e --LEFT JOIN de la tabla escuelas, alias e
on st_dwithin(e.geom,b.geom,0.001) --ON:usando el predicado st_dwithin() ambas geometrias
--deben intersectarse dentro de una distancia de 0.001 metros
order by b.barrio,e.escuela; --ordena por barrio asc y por nombre de sector asc
```

g03_legales_barrios_2015_villalba(+) 1 X

```
select b.barrio, e.escuela from g03_legales_barrios_2015_villalba b
left join g33_dotacional_educacion_escuelas_2021_villalba e
on st_dwithin(e.geom,b.geom,0.001)
order by b.barrio,e.escuela;
```

barrio	escuela
Barrio Pueblo	FRANCISCO ZAYAS SANTANA
Caonillas Abajo	
Caonillas Arriba	
Hato Puerco Abajo	
Hato Puerco Arriba	CRISTINA (AMADA) MARTINEZ MARTINEZ
Hato Puerco Arriba	CROEV: ESPECIALIZADA EN CIENCIAS Y MATEMATICAS
Hato Puerco Arriba	ISABEL ALVARADO ALVARADO
Hato Puerco Arriba	NORMA I TORRES COLON
Hato Puerco Arriba	SU HATILLO SILVIA TORRES TORRES
Vacas	RAMON LOPEZ BERRIOS
Villalba Abajo	DANIEL SERRANO RIVERA
Villalba Arriba	LYSANDER BORRERO TERRY

Devuelve 12 filas. Aparecen todos los barrios, independientemente tengan escuelas o no. Útil para saber cuáles barrios no tienen escuelas. Esto se puede hacer con **sectores censales** y añadir datos de población en edad escolar para saber dónde están concentradas estas edades.



PostGIS, 3.x

En el query anterior puede hacerse una lista de los ocho barrios y concatenando los nombres de escuelas usando **LEFT JOIN**, **GROUP BY**, la función [string_agg\(\)](#) y el predicado [ST_Covers](#).

Por ejemplo:

- Escriba y ejecute el siguiente query en la pestaña **Script-1 de DBeaver**:

The screenshot shows the DBeaver SQL editor with a query in the 'Script-1' tab. The query is as follows:

```
select b.barrio, --trae barrio-tabla barrios
count(e.gid) as cuantas_escuelas, --núm de escuelas as cuantas_escuelas
string_agg(e.escuela,',') as escuelas --lista de escuelas, separadas por coma ','
from g03_legales_barrios_2015_villalba b --de la tabla barrios, alias b, -no usar coma-
left join g33_dotacional_educacion_escuelas_2021_villalba e --JOIN de la tabla escuelas, alias e
on st_covers(b.geom,e.geom) --ON:usando el predicado st_covers(barrios cubren las escuelas)
group by b.barrio --agrupa conteos por barrio
order by 2 desc,1; --ordena por número de escuelas, desc, luego por barrio, asc
```

Below the query, the results are displayed in a table with columns: barrio, cuantas_escuelas, and escuelas. The results are as follows:

	barrio	cuantas_escuelas	escuelas
1	Hato Puerco Arriba	5	NORMA I TORRES COLON, SU HATILLO SILVIA TORRES TORRES, ISABEL ALVARADO A
2	Barrio Pueblo	1	FRANCISCO ZAYAS SANTANA
3	Vacas	1	RAMON LOPEZ BERRIOS
4	Villalba Abajo	1	DANIEL SERRANO RIVERA
5	Villalba Arriba	1	LYSANDER BORRERO TERRY
6	Caonillas Abajo	0	[NULL]
7	Caonillas Arriba	0	[NULL]
8	Hato Puerco Abajo	0	[NULL]

- Note que podemos usar el predicado **st_covers(geom,geom)** para comparar polígonos con puntos. Barrios que 'cubren' escuelas.
- Usamos **GROUP BY** para hacer el conteo de escuelas por barrio.
- **LEFT JOIN**: Los últimos tres barrios no tienen escuelas públicas.

SUBQUERIES:

En esta parte, traemos la idea de un query anidado dentro de otro para poder resolver un problema de dos maneras.

ST_Disjoint Y SUBQUERIES

Ejemplo: Queremos saber **cuáles son los municipios que NO tienen áreas naturales protegidas terrestres**, según el **geodato de áreas naturales protegidas terrestres de 2019 (ANP)**.

En PostGIS, el predicado **st_disjoint(geom,geom)** es **costoso**, ya que compara todas las geometrías de A con el exterior de todas las geometrías de B. Por lo tanto, devolverá múltiples filas comparadas. Más aún, **para que PostGIS nos devuelva solo los municipios que no tienen áreas naturales protegidas (ANP) terrestres, tendremos que combinar/unir todas las geometrías de ANP usando ST_Union(geom) en un subquery, y con esta geometría agregada o total, se hace la comparación con el geodato de municipios.**

Esta manera es mucho menos eficiente, pero se usa aquí para demostrar el uso del predicado **st_disjoint** y cómo **hacer un subquery** dentro de otro.



PostGIS, 3.x

- Escriba y ejecute el siguiente query en la pestaña **Script-1 de DBEaver**:

```
--QUERY COSTOSO: ST_Disjoint(geom, geom)
select m.municipio --trae nombres de municipios
from pr_geodata.g03_legales_municipios_2015 m
--tabla municipios
join ( --en este join (intersección) devuelve...
select ST_Union(geom)
--unión de geometrías...
--del geodato
--áreas naturales protegidas terrestres
as geom --alias geom
from --de esta tabla:
pr_geodata.g11_conserv_areas_naturales_protegidas_terrestres_2019
/* En esta parte entre paréntesis, (subquery) se realiza la unión
* de todas las geometrías del geodato ANP. De lo contrario,
* PostGIS hará el pareo de cada fila de municipio con cada una
* de las geometrías de ANP. El resultado de este subquery debe
* tener alias y se llama "anp".
*/
)as anp --alias del query: anp
on
ST_Disjoint(m.geom,anp.geom);
--test:geometrías disjuntas, lo contrario de st_intersects
```

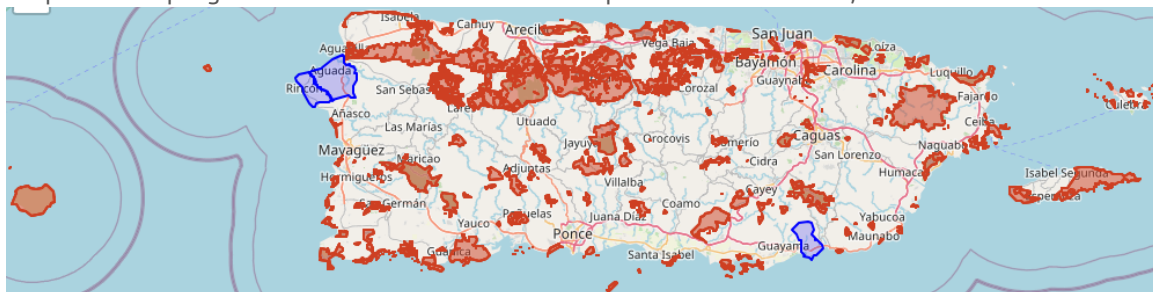
Resultado:

	A-Z municipio ▼
1	Aguada
2	Arroyo
3	Rincón

Como se mencionó, usaremos un subquery. Dentro de este subquery se realizará la unión `st_union(geom)` de las geometrías de las ANP. Los subqueries se encierran entre paréntesis. **Si queremos ver las geometrías en DBEaver, inserte los campos `m.geom` y `anp.geom`.**

```
select municipio, m.geom,anp.geom
```

Tendrá que esperar un rato (30 segundos o más) para que DBEaver produzca el resultado y que haga después el despliegue. Las áreas en azul son los municipios sin ANP terrestres, 2019.



Los municipios en azul no intersecan con ninguna área natural protegida *terrestre*: (Aguada, Arroyo y Rincón).



Una alternativa es usar **LEFT JOIN** con la tabla de municipios.

```
select --trae municipios
      municipio, anp.gid as "anp.gid"
from --de la tabla de municipios as m
      pr_geodata.g03_legales_municipios_2015 m
left join --traer "todos los récords de municipios.."
      pr_geodata.g11_conserv_areas_naturales_protegidas_terrestres_2019 anp
on --...que al hacer el test de intersección..
      st_intersects(m.geom,anp.geom)
where --...devuelva falso, por lo tanto, anp.gid estará vacío
      anp.gid is null;
```

Resultado: se incluyó **anp.gid** para demostrar que no hay coincidencia entre **geometrías anp** vs **municipios** de **Aguada, Arroyo y Rincón**. Por lo tanto, devuelve **NULL** en cada fila.

	A-Z municipio	123 anp.gid
1	Aguada	[NULL]
2	Arroyo	[NULL]
3	Rincón	[NULL]

Este query tardó 5 segundos en completar.

Una **alternativa** al uso de **st_disjoint** y **más eficiente** puede ser la siguiente:

Este query devuelve los mismos datos, aprovechando que **st_intersects** es el **"inverso"** del predicado **st_disjoint**.

☐ Escriba y ejecute el siguiente query en la pestaña **Script-1 de DBeaver**:

```
select --trae municipios
      m.municipio
from --de la tabla municipios
      pr_geodata.g03_legales_municipios_2015 m
where --test...
      not exists
      --devuelve récords de anp donde que no haya ...
      (select --no hay que escribir nombres de
            --columnas aquí
            from --de la tabla anp terrestres, 2019
            pr_geodata.g11_conserv_areas_naturales_protegidas_terrestres_2019 anp
            where --...intersección geométrica
                  st_intersects(m.geom,anp.geom)
      );
```

En otras palabras, **el subquery** devuelve las filas donde no exista intersección (que no haya puntos en común entre ambas geometrías).

No fue necesario hacer la unión de geometrías como en el query y subquery anterior. Por lo tanto, el query se completó en menos de 4 segundos.

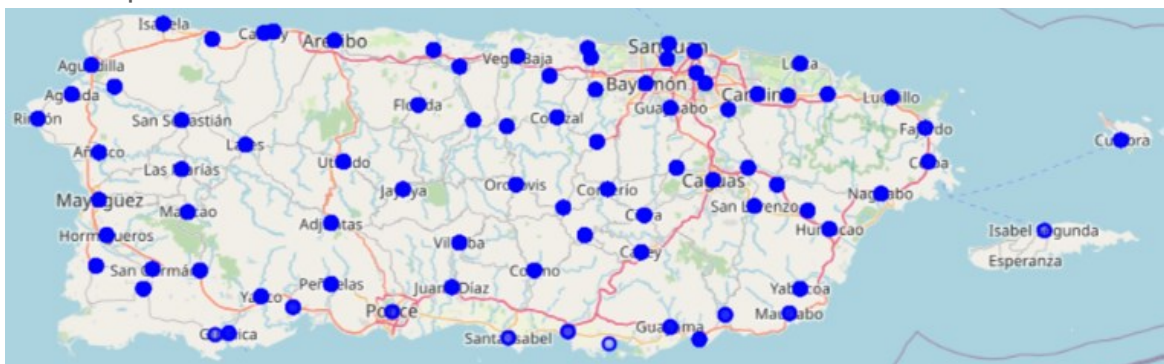
Esto demuestra que puede haber varias maneras de resolver un problema o situación. Se recomienda que busquemos información de las propiedades de los predicados topológicos: para qué están diseñados (propósito), si usan algún índice para agilizar búsquedas, y si hay otras alternativas de predicados más eficientes. En este y muchos casos **ST_Intersects** se ha estado mejorando constantemente para hacerlo más eficiente y es uno de los predicados más utilizados.



TABLAS DE DISTANCIAS: DISTANCIAS ENTRE PLAZAS DE RECREO

El siguiente mapa muestra la distribución de plazas públicas (de recreo). Estas son “el centro” del centro urbano histórico de cada municipio y algunos “*protomunicipios*” como Aguirre, Castañer y comunidades con “personalidad” como Roosevelt. Las plazas en el Viejo San Juan no necesariamente son plazas de recreo tradicionales.

La plaza de Armas del Viejo San Juan vendría siendo el equivalente a las plazas de recreo en los demás municipios. La plaza de Río Piedras es la excepción, ya que Río Piedras fue un municipio hasta 1951 cuando fue anexado al Municipio de San Juan. Algo parecido hay en Santurce, que también fue un municipio pero se fusionó con San Juan también en 1880 bajo el gobierno español.



Ubicación de las plazas de recreo (2010).

El siguiente query nos dará las **primeras 20 plazas de recreo públicas más cercanas**.

☐ Escriba y ejecute el siguiente query en la pestaña **Script-1 de DBaiver**:

```
select
  concat(p.nombre, ', ', p.municipio) as plaza_origen,
  --unir nombre plaza con ', ' como separador más el municipio
  --alias plaza_origen
  concat(p2.nombre, ', ', p2.municipio) as plaza_destino,
  --unir nombre plaza2 con ', ' como separador más el municipio2
  --alias plaza_destino
  round((ST_Distance(p.geom, p2.geom)/1000)::numeric,3) as distancia_kms
  --calcula las distancias entre geometrías
  --/1000 = kilómetros
from
  pr_geodata.g33_dotacional_plazas_publicas_2010 p
  --plazas públicas
join
  pr_geodata.g33_dotacional_plazas_publicas_2010 p2
  --copia, plazas públicas
on
  p.gid < p2.gid
  --importante: "<" para que
  --no devuelva la misma plaza
order by
  p.geom <-> p2.geom --ordena por distancia, ascendente.
  limit 20; --devuelve las 20 plazas más cercanas
```



	A-Z plaza_origen	A-Z plaza_destino	123 distancia_kms
1	Plaza de Recreo, Camuy	Plaza de Recreo, Hatillo	2.003
2	Plaza de Recreo, Dorado	Plaza de Recreo, Toa Baja	2.036
3	Plaza de Recreo, Guánica	Plaza Ensenada, Guánica	2.59
4	Plaza Roosevelt, San Juan	Plaza de Convalecencia, San Juan	2.624
5	Plaza de Recreo, Cataño	Plaza de Armas (Viejo San Juan), San Juan	2.705
6	Plaza de Recreo, Lajas	Plaza de Recreo, San Germán	3.931
7	Plaza Antonia Quiñones (El Condado), San Juan	Plaza Roosevelt, San Juan	4.16
8	Plaza de Armas (Viejo San Juan), San Juan	Plaza Antonia Quiñones (El Condado), San Juan	5.091
9	Plaza de Recreo, Yabucoa	Plaza de Recreo, Maunabo	5.229
10	Plaza de Recreo, Humacao	Plaza de Recreo, Las Piedras	5.352
11	Plaza de Recreo, Cataño	Plaza Antonia Quiñones (El Condado), San Juan	5.399
12	Plaza de Recreo, Barceloneta	Plaza de Recreo, Manatí	5.712
13	Plaza de Recreo, Moca	Plaza de Recreo, Aguadilla	5.782
14	Plaza de Recreo, Juncos	Plaza de Recreo, San Lorenzo	5.855

En esta ocasión usaremos un CROSS JOIN LATERAL en un subquery para hacer la comparación de distancias entre cada plaza. Para este query, en lugar de devolver miles de filas, devolverá exactamente 84 filas. Este es el número total de plazas en el geodato (2010).

CROSS JOIN LATERAL permite que un subquery acceda a columnas de la tabla que está en la parte FROM. *LATERAL* puede ser parte de un LEFT JOIN, RIGHT JOIN y otros tipos de JOIN.

Si no limitamos con *LIMIT 1*, obtendríamos un producto cartesiano, comparando cada plaza con las demás plazas. Eso puede ser útil para otros análisis, pero este no es el caso en nuestro query.



- Escriba y ejecute el siguiente query en la pestaña **Script-1 de DBeaver**:

```
select
  concat(p.nombre, ', ', municipio) as plaza_origen,
  p2.nombre as plaza_mas_cercana,
  round((ST_Distance(p.geom, p2.geom)/1000)::numeric,3) as distancia_kms
--Usar ST_Distance para distancia real/1000 para devolver kilómetros
--::numeric: convertir el resultado de distancias 'double precision'
--y convertirlo a tipo numeric.
--round() para que las distancias estén a tres lugares decimales
from
  pr_geodata.g33_dotacional_plazas_publicas_2010 p
  cross join lateral (
    select
      concat(p2.nombre, ', ', p2.municipio) as nombre,
      p2.geom --p2.geom es necesario para comparar con p.geom
    from
      pr_geodata.g33_dotacional_plazas_publicas_2010 p2
    where
      p.gid <> p2.gid --no comparar con la misma plaza
    order by
      p.geom <-> p2.geom -- operador <-> para ordenar por dists
    limit 1
  ) as p2 --copia tabla plazas
order by 3; --ordena ascendente por columna #3
```

	A-Z plaza_origen	A-Z plaza_mas_cercana	123 distancia_kms
1	Plaza de Recreo, Camuy	Plaza de Recreo, Hatillo	2.003
2	Plaza de Recreo, Hatillo	Plaza de Recreo, Camuy	2.003
3	Plaza de Recreo, Dorado	Plaza de Recreo, Toa Baja	2.036
4	Plaza de Recreo, Toa Baja	Plaza de Recreo, Dorado	2.036
5	Plaza Ensenada, Guánica	Plaza de Recreo, Guánica	2.59
6	Plaza de Recreo, Guánica	Plaza Ensenada, Guánica	2.59
7	Plaza de Convalecencia, San Juan	Plaza Roosevelt, San Juan	2.624
8	Plaza Roosevelt, San Juan	Plaza de Convalecencia, San Juan	2.624
9	Plaza de Recreo, Cataño	Plaza de Armas (Viejo San Juan), San Juan	2.705

Aquí la lista devuelta está en orden por distancia ascendente.

Aunque este query devuelve exactamente 84 filas de cada plaza, puede notar también que hay una “aparente” repetición. Sin embargo, nos parece que están repetidas las relaciones de distancias, ya que la distancia $a \leftrightarrow b = b \leftrightarrow a$.

Para evitar esta duplicidad, podemos seguir usando CROSS JOIN LATERAL modificando el test bajo el WHERE del outer query. Añadiremos una condición adicional para filtrar las filas devueltas.

- En la pestaña de escribir queries pulse las teclas **ctrl+] para generar un nuevo query.**
- En la ventana nueva escriba el siguiente query:



PostGIS, 3.x

```

select concat(p.nombre,', ',p.municipio) as plaza_origen,
--concatena cols nombre y municipio usando ', ' tabla alias p
p2.plaza as plaza_destino,
--trae el campo plaza de la tabla alias p2
round(p2.dist_kms::numeric,3) as kms
--trae las distancias en kms de la tabla p2
--redondéalos a dos lugares decimales

from
pr_geodata.g33_dotacional_plazas_publicas_2010 p --tabla plazas as p
cross join lateral (
--cross join lateral compara una fila con todas las otras de p2
select p2.gid, --trae el campo p2.gid
concat(p2.nombre,', ',p2.municipio) as plaza,
--concatena cols nombre y municipio alias p2
(p.geom->p2.geom)/1000.0 as dist_kms
--calcular distancias, convertirlas a kms, alias kms
from
pr_geodata.g33_dotacional_plazas_publicas_2010 p2
--copia la tabla plazas as p2
order by dist_kms limit 2
) as p2
--ordena por campo dist_kms de la tabla p2 y
--limita solo a dos filas porque una será dist=0
--y la segunda fila será la más cercana <>0
where p.gid<p2.gid
--no deben tener el mismo identificador gid
order by dist_kms;
--ordena por campo "municipio" asc, tabla p
--luego ordena por campo dist kms asc, tabla p2

```

Resultados:

SQL: select concat(plz.nombre,', ',plz.munici) Enter a SQL expression to filter results (use Ctrl+Space)

	A-Z plaza_origen	A-Z plaza_destino	123 kms
1	Plaza de Recreo, Camuy	Plaza de Recreo, Hatillo	2.003
2	Plaza de Recreo, Dorado	Plaza de Recreo, Toa Baja	2.036
3	Plaza de Recreo, Guánica	Plaza Ensenada, Guánica	2.59
4	Plaza Roosevelt, San Juan	Plaza de Convalecencia, San Juan	2.624
5	Plaza de Recreo, Cataño	Plaza de Armas (Viejo San Juan), San Juan	2.705
6	Plaza de Recreo, Lajas	Plaza de Recreo, San Germán	3.931
7	Plaza Antonia Quiñones (El Condado), San Juan	Plaza Roosevelt, San Juan	4.16
8	Plaza de Recreo, Yabucoa	Plaza de Recreo, Maunabo	5.229
9	Plaza de Recreo, Humacao	Plaza de Recreo, Las Piedras	5.352
10	Plaza de Recreo, Barceloneta	Plaza de Recreo, Manatí	5.712
11	Plaza de Recreo, Moca	Plaza de Recreo, Aguadilla	5.782
12	Plaza de Recreo, Juncos	Plaza de Recreo, San Lorenzo	5.855
13	Plaza de Recreo, Canóvanas	Plaza de Recreo, Carolina	5.903
14	Plaza de Recreo, Arroyo	Plaza de Recreo, Guayama	5.954
15	Plaza de Recreo, Toa Alta	Plaza de Recreo, Toa Baja	6.05
16	Plaza de Recreo, Bavamón	Plaza de Recreo, Cataño	6.056

Refresh Save Cancel Export data 200 40

Este query devuelve 40 filas. Notará que si aparece Camuy – Hatillo en una fila, no aparecerá otra fila Hatillo – Camuy.



PRÁCTICAS:

1. ¿Cuál es el barrio que está al extremo este de Puerto Rico sin contar a Vieques ni Culebra? Dejar fuera Vieques y Culebra, ¿cuál es el barrio y su municipio más al extremo este? Use el [query de ejemplo ST_XMin](#) anterior. Use la función ST_XMax. En la cláusula Where, puede usar NOT municipio IN ('Vieques', 'Culebra') o usar NOT (barrio = 'Vieques' OR barrio = 'Culebra'). Limite los resultados a 10 filas. Resultado:

```
select
    ____,          --traer columna barrio
    ____,          --traer municipio
    ____ (geom)    --traer coordenada máxima x (oriental) de cada barrio
from
    pr_geodata.____ --tabla de todos los barrios PR
where
    municipio not in ('____', '____') --excluir Vieques y Culebra
order by ____ --ordenar por columna 3, descendente
limit ____;      --trae los primeros 10 más orientales
```

barrio	municipio	st_xmax
Cabezas	Fajardo	295747.4422
Machos	Ceiba	290707.0613
Guayacán	Ceiba	285808.9294
Quebrada Vueltas	Fajardo	285502.6251
Barrio Pueblo	Fajardo	285268.2501
Sardinera	Fajardo	284810.3437
Demajagua	Fajardo	284764.9375
Barrio Pueblo	Ceiba	284092.6221
Quebrada Fajardo	Fajardo	283379.4742
Chupacallos	Ceiba	282757.2186

¿Qué pasaría si usamos la función ST_XMin?
No se recomienda en este caso porque devuelve el mínimo x (la parte más occidental de cada barrio). Esto quiere decir que la lista de barrios se va a parecer, pero las coordenadas devueltas serán las del extremo oeste, no del extremo este, que es el que estamos buscando

2. Haga una lista donde cada barrio del Municipio de Villalba muestre el número de sectores dentro del barrio y una lista horizontal de sectores por cada barrio.

```
select
    ____,          --traer columna barrio
    count(____) as num_sectores,          --conteo de sectores
    string_agg(____, '____') as sectores --lista horizontal de sectores
                                         --separados por coma_espacio
from
    ____ as ____,          --tabla barrios, Villalba
    pr_geodata.____ s     --tabla sectores, Villalba
where
    st____(____.geom, ____.geom) --predicado geoespacial a usar
group by ____ --agrupar conteo sectores por barrio
order by ____; --ordenar por columna 1, asc
```

Resultado parcial:

barrio	num_sectores	sectores
Barrio Pueblo	4	San Cristóbal, Borinquen, Cooperativa, Efraín Suárez
Caonillas Abajo	4	Los Fondos, La Cruz, El Rincón, Cerro Gordo



PostGIS, 3.x

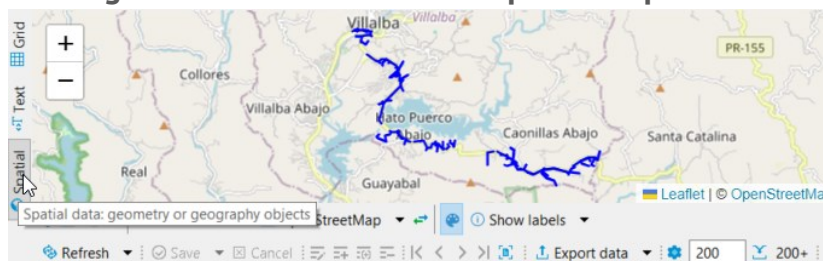
- Use el geodato ***g35_viales_mapa_base_tiger_rds_2006se_villalba*** para **mostrar una lista de nombres de vías que conectan con la carretera PR-150.**

```
select
  __.__ as id_v1, --traer id viales-copia1
  v2.gid as id_v2, --traer id viales-copia2
  v2.__, --traer feature name, copia2
  __.geom --traer geometría copia2
from
  pr_geodata.g35_viales_mapa_base_tiger_rds_2006se_villalba v1,
  __ v2 --tabla viales as v2
where
  __.fename = 'PR-150' --importante: carretera PR-150, v1
  and __.gid<>__. --que los id entre vías no sean iguales
  and st__(v1.geom,v2.geom) --función geoespacial
order by __.gid; --ordena por primera columna, tabla 1
```

Predicado: **ST_Touches(v1.geom,v2.geom)**. Puede usar también **st_intersects**.
Debe devolver 340 filas (si los datos no han cambiado)

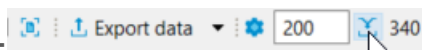
gid	gid	fename	geom
6	474	PR-150	MULTILINESTRING ((200489.
164	987	PR-150	MULTILINESTRING ((193498.
164	287	Calle Borinquen	MULTILINESTRING ((193556.
164	2427	PR-150	MULTILINESTRING ((193600.
319	992		MULTILINESTRING ((193206.
319	3152	PR-150	MULTILINESTRING ((193645.
319	699	PR-150	MULTILINESTRING ((193396.
389	818	PR-150	MULTILINESTRING ((199570.

- ☐ **Vea el geodato haciendo click en la pestaña Spatial.**



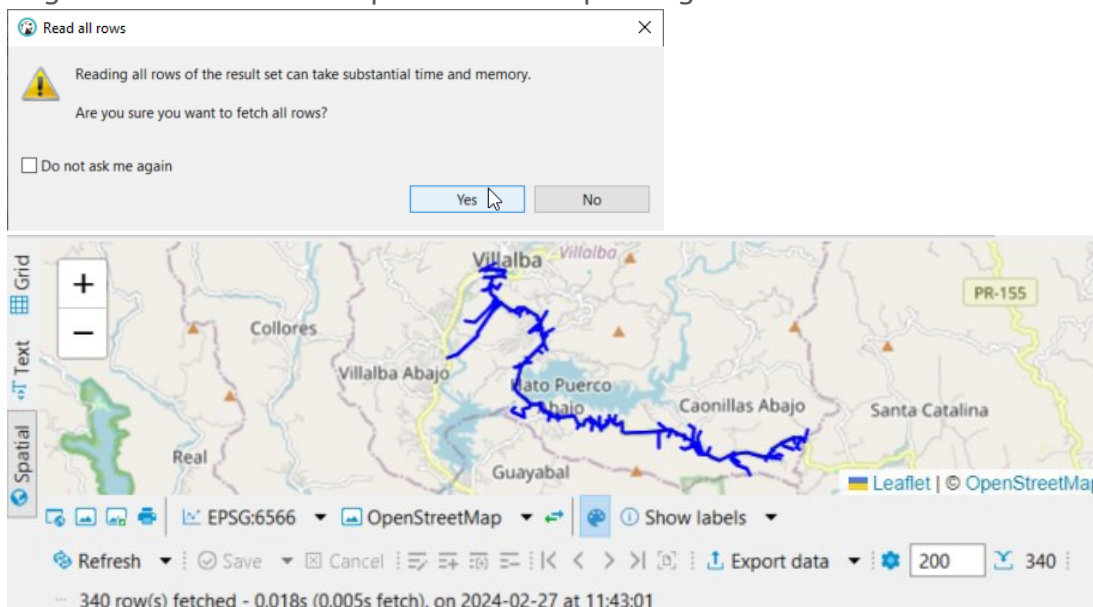
¿Está esto correcto? No completamente. Por defecto solo muestra las primeras 200 filas.

- ☐ Haga **click** en el botón **Calculate total row count** para saber cuántas filas genera este query.
- ☐ Luego haga **click** en el botón **Fetch All Data** para que pueda ver todos los elementos seleccionados en el mapa



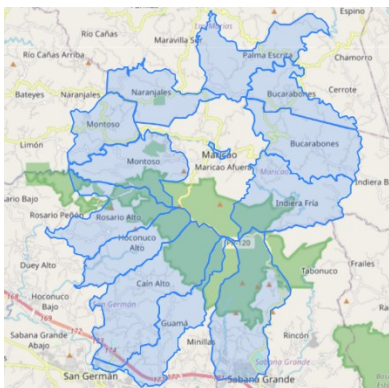


- ☐ Haga **click** en el botón **Yes** para confirmar que traiga toda la selección de filas.



4. ¿Cuáles son los barrios con más vecinos adyacentes? Use el [query para visualizar el número de barrios adyacentes con ST_Touches](#) como referencia. Esta vez, **sustituya** el predicado ST_Touches(b1.geom,b2.geom) con el predicado **ST_DWithin(b1.geom, b2.geom, 50)**, estableciendo una distancia de búsqueda de 50 m. **Para que no repita el barrio de referencia**, inserte la condición **and b1.gid<>b2.gid**. ¿Cuáles son los barrios?

```
select b1._____|'|, Bo. '||b1.____ as munic_barrio,
--1:une municipio y barrio
count(b2.____) as cuántos_adyacentes,
--2:cuenta cuántos barrios son adyacentes
string_agg('(Bo. '||b2._____|'|, '||b2._____|'|)',', ' ') as "adyacentes"
--concatena
paréntesis+Bo.espacio+barrio+coma_con_espacio+municipio+paréntesis+coma_con_espacio
from --geodato de barrios, simplificado
pr_geodata.g03_legales_barrios_2015_simpl_5m as b1
join
pr_geodata.____ as b2 --copia, geodato de barrios, simplificado
on
st_dwithin(__.geom,__.geom,50) --distancia búsqueda:50 metros
and __.gid<>__.gid
--a diferencia de st_touches, st_dwithin compara interior; excluir id
--iguales
group by
____ --agrupar por columna #1
having count(__.barrio)>=10 --devuelve solo los que tienen 10 o más vecinos
order by __ desc; --ordenar por columna #2 descendente
```

Estos son los **13 barrios adyacentes** (comparten límites) con el barrio Maricao Afuera del Municipio de Maricao. Podrá ver que el barrio Minillas está bien próximo pero no comparten límites ni en un punto.

Si aplicamos **ST_DWithin** con una **distancia de 50 metros**, podríamos obtener una lista de los barrios que se tocan, además de los que están cerca de tocarse.

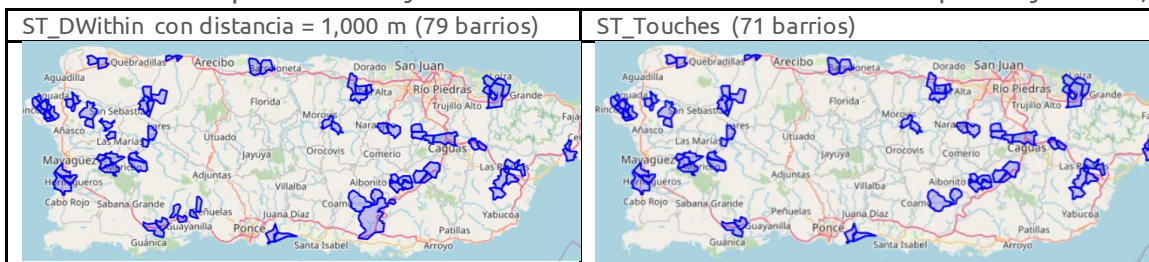
En la lista de salida de este nuevo query debe haber **14 filas**.

Advertencia: El tiempo de ejecución de este query usando ST_DWithin() puede tardar un poco.

- Haga una lista de barrios que tengan nombres iguales y que sean adyacentes. Por ejemplo, los municipios de Caguas, Cidray y Cayey son adyacentes y además tienen cada uno un barrio "Beatriz" que es adyacente o colindante con cada uno de estos municipios. Así hay cierto número de otros barrios adyacentes con el mismo nombre en municipios diferentes. Puede usar el predicado **ST_Touches** o usar el predicado **ST_DWithin(geom,geom,1000)** para establecer una distancia de búsqueda. **Escriba lo que falta en este query y ejecútelo en DBeaver**. Visualice el resultado en el tab **Spatial**

```
select b1.barrio || ', ' || b1.municipio as "barrio_municipio: ", --1:une barrio y municipio
count(b2.____) as cuántos_adyacentes, --2:cuenta cuántos barrios son adyacentes
string_agg(b2.____ || '-' || b2.____, ', ') --3:agrega barrio-municipio a la lista
as "barrios_municipios_adyacentes:", --3:alias de la columna
round(sum((b1.____<->b2.____)::numeric/1000.0,3) as dist_kms, --4:redondea la dist_kms
b1.____ --añade las geometrías para verlas en el tab Spatial de DBeaver
from pr_geodata.g03_legales_barrios_2015 as b1 --geodato de barrios
join _____ as b2 --copia del geodato de barrios
on ST_DWithin(b1.____,b2.____,1000) --dist de búsqueda: 1km
where __.barrio=__barrio --nombres de barrio idénticos entre bo. original y copia
and __.gid<>__.gid --el id no debe ser igual para no repetir el mismo barrio
group by b_.barrio,b_.municipio,b_.geom --agrupar por estos campos, geodato no copiado
order by 4 desc; --ordena por columna 4: dist_kms, descendente
```

Resultado en mapa. Barrios adyacentes con nombres idénticos en municipios adyacentes,



barrio_municipio:	cuántos_adyacentes	barrios_municipios_adyacentes:	dist_kms	geom
Beatriz, Caguas	2	Beatriz-Cayey, Beatriz-Cidra	0.00	MULT
Beatriz, Cayey	2	Beatriz-Cidra, Beatriz-Caguas	0.00	MULT



6. Haga una **lista de escuelas y las distancias a hospitales en Villalba** y del área.

Geodatos a usar: g33_dotacional_educacion_escuelas_2021_villalba as e,
g33_dotacional_salud_hospitales_cerca_villalba as h

Función: ST_Distance o el operador <->.

Use la función :: para convertir el tipo de dato numérico FLOAT a **NUMERIC**

Escriba lo que falta en este query y ejecútelo en DBeaver.

```
select e.escuela, --trae nombre de escuelas
       h.nombre, --trae nombre de hospitales
       round((e.geom<->h.geom)::numeric/1000,2) as dist_kms --redondea distancia kms entre geoms
from g33_dotacional_educacion_escuelas_2021_villalba e, --tabla escuelas
     g33_dotacional_salud_hospitales_cerca_villalba h --tabla hospitales
order by e.escuela, --orden alfabético ascendente escuela
         e.geom<->h.geom; --orden por distancia ascendente
```

Resultado:

escuela	nombre	dist_kms
CRISTINA (AMADA) MARTINEZ MARTINEZ	MedCentro Villalba	1.53
CRISTINA (AMADA) MARTINEZ MARTINEZ	Centro San Cristóbal Villalba	2.37
CRISTINA (AMADA) MARTINEZ MARTINEZ	CDT San Cristóbal Juana Díaz	6.94
CRISTINA (AMADA) MARTINEZ MARTINEZ	Centro Médico San Lucas	13.02
CRISTINA (AMADA) MARTINEZ MARTINEZ	Hospital Menonita Coamo	14.84
CRISTINA (AMADA) MARTINEZ MARTINEZ	Hospital Metropolitano Dr Pila	16.41
CRISTINA (AMADA) MARTINEZ MARTINEZ	Hospital Damas	17.46
CRISTINA (AMADA) MARTINEZ MARTINEZ	CDT Santa Isabel	18.52
CROEV: ESPECIALIZADA EN CIENCIAS Y MATEMATICAS	MedCentro Villalba	0.44
CROEV: ESPECIALIZADA EN CIENCIAS Y MATEMATICAS	Centro San Cristóbal Villalba	1.08

7. El query anterior nos devuelve las distancias de entre cada escuela con todos los hospitales registrados en el geodato de hospitales. Por eso tiene 72 filas: 9 escuelas y 8 hospitales.

Pero si nos piden: **Lista solamente los primeros 3 centros de salud (CDT y hospitales) más cercanos a cada escuela**. Para hacer esa lista, debemos hacer un **CROSS LATERAL JOIN** como el [query de ejemplo](#) de esa sección.

Escriba lo que falta en este query y ejecútelo en DBeaver.

```
select e.escuela, --trae nombre escuela
       h.nombre, --trae nombre hospital
       round(h.dist::numeric/1000,2) as kms --calc y redondea distancia en m as kms
from g33_dotacional_educacion_escuelas_2021_villalba e --tabla escuelas
cross join lateral( --uso cross join lateral
    select h.nombre, --trae nombre hospital para devolverlo al outer query
           e.geom<->h.geom as dist --calc distancias entre geoms escuelas y hospst
    from g33_dotacional_salud_hospitales_cerca_villalba h --tabla hospitales
    order by dist --ordena por distancia, si no ordenas no sirve el limit 3
    limit 3) --trae solo __ hospitales más cercanos
as h --necesario poner nombre o alias al subquery: h
order by e.escuela, h.kms; --outer query: ordena resultados por cols 1 asc y 3 asc
```

escuela	nombre	kms
CRISTINA (AMADA) MARTINEZ MARTINEZ	MedCentro Villalba	1.53
CRISTINA (AMADA) MARTINEZ MARTINEZ	Centro San Cristóbal Villalba	2.37
CRISTINA (AMADA) MARTINEZ MARTINEZ	CDT San Cristóbal Juana Díaz	6.94
CROEV: ESPECIALIZADA EN CIENCIAS Y MATEMATICAS	MedCentro Villalba	0.44
CROEV: ESPECIALIZADA EN CIENCIAS Y MATEMATICAS	Centro San Cristóbal Villalba	1.08
CROEV: ESPECIALIZADA EN CIENCIAS Y MATEMATICAS	CDT San Cristóbal Juana Díaz	8.14
DANIEL SERRANO RIVERA	MedCentro Villalba	1.96
DANIEL SERRANO RIVERA	Centro San Cristóbal Villalba	2.83
DANIEL SERRANO RIVERA	CDT San Cristóbal Juana Díaz	6.54
FRANCISCO ZAVAS SANTANA	Centro San Cristóbal Villalba	0.35



PostGIS, 3.x

Deben aparecer 27 filas ordenadas por escuela y distancia, ambos de forma ascendente.

8. Encuentre la cantidad de edificios que están en la subzona inundable **zone_subty='FLOODWAY'** en el Municipio de **Villalba**.

```
select f._____, --sub-clase zonas inundables
       count(e._____) as num_edifs --conteo edificios
from   g23_riesgo_inunda_floodzones_2017_villalba f --zonas inundables
join   g33_mapa_base_building_centroids_1998_villalba e --centroides-edificios
on     ST_CoveredBy(_.geom,_.geom) --edificios cubiertos por zona inundable
where  f._____ = 'FLOODWAY' --en la zona marcada como 'FLOODWAY'
group by f._____; --resume por sub-zona inundable
```

Practique usar st_intersects, st_covers y st_coveredby

Resultado: debe ser idéntico para los tres predicados

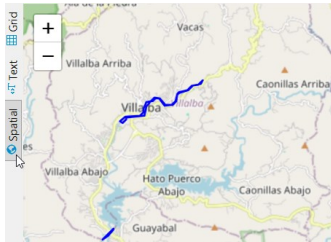
Grid	ABC zone_subty	123 num_edificios
1	FLOODWAY	47

9. Ejecute el siguiente query en el tab <gisdb_lab> Script-1

```
select f.zone_subty,geom
from g23_riesgo_inunda_floodzones_2017_villalba as f
where f.zone_subty = 'FLOODWAY';
```

Verá que hay dos áreas 'FLOODWAY' en el municipio.

Grid	ABC zone_subty	geom
1	FLOODWAY	MULTIPOLYGON (((196180.7
2	FLOODWAY	MULTIPOLYGON (((192611.1



¿Por qué crees que la otra zona **Floodway** no aparece en el query anterior #8?
Hint: Tiene que ver con GROUP BY y los nombres de las zonas FLOODWAY



10. Usando el query de la práctica anterior #8, haga los siguientes cambios:

```
select distinct f._____, --subclse zonas inundables
               count(e.____) as num_edifs --conteo edificios
from
  g23_riesgo_inunda_floodzones_2017_villalba f, --zonas inundables
  g33_mapa_base_building_centroids_1998_villalba e --edificios
where
  ST_DWithin(_.geom,f.____,1)
  and f.____ = 'FLOODWAY' --en la zona marcada como 'FLOODWAY'
group by
  f.gid; --resume usando el gid de la tabla f
```

¿Cuántos puntos aparecen si aplica una distancia de 1 metro, de 10, o de 20?. ¿Aparece más de una zona Floodway? _____

¿Por qué aparece otra zona Floodway cuando aplicamos distancia mayor? _____

Importante: Cuando hacemos GROUP BY, PostgreSQL va a **agregar**, contar, sumar por todos los campos incluidos en el GROUP BY **con valores iguales**. Por ejemplo, el campo **zone_subty** solamente tiene el valor 'FLOODWAY' para todas las áreas, aunque estén separadas. Por lo tanto, para que nos devuelva conteos de edificios por cada geometría que sea FLOODWAY, tendremos que **incluir** en el GROUP BY el campo **f.gid**. Este campo identifica cada geometría del geodato de zonas inundables, de manera única.

11. Usando el query de la práctica anterior #10:

¿Qué podemos hacer para que siempre aparezcan todas las zonas f.zone_subty = 'FLOODWAY' en el query? Hint: revise la sección [Spatial Joins](#). ¿Qué tipo de JOIN aplicaría?

```
select f.zone_subty, --subclse zonas inundables
       count(e.____) as num_edifs --conteo edificios
from
  _____villalba f --zonas inundables
left join
  g33_mapa_base_building_centroids_1998_villalba e --centroides-edificios
on
  st_intersects(_.geom,f.____) --edificios cubiertos por zona inundable
where
  f.____ = 'FLOODWAY' --en la zona marcada como 'FLOODWAY'
group by
  f.____; --resume por sub-zona inundable
```

Resultado:

Grid	A-Z zone_subty	123 num_edifs
1	FLOODWAY	0
2	FLOODWAY	47



Consideraciones importantes antes del análisis geográfico y geoprocursos

ÍNDICES ESPACIALES

La búsqueda de datos se hace más eficiente al establecer índices. Por ejemplo, es más fácil buscar un tema en un libro usando el índice. El índice del libro tendrá una lista de temas y la localización (en qué página se encuentra). De la misma manera trabajará un índice espacial, reduciendo el número de comparaciones entre filas de una o más tablas.

En PostgreSQL el algoritmo de indexación espacial se le llama [GiST \(Generalized Search Tree\)](#), el cual en PostGIS se aplica a geometrías/geografías también. Cuando los datos geográficos comienzan a tener miles de filas, es altamente recomendable asegurarse de que tengan índice geoespacial.

Al importar datos a PostGIS desde la interfaz DB Manager de QGIS, **recomendamos** crear índices espaciales.

Ya hemos conocido el operador **geom_a<->geom_b**, el cual devuelve la distancia entre las geometrías de dos geodatos. Este operador y otros no discutidos aquí, utilizan la indexación GiST.

Para aprovechar el uso de índices, podemos usar el operador doble ampersand && para comparar entre cajas de extensión territorial (bounding boxes) que encierran cada objeto geometry o geography. Este es el operador a utilizado cuando un predicado espacial no integra el uso de índices disponibles.



PREDICADOS/FUNCIONES ESPACIALES QUE USAN ÍNDICE GEOESPACIAL POR DEFECTO

Ya que los índices son tan importantes, deberíamos tener idea de **cuáles predicados topológicos/espaciales integran el uso de índices**.

Esto es, comparar entre bounding boxes de geometrías.

ST_Intersects	ST_Equals
ST_Crosses	ST_OrderingEquals
ST_Within	ST_DWithin
ST_Contains	ST_3DDWithin
ST_Covers	ST_3DDFullyWithin
ST_CoveredBy	ST_3DDIntersects
ST_ContainsProperly	ST_Touches
ST_Overlaps	

PostGIS nos da la opción de usar el predicado sin utilizar el índice. En esos casos, ponemos una barra baja `_` al inicio del predicado. Por ejemplo `_ST_Intersects` no usa índices. En general, no es común no utilizar índices geoespaciales, excepto en ciertas ocasiones específicas.

Por ejemplo, la función `ST_Relate(geom, geom, 'patrón')` no usa índices. Esta función la veremos más adelante y será necesario incluir el uso de índices de manera explícita.

CREACIÓN DE TABLAS A PARTIR DE UN QUERY

En ocasiones podemos aprovechar las filas y columnas devueltas por un query y convertirlos en una tabla. La forma recomendada por Martínez-Llario (2020) (pp. 114-117), es la siguiente:

1. Crear la tabla espacial receptora de los resultados de un query, comenzando con la sentencia `CREATE TABLE`, añadir las columnas/campos de interés, y la geometría.
2. Luego se insertan las filas y columnas mediante el comando `INSERT INTO`.

Ejemplo 1: Crear tabla definiendo los campos

```
CREATE TABLE workspaceXX.g15_suelos_agricolas_prime_2018
(gid serial primary key,
 farm_class varchar,
 geom geometry (multipolygon, 6566));

INSERT INTO g15_suelos_agricolas_prime_2018
(farm_class, geom)
SELECT farm_class, geom
FROM g15_suelos_soil_map_units_2018_villalba
WHERE farm_class <> 'Not prime farmland';
```

Explicación:

Creación de tabla:	
create table workspaceXX.g15_suelos_agricolas_prime_2018	create table Comando para crear la tabla con nombre <code>g15_suelos_agricolas_prime_2018</code> dentro del schema <code>workspaceXX</code> . Donde la <code>XX</code> corresponde a cada participante del taller.



<pre>(gid serial primary key, farm_class varchar, geom geometry (MULTIPOLYGON,6566));</pre>	Entre paréntesis van los nombres de los campos/columnas que van a crearse en la nueva tabla. gid serial primary key será el campo gid tipo serial y primary key. Serial es como el tipo <i>autonumber</i> en MS Access; números enteros que se incrementan automáticamente y no se repiten. farm_class será un campo tipo varchar, geom será el campo tipo geometry, que a su vez se definirá como 'MULTIPOLYGON' y con el sistema de referencia espacial EPSG:6566, que corresponde con STATE PLANE 5200, PR USVI, NAD83(2011), unidades en metros.
---	---

Insertar filas en la nueva tabla:	
<pre>insert into workspaceXX.g15_suelos_agricolas_prime_2018 (farm_class, geom)</pre>	Instrucción SQL para insertar valores en las filas de la nueva tabla <code>workspaceXX.g15_suelos_agricolas_prime_2018</code> en los campos <i>farm_class</i> y <i>geom</i>. No se incluye gid, ya que es tipo serial, como se explicó arriba.
<pre>select farm_class, geom from g15_suelos_soil_map_units_2018_villalba where farm_class <> 'Not prime farmland';</pre>	Luego se escribe el query que deseamos usar para insertar las filas. La condición en WHERE es que los suelos no estén clasificados como 'Not prime farmland'. Dicho de otro modo, todos los que sean tipo prime en todas sus variantes, y otros que están catalogados como suelos agrícolas de importancia estatal.

* Si no especificamos el **schema workspaceXX.tabla**, la tabla se va a crear dentro del schema "public".

¿Y si necesitamos todos los campos de la tabla original y pasarlos a la nueva tabla? En el próximo ejemplo, veremos **cómo copiar la estructura de una tabla existente a una tabla vacía**.

Ejemplo 2: Copiar la estructura de una tabla existente a una tabla nueva

El siguiente código sirve para copiar solamente la estructura de una tabla existente y generar una nueva tabla basada en esa estructura. Vamos a copiar la estructura de una tabla, incluyendo campos, y todas sus restricciones a la nueva tabla vacía.

Este procedimiento es análogo a hacer una selección de datos en un geodato y guardarlo con otro nombre (Save selected features as...).



```
CREATE TABLE workspaceXX.g15_suelos_agricolas_prime_2018
(like pr_geodata.g15_suelos_soil_map_units_2018_villalba
 including defaults
 including constraints
 including indexes);
```

```
INSERT INTO g15_suelos_agricolas_prime_2018
SELECT *
FROM g15_suelos_soil_map_units_2018_villalba
WHERE farm_class <> 'Not prime farmland';
```

**Explicación:

Creación de tabla:	
create table workspaceXX.g15_suelos_agricolas_prime_2018	create table: Comando para crear la tabla con nombre g15_suelos_agricolas_prime_2018 dentro del schema workspaceXX . *
(like g15_suelos_soil_map_units_2018_villalba including defaults including constraints including indexes);	"like" es la palabra clave para copiar estructura, incluyendo <i>defaults</i> o valores que adopta por defecto, <i>constraints</i> o restricciones y todos los índices que existan relacionados con la tabla fuente, que en este caso es la tabla g15_suelos_soil_map_units_2018_villalba

Insertar filas en la nueva tabla:	
insert into workspaceXX.g15_suelos_agricolas_prime_2018	Instrucción SQL para insertar valores en las filas de la tabla g15_suelos_agricolas_prime_2018 en todos los campos .
select * from g15_suelos_soil_map_units_2018_villalba where farm_class <> 'Not prime farmland';	Luego se escribe el query que deseamos usar para insertar las filas. La condición en WHERE es que los suelos no estén clasificados como 'Not prime farmland'. Dicho de otro modo, todos los que sean tipo prime en todas sus variantes, y otros que están catalogados como suelos agrícolas de importancia estatal.

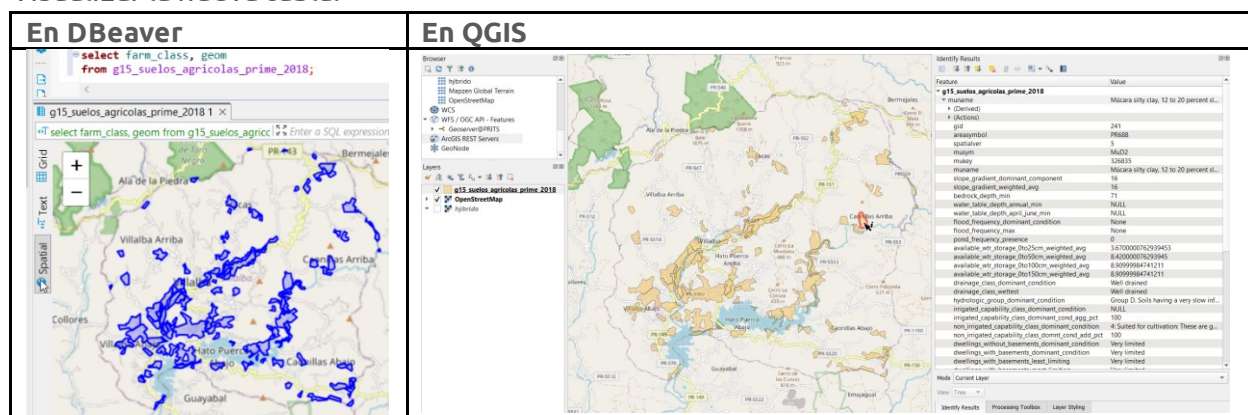
** Si no especificamos el **schema workspaceXX.tabla**, la tabla se va a crear dentro del schema "**public**".

Nota: Tener en cuenta que las secuencias del campo *gid* **serial** de la nueva tabla serán iguales a los de la tabla original.



PostGIS, 3.x

Visualizar la nueva tabla:



HOMOGENIZAR GEOMETRÍAS DEVUELTAS AL SOBREPONER GEODATOS

Al sobreponer geometrías en PostGIS es posible que el resultado del query esté compuesto de varios tipos de geometrías. Por ejemplo:

Sobreposiciones geométricas	Geometrías devueltas al sobreponer
P/P	Punto
P/L, L/P	Punto
P/S, S/P	Punto
L/L	Punto y/o línea
L/S, S/L	Punto y/o línea
S/S	Punto, línea, polígono(S)

* En algunos casos, PostGIS puede devolver filas con geometrías vacías.

Los SIG de interfaz gráfica, QGIS, ArcGIS y otros, **no** están diseñados por el momento para manejar una columna con *más de un tipo de geometría*. Por lo tanto, es preferible homogenizar los resultados geométricos devueltos por las funciones de superposición tales como [ST_Intersection](#)(geom, geom) entre otras. (Martínez-Llario, 2020, pp. 131-134)



Hagamos un ejemplo que involucre dos geodatos con geometría de polígonos.

❑ Escriba y ejecute el siguiente query en la pestaña **Script-1 de DBEaver**:

```
/* Tipos de geometría que devuelve ST_Intersection()
 * al hacer overlay de landslides y plan de usos de terrenos, 2015
 * condición: where st_intersects(ls.geom,put.geom)
 */
select --1a parte:
    geometrytype(geom) as tipo_geom,    --listar tipos de geometría
    count(*)                          --conteo de elementos geométricos
from (--2a parte:
    select ST_Intersection(ls.geom,put.geom) as geom --overlay
    from
        g15_riesgo_geol_landslides_1979_villalba ls, --landslides
        g29_planes_put_2015_villalba put --plan de usos terr, 2015
    where
        st_intersects(ls.geom,put.geom) --donde haya intersección
    ) as tabla_subquery                --nombre del subquery
group by tipo_geom;                   --usa los tipos de geom para hacer conteo
```

tipo_geom	count
GEOMETRYCOLLECTION	4
LINESTRING	3
MULTIPOLYGON	79
POINT	6
POLYGON	1027

1ª parte: usar la función **GeometryType(geom)** as **tipo_geom** y el conteo de elementos.

2ª parte: Hacer **subquery**. Uso de **ST_Intersection(a.geom, b.geom)** as **geom** entre las tablas **g23_riesgo_geol_landslides_villalba** as **a** y **g29_planes_put_2015_villalba** as **b**

Condición: (WHERE) **st_intersects(a.geom, b.geom)** alias "tabla_subquery"

3a parte: **GROUP BY**, agrupa los resultados **por** la columna **tipo_geom**, que está en la primera parte o query exterior.

Como parte de este query de intersección geométrica **ST_Intersection**, PostGIS devuelve en la columna **tipo_geom**: **GEOMETRYCOLLECTION**, **LINESTRING**, **MULTILINESTRING**, **MULTIPOLYGON**, **POINT** y **POLYGON**.

Una de las formas para tratar de evitar que se produzcan distintos tipos de geometría devueltos en un query es utilizar el predicado [ST_Relate](#)(geom, geom, 'patrón'). 'Patrón' se refiere a uno de los múltiples patrones existentes al relacionar o sobre imponer geometrías, según la matriz de intersección Egenhofer/Clementini discutidas en los apéndices [A-1](#) y [A-2](#).

Si vamos a **intersecar áreas superficiales o polígonos (S)** y queremos obtener solo polígonos, podemos usar **ST_Relate(geom, geom, 'T*****')** para solo intersecar el interior de una geometría con el interior de la otra geometría. En el caso de intersecar los interiores de S/S, estos devolverán polígonos (S).

Ya que **ST_Relate** no utiliza por defecto los índices, antes será necesario forzar su utilización mediante el **operador de comparación de bounding boxes** doble ampersand **&&**.



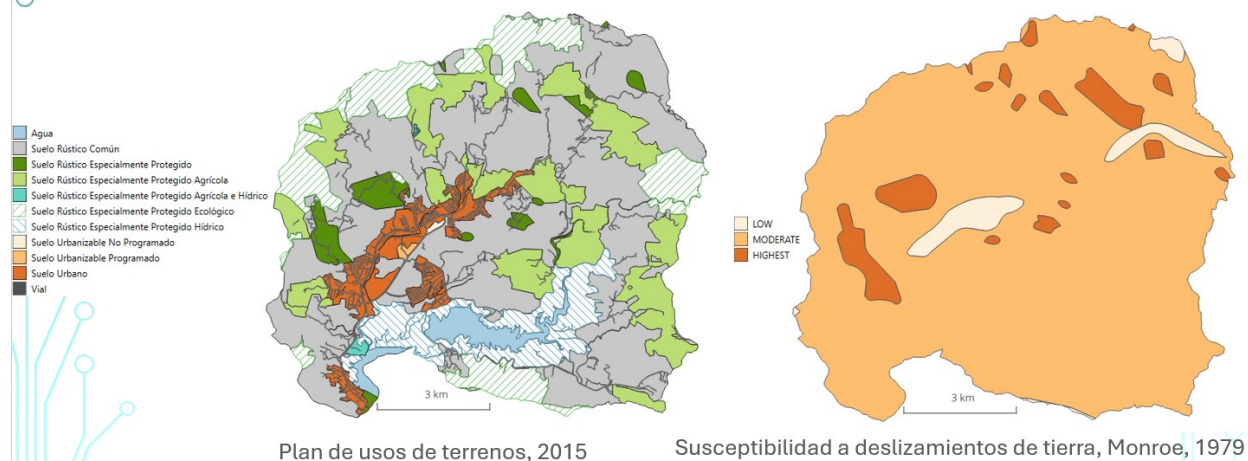
Por ejemplo, en la cláusula WHERE:

WHERE a.geom && b.geom AND ST_Relate(a.geom, b.geom, 'T***');**

Los resultados devueltos pueden combinarse en geometrías vacías, tipo MULTI o SIMPLE y GeometryCollection.

Hagamos un ejemplo que involucre dos geodatos con geometría de polígonos y usar el predicado **ST_Relate** en la cláusula WHERE y añada las geometrías usando el campo geom.

OVERLAY = ?



☐ Escriba y ejecute el siguiente query en la pestaña **Script-1 de DBEaver:**

```
/* Tipos de geometría que devuelve ST_Intersection()
 * al hacer overlay de landslides y plan de terrenos, 2015
 * condición: where st_intersects(ls.geom, put.geom)
 */
select                                --1a parte:
    geometrytype(geom) as tipo_geom,  --listar tipos de geometría
    geom                             --traer elementos geométricos
from (                                --2a parte:
    select ST_Intersection(ls.geom, put.geom) as geom    --overlay
    from
        g15_riesgo_geol_landslides_1979_villalba ls,    --landslides
        g29_planes_put_2015_villalba put                --plan de usos terr, 2015
    where
        ls.geom && put.geom                             --usar índice geoespacial
        and ST_Relate(ls.geom, put.geom, 'T*****')    --intersección de interiores
    ) as tabla_subquery                                --nombre del subquery
group by tipo_geom, geom;                             --usa los tipos de geom para hacer conteo
```



PostGIS, 3.x

select geometrytype(geom) as tipo_geom, geom from (select st_inter... Enter a SQL expression to filter results (use Ctrl+Space)

	tipo_geom	geom
1	GEOMETRYCOLLECTION	GEOMETRYCOLLECTION (POLYGON ((196892.2
2	GEOMETRYCOLLECTION	GEOMETRYCOLLECTION (POLYGON ((195166.7
3	GEOMETRYCOLLECTION	GEOMETRYCOLLECTION (POLYGON ((201249.1
4	GEOMETRYCOLLECTION	GEOMETRYCOLLECTION (POLYGON ((197448.0
5	MULTIPOLYGON	MULTIPOLYGON (((191960.1195999998 230641.1
6	MULTIPOLYGON	MULTIPOLYGON (((192339.9298999998 230609.1
7	MULTIPOLYGON	MULTIPOLYGON (((192405.54461897278 23120
8	MULTIPOLYGON	MULTIPOLYGON (((191681.19637130725 23084
9	MULTIPOLYGON	MULTIPOLYGON (((191701.7032908257 231021
10	MULTIPOLYGON	MULTIPOLYGON (((191083.43590000272 23246

Refresh Save Cancel Export data 200 200+

☐ Haga **click** en la **celda 1** bajo la columna geom.

1ª **parte**: traer tipos de geometría **GeometryType(geom)** as **tipo_geom** y las geometrías.

2ª **parte**: Hacer **subquery**. Uso de **ST_Intersection(a.geom, b.geom)** as **geom** entre las tablas **g23_riesgo_geol_landslides_villalba** as **a** y **g29_planes_put_2015_villalba** as **b**

Condición: (WHERE) **a.geom && b.geom** para inducir uso de comparador de bounding boxes

ST_Relate(a.geom, b.geom, 'T***')** alias "tabla_subquery"

3ª **parte**: **GROUP BY**, agrupa los resultados **por** las columnas **tipo_geom** y **geom**, que están en la primera parte o query exterior.

Usamos **ST_Relate** con el patrón **'T*****'** para evitar que nos devuelva geometrías lineales (LINESTRING) ni puntos (POINT). En los resultados, el campo **tipo_geom** no vemos ni POINT ni LINESTRING y vemos que devolvió tres tipos de geometría: **GEOMETRYCOLLECTION**, **MULTIPOLYGON** y **POLYGON**. **Vamos a usar la interfaz gráfica de DBeaver para indagar.**

- ☐ En la pestaña **Results 1**, vaya al campo **geom** y haga **click** en la **cuarta celda** **GEOMETRYCOLLECTION (POLYGON....**
- ☐ En la pestaña **Value** aparecerán los elementos que componen esta **GEOMETRYCOLLECTION**. Si deja el cursor encima de esta celda, le mostrará los elementos y la lista de coordenadas

select geometrytype(geom) as tipo_geom, geom from (select st_inter... Enter a SQL expression to filter results (use Ctrl+Space)

	tipo_geom	geom
1	GEOMETRYCOLLECTION	GEOMETRYCOLLECTION (POLYGON ((196892.2
2	GEOMETRYCOLLECTION	GEOMETRYCOLLECTION (POLYGON ((195166.7
3	GEOMETRYCOLLECTION	GEOMETRYCOLLECTION (POLYGON ((201249.1
4	GEOMETRYCOLLECTION	GEOMETRYCOLLECTION (POLYGON ((197448.0
5	MULTIPOLYGON	MULTIPOLYGON (((191960.1195999998 230641.1
6	MULTIPOLYGON	MULTIPOLYGON (((192339.9298999998 230609.1
7	MULTIPOLYGON	MULTIPOLYGON (((192405.54461897278 23120
8	MULTIPOLYGON	MULTIPOLYGON (((191681.19637130725 23084
9	MULTIPOLYGON	MULTIPOLYGON (((191701.7032908257 231021
10	MULTIPOLYGON	MULTIPOLYGON (((191083.43590000272 23246

Refresh Save Cancel Export data 200 1,110

- ☐ Podrá ver que **aun cuando se usó ST_Relate y el patrón 'T*****'**, esto no impidió que devolviera **Linestrings** en dos filas tipo **GEOMETRYCOLLECTION**. Además el query devuelve tanto **MULTIPOLYGON** como **POLYGON**.

En PostGIS no hay inconvenientes con esto en cuanto a su almacenamiento y visualización en DBeaver. Sin embargo, **este resultado no es compatible con un SIG de interfaz gráfica como QGIS o ArcGIS**. Por tal motivo, **se hace necesario homogenizar las geometrías**



devueltas por los queries. Existe una función **ST_CollectionExtract**, sin embargo, esta función puede devolver geometrías vacías (NULL) tipo GEOMETRYCOLLECTION EMPTY y sólo convierte a MULTI los elementos dentro de un GEOMETRYCOLLECTION (Martínez Llario, 2020, pp. 131-34).

Otro problema de fondo es la **cantidad de geometrías insignificantes** que devuelven estas funciones. Casi todas las aplicaciones que usan el estándar Simple Features, pueden producir estos elementos ya que en sus procesos, no necesariamente están aplicando tolerancias al hacer overlay analysis. Sería ideal que PostGIS utilizara algún tipo de “cluster tolerance” como las aplicaciones como GRASS o ArcGIS. PostGIS incluye el uso de topología, pero su manejo está fuera del alcance de este tutorial.



UTILIZACIÓN DE FUNCIÓN EXTRA: STX_EXTRACT

Para evitar las situaciones mencionadas anteriormente, el Prof. Martínez Llario desarrolló una función en lenguaje PL/PgSQL llamada **STX_Extract()**, la cual modifica el comportamiento de la función **ST_CollectionExtract**, permitiendo que este tipo de queries devuelvan geometrías de manera homogénea.

- Si la geometría devuelta está vacía, devolverá un null (la columna geom puede almacenar null).
- Solo devuelve entidades **MULTI**.
- Tiene un **segundo argumento para** especificar la **dimensión de las geometrías de salida: 0 para puntos, 1 para líneas y 2 para polígonos/superficies**.

```
DECLARE
geom alias for $1;
dimension alias for $2;
sencillo alias for $3;
out Geometry;
tipo Varchar;
dimensioncal integer;
BEGIN
IF (ST_IsEmpty (geom)) THEN
RETURN null;
END IF;

tipo:= geometrytype(geom);
dimensioncal:=st_dimension(geom);

--Una colección es: multipoint, multilinestring, multipolygon,
--geometrycollection, multisurface, multicurve, compoundcurve

--Si no es una colección, devuelve el elemento si coincide
--la dimensión solicitada con la dimensión del elemento
IF (st_iscollection(geom) = false) THEN
IF (dimension <> dimensioncal) THEN RETURN null; END IF;
IF (sencillo = false) THEN RETURN ST_Multi(geom); END IF;
RETURN geom;
END IF;

--Si Es una colección extrae los subelementos que tienen la
--dimensión solicitada. No extrae subelementos curvos.

out:= ST_collectionextract (geom, dimension + 1);
if (sencillo = false) THEN out:= ST_Multi(out);

--Devuelve un elemento sencillo
ELSE out:= ST_GeometryN (out, 1);
END IF;

IF (ST_IsEmpty (out)) THEN
RETURN null;
END IF;

RETURN out;
END;
```

Este es el script de la función STX_Extract. Esta base de datos tiene esta función instalada. El código puede encontrarlo en <https://cartosig.web.upv.es> (Universidad Politécnica de Valencia, España).

La dirección contiene este y otros scripts. Las funciones SQL deberán ejecutarse **en Windows desde el usuario Administrator** o **"run as Administrator"**.

Consola> psql -U postgres -f ruta_al_archivo_funcionesextra.sql base_de_datos

psql se ejecuta desde el folder de instalación de Postgres/PostGIS o haber declarado el path en las variables de System en Windows. Puede ser:
...\\PostgreSQL\\release_number\\bin (Martínez Llario, 2017, pp. 31-32).

Vamos a usar el query anterior con la función extra **STX_Extract** para demostrar el resultado.

- ☐ Modifique el query anterior en la pestaña **Script-1 de DBeaver**:

```
* Tipos de geometría que devuelve ST_Intersection()
* al hacer overlay de landslides y plan de usos de terrenos, 2015
* condición: where st_intersects(ls.geom,put.geom)
*/
```




```

select                                --1a parte:
  geometrytype(geom) as tipo_geom,    --listar tipos de geometría
  count(*)                            --conteo de geometrías
from (                                --2a parte:
  select STX_Extract(                  --función extra ...
    ST_Intersection(ls.geom,put.geom)
    ,2) as geom                        --overlay: opción 2=devolver polígonos
  from
    g15_riesgo_geol_landslides_1979_villalba ls, --landslides
    g29_planes_put_2015_villalba put --plan de usos terr, 2015
  where
    ls.geom && put.geom
    and ST_Relate(ls.geom,put.geom,'T*****') --intersección
                                           --de interiores de geometrías
  ) as tabla_subquery                  --nombre del subquery
group by tipo_geom--,geom;             --usa los tipos de geom para hacer conteo

```

Resultados:	Grid	1	MULTIPOLYGON	1,110
-------------	------	---	--------------	-------

Podrá notar que el query anterior le devuelve solamente 1,100 MULTIPOLYGONS. Esta vez no devolvió elementos vacíos (NULL). Recuerde que PostGIS permite guardar geometrías vacías (NULL).

Si no desea que le devuelva filas NULL, inserte lo siguiente en la cláusula WHERE, en el query exterior: **WHERE geometrytype(geom) notnull**

```

select                                --1a parte:
  geometrytype(geom) as tipo_geom,    --listar tipos de geometría
  count(*)                            --conteo de geometrías
from (                                --2a parte:
  select stx_extract(                  --función extra ...
    st_intersection(ls.geom,put.geom) --intersección geométrica
    ,2) as geom                        --overlay opción 2=devolver polígonos
  from
    g15_riesgo_geol_landslides_1979_villalba ls, --landslides
    g29_planes_put_2015_villalba put --plan de usos terr, 2015
  where
    ls.geom && put.geom --usar índice geoespacial
    and st_relate(ls.geom,put.geom,'T*****') --intersección
                                           --de interiores de geometrías
  ) as tabla_subquery                  --nombre del subquery
where
  geometrytype(geom) notnull           --no devolver geometrías null
group by tipo_geom;                   --usa los tipos de geom para hacer conteo

```




TOLERANCIA EN EL ANÁLISIS ESPACIAL

Ya hemos visto en el ejemplo anterior que la función de intersección **ST_Intersection(geomA,geomB)**. No debemos confundir **ST_Intersection** con el predicado **ST_Intersects**. El predicado **ST_Intersects** devuelve TRUE o FALSE al comparar dos geometrías. La función **ST_Intersection** devolverá geometría(s) o null.

¿Qué situación encontramos al usar **ST_Intersection**? Primero, al hacer la prueba CIERTO o FALSO el predicado **ST_Intersects** da paso a cualquier tipo de geometría que pase la prueba de intersección: ya sea de punto, línea o área. Vimos también cómo intentamos “cerrar el paso” a otras geometrías al usar **ST_Relate** con el patrón ‘T*****’. El patrón se usó para especificar que solo las geometrías que se intersecan interiores serán seleccionadas. Entonces, nos devolvió cuatro filas de geometrías tipo GEOMETRYCOLLECTION que tenían polígonos y segmentos de línea de dos vértices y puntos. Además, el resto de los polígonos fueron tanto MULTIPOLYGON como POLYGON.

Luego vimos cómo la función extra **STX_Extract** nos ayudó a poder ser más selectivos y solamente devolver MULTIPOLYGON. Sin embargo, esto no resuelve el problema de fondo. Lo que hace es proveer un remedio a un problema generado desde PostGIS y todos los programas SIG que usan las bibliotecas [GEOS](#) y que operan con el estándar discutido anteriormente en la sección [Simple Features](#) de este manual. Las operaciones de intersección nos pueden devolver geometrías insignificantes. Por tal razón el modelo de tolerancias ha sido un buen recurso en los geoprocесamientos, tanto para ArcGIS, ArcInfo y para GRASS. Estos programas usan un modelo topológico persistente y ‘planarizan’ las geometrías, forzando la creación de nodos en cada intersección entre geometrías. En estos programas se establece un umbral de tolerancia, con el cual podemos eliminar geometrías insignificantes. Las cuatro GEOMETRYCOLLECTION que nos generó **ST_Intersection** eran geometrías insignificantes, al igual que otras geometrías minúsculas que pudieron haber sido generadas.

¿Qué criterio usamos para definir umbrales de tolerancia?

Dependerá principalmente de la **escala de compilación** en que se preparó el geodato. Originalmente los mapas eran preparados en un medio fijo, ya fuera papel, acetato (mylar) y otros. La escala de compilación era fija y se establecía una equivalencia entre distancias en el mapa y en el terreno.

En los [estándares de exactitud de mapas de los EE. UU.](#) Se establece que para mapas a escala 1:24,000 de la serie de mapas topográficos de 7.5 minutos la exactitud horizontal requerida para el 90% de los puntos escogidos para prueba, estos deben tener una exactitud alrededor de 1/50 de pulgada (0.05 cm) en el mapa.

A una escala de 1:24,000, 1/50 de pulgada son 40 pies (12.2 m). Más adelante continúa este escrito: **para mapas con escala de publicación mayor de 1:20,000, no más del 10% de los puntos para prueba deberán tener un error de 1/30 de pulgada, medido en la escala de publicación.** Estos puntos aplican solamente a “puntos bien definidos”, tales como **benchmarks, intersecciones de carreteras y líneas de ferrocarril, y esquinas de edificios.**



Martínez Llario (2020), pág. 215 muestra una **relación de exactitud cartográfica** entre los estándares de **EEUU** y de **España**:

Escala	Exactitud cartográfica en metros, según el intervalo de confianza	
	EE UU (95%)	España (99%)
1:100	0.06	0.02
1:500	0.31	0.10
1:1,000	2.45	0.20
1:5,000	3.06	1.00
1:20,000	12.25	4.00

Tomado de Martínez Llario (2020), p. 215.

Tomando esto en consideración, **la exactitud varía en función a la escala**. En muchos casos, dependerá del nivel de detalle del dato fuente, que puede ser una ortofotografía de alrededor de 0.3 x 0.3 m de resolución, así como el nivel de acercamiento utilizado al derivar datos de la ortofoto. Debemos buscar en los metadatos de la fuente para saber cuál es la escala de salida de este tipo de imagen para tener una idea del nivel de exactitud. A ese error se le sumarán otros errores al derivar datos de este tipo de imágenes.

Como mencionamos antes, tanto PostGIS y todos los programas que utilizan las bibliotecas [GEOS](#) y el estándar simple features, no tienen procedimientos para establecer umbrales de tolerancia como en GRASS o ArcInfo (ahora ArcGIS con [cluster tolerance](#)).

Sin embargo, esto no es excusa para descartar estos productos. Las funcionalidades que nos ofrecen han sido de mucha utilidad. Por otro lado, PostGIS está desarrollando topología persistente y se espera que fortalezcan este tipo de modelo para representar geodatos en un futuro no muy lejano.

Una opción que se ofrece es usar la función [ST_Snap](#)(geomA,geomB, dist). Esto es bien parecido al caso de usar cluster tolerance en ArcGIS, dando prioridad a la geometría de A. **Los vértices de la geometría de B serán “pegados” a los vértices de la geometría de A, siempre y cuando estén dentro de la distancia de tolerancia.**



Si el query anterior nos devolvió múltiples tipos de geometrías:

```
/* Tipos de geometría que devuelve ST_Intersection()
 * al hacer overlay de landslides y plan de usos de terrenos, 2015
 * condición: where st_intersects(ls.geom,put.geom)
 */
select                                --1a parte:
    geometrytype(geom) as tipo_geom,  --listar tipos de geometría
    count(*)                        --conteo de elementos geométricos
from (                               --2a parte:
    select ST_Intersection(ls.geom,put.geom) as geom      --overlay
    from
        g15_riesgo_geol_landslides_1979_villalba ls,    --landslides
        g29_planes_put_2015_villalba put                --plan de usos terr, 2015
    where
        ls.geom && put.geom                            --usar índice geoespacial
        and ST_Relate(ls.geom,put.geom,'T*****')      --intersección
                                                -- de interiores
    ) as tabla_subquery                                --nombre del subquery
group by tipo_geom;                                   --usa los tipos de geom para hacer conteo
```

tipo_geom	count
GEOMETRYCOLLECTION	4
MULTIPOLYGON	79
POLYGON	1027

Qué hace ST_Snap: Esta función **mueve** los **vértices** y **segmentos** de una geometría de un geodato y los **engancha** a las **geometrías de otro geodato**.

Por ejemplo, en el siguiente query se usa ST_Snap para mover los vértices y segmentos de las geometrías de landslides y engancharlos en las geometrías del plan de usos de terrenos. Las geometrías del geodato del plan de usos de terrenos no son modificadas.

Al usar [ST_Snap](#) con un umbral de tolerancia escogida de 0.25 metros para una escala de 1:1,000, solo devuelve geometrías tipo POLYGON si lo insertamos en el siguiente query modificado del anterior.



- Modifique el query anterior en insertando la función ST_Snap en el **Script-1 de DBeaver**

```
/* Tipos de geometría que devuelve ST_Intersection()
 * al hacer overlay de landslides y plan de usos de terrenos, 2015
 * condición: where st_intersects(ls.geom,put.geom)
 */
select                                --1a parte:
    geometrytype(geom) as tipo_geom, --listar tipos de geometría
    count(*)                      --conteo de elementos geométricos
from (                             --2a parte:
    select ST_Intersection(--intersección geométrica put &
                           --resultado de st_snap
                           ST_Snap(ls.geom,put.geom,0.25), --st_snap:
                           --enganchar vértices de ls en put
                           ls.geom) --realiza intersección ls y put
        as geom --alias geom

    from

        g15_riesgo_geol_landslides_1979_villalba ls, --landslides
        g29_planes_put_2015_villalba put --plan de usos terr, 2015

    where

        ST_Intersects(ls.geom,put.geom) --test intersección geometrías
    ) as tabla_subquery                --nombre del subquery
group by tipo_geom;                   --usa los tipos de geom para hacer conteo
```

tipo_geom	count
MULTIPOLYGON	864
POLYGON	255
Suma de geometrías: 1,119	

Si cambiamos, el orden de geodatos, los resultados pueden ser diferentes. Veamos.

- Modifique el query anterior en insertando la función ST_Snap en el **Script-1 de DBeaver**

```
/* Tipos de geometría que devuelve ST_Intersection()
 * al hacer overlay de landslides y plan de usos de terrenos, 2015
 * condición: where st_intersects(ls.geom,put.geom)
 */
select                                --1a parte:
    geometrytype(geom) as tipo_geom, --listar tipos de geometría
    count(*)                      --conteo de elementos geométricos
from (                             --2a parte:
    select ST_Intersection( --intersección geométrica put &
                           --resultado de st_snap
                           ST_Snap(put.geom,ls.geom,0.25), --st_snap: enganchar vértices
                           --de put en ls
                           put.geom) --realiza intersección ls y put
        as geom --alias geom
```



```
from
    g15_riesgo_geol_landslides_1979_villalba ls, --landslides
    g29_planes_put_2015_villalba put --plan de usos terr, 2015
where
    ST_Intersects(ls.geom,put.geom) --test intersección geometrías
) as tabla_subquery --nombre del subquery
group by tipo_geom; --usa los tipos de geom para hacer conteo
tipo_geom|count|
-----+-----+
POLYGON | 1119|
```

Al invertir el orden de geodatos en **ST_Snap**, la cantidad de geometrías devueltas es igual: 1,119. Mover las geometrías de *landslides* (ls.geom) a las del *plan de usos de terrenos* (put.geom) tarda un poco más.

Geodatos	Tipo de geometría devuelta	Conteo	Tiempo de ejecución
ST_Snap(put.geom,ls.geom)	POLYGON	1,119	41 segundos
ST_Snap(ls.geom,put.geom)	MULTIPOLYGON POLYGON	864 255	59 segundos

Comparación del tiempo de ejecución al intercambiar el orden de geodatos al usar la función ST_Snap.

El problema de usar ST_Snap es que puede **alterar la integridad de las adyacencias de las geometrías**. No se recomienda por ejemplo, **usar ST_Snap en geodatos como parcelas o límites administrativos**, ya que alterar los vértices puede **afectar la validez** de los datos: crear solapes (slivers) y aberturas entre áreas (gaps).

Desde la versión 2.0 de PostGIS, se incluyó una geometría diferente “topogeometry” para integrar topología. Sin embargo, las funciones geométricas no están diseñadas para topologías, por lo menos, todavía. Una **solución intermedia** es **utilizar la función STX_Extract** preparada por Martínez Llario (2020, pp. 131-34). La podemos integrar cuando utilicemos las funciones ST_Intersection, ST_Buffer, ST_Difference, etcétera.



Como vimos en ejemplos anteriores, la función **STX_Extract** “filtra” del resultado de ST_Intersection solamente las geometrías tipo “#2” = polígonos y las **convierte a MULTIPOLYGON**. .

- Modifique el query anterior en insertando la función STX_Extract en el **Script-1 de DBever**

```
select --1a parte:
  geometrytype(geom) as tipo_geom, --listar tipos de geometría
  count(*) --conteo de geometrías
from ( --2a parte:
  select STX_Extract( --función extra ...
    ST_Intersection(ls.geom,put.geom) --intersección geométrica
    ,2) as geom --overlay opción 2=devolver polígonos
  from
    g15_riesgo_geol_landslides_1979_villalba ls, --landslides
    g29_planes_put_2015_villalba put --plan de usos terr, 2015
  where
    ST_Intersects(ls.geom,put.geom) --intersección
  ) as tabla_subquery --nombre del subquery
where
  geometrytype(geom) notnull --no devolver geometrías null
group by tipo_geom; --usa los tipos de geom para hacer conteo
```

Grid	1	MULTIPOLYGON	1,110
Tiempo de ejecución: 1.8 segundos			

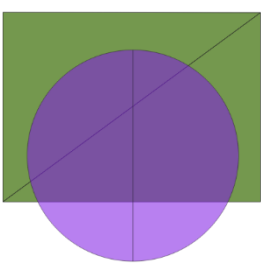
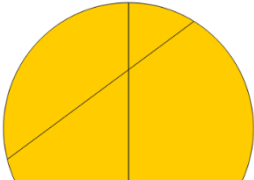
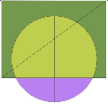
La solución ideal sería que se pudiese establecer tolerancias en estas operaciones y hacerlas más eficientes.



Análisis geoespacial

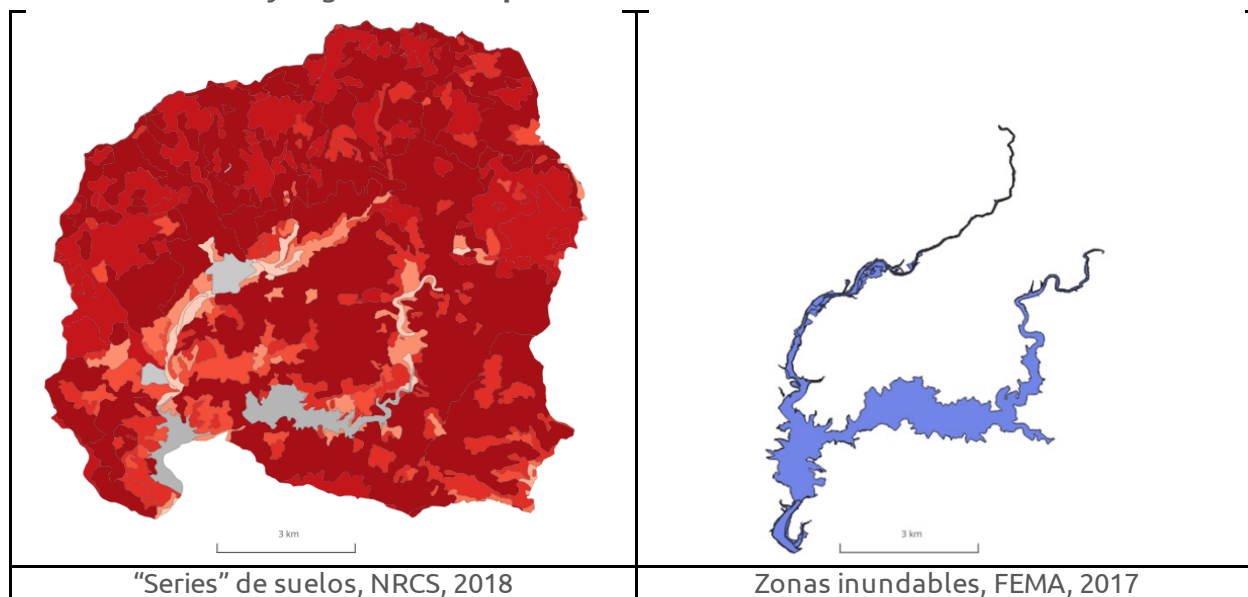
En adelante vamos a usar la función **STX_Extract** cuando sea pertinente en los queries de esta sección de análisis geoespacial, como vimos, para homogenizar las geometrías de los resultados.

INTERSECCIÓN

Layers de entrada		Layer resultante ($A \cap B$)
 Layer A		
 Layer B		 Geometrías originales y resultado en amarillo
Dimensiones	Dimensiones posibles: 0, 1 o 2 Los inputs pueden tener geometrías diferentes	La dimensión de salida va a ser igual o menor que los layers de entrada
Atributos, campos, o columnas	--	Los campos de salida serán la combinación de los campos de A y B .

Ejemplo:

- ☐ Realice un **cruce de geodatos** (mediante intersección) entre el geodato de **zonas inundables** y el geodato de **tipos de suelos**.



¿Cómo debe ser el resultado de este proceso?



□ Escriba y ejecute los siguientes queries en el **Script-1 de DBeaver**:

```
--sección crear tabla geoespacial
create table g00_overlays_intersect_soil_zi
(
  gid serial primary key,
  su_gid integer, --recibirá valores de su.gid
  musym varchar,
  muname varchar,
  farm_class varchar,
  zi_gid integer, --recibirá valores de zi.gid
  fld_zone varchar,
  zone_subty varchar,
  geom geometry(multipolygon,6566)
);

--sección análisis geo: INTERSECT
insert into g00_overlays_intersect_soil_zi --nuevo geodato
(su_gid,musym,muname,farm_class, --campos de suelos
zi_gid,fld_zone,zone_subty, --campos de zonas inundables
geom) --recibirá geometrías del proceso intersect
select su.gid, --se refiere a gid de tabla origen suelos
musym,muname,farm_class, --campos suelos
zi.gid, --se refiere a gid de tabla origen ziundables
fld_zone,zone_subty, --campos zonas inundables
STX_Extract( --función, recibe result de ST_Intersection...
  ST_Intersection(su.geom,zi.geom) --intersección geom
  ,2) --filtra resultado. 2=MULTIPOLYGON
as geom --alias geom

from
pr_geodata.g15_suelos_soil_map_units_2018_villalba as su,
pr_geodata.g23_riesgo_inunda_floodzones_2017_villalba as zi
where su.geom && zi.geom --inducir uso de índice geosp
and ST_Relate(su.geom,zi.geom,
'T*****'); --test: intersección de interiores geoms

select * from g00_overlays_intersect_soil_zi; --ver resultado en DBeaver
```

Resultado en DBeaver:



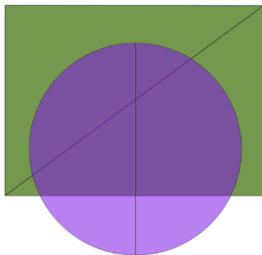
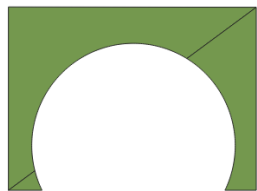
La intersección solo incluye áreas comunes y los valores correspondientes de los campos incluidos en el query



El proceso de escribir código al inicio es tedioso y susceptible a cometer errores. Por el lado positivo, hay mayor control sobre el resultado y además el procedimiento queda registrado en el código. Si comenta el código es mejor aún porque cuando lo vuelva a ver, entenderá qué hace cada parte, además que otros podrán entender la lógica del código escrito.

ERASE

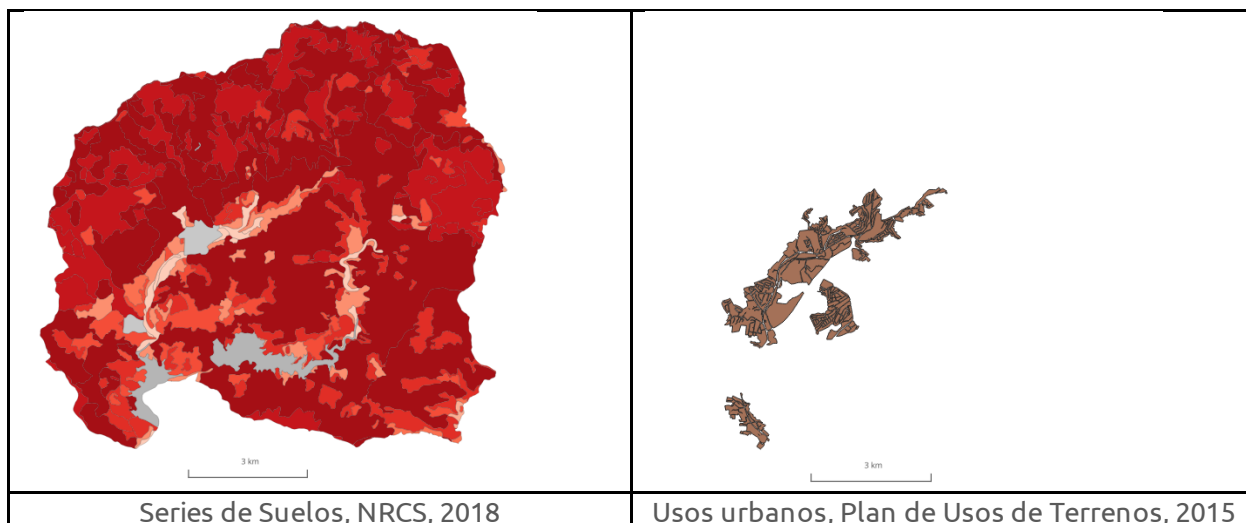
[Erase](#) es un geoproceto que remueve el área de solape entre la geometría de un geodato A y un geodato B.

Layers de entrada		Layer resultante (A-B)
 Layer A		
 Layer B		
Dimensiones geométricas	Dimensiones posibles: 0, 1 o 2 La dimensión del layer B debe ser igual a, o superior a la dimensión del layer A .	El resultado será de la misma dimensión de la capa A.
Conmutativa	Es conmutativa. No importa el orden.	--
Atributos, campos, o columnas	--	Conserva los campos del layer A .

Este geoproceto se realiza en PostGIS mediante la combinación de dos funciones: [ST_Difference](#) y [ST_Union](#). No debe usarse solamente ST_Difference (no es como el geoproceto *Difference* de QGIS), porque ST_Difference usada sola va a producir polígonos erróneos. Se recomienda usar ST_Union para crear solo una geometría en el layer B y luego sobre imponerla al layer A. Este proceso de unión puede ser ineficiente si generamos polígonos grandes con miles de vértices.

Lo más **recomendable** es: **quitar a cada polígono del layer A el área de intersección de la unión de los polígonos del layer B**. En otras palabras, es **enfocar la búsqueda y geoproceto solo en el área de solape**.

Ejemplo: Usar geometrías de **suelos urbanos, urbano-programado, urbano-no-programado** del geodato del **plan de usos de terrenos, 2015** para **recortarlas (erase)** del geodato del **catastro de suelos de USDA-NRCS, 2018**.



☐ Escriba y ejecute los siguientes queries en el **Script-1 de DBeaver**:

```
--Generar nuevo geodato
create table g00_overlay_erase_suelos_menos_urbanos
(
    --sección para crear campos de la tabla, dentro de paréntesis
    gid serial primary key,      --gid para id interno primary key
    musym varchar, muname varchar, farm_class varchar,
    -- campos, tabla suelos
    geom geometry(multipolygon,6566) --geom multipolygon, epsg:6566
); -- fin de enunciado create table

--parte analítica: ERASE
insert into g00_overlay_erase_suelos_menos_urbanos --insertar filas en tabla
(musym,muname,farm_class,geom) --lista de campos suelos
select musym,muname,farm_class, geom --trae estos campos de outer query
from
(select
    s.gid,musym,muname,farm_class, --trae campos del subquery
    STX_Extract( --extraer geometrías
    ST_Difference(s.geom, --función difference
    st_union(u.geom)) --unir polígonos usos urbanos
    ,2) as geom --opción 2 = multipolygons
from
    g15_suelos_soil_map_units_2018_villalba s, --de la tabla suelos
    g29_planes_put_2015_villalba u --de la tabla plan usos terrenos
where clasiput like 'SU%' --solo suelos urbanos
    or clasiput = 'VIAL' --y viales
    and st_intersects(s.geom,u.geom)
group by s.gid --Dependencia funcional
) as subq --nombre alias de este subquery
where geom is not null; --no devolver geometrías nulas.

select * from g00_overlay_erase_suelos_menos_urbanos; --ver los resultados
```



¿Qué es group by s.gid?

En la sección GROUP BY del pasado query no se listaron los campos musym, muname, etc. Por el contrario, solamente se añadió el campo gid de la tabla "s" o geodato de suelos a esto se le llama Dependencia Funcional. Esto lo discutimos en la siguiente sección.

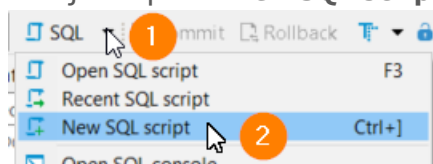
QUÉ ES DEPENDENCIA FUNCIONAL

Antes de seguir, aprovechamos para explicarlo. Desde la versión 9.1 de PostgreSQL se permite usar solo el campo de identificadores, en este caso, gid dentro de la cláusula GROUP BY para no tener que declarar todos los campos que están contenidos en la cláusula SELECT. Dependencia funcional tiene que ver con la lógica de normalización de datos en las tablas. PostgreSQL toma como cierto que solo existe un valor para cada columna de la tabla. Por lo tanto, se puede usar el identificador dentro del Group By cuando nos referimos a todas las columnas de la tabla.

CALCULAR ÁREAS POTENCIALMENTE AGRÍCOLAS SUSTITUIDAS POR USO URBANO

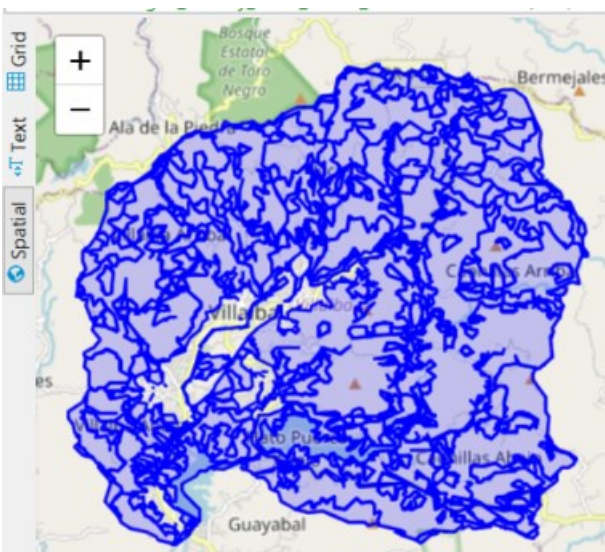
Para ver la tabla que acabamos de generar escriba lo siguiente, en la próxima línea debajo del último query:

- ☐ Abra otra pestaña (tab) para escribir un nuevo script. Haga **click** en el botón **SQL**, y escoja la opción **New SQL script** o teclee **ctrl+]**.

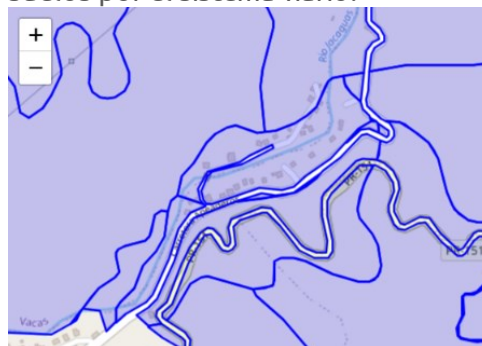


- ☐ Escriba: **select * from g00_overlay_erase_suelos_menos urbanos;**

g00_overlay_erase_suelos_menos urbanos 1					
Statistics 2					
	gid	musym	muname	farm_class	geom
1	1	LuE	Los Guineos clay, 20 to 40 percent slopes	Farmland of statewide importance	MULTIPOLYGON (((195429.542 236015.619
2	2	LuF	Los Guineos clay, 40 to 60 percent slopes	Not prime farmland	MULTIPOLYGON (((190094.6931989159 23
3	3	QeD2	Quebrada silty clay loam, 12 to 20 percent	Farmland of statewide importance	MULTIPOLYGON (((200514.9299 235987.0
4	16	LeC	Lares clay, 5 to 12 percent slopes	All areas are prime farmland	MULTIPOLYGON (((191868.105 231244.64
5	4	CbF2	Caguabo gravelly clay loam, 20 to 60 percent	Not prime farmland	MULTIPOLYGON (((197257.92 235870.308,
6	5	QeF2	Quebrada silty clay loam, 40 to 60 percent	Not prime farmland	MULTIPOLYGON (((191177.035 230308.63



Note los espacios vacíos en el área urbana del Municipio. Estos corresponden a los suelos reglamentados para uso urbano actual, programado y no programado, si acerca más, verá que también se recortaron geometrías de suelos por el sistema viario:



- ☐ Comparamos área original...
- ☐ En una nueva línea bajo el query anterior y escriba:

```
--calcula el área total en sqkm del geodato suelos
select sum(st_area(o.geom))/1000000 as area_orig_sqkm
from pr_geodata.g15_suelos_soil_map_units_2018_villalba as o;
```

Use ctrl+\ para que el query aparezca en otro tab

```
Grid area_orig_sqkm |
-----+
95.87616038282688 |
```

- ☐ Más abajo escriba este otro query para calcular el área del con áreas recortadas...

```
--calcula área total del geodato recortado
select sum(st_area(n.geom))/10000 area_wo_urban_sqkm
from g00_overlay_erase_suelos_menos_urbanos n;
```

- ☐ Sombree solo este último query y teclee **ctrl+** para ejecutar el script SQL en otra pestaña de resultados, para no borrar el resultado anterior.

```
26 --calcula área total del geodato recortado
27 select sum(st_area(n.geom))/10000 area_wo_urban_sqkm
28 from g00_overlay_erase_suelos_menos_urbanos n;
29
Grid area_wo_urban_sqkm |
-----+
88.10095514603212 |
```

¿Cuánta área de suelos potencialmente agrícolas han sido o serán sustituidas por suelos urbanizados?

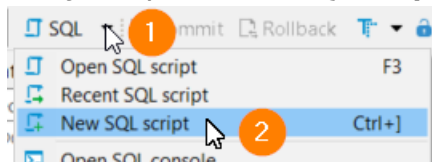
Ya que hicimos el proceso **Erase**, podemos contestar la pregunta con dos queries:

Área en **prime farm land en todas sus variedades y farm land of statewide importance**:



PostGIS, 3.x

- Abra otra pestaña (tab) para escribir un nuevo script. Haga **click** en el botón **SQL**, y escoja la opción **New SQL script** o teclee **ctrl+]**.



- Escriba el siguiente query combinado usando la cláusula UNION en la nueva pestaña.

```
--calcula cuántas hectáreas y cuerdas hay en prime farmland, dato original
select 'erased' as geodato, --nombre 'dummy' del campo
       round((sum(st_area(n.geom))/10000)::numeric,2)
       as prime_farmland_ha, --calc hectáreas a 2 lugares decimales
       round((sum(st_area(n.geom))/3930.395625)::numeric,2)
       as prime_farmland_cuerdas --calc cuerdas a 2 lugares decimales
from
    g00_overlay_erase_suelos_menos_urbanos n --tabla generada del
                                           -- geoproceso erase anterior
where
    farm_class not like 'Not prime%' --suelos NO "prime-farmland
                                     --ni sus variantes"
union --unión de queries, deben tener estructura idéntica
select 'original' as geodato, --nombre 'dummy' del campo
       round((sum(st_area(o.geom))/10000)::numeric,2)
       as prime_farmland_ha, --calc hectáreas a 2 lugares decimales
       round((sum(st_area(o.geom))/3930.395625)::numeric,2)
       as prime_farmland_cuerdas --calc cuerdas a 2 lugares decimales
from
    pr_geodata.g15_suelos_soil_map_units_2018_villalba o
                                           --tabla original suelos
where
    farm_class not like 'Not prime%'; --suelos NO "prime-farmland
                                     --ni sus variantes"
```

UNION: La cláusula UNION del lenguaje SQL usada aquí, sirve para unir dos conjuntos de filas que tengan igual estructura (las mismas columnas). Lo ejecutamos en una sola corrida.

Resultados, query: Área estimada en suelos potencialmente agrícolas:

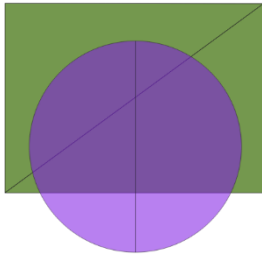
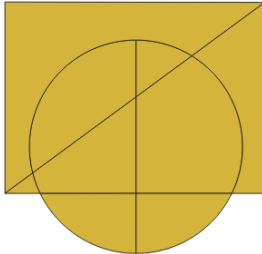
geodato	prime_farmland_ha	prime_farmland_cuerdas
erased	727.10	1849.95
original	1066.45	2713.34

En vista de esto, **se ha sustituido o se sustituirán hasta 863.4 cuerdas de suelo potencialmente agrícola** para ser usados como suelo urbanizado.



UNIÓN DE LAYERS

Este geoproceso se usa para **unir** las **geometrías** de dos geodatos, además de **integrar** los **atributos** que queramos extraer de **ambas tablas** de atributos asociados.

Layers de entrada		Layer resultante (A U B)
 Layer A		
 Layer B		
Dimensiones geométricas	Dimensiones posibles: 0, 1 o 2 Los layers deben tener igual dimensión geométrica. (p/p, l/l, s/s)	El resultado será de la misma dimensión de los layers A y B .
Conmutativa	Es conmutativa. No importa el orden.	--
Atributos, campos, o columnas	--	Los campos de salida serán la combinación de los campos de los layers A y B .

Este proceso no debe confundirse con la función [ST_Union](#). Ya usamos ST_Union en queries anteriores para integrar todas las geometrías de un geodato en una sola.

En cambio, el geoproceso '**Unión de layers**' trabaja como un Intersect de layers, pero conservará además las geometrías que no intersequen ente ambos layers. Conserva todas las geometrías de A, más todas las geometrías de B, además de los atributos que queramos conservar de ambos layers.

El proceso **Unión de layers** sería descrito gráficamente de la siguiente forma:



Estos pasos se realizan en queries separados y corridos secuencialmente. **El geoproceso Unión de layers es conmutativo**, pero las funciones ST_Difference a usarse no lo serán, porque va a realizarse A - B y B - A, luego de la intersección $A \cap B$. La intersección $A \cap B$ sí es conmutativa.



PostGIS, 3.x

Hagamos la unión de los geodatos de **riesgo a deslizamientos** con el geodato de **riesgo por inundaciones**. Las tablas tendrán **alias: deslizamientos as d** y **zonas inundables as i**. Estos son los atributos de ambas tablas:

g15_riesgo_geol_landslides_1979_villalba	g23_riesgo_inunda_floodzones_2017_villalba
Columns	Columns
gid (int4)	gid (int4)
geom (public.geometry(multipolygon, 6566))	geom (public.geometry(multipolygon, 6566))
risk_level (varchar(1))	fld_zone (varchar(17))
description (varchar(80))	st_bfe_m (float8)
	v_datum (varchar(17))
	fzn_1pct_id (int4)
	depth_m (float8)
	zone_subty (varchar(100))

Los **campos** que deseamos **conservar** serán en el geodato *landslides*: **gid, description, risk_level**, y en *floodzones*, **gid, fld_zone, zone_subty**. La **geometría resultante** de ambos campos geom de los layers de entrada se guardará en el nuevo geodato combinado.

**Riesgo por deslizamientos de terreno, Riesgo por inundaciones, FEMA, 2017
Monroe, 1979**

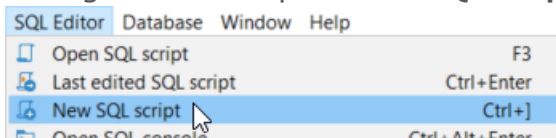


Como mencionamos anteriormente, el geoproceso Unión de layers está compuesto de tres pasos:

- 1) $A - B$, usando **ST_Difference(geomA, ST_Union(geomB))** +
- 2) $A \cap B$, usando **ST_Intersection(geomA, geomB)** +
- 3) $B - A$, usando **ST_Difference(geomB, ST_Union(geomA))**

Hagamos el proceso por partes:

- ☐ Primero: Definir la nueva tabla que recibirá los datos desde queries que vamos a ejecutar.
- ☐ Hagamos esto en una nueva pestaña-script. Vaya al **menú principal** y haga **click** en **SQL Editor**.
- ☐ Haga **click** en la opción **New SQL script**





- ☐ Escriba el siguiente query para definir la nueva tabla:

```
create table g00_overlay_union_desliz_inunda --nueva tabla
(gid serial primary key, --primary key de nueva tabla
d_gid integer, -- guardar gid del geodato landslides
d_risk_level varchar, --código risk level landslides
d_description varchar, --descripción
i_gid integer, --guardar gid del geodato floodzones
i_fld_zone varchar, --zonas inundables
i_zone_subty varchar, --subtipo de zona inundable
geom geometry (multipolygon,6566)); --geometría EPSG:6566 NAD83(2011)
```



- ☐ **Parte A – B** consiste en **dos subpartes**:

Subparte 1: **Extraer las geometrías que intersecan ...**

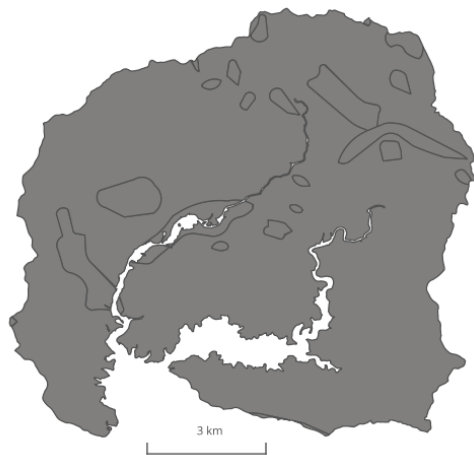
```
select gid, risk_level, description, geom --campos de tablas originales
from
(select d.gid, --campo gid de la tabla "d"
d.risk_level, --campo código nivel de riesgo de la tabla "d"
d.description, --campo descripción de la tabla "d"
STX_Extract( --función STX_Extract para extraer multipolygons
ST_Difference(d.geom, --devuelve lo que no interseca las geometrías "d"
st_union(i.geom) --función para unir todas las geometrías de "i"
),2 -- opción #2 de STX_Extract: extrae solo multipolygons
) as geom --alias del proceso combinado stx_extract,st_difference,st_union
from g15_riesgo_geol_landslides_1979_villalba as d, --tabla landslides
g23_riesgo_inunda_floodzones_2017_villalba as i
--tabla zonas inundables
where
st_intersects(d.geom, i.geom) --geometrías que intersecan
group by d.gid --agrupa por campos tabla d (dependencia funcional)
) as subq --el nombre o alias del subquery:subq
where geom is not null; --que las geometrías no estén vacías (null)
-- Fin del primer paso...
```



□ Subparte 2

```
--2a parte: polígonos tabla d que no son tocados por polígonos tabla i
insert into g00_overlay_union_desliz_inunda --insertar filas en esta tabla
(d_gid, d_risk_level, d_description, geom) --para los campos
d_gid,d_risk_level,d_descripcion,geom tabla nueva
select gid, risk_level, description, geom --campos tabla landslides
from --iniciar subquery
    (select d.gid,
        d.risk_level,
        d.description,
        i.gid as i_gid, --gid de tabla floodzones
        st_multi(d.geom) as geom
        --convertir geometrías a tipo multipolygon
    from g15_riesgo_geol_landslides_1979_villalba as d
        --tabla landslides
    left join g23_riesgo_inunda_floodzones_2017_villalba as i
        --tabla zonas inundables
    on st_intersects(d.geom, i.geom) --geometrías que interseccionan
    ) as subq --el nombre o alias del subquery:subq
where i_gid is null; --que en el campo i_gid no haya filas vacías (null)
--Fin de paso Difference (A-B) o (d-i)
```

Resultado del proceso (A-B) deslizamientos- inundaciones (d-i)



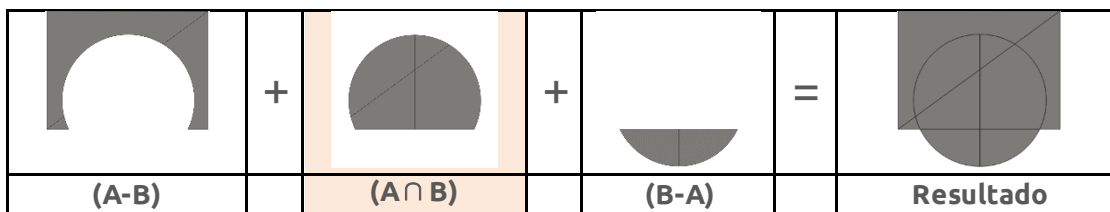
En esta parte, estamos mostrando el resultado parcial con los dos queries que completan la parte A-B para simular el geoproceso UNION de otras aplicaciones GIS Desktop.

A la izquierda tenemos **Diferencia Geométrica**: eliminación de las geometrías de "d" *landslides* que coinciden con las geometrías de "i" *floodzones*.

La primera parte del query A-B utiliza ST_Difference pero solo para la parte donde hay intersección.

La segunda parte del query es necesaria para integrar geometrías que no interseccionan las geometrías de "i"

En la próxima parte, continuamos añadiendo más código a este query. Recordamos que es prudente seleccionar el código que se va a ejecutar, hasta el símbolo ;. Pasemos a la próxima parte del query donde vamos a insertar las geometrías que interseccionan o que coinciden ($A \cap B$), "d.geom" intersección "i.geom".



```
--Paso Intersection (A ∩ B)
insert into g00_overlay_union_desliz_inunda --insertar filas en esta tabla
(d_gid,d_risk_level,d_description,
--campos tabla g00_overlay_union_desliz_inunda
--reciben valores de la tabla landslides
i_gid,i_fld_zone,i_zone_subty,geom)
--campos tabla g00_overlay_union_desliz_inunda,
-- reciben valores de tabla floodzones y
--geometrías (d n i)

select d.gid,
d.risk_level,d.description, --campos tabla d
i.gid,i.fld_zone,i.zone_subty,--campos tabla i
STX_Extract( --función stx_extract recibe geometrías de
--st_intersection...
st_intersection(d.geom,i.geom),2
) as geom --...y las convierte en multipolygons de (d n i)
from g15 riesgo geol landslides 1979 villalba as d,
--tabla landslides
g23 riesgo inunda floodzones 2017 villalba as i
--tabla zonas inundables
where
st_intersects(d.geom,i.geom); --geometrías que intersequen
--fin del paso Intersection
```

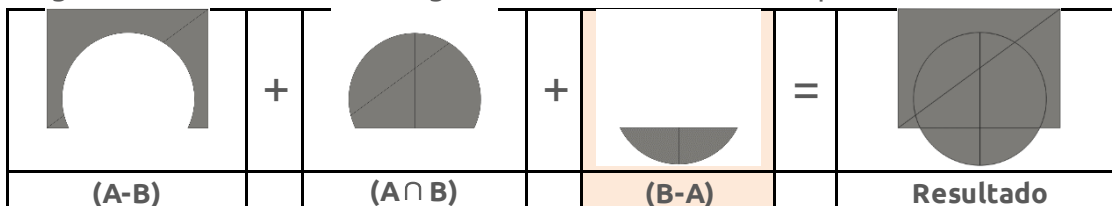
Resultado del proceso $(A \cap B)$ deslizamientos $\cap$ inundaciones $(d.geom \cap i.geom)$



Podemos ver cómo con el uso de ST_Intersection, obtenemos las geometrías que coinciden entre ambos geodatos y las “pegamos” en el nuevo geodato. El uso de STX_Extract() es para asegurar la homogeneidad de geometrías multipolygon en estos procesos.



En la próxima parte del query completaremos el proceso haciendo la **parte (B-A)**, trayendo al nuevo geodato las geometrías de landslides “i.geom” que están fuera de las geometrías de landslides “d.geom”. También tendrá dos partes.



```
--Paso Difference (B-A)
--1a parte: polígonos tabla i que son borrados parcial o totalmente por
polígonos tabla d
insert into g00 overlay union desliz inunda --insertar filas en esta tabla
(i_gid, i_fld_zone, i_zone_subty, geom) --para los campos
--i_gid, i_fld_zone, i_zone_subty, geom en tabla nueva
select gid, fld_zone, zone_subty, geom --la inserción se basa en
--el query a continuación:
--los campos corresponden a campos de tablas originales
from --de este subquery:
(select i_gid, i_fld_zone, i_zone_subty, --campos, tabla i
STX_Extract(
ST_Difference(i.geom, --extrae multipolígonos de
--la diff entre geoms de i - unión geoms de d
ST_Union(d.geom)), 2 --guárdalas como multipolygon
) as geom --guárdalas con alias geom
from g23_riesgo_inunda_floodzones_2017_villalba as i,
--tabla zonas inundables “i”
g15_riesgo_geol_landslides_1979_villalba as d
--tabla landslides “d”

where
st_intersects(i.geom, d.geom) --geometrías que interseccionan
group by i_gid --agrupa por campos tabla i (dependencia funcional)
) as subq --el nombre o alias del subquery: subq
where geom is not null; --condición: que las geometrías no estén vacías
```

```
--2a parte: polígonos tabla i que no son tocados por polígonos tabla d
insert into g00 overlay union desliz inunda --insertar filas en esta tabla
(i_gid, i_fld_zone, i_zone_subty, geom) --para los campos
-- i_gid, i_fld_zone, i_zone_subty, geom en tabla nueva
select gid, fld_zone, zone_subty, geom --la inserción se basa en
--el query a continuación
from --de este subquery:
(select i_gid, i_fld_zone, i_zone_subty, d_gid as left_gid,
--campos, tabla i y campo d_gid usado para contar filas
st_multi(i.geom) as geom --convierte geom de i en tipo multi
from g23_riesgo_inunda_floodzones_2017_villalba as i
```



```

--tabla zonas inundables
left join      --conservar todas las filas de la tabla i
               --aunque no haya intersección
g15_riesgo_geol_landslides_1979_villalba as d
               --tabla landslides "d"
on
  st_intersects(i.geom,d.geom) --geometrías que intersequen
) as subq --el nombre o alias del subquery:subq
where left_gid is null; --condición: que el campo left_gid esté vacío.
--Fin de paso Difference (B-A)
```

A continuación mostramos el resultado:

Resultado:

- ☐ Escriba el siguiente query **en otra pestaña de queries, ctrl+] select * from g00_overlay_union_desliz_inunda;**
- ☐ Use la pestaña **Spatial** para ver el geodato en el mapa:



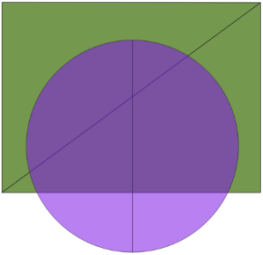
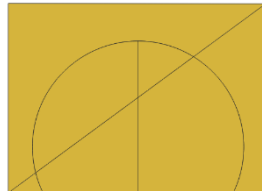
El geodato resultante puede mejorarse. Las áreas de lagos están catalogadas como de riesgo moderado. Esa era la clasificación que heredada del geodato original de deslizamientos. Al unirle el geodato de zonas inundables, podemos modificar o actualizar el resultado. Será apropiado entonces usar la cláusula UPDATE en esta tabla/geodato y modificar/actualizar los campos **d_risk_level**, cambiar el valor a **L**, igualmente, cambiar el valor **d_description** a **'AREA OF LOW SUSCEPTIBILITY TO LANDSLIDING'**. De igual manera pueden actualizarse las columnas de zonas inundables y añadir información a las áreas que no tienen datos.

- ☐ Guarde su código SQL con el nombre **g00_overlay_union_desliz_inunda.sql**. Vaya al menú principal en DBEaver y escoja la opción Save As para guardar este script.



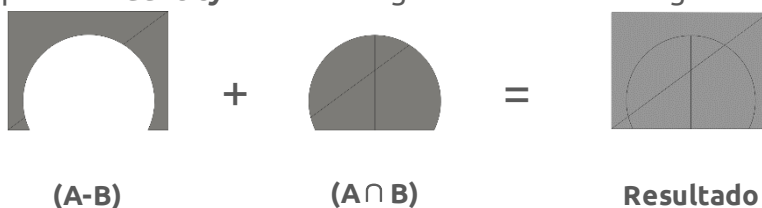
IDENTITY

Para el geoproceso Identity, lo que interesamos es preservar la geometría y atributos de uno de los layers de entrada: el input layer. Dicho de otro modo, es **obtener del layer Identity solo lo que corresponde geométicamente con el layer Input, el cual mantendrá su extensión espacial.**

Layers de entrada		Layer resultante (A identity B)
 Layer A		
 Layer B		
Dimensiones geométricas	El layer A (input) puede ser dimensión: 0, 1 o 2 El layer B (identity) será de igual dimensión que A.	El resultado será de la misma dimensión del layer A .
Conmutativa	No es conmutativa. El layer B es de <i>identidad</i> .	--
Atributos, campos, o columnas	--	Los campos de salida serán la combinación de los campos de los layers A y B .

El geoproceso 'Identity' se trata de una **diferencia $A - B$** , seguido por la **intersección $A \cap B$** . **Conserva todas las geometrías de A, más solo las geometrías de B que intersequen con A.**

El proceso **Identity** se describe gráficamente de la siguiente forma:

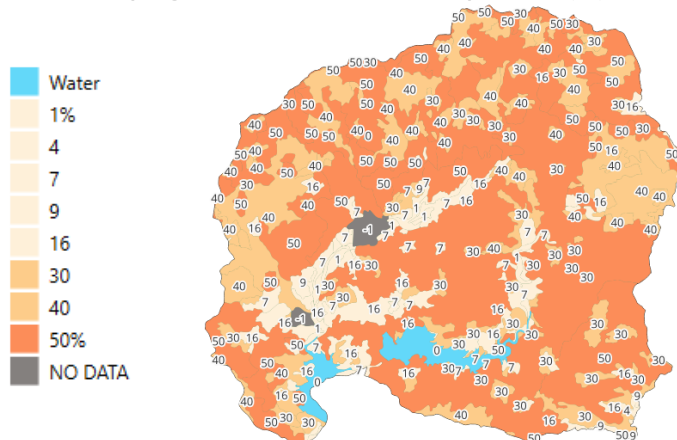


Estos pasos se realizan en queries separados y corridos secuencialmente.

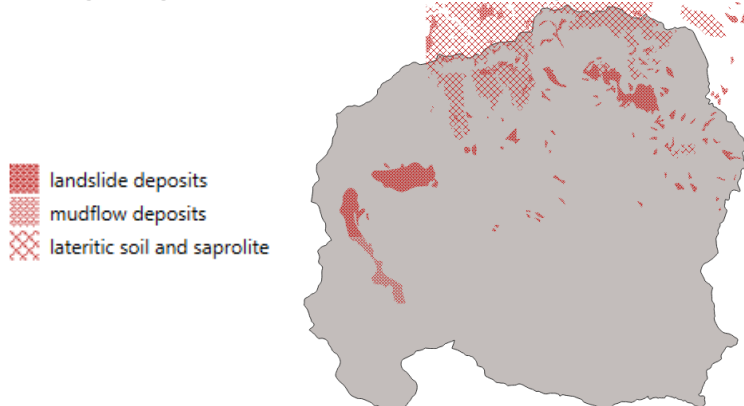
Para demostrar el geoproceso **Identity**, vamos a **producir un mapa de susceptibilidad a deslizamientos**, basado en el geodato del catastro de suelos del USDA-NRCS y el geodato de unidades geológicas de los cuadrángulos geológicos a escala 1:20,000.



Catastro de suelos: Slope gradient dominant component (%)



Unidades geológicas: Landslides, mudslides, saprolites, laterites debris.



La figura gris (límites municipales) no es parte del geodato geológico.

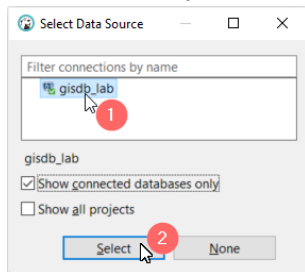
Ya que vamos a emular el geoproceso Identity, el geodato **input** será el de **suelos** y el geodato **identity** será el de **unidades geológicas**.

```
g15_suelos_soil_map_units_2018_villalba
├── Columns
│   ├── gid (serial4)
│   ├── geom (public.geometry(multipolygon, 6566))
│   ├── areasybol (varchar)
│   ├── spatialver (int4)
│   ├── musym (varchar)
│   ├── mukey (varchar)
│   ├── muname (varchar)
│   └── slope_gradient_dominant_component (float8)
└── slope_gradient_weighted_sum (float8)
```

Fuente: USDA-NRCS, 2018



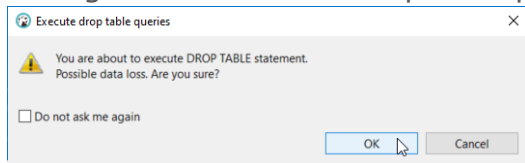
- ☐ Para este proceso, comience con un nuevo script. Pulse las teclas **ctrl +]** para generar un nuevo script en DBeaver.



- ☐ En la **pestaña** para escribir queries, escriba los siguientes queries:
- ☐ Escriba lo siguiente para generar la tabla nueva:

```
/* Geoproceso IDENTITY
* Primera parte: Crear una tabla
* Esta tabla contendrá las geometrías que resulten del geoproceso,
* además de las columnas que hayan sido escogidas para describir las geometrías
*/
drop table if exists g00_overlay_identity landslides 2025 villalba
-- si la tabla existe ...
cascade; --bórrala con la opción cascade para eliminar sus objetos asociados
create table g00_overlay_identity_landslides_2024_villalba
--crear nueva tabla/geodato
(gid serial primary key, --identificador único del nuevo geodato
g_gid integer, --identificador, geodato geología
g_quad_code varchar, --código, cuadrángulo geológico
g_unit_code varchar, --código, unidad geológica
g_unit_name varchar, --nombre, unidad geológica
s_gid integer, --identificador, geodato suelos
s_slope_gradient_dominant_component integer,
--valores, pendiente en por ciento del componente suelos dominante
risk_level varchar, --nivel de riesgo:
--luego de terminar el geoproceso, será calculado según valores
--de geología y pendientes
geom geometry (multipolygon,6566) --geometría, SPCS-PR NAD83(2011),
--EPSG:6566
);
```

- ☐ Sombree el código desde el inicio hasta el **);** y corra el script.
- ☐ Haga **click** en el botón **OK** para aceptar la ejecución del **DROP TABLE**.



Si la tabla no existe, aparecerá un mensaje OUPUT que le avisará que brincó el paso DROP TABLE.

- ☐ Continúe el script en la próxima página



□ Sección Difference (A-B)

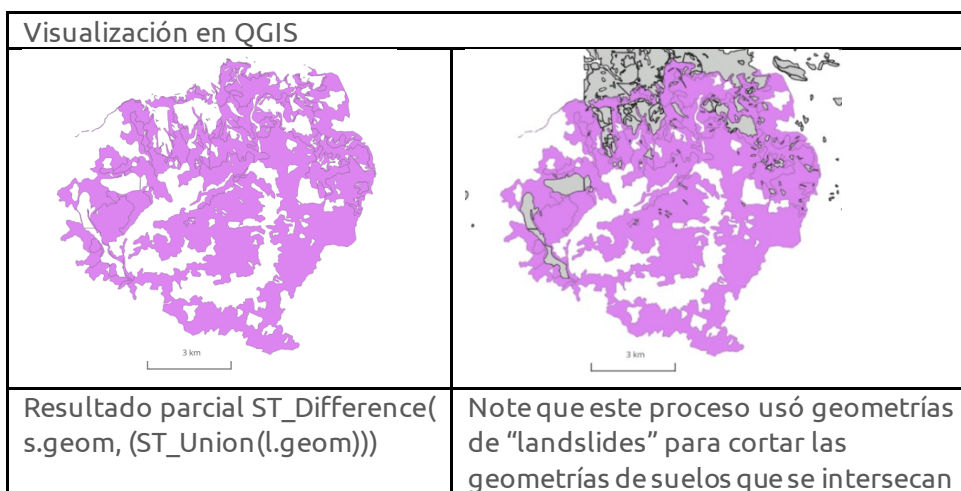
```
/* Iniciar geoprocesamiento IDENTITY
 * Procesos: Difference:(A-B)
 * El proceso Difference (A-B) consiste en dos pasos:
 * 1ro: Difference de Áreas que solapan entre geodatos
 * 2do: + Añadir áreas que no solapan
 * Paso "Difference" (A-B), en adelante (s-l) s=suelo, l=landslides
 */
--1er paso: borrar áreas en tabla "s" que son intersecadas parcial o
--totalmente por áreas de tabla "l"
insert into g00 overlay identity landslides 2025 villalba
    --insertar filas en esta tabla
    (s_gid,s_slope_gradient_dominant_component,geom)
    --campos donde vamos a copiar valores de tabla "s"
    --no incluimos gid porque se calculará automáticamente (serial)
select gid, --gid de la tabla s suelos.
    slope_gradient_dominant_component,geom --campos de la tabla suelos
from
    (select s.gid,s.slope gradient dominant component,
    --campos, tabla suelos "s"
    STX_Extract( --función STX_Extract para homogeneizar geometrías...
        ST_Difference(s.geom, --devolverá la diferencia entre
            --geoms de s y
            ST_Union(l.geom)) --la unión de geometrías geodato "l"
        ,2) as geom -- STX_Extract: opción #2:
        --devuelve multipolygons, alias geom

    from
    pr_geodata.g15_riesgo_soil_slopes_2018_villalba as s,
        --tabla suelos
    pr_geodata.g15_riesgo_geol_mapping_units_20k_2018_landslides as l
        --tabla geología unidades landslides

    where
        st_intersects(s.geom,l.geom) --test: geometrías que intersequen
    group by s.gid --agrupar por campo gid, tabla suelos
    ) as subq --alias del subquery
where geom is not null; --que las geometrías no estén vacías(null)
```

□ Luego de terminar, escriba la sentencia para corroborar:

```
select * from g00 overlay identity landslides 2025 villalba;
```



- ☐ En **DBeaver** use la pestaña **Spatial** al lado izquierdo del grid de columnas y filas.



Si no ve geometrías, debe volver al código y cotejar dónde está el error.

- ☐ Continuemos con la segunda parte del proceso Difference (A-B).

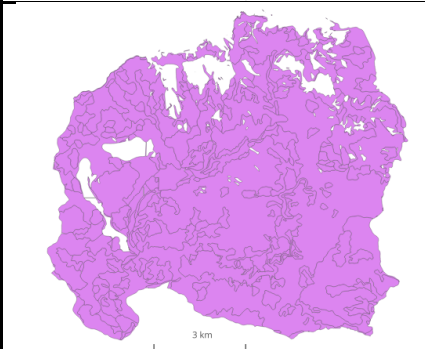
```
-- 2a parte del proceso (A-B):
-- añadir geometrías de tabla s que no intersectan áreas de l
insert into g00_overlay_identity landslides 2025 villalba
(s_gid,s_slope_gradient_dominant_component,geom)
--reciben valores de tabla s
select gid,slope_gradient_dominant_component,geom
--campos tabla suelos
from
(select s.gid,s.slope_gradient_dominant_component,
--campos de tabla suelos
l.gid as l_gid, --identificador de tabla l
st_multi(s.geom) as geom --conv geoms a multipolygon
from
pr_geodata.g15_riesgo_soil_slopes_2018_villalba as s
--tabla suelos
left join
pr_geodata.g15_riesgo_geol_mapping_units_20k_2018_landslides l
on
st_intersects(s.geom,l.geom) --test:intersección geom
) as subq --alias del query
where l_gid is null;
```



PostGIS, 3.x

- `--donde valores del campo l_gid o l_gid original sean null`
- ☐ **Sombree la última sentencia** que acabó de escribir **desde** el último **Insert into** hasta **where l_gid is null**;
 - ☐ **Corra el script** pulsando **ctrl+enter** o **click** en el **botón**  para correr el script.
 - ☐ Use el tab **Spatial** para ver las geometrías.



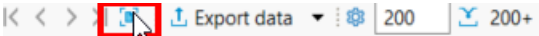
Visualización en QGIS	
	
Resultado parcial #2 left join: traer todos los récords tabla suelos que intersecan tabla landslides y que tengan valores null en el campo l_gid o l_gid original	Note que en este proceso solo falta “rellenar” los espacios vacíos de la tabla suelos con las áreas de landslides que van en esos espacios. Hará falta una intersección geométrica para descartar áreas de <i>landslides</i> que estén fuera de la tabla <i>suelos</i> .

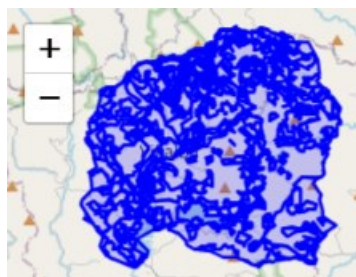
- ☐ Continuemos, escribiendo la sección o proceso **INTERSECTION** que servirá para insertar las áreas que faltan.

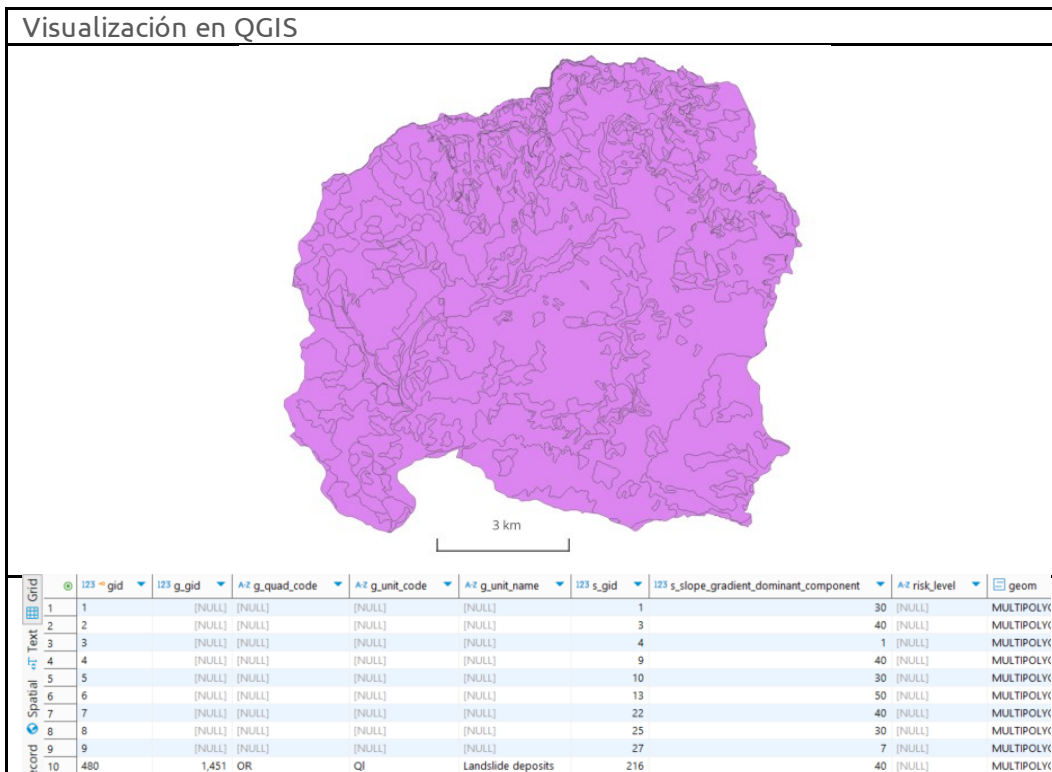


```
-- 3er paso: proceso Intersection suelos n landslides
insert into g00_overlay_identity_landslides_2025_villalba
(s_gid,s_slope_gradient_dominant_component, --campos, tabla s
g_gid,g_quad_code,g_unit_code,g_unit_name, --campos, tabla l
geom) --geometrías resultado de proceso s n l

select
s.gid,s.slope_gradient_dominant_component, --campos, tabla s
l.gid,quad_code,unit_code,unit_name,
STX_Extract( --función homogeneizar geometrías
ST_Intersection(s.geom,l.geom) --intersección geometrías
,2) as geom --opción #2: multipolygons alias geom
from
pr_geodata.g15_riesgo_soil_slopes_2018_villalba s,
--tabla suelos
pr_geodata.g15_riesgo_geol_mapping_units_20k_2018_landslides l
--tabla landslides
where st_intersects(s.geom,l.geom); --donde haya intersección
--Fin del proceso INTERSECTION para generar Identity
--Fin de geoprocso IDENTITY
```

- ☐ **Sombree la última sentencia** que acabó de escribir **desde** el último **Insert into** hasta **where st_intersects(s.geom,l.geom);**
 - ☐ **Corra el script** pulsando **ctrl+enter** o **click** en el **botón**  para correr el script.
 - ☐ Use el tab **Spatial** para ver las geometrías. Si no aparecen todas las geometrías
 - ☐ Haga **click** en el botón **Fetch All Data**
- 
- ☐ Haga **click** en el botón **Yes** en la forma de advertencia **Read all rows**.





Los espacios NULL en la tabla corresponden a valores que no existían en alguno de los dos geodatos antes de unirlos.

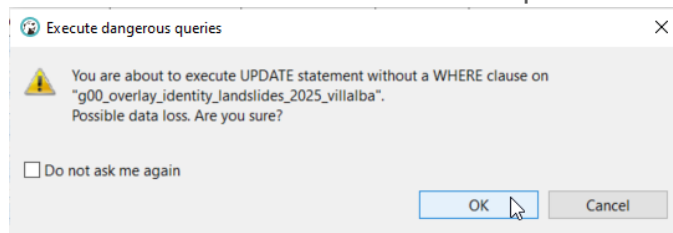
- ☐ Si todo está bien, el próximo paso es **actualizar** el campo **risk_level** con los niveles que asignaremos mediante la cláusula **UPDATE** y la función **CASE-WHEN-END**.



- Continuemos en este script para añadir otra sentencia **CASE-WHEN-END**.

```
--Procedimiento CASE-WHEN-END
--Propósito: actualizar valores risk_level
--Riesgo por deslizamientos de terrenos
--El cálculo está basado en el porcentaje de pendientes
--y geologías previamente identificadas como de alto riesgo
update g00_overlay_identity_landslides_2025_villalba
--actualiza esta tabla
set risk_level = --el campo que vamos a actualizar
  case when g_gid is not null
    --selecciona valores que no sean null
    --estos corresponden con áreas identificadas
    --como unidades geológicas de más alto riesgo
  then 'Very High' --asígnale esta etiqueta al campo risk_level
  when s_slope_gradient_dominant_component >=50
    --cuando la pendiente sea >= 50%
  then 'High' --asigna etiqueta High al campo risk_level
  when s_slope_gradient_dominant_component between 30 and 49
  then 'Moderate' --asigna etiqueta Moderate al campo risk_level
  when s_slope_gradient_dominant_component between 1 and 29
  then 'Low' --riesgo bajo
  when s_slope_gradient_dominant_component =0
  then 'Water' --corresponde a embalses
  when s_slope_gradient_dominant_component =-1
  then 'Unknown' --áreas donde no hay valores de pendientes.
  end;
```

- **Seleccione** el código que acaba de escribir, **desde** la parte **update** hasta la palabra **end**;
- **Corra** el **script** pulsando **ctrl+enter** o **click** en el botón  para correr el script.
- Haga **click** en el botón **OK** de la forma **Execute dangerous queries**. Esta forma aparece porque este query MODIFICA los datos. Se van a sustituir valores existentes con otros que estamos asignando.

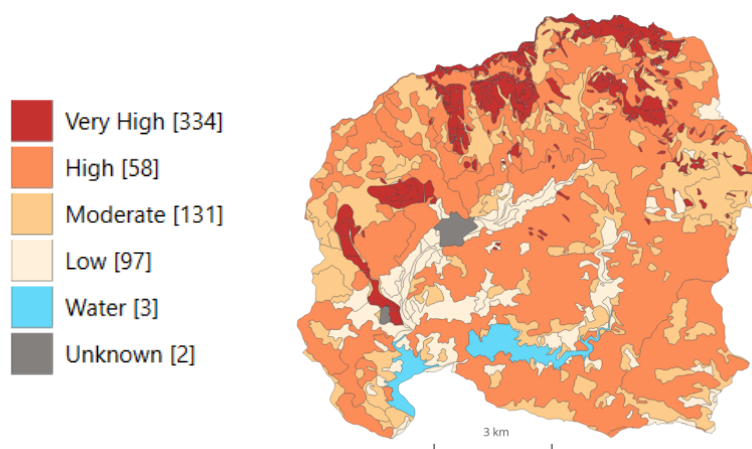


- Escriba la sentencia **select * from g00_overlay_identity_landslides_2025_villalba**;
- Vea cómo cambiaron los valores en la columna **risk_level**

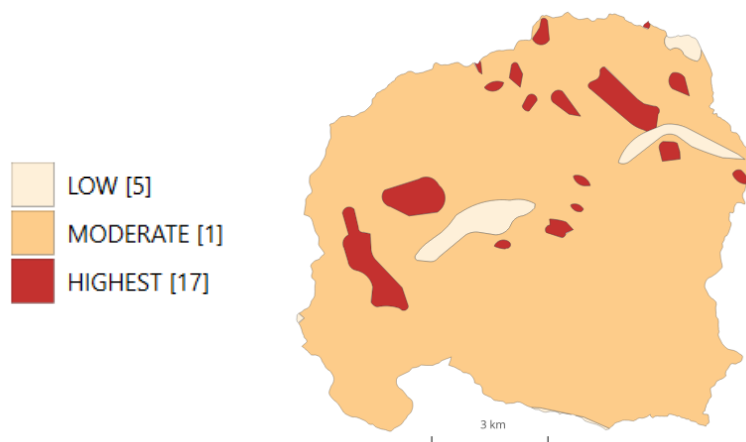


PostGIS, 3.x

Luego de aplicar simbología en QGIS, con datos compilados a escala 1:20,000:



Comparar con el mapa original de 1979 compilado a escala 1:250,000:

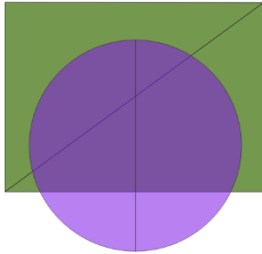
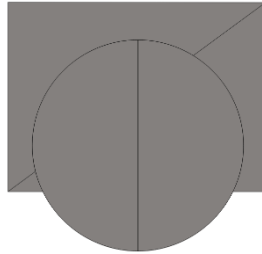


El geodato combinado de geología y suelos puede ser actualizado en términos de pendientes, si le hacemos un overlay con un ráster de pendientes y aplicamos [ST_SummaryStats](#).

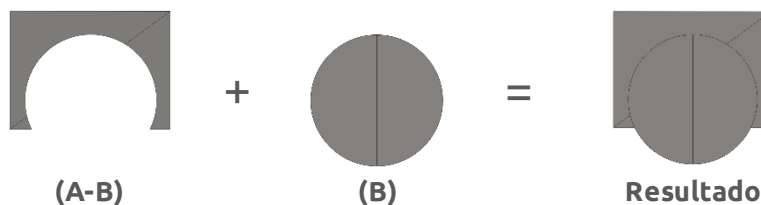


UPDATE

Este geoproceso produce la intersección geométrica de las geometrías de entrada y añade las geometrías del geodato de actualización. Los atributos (que sean iguales) y geometrías del layer actualizado serán integrados en el layer de salida.

Layers de entrada		Layer resultante (A update B)
 Layer A  Layer B		
Dimensiones geométricas	El layer A 'input' y layer B 'update' deben ser de dimensión 2 .	El resultado será de la misma dimensión que el layer A .
Conmutativa	No es conmutativa. El layer B es de <i>actualización</i> .	--
Atributos, campos, o columnas	--	Los campos de salida serán los que contiene el layer A y los campos de B que sean iguales a los del layer A .

El proceso **Update** sería descrito gráficamente de la siguiente forma:

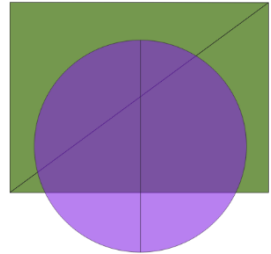
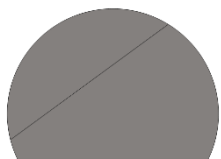


Estos pasos se realizan mediante queries corridos secuencialmente. El geoproceso UPDATE es parecido al proceso ERASE en su primera parte (que a su vez está compuesta de dos subpartes). Luego que las áreas de intersección estén “eliminadas”, se añade entonces las geometrías del layer B.



CLIP

Este es un proceso de **extracción** donde la forma o geometría de layer-clip (B) servirá para **recortar** las geometrías del layer input (A).

Layers de entrada		Layer resultante (A update B)
 Layer A  Layer B		
Dimensiones geométricas	El layer A 'input' puede ser de dimensión 0, 1 o 2 . El layer B debe ser de dimensión 2 .	El resultado será de la misma dimensión que el layer A .
Conmutativa	No es conmutativa . El layer B es el layer de <i>recorte</i> .	--
Atributos, campos, o columnas	--	Los campos de salida serán los que contiene el layer A .

Antes de realizar el corte, se aplicará **ST_Union** para consolidar las geometrías de B, si es que contiene más de un polígono. **Si el layer B es de un solo polígono**, podemos usar solamente **ST_Intersection(A,B)**.

Vamos a llevar a cabo dos queries emulando el geoproceso Clip sin hacer una tabla permanente como las anteriores. Solo para propósitos de demostración. Uno se hará sin la función ST_Union porque ya es un solo polígono (área municipal). El segundo query utilizará ST_Union (geodato de barrios, Villalba).

Query sin usar ST_Union:

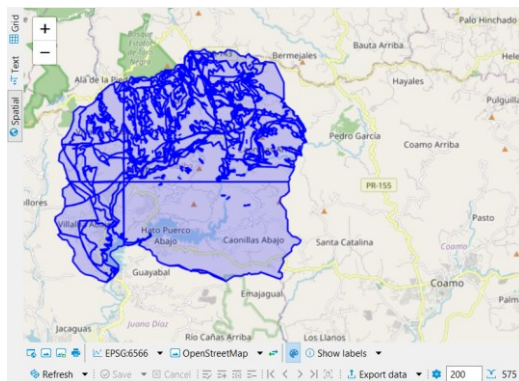
En la pestaña para escribir queries **<gisdb_lab>** escriba el siguiente query

```
/* geoproceso CLIP sin hacer
tabla permanente y sin usar ST_Union
*/
select g.gid as g_gid,
g.quad_code,g.unit_code,g.unit_name,
g.period,g.epoch,g.unit_descript,
g.quad_name,g.geol_id,          --campos, tabla geología
STX_Extract(                    --función para homogenizar geoms
    ST_Intersection(g.geom,v.geom)
    ,2)                          --devuelve dimensión 2: multipolygons
as geom --alias geom
from
    pr_geodata.g15_geologia_mapping_units_20k_2018 g,
    --tabla geología
```



PostGIS, 3.x

```
pr_geodata.g03_legales_municipios_2015_villalba v
--tabla con solo el área municipal, Villalba
where
st_intersects(g.geom,v.geom);
--donde las geoms g y v intersequen
```

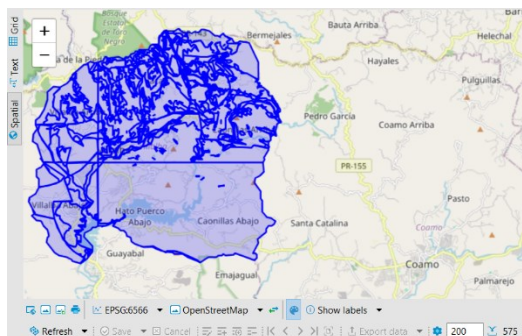


Si queremos hacer un geodato permanente, debemos usar CREATE TABLE como en los procesos anteriores. Luego use INSERT INTO, usando el query que está escrito arriba. Note que el nombre del campo “period” va encerrado en comillas dobles.

Query con ST_Union:

En la pestaña para escribir queries **<gisdb_lab>** escriba el siguiente query:

```
/* geoprocso CLIP sin hacer
tabla permanente usando ST_Union
*/
select g.gid as g_gid,
       g.quad_code,g.unit_code,g.unit_name,
       g.period,g.epoch,g.unit_descript,
       g.quad_name,g.geol_id, --campos, tabla geología
STX_Extract(
    ST_Intersection(g.geom,(ST_Union(b.geom)))
    ,2) --devuelve dimensión 2: multipolygons
as geom --alias geom
from
pr_geodata.g15_geologia_mapping_units_20k_2018 g,
--tabla geología
pr_geodata.g03_legales_barrios_2015_villalba b
--tabla “ST_Union=Dissolve” barrios,Villalba
where
st_intersects(g.geom,b.geom)
--donde las geoms g y v intersequen
group by g.gid;
--ya que usamos ST_Union, debemos agrupar por g.gid
--Dependencia Funcional
--use ctr+shift+= (fetch all data)
```



El resultado debe ser el mismo, pero en el proceso se utilizó ST_Union para consolidar los polígonos de los barrios. Además, al usar ST_Union, por ser una función de agregación, debemos usar GROUP BY al final usando solo el campo g.gid ([ver explicación sobre Dependencia funcional](#)).

****Resista la tentación de disolver los polígonos adyacentes entre cuadrángulos geológicos. No necesariamente son lo mismo.****

Cada cuadrángulo es una interpretación independiente de cuadrángulos adyacentes. Las interpretaciones están basadas en estudios científicos, pero no siempre coinciden. En ocasiones hay formaciones geológicas que han cambiado de nombre y se han quedado así en los cuadrángulos. Sin duda, hace falta actualizar y completar estos mapas, ya que muchos son antiguos, otros son preliminares y otros no existen a escala 1:20,000.

BUFFER

Un buffer es un área de proximidad o influencia desde una geometría de dimensión 0, 1, o 2 y una distancia escogida. Su uso es muy común en aplicaciones SIG de interfaz gráfica. Se debe usar [ST_Buffer](#) por ejemplo, cuando queremos extraer un área de influencia a una geometría.

Geometrías	Input layers	Output
Punto		
Linestring		
Polygon		
Dimensiones requeridas	Un layer de una de tres dimensiones: 0, 1 o 2	Dimensión 2
Atributos	--	Solo atributos del input layer
Notas:	Luego de hacer el buffer, se puede aplicar la función ST_Union para consolidar los polígonos de salida, según se necesario.	



Sintaxis:

```
ST_Buffer(geometry, radio_del_buffer float, buffer_style_parameters = '' text);  
ST_Buffer(geometry, radio_del_buffer float, num_segmentos_ct_círc integer);
```

El **tercer argumento** o parámetro sirve para controlar la exactitud y estilo del buffer. La exactitud del cuarto de círculo es dada por **el número de segmentos**, que **por defecto es 8**. El estilo del buffer puede especificarse mediante **parámetros separados por un espacio**:

- **'quad_segs=16'**: número de segmentos usados para la aproximación del cuarto de círculo.
- **'endcap=round|flat|square'**: forma de terminación del buffer, por defecto, round.
- **'join=round|mitre|bevel'**: estilo del join, por defecto round. Se puede escribir 'miter'.
- **'mitre_limit=#.#'**: razón de límite del mitre "pico", el cual solo afecta al join mitre
- **'side=both|left|right'**: left o right realiza el buffer de un solo lado de la geometría, con el lado del buffer relativo a la dirección de la línea. Esto solo aplica a Linestrings y no afecta a geometrías de punto y de polígonos. Por defecto, los end caps son cuadrados.

El resultado de salida es un polígono o un multipolígono que representa todos los puntos en los cuales la distancia de una geometría/geografía es igual o menor a la distancia dada. Una distancia negativa puede reducir el polígono completamente y en tal caso, se devuelve un polígono vacío. Para puntos y linestrings las distancias negativas siempre devolverán geometrías/geografías vacías.

ST_Buffer siempre devuelve geometrías válidas. Por tal razón, es usado a veces como forma de validar geometrías, mediante ST_Buffer(geom,0).

Ejemplo: Generar un geodato/tabla que contenga un buffer con **radio de 5 metros** de **cada elemento del sistema vial** del Municipio de Villalba. **Consolide el buffer** en una sola geometría. **Conviértala a geometría MultiPolygon**. Los **end caps** deben ser **flat**/planos

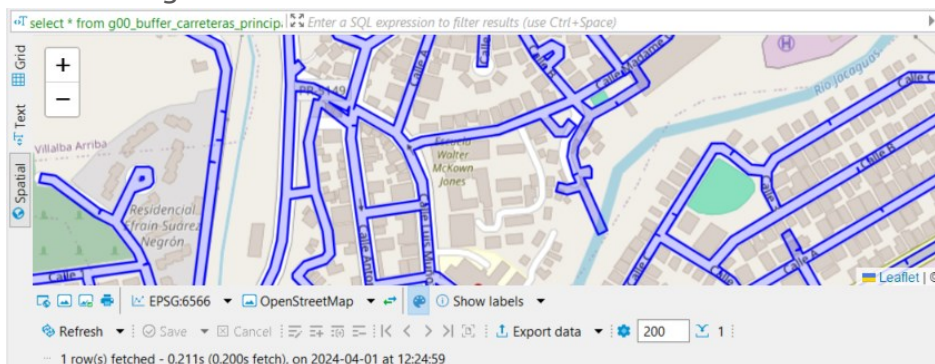


PostGIS, 3.x

□ En un nuevo tab/script, escriba el siguiente código:

```
--script para generar una tabla
--que contenga un buffer alrededor
--de sistema vial de Villalba
drop table if exists workspace01.g00_buffer_carreteras_principales_villalba
cascade;
--si la tabla existe, bórrala con opción cascade para borrar objetos asociados
create table g00_buffer_carreteras_principales_villalba
(gid serial primary key,
geom geometry (multipolygon,6566));
--crear tabla/geodato buffer vacío
insert into g00_buffer_carreteras_principales_villalba --insertar récords
(geom) --solo geometrías
select st_union( --une/disuelve en una geometría
st_multi( --convertir las geoms a multipolygons
ST_Buffer(v.geom, --realizar buffer
5, --de un radio de 5 metros
'endcap=flat')) --que el endcap (rabo) sea cuadrado
from
pr_geodata.g35_viales_mapa_base_tiger_rds_2006se_villalba v;
--tabla de sistema vial municipal
```

Detalle del geodato:





SELECCIÓN POR DISTANCIA

En programas SIG de interfaz gráfica estamos acostumbrados a utilizar un buffer para hacer una búsqueda por distancia, o utilizar una función de selección localización (por intersección geométrica) que conlleve una distancia. En el caso de PostGIS, hacer ese proceso como parte de un query es bastante costoso en recursos de memoria (tarda mucho). Para búsquedas por distancia es más eficiente utilizar [ST_DWithin](#). Esta función de PostGIS utiliza el índice espacial y es mucho más eficiente y rápida.

Ya hemos utilizado ST_DWithin anteriormente en queries espaciales en las secciones anteriores.

APPEND

El geoproceso Append usa la cláusula INSERT INTO como ha sido realizado en las secciones anteriores. Durante este proceso, se añaden nuevas filas a la tabla, ya sea nueva sin filas o con datos.

Se debe recordar que al hacer la inserción, no necesariamente se realiza una validación topológica y por ende, puede haber solape entre geometrías, con las consecuencias que esto pueda implicar.

DISSOLVE

Ya hemos visto indirectamente el proceso **Dissolve** en secciones anteriores cada vez que utilizamos la función [ST_Union](#). Por ejemplo, lo vimos en el proceso Buffer anterior cuando consolidamos los buffers de cada segmento de vía.

	Input Layer	Resultado
Consolidación de todas las geometrías		
Consolidación basada en un campo; generalizar/agregar geometrías. (Usar GROUP BY)		
Dimensiones requeridas	Solo un input layer de dimensión 0, 1 o 2.	Igual a la dimensión del input layer.

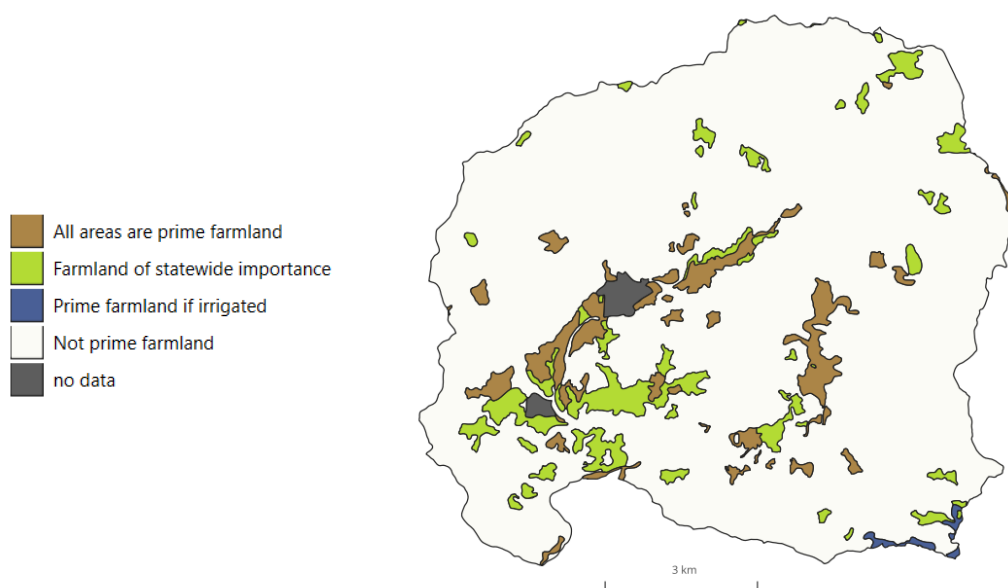


Hagamos un ejemplo de **Dissolve**, consolidando geometrías basado en un campo de la tabla.

□ En la pestaña **<gisdb_lab> Script**, escriba el siguiente script SQL:

```
drop table if exists
  workspace01.g00_dissolve_prime_farmland_villalba
  cascade;
--borra la tabla si existe on opción cascade
create table workspace01.g00_dissolve_prime_farmland_villalba
(gid serial primary key,
 farm_class varchar,
 geom geometry (multipolygon, 6566));
--crea tabla Dissolve basado en el campo farm_class
--de la tabla original de suelos
insert into workspace01.g00_dissolve_prime_farmland_villalba
(farm_class,geom)
--insertar los récords en esta tabla nueva
select
  s.farm_class, --campo farm_class de tabla suelos
  st_multi( --convierte a tipo multi
    st_union(s.geom))
    --une/disuelve geometrías de suelos
from
  pr_geodata.g15_suelos_soil_map_units_2018_villalba s
  --tabla/geodato suelos NRCS-Villalba
group by
  s.farm_class; --agrega/consolida por campo farm_class
select * from g00_dissolve_prime_farmland_villalba;
--corroborar/ver en DBeaver
```

Resultado:





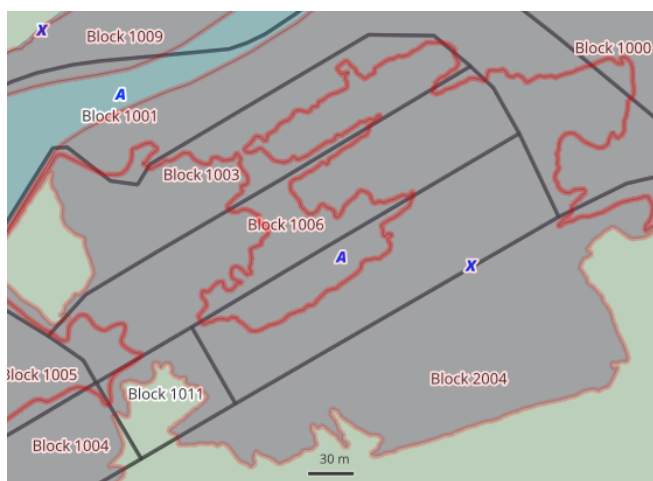
INTERPOLACIÓN AREAL

En esta parte veremos un ejemplo de uso del mecanismo de **interpolación areal** para **estimar población según el geodato de bloques censales, Censo-2020 que están en zonas inundables, FEMA-2017**. La manera más exacta de conocer la población afectada por ubicarse en zonas inundables es cuando las viviendas están identificadas por **puntos**. Solo haría falta **seleccionar** los puntos que intersequen o estén a cierta distancia de los bordes de zonas inundables y luego **multiplicar** por el **factor de personas por vivienda** para el Censo 2020 (**2.65 personas por vivienda**).

Cuando no disponemos de puntos, podemos **usar la unidad cartográfica censal más pequeña y solaparla con el geodato de interés**. El geodato de **bloques censales 2020** tiene la unidad de área más pequeña con los conteos de 100% de los datos poblacionales.

Usaremos el geodato de **zonas inundables** como ejemplo para solapar este mapa con el mapa de bloques y utilizar el mecanismo de **interpolación areal** para **estimar la población afectada**. La estimación de población se basa en el principio de **proporción de área ocupada** del geodato de inundaciones en este caso, dividido por el **área original del bloque censal**. Esta proporción de área (de 0 hasta 1) es multiplicada por el dato original, de población en este caso y nos dará indirectamente el estimado de población afectada.

Este método funciona mejor cuando la población está repartida de manera uniforme a través del área. Por otro lado, cabe la posibilidad de que la población **esté concentrada** en la zona inundable pero el bloque censal sea grande y el área de solape sea pequeño. En ese caso, la población estaría **subestimada**. **Otra desventaja** es que a medida que **nos alejamos del año censal decenal**, el conteo de población puede dejar de ser real.



Bloques censales (2020) y zonas inundables A y X, (2017) en una sección del Municipio de Villalba.

El siguiente query mostrará áreas originales de bloques, zonas inundables, nombre de bloque, zona inundable, y las proporciones de área que serán multiplicadas por la población 2020 de cada bloque censal.



```
/* Estimar población en zonas inundables
 * usando interpolación areal,
 * unidad censal mínima usada: census block, 2020
 * Municipio de Villalba
 */
drop table if exists
    workspace01.g00_interpol_areal_boques_inundables
    cascade; --borra la tabla si existe con opción cascade
create table workspace01.g00_interpol_areal_boques_inundables
    (gid serial primary key,
    b_geoid20 varchar,b_name20 varchar,
    b_pop100 integer,i_fld_zone varchar,
    area_blk_sqm numeric,inters_area numeric,
    proporcion numeric,pob_estim integer,
    geom geometry (multipolygon, 6566));
    --crear tabla vacía y definir sus columnas

insert into workspace01.g00_interpol_areal_boques_inundables
    (b_geoid20, b_name20,b_pop100,i_fld_zone,
    area_blk_sqm,inters_area,proporcion,pob_estim,
    geom)
    --insertar los récords en esta tabla vacía

select
    b.geoid20,b.name20,b.pop100,        --campos, tabla bloques censales, 2020
    i.fld_zone,                        --tipos de zonas inundables
    st_area(b.geom) as area_blk_sqm,   --área original bloques censales m2
    st_area(
        st_intersection(b.geom,i.geom) --la intersección de geoms
    ) as inters_area,                  --alias inters_area en m2
    st_area(
        st_intersection(b.geom,i.geom)) --calcula el área m2 de...
    /st_area(b.geom)                  --la intersección de geoms
    as proporcion_area,               --área original del bloque censal2020
    --alias. Proporción área intersecada
    ceil(                             --redondea al próximo entero...
        b.pop100* --multiplicación de población original bloque censal por...
        st_area( --...la proporción, calculada como el área de...
            st_intersection(b.geom,i.geom)) --intersección geoms
        /st_area(b.geom) --dividida entre el área original, census block
    ) as pob_estim,                    --alias pob_estim
    STX_Extract(                       --homogenizar geometrías ...
        st_intersection(b.geom,i.geom) --de la intersección geoms
        ,2)                            --opción #2 de STX_Extract: multipolygons
    as geom --alias geom

from
    pr_geodata.g31_censo2020_blk_villalba b,
    --tabla bloques censales 2020
    pr_geodata.g23_riesgo_inunda_floodzones_2017_villalba i
    --tabla zonas inundables 2018

where
    st_intersects(b.geom,i.geom) --que las geometrías intersequen

group by
    b.gid,i.gid --dependencia funcional

order by
    pob_estim desc; --población estimada descendente
```



PostGIS, 3.x

Resultados:

	Grid	123 gid	A-Z b_geoid20	A-Z b_name20	123 b_pop100	A-Z i_fld_zone	123 area_blk_sqm	123 inters_area	123 proporcion	123 pob_estim	123 gec
1		1	721497201003026	Block 3026	92	A	985,167.921312591	870,001.098239449	0.8830992965	81	MULTIP
2		45	721497204001009	Block 1009	60	A	74,218.932104953	9,271.3334974	0.1249187133	7	MULTIP
3		46	721497205023029	Block 3029	91	A	236,568.889352053	15,286.5905260795	0.0646179241	6	MULTIP
4		47	721497205041018	Block 1018	30	X	57,796.4016538049	10,657.5264604598	0.1843977506	6	MULTIP
5		2	721497204001002	Block 1002	93	X	13,951.0111884635	11,057.2384338947	0.7925761283	74	MULTIP
6		3	721497204001006	Block 1006	94	X	16,524.6916918515	11,501.944930172	0.6960459623	65	MULTIP
7		4	721497204001005	Block 1005	72	A	9,596.4744715726	8,328.6774983977	0.8678892986	62	MULTIP
8		5	721497204001001	Block 1001	101	A	50,225.9714555758	25,818.9758917195	0.5140562769	52	MULTIP
9		16	721497204001004	Block 1004	59	A	12,494.2469213945	3,627.5444776209	0.2903371848	17	MULTIP
10		6	721497205023020	Block 3020	104	A	167,489.036503097	72,692.0486791566	0.4340107878	45	MULTIP
11		7	721497204001004	Block 1004	59	X	12,494.2469213945	3,627.5444776209	0.2903371848	17	MULTIP

Ver usando el tab Spatial en DBeaver:



- ☐ Para un estimado de cuántos habitantes hay en zonas inundables en el Municipio de Villalba:

```
select sum(pob_estim) from workspace01.g00_interpol_areal_boques_inundables;
--suma: cuántos habitantes en Villalba están en zonas inundables
```

	123 sum
1	1,128

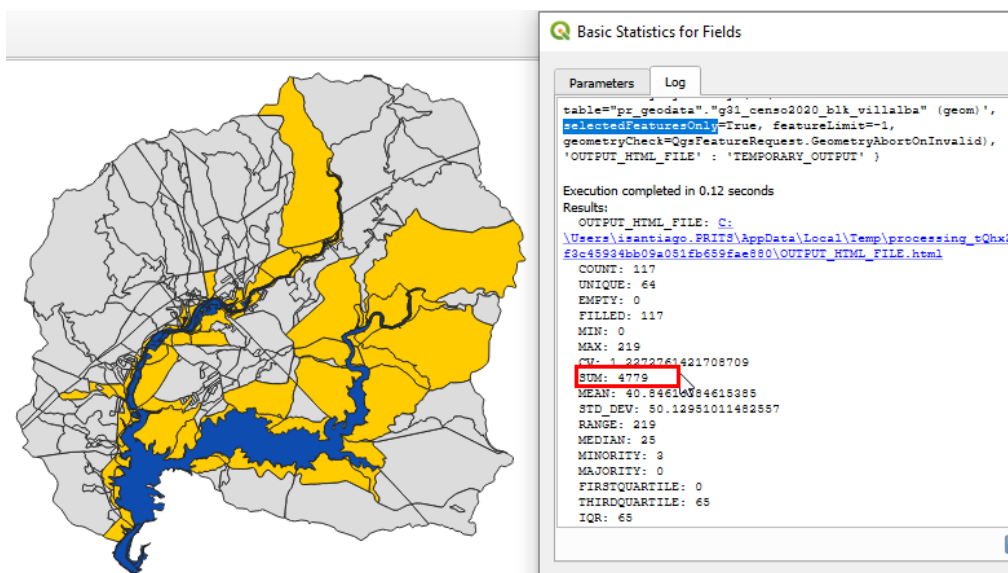
- ☐ Un estimado de población por tipo de zona inundable:

```
select i_fld_zone, sum(pob_estim)
from workspace01.g00_interpol_areal_boques_inundables
group by i_fld_zone;
--sumar por tipo de zona inundable
```

	A-Z i_fld_zone	123 sum
1	X	373
2	A	755

Estos estimados tienen las ventajas de basarse en el uso de la unidad cartográfica censal más pequeña y aplicar una proporción de área afectada por bloque.

Si usáramos un select by Location en QGIS el estimado es 4.2 veces mayor porque considerará toda el área de los bloques intersecados por zonas inundables.



El uso de puntos de estructuras sigue siendo la mejor opción pero deben distinguirse por uso comercial, residencial, etc. y diferenciarlos de estructuras menores auxiliares para no sobreestimar la población.



PRÁCTICAS

1. Utilice la función **ST_Intersection** para calcular el **kilometraje agregado de vías por tipo CFCC** (geodato de vías del Censo) **por barrio** en el municipio de Villalba. Este query se parece a otros hechos anteriormente usando **GROUP BY**. En este ejercicio debe usar **ST_Intersection** para obtener los **kilometrajes por barrio**.

Geodatos a usar: **g35_viales_mapa_base_tiger_rds_2006se_villalba** as **v**
g03_legales_barrios_2015_villalba as **b**

Query **sin usar ST_Intersection** y **sin usar** el campo **CFCC**.

```
1 select b.barrio, sum(st_length(v.geom))/1000 as kms
2 from g35_viales_mapa_base_tiger_rds_2006se_villalba v,
3 g03_legales_barrios_2015_villalba b
4 group by b.barrio
5 order by b.barrio;
```

Resultado erróneo. Agrupa por barrio, pero el cómputo no es correcto.

Grid	barrio	kms
1	Barrio Pueblo	350.073673643
2	Caonillas Abajo	350.073673643
3	Caonillas Arriba	350.073673643
4	Hato Puerco Abajo	350.073673643
5	Hato Puerco Arriba	350.073673643
6	Vacas	350.073673643
7	Villalba Abajo	350.073673643
8	Villalba Arriba	350.073673643

¿Dónde va la función **ST_Intersection**? Esta **debe** colocarse en la parte declarativa, **dentro del paréntesis** de la función **ST_Length(geom)** para que devuelva el cómputo correcto: **sum(largo(intersección(v.geom,b.geom)))**. Recuerde que debe añadir el campo **cfcc** después del campo **b.barrio** y añadirlo en las cláusulas **GROUP BY** y en **ORDER BY**, ambos ascendentes.

Resultado:

Grid	barrio	cfcc	kms
1	Barrio Pueblo	A31	1.750017493
2	Barrio Pueblo	A35	0
3	Barrio Pueblo	A41	4.371139844
4	Barrio Pueblo	A51	0
5	Barrio Pueblo	A63	0.0434059215
6	Caonillas Abajo	A31	8.1746037286
7	Caonillas Abajo	A35	0
8	Caonillas Abajo	A41	31.228903188
9	Caonillas Abajo	A51	1.5284468051
10	Caonillas Abajo	A63	0
11	Caonillas Arriba	A31	16.708309044
12	Caonillas Arriba	A35	0
13	Caonillas Arriba	A41	28.9282363404

Si quiere ver el **cómputo por barrio sin categorizar por cfcc**, elimine ese campo de la cláusula **SELECT**, de **GROUP BY** y de **ORDER BY**.

Grid	barrio	kms
1	Barrio Pueblo	6.1645632584
2	Caonillas Abajo	40.9319537217
3	Caonillas Arriba	48.5927614639
4	Hato Puerco Abajo	25.3858483654
5	Hato Puerco Arriba	76.770263829
6	Vacas	42.6120686582
7	Villalba Abajo	42.9107030151
8	Villalba Arriba	66.7055113312

2. Use el [geoproceso Clip](#) para **generar una nueva tabla** que sea el producto de **extraer** del geodato de **catastro de suelos** el **área del Municipio de Barranquitas**. Utilice el geodato de municipios, donde la condición sea **where m.municipio = 'Barranquitas'**.



PostGIS, 3.x

Primero, use el comando **DROP TABLE IF EXISTS g00_clip_soils_barranquitas CASCADE**; como preámbulo a generar la nueva tabla.

```
1 DROP TABLE IF EXISTS g00_clip_soils_barranquitas CASCADE;
```

El nuevo geodato de suelos **g00_clip_soils_barranquitas** solo debe tener los siguientes campos: **CREATE TABLE g00_clip_soils_barranquitas** (**gid** serial primary key, **musym** varchar, **muname** varchar, **slope_gradient_dominant_component** integer, **geom** geometry (multipolygon, 6566)).

```
1 DROP TABLE IF EXISTS g00_clip_soils_barranquitas CASCADE;
2 CREATE TABLE g00_clip_soils_barranquitas
3 (gid serial primary key,
4 musym varchar, muname varchar,
5 slope_gradient_dominant_component integer,
6 geom geometry (multipolygon, 6566));
```

Geodatos a utilizar: **g15_suelos_soil_map_units_2018** as **s**
g03_legales_municipios_2015 as **m**.

Use el comando **INSERT INTO** para insertar filas en la tabla nueva mediante un query que haga el CLIP, según discutido en [esa sección del tutorial](#).

```
8 insert into g00_clip_soils_barranquitas
9 --estos son los campos de la tabla que recibirán datos...
10 (musym, muname, slope_gradient_dominant_component, geom)
```

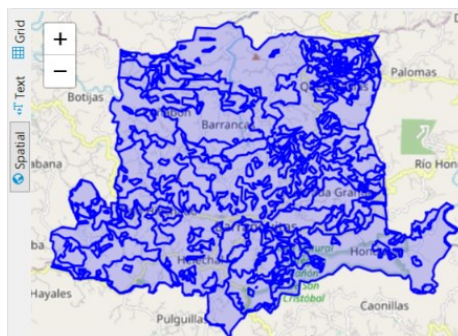
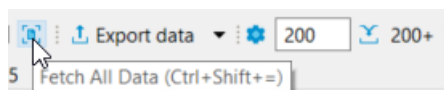
Y este será el query para insertar las filas a la nueva tabla. Recuerde usar la función extra STX_Extract,2 para homogenizar las geometrías y devolver sólo multipolygons.

```
11 --usando el siguiente query:
12 select musym, muname, slope_gradient_dominant_component,
13 stx_extract(st_intersection(s.geom, m.geom), 2) as geom
14 --no hay que usar st_union porque m.geom tendrá
15 -- solo una geometría: el munic de Barranquitas
16 from g15_suelos_soil_map_units_2018 as s,
17 g03_legales_municipios_2015 as m
18 where m.municipio = 'Barranquitas'
19 and s.geom && m.geom
20 and st_relate(s.geom, m.geom, 'T*****');
21 --comprobar:
22 select * from g00_clip_soils_barranquitas;
```

Resultado:

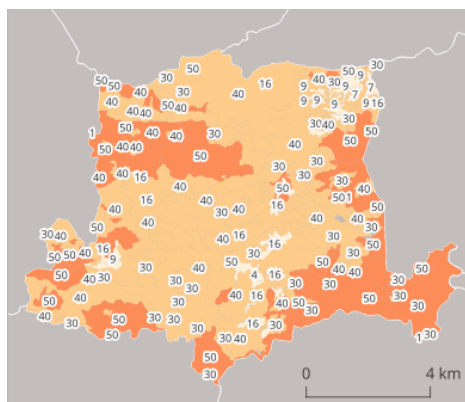
	Grid	gid	musym	muname	slope_gradient_dominant_component	geom
	1	1	MuE2	Múcara silty clay, 20 to 40 percent slopes, erod	30	MULTIPOLYGON (((21
	2	2	MuE2	Múcara silty clay, 20 to 40 percent slopes, erod	30	MULTIPOLYGON (((21
	3	3	MkF2	Maricao clay, 20 to 60 percent slopes	40	MULTIPOLYGON (((21
	4	4	MxD	Múcara clay, 12 to 20 percent slopes	15	MULTIPOLYGON (((21
	5	5	CuE	Consumo clay, 20 to 40 percent slopes	30	MULTIPOLYGON (((21

Cuando vaya a la pestaña **Spatial**, recuerde hacer **click** en el botón **Fetch All Data** porque la cantidad de filas es más de 200:

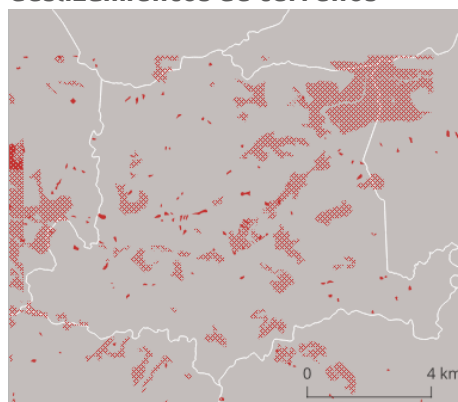


3. Use el [geoproceso Identity](#) para entrecruzar el **nuevo geodato derivado de suelos** para el **Municipio de Barranquitas** con el **geodato geológico** que contiene áreas de **muy alta susceptibilidad a deslizamientos de terreno**.

Pendientes en porciento



Muy alta susceptibilidad a deslizamientos de terrenos



- Haga una nueva tabla **g00_identity_suelos_landslides_barranquitas** que contenga los campos **g_gid**, **g_geol_id**, **g_unit_name** del **geodato geológico original**, más los campos del **geodato de suelos recortado** en el proceso **Clip** realizado en la práctica anterior: **s_gid**, **s_musym**, **s_muname** y **s_slope_gradient_dominant_component**. **Añada el campo risk_level**, que va a contener los niveles de riesgo. Esos niveles se asignarán después de este proceso. Añada al final el campo **geom** multipolygon con CRS código 6566.

Sección para generar la tabla nueva:

```
1 drop table if exists g00_identity_suelos_landslides_barranquitas cascade; --si existe, bórrala
2 create table g00_identity_suelos_landslides_barranquitas --crear tabla vacía
3 (gid serial primary key, --campo id serial primary key
4  g_gid integer, --id numérico original de cada fila del geodato geológico
5  g_geol_id varchar, --código id de cada mapping unit geológica
6  g_unit_name varchar, --nombre del mapping unit
7  s_gid integer, --id numérico original de cada fila del geodato suelos clip
8  s_musym varchar, --código id de unidad de suelo
9  s_muname varchar, --nombre de la unidad de suelo
10 s_slope_gradient_dominant_component integer, --slope percent
11 risk_level varchar, --nuevo campo: nivel de riesgo
12 geom geometry (multipolygon,6566)); --geometría, multipolígono, NAD83(2011)
```



Resultado:

Name	Value
Updated Rows	0
Query	<pre>--si existe, bórrala create table g00_identity_suelos_landslides_barranquitas --crear tabla vacía (gid serial primary key, --campo id serial primary key g_gid integer, --id numérico original de cada fila del geodato geológico g_geol_id varchar, --código id de cada mapping unit geológica g_unit_name varchar, --nombre del mapping unit s_gid integer, --id numérico original de cada fila del geodato suelos clip s_musym varchar, --código id de unidad de suelo s_muname varchar, --nombre de la unidad de suelo s_slope_gradient_dominant_component integer, --slope percent risk_level varchar, --nuevo campo: nivel de riesgo geom geometry (multipolygon,6566)) Start time Mon Apr 08 10:52:11 BOT 2024 Finish time Mon Apr 08 10:52:11 BOT 2024</pre>

Creó una tabla vacía, solo con la estructura de campos. Updated rows = 0

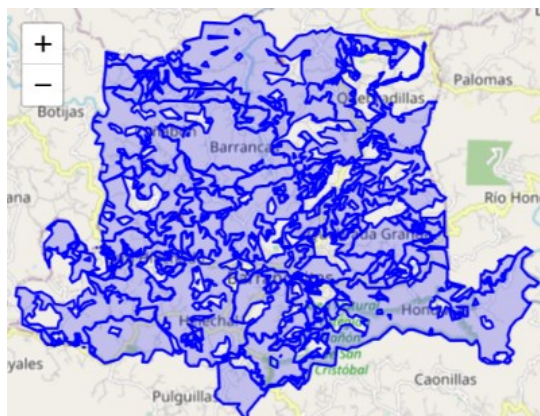
Realice ahora el paso Difference: A-B

Sección Difference (A-B), primera parte: Insertar polígonos del geodato clip de suelos que no solapan las geometrías del geodato geológico de deslizamientos.

```
14--Paso Difference (A-B) o (s-g)
15 --Primera parte: Devolver polígonos de la tabla "s"
16 --que son borrados parcial o totalmente por polígonos de tabla g
17 insert into g00_identity_suelos_landslides_barranquitas
18 (s_gid, s_musym, s_muname, s_slope_gradient_dominant_component, geom) --campos para llenar filas
19 select gid, musym, muname, slope_gradient_dominant_component, geom --campos, tabla suelos
20 from -- en lugar de una tabla física se obtienen de un subquery
21 (select s.gid,s.musym,s.muname,s.slope_gradient_dominant_component, --campos
22 stx_extract( --extraer multipolígonos "2" de...
23 st_difference(s.geom, --diferencia s menos
24 st_union(g.geom)),2) as geom --(la unión geometrías de tabla g)
25 from g00_clip_soils_barranquitas s, --tabla suelos_clip
26 g15_geologia_mapping_units_20k_2018_landslides g --tabla geología landslides
27 where s.geom && g.geom --comparador de bounding boxes
28 and st_relate(s.geom,g.geom, 'T*****')--solo interiores geoms intersequen
29 group by s.gid --agrupar por s.gid (para no repetir el s.gid)
30 ) as subq --alias del subquery, como si fuera una tabla temporal
31 where geom is not null; --condición: que las geoms no estén vacías
32 --corroborar
33 select * from g00_identity_suelos_landslides_barranquitas;
```

Resultado parcial, paso 1:

Name	Value
Updated Rows	130
Query	<pre>--Paso Difference (A-B) o (s-g) --Primera parte: Devolver polígonos de la tabla "s" --que son borrados parcial o totalmente por polígonos de tabla g insert into g00_identity_suelos_landslides_barranquitas (s_gid, s_musym, s_muname, s_slope_gradient_dominant_component, geom) --campos para llenar filas select gid, musym, muname, slope_gradient_dominant_component, geom --campos, tabla suelos from -- en lugar de una tabla física se obtienen de un subquery (select s.gid,s.musym,s.muname,s.slope_gradient_dominant_component, --campos stx_extract(--extraer multipolígonos "2" de... st_difference(s.geom, --diferencia s menos st_union(g.geom)),2) as geom --(la unión geometrías de tabla g) from g00_clip_soils_barranquitas s, --tabla suelos_clip g15_geologia_mapping_units_20k_2018_landslides g --tabla geología landslides where s.geom && g.geom --comparador de bounding boxes and st_relate(s.geom,g.geom, 'T*****')--solo interiores geoms intersequen group by s.gid --agrupar por s.gid (para no repetir el s.gid)) as subq --alias del subquery, como si fuera una tabla temporal where geom is not null Start time Mon Apr 08 10:58:28 BOT 2024 Finish time Mon Apr 08 10:58:29 BOT 2024</pre>



Paso #2 Difference: Insertar polígonos del geodato clip de suelos que coinciden con las geometrías del geodato geológico de deslizamientos. En este paso #2, ST_Difference creará espacios vacíos donde coincidan las geometrías del geodato

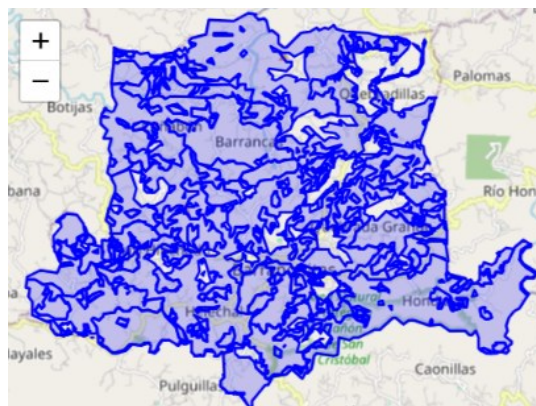


PostGIS, 3.x

geológico. Esos espacios se llenarán más adelante en el paso #3 ST_Intersection.

```
34 --Paso Difference (A-B) o (s-g)
35 --Segunda parte: Devolver polígonos de la tabla "s"
36 --donde las geometrías de los interiores de "s"
37 --Si intersecan geometrías de tabla "g"
38 insert into g00_identity_suelos_landslides_barranquitas
39 (s_gid,s_slope_gradient_dominant_component,geom) --campos para llenar filas
40 select gid, slope_gradient_dominant_component, geom --campos, tabla suelos
41 from -- en lugar de una tabla física se obtienen de un subquery
42 (select s.gid,s.slope_gradient_dominant_component, --campos tabla suelos
43 g.gid as g_gid, --id tabla geometría solo para comparar al final
44 st_multi(s.geom) as geom --homogenizar:convertir geometría en multipolígono
45 from g00_clip_soils_barranquitas s --tab suelos_clip left join guardar sus records
46 left join g15_geologia_mapping_units_20k_2018_landslides g --tabla geologia
47 on s.geom && g.geom --condiciones sin usar where: comparar bounding boxes
48 and st_relate(s.geom,g.geom, 'T*****')--solo intersecar interior geometrías
49 ) as subq --alias del subquery, como si fuera una tabla temporal
50 where g_gid is null; --condición: que el campo g_gid no esté vacío
51 --corroborar
52 select * from g00_identity_suelos_landslides_barranquitas;
```

Name	Value
Updated Rows	101
Query	--Paso Difference (A-B) o (s-g) --Segunda parte: Devolver polígonos de la tabla "s" --que no solapan con geometrías de la tabla "g" y --se conservan íntegros insert into g00_identity_suelos_landslides_barranquitas (s_gid,s_musym,s_muname,s_slope_gradient_dominant_component,geom) --campos para llenar filas select gid,musym,muname,slope_gradient_dominant_component,geom --campos, tabla suelos from -- en lugar de una tabla física se obtienen de un subquery (select s.gid,s.musym,s.muname,s.slope_gradient_dominant_component, --campos tabla suelos g.gid as g_gid, --id tabla geometría solo para comparar al final st_multi(s.geom) as geom --homogenizar:convertir geometría en multipolígono from g00_clip_soils_barranquitas s --tab suelos_clip left join guardar sus records left join g15_geologia_mapping_units_20k_2018_landslides g --tabla geologia on s.geom && g.geom --condiciones sin usar where: comparar bounding boxes and st_relate(s.geom,g.geom, 'T*****')--solo intersecar interior geometrías) as subq --alias del subquery, como si fuera una tabla temporal where g_gid is null Start time Mon Apr 08 11:00:03 BOT 2024 Finish time Mon Apr 08 11:00:03 BOT 2024



A este punto, nos queda **realizar la intersección geométrica** para completar los polígonos que faltan. Esos corresponden al geodato de geología que contiene solo deslizamientos, lateritas y saprolitas.

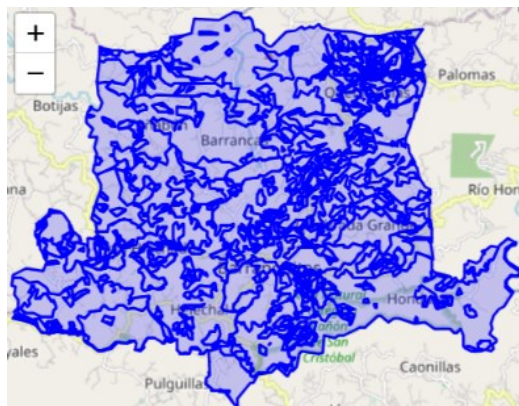


- Use la función **ST_Intersection** para **insertar los polígonos geológicos** que faltan en el geodato `g00_overlay_identity_suelos_landslides_barranquitas`

Sección intersection:

```
56 --Paso intersección A n B
57 insert into g00_identity_suelos_landslides_barranquitas
58 (g_gid, g_geol_id, g_unit_name,      --campos de geología
59  s_gid, s_musym, s_muname,          --datos de suelos
60  s_slope_gradient_dominant_component, --campos de suelos
61  geom)                               -- incluye geometría de los geoprocetos
62 select g_gid, g_geol_id, g_unit_name, --campos t geología
63        s_gid, s_musym, s_muname,      --campos t suelos
64        s_slope_gradient_dominant_component, --campos t suelos
65        stx_extract(                   --extraer solo polígonos "opción #2"
66        st_intersection(g.geom, s.geom), 2) as geom --intersección gns as geom
67 from g15_geologia_mapping_units_20k_2018_landslides g, --tab geología
68      g00_clip_soils_barranquitas s                --tab suelos
69 where g.geom && s.geom                             --comparador bounding boxes
70 and st_relate(g.geom, s.geom, 'T*****'); --solo las que intersequen interiores
71 --corroborar
72 select * from g00_identity_suelos_landslides_barranquitas;
```

Statistics 1 X	
Name	Value
Updated Rows	400
Query	--
	--Paso intersección A n B
	insert into g00_identity_suelos_landslides_barranquitas
	(g_gid, g_geol_id, g_unit_name, --campos de geología
	s_gid, s_musym, s_muname, --datos de suelos
	s_slope_gradient_dominant_component, --campos de suelos
	geom) -- incluye geometría de los geoprocetos
	select g_gid, g_geol_id, g_unit_name, --campos t geología
	s_gid, s_musym, s_muname, --campos t suelos
	s_slope_gradient_dominant_component, --campos t suelos
	stx_extract(--extraer solo polígonos "opción #2"
	st_intersection(g.geom, s.geom), 2) as geom --intersección gns as geom
	from g15_geologia_mapping_units_20k_2018_landslides g, --tab geología
	g00_clip_soils_barranquitas s --tab suelos
	where g.geom && s.geom --comparador bounding boxes
	and st_relate(g.geom, s.geom, 'T*****')
Start time	Mon Apr 08 11:08:48 BOT 2024
Finish time	Mon Apr 08 11:08:49 BOT 2024





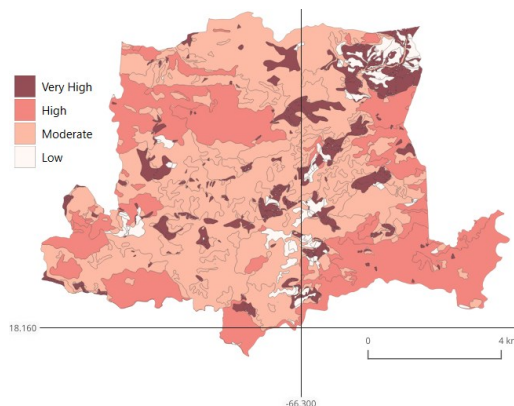
PostGIS, 3.x

4. **Actualice** el contenido del campo **risk_level** basado en el siguiente script usando la cláusula CASE-WHEN-THEN. Usar **SET risk_level =**

```
74 --UPDATE risk_level
75 update g00_identity_suelos_landslides_barranquitas --actualiza esta tabla
76 set risk_level = --en el campo risk_level...
77 --cuando g_gid no esté vacío y slope >29%,
78 CASE WHEN g_gid is not NULL and s_slope_gradient_dominant_component > 29
79 THEN 'Very High' --asigna 'Very High' a risk_level
80 --cuando g_gid esté vacío y slope >=50%,
81 WHEN g_gid is null and s_slope_gradient_dominant_component >= 50
82 then 'High' --asigna 'High' a risk_level
83 --cuando g_gid esté vacío y slope esté entre 30 y 49%,
84 WHEN g_gid is null and s_slope_gradient_dominant_component between 30 and 49
85 then 'Moderate' --asigna 'Moderate' a risk_level
86 --cuando g_gid esté vacío y slope esté entre 1 y 29%,
87 WHEN g_gid is null and s_slope_gradient_dominant_component between 1 and 29
88 then 'Low' --asigna 'Low' a risk_level
89 --cuando g_gid no esté vacío y slope esté entre 1 y 29%,
90 WHEN g_gid is not null and s_slope_gradient_dominant_component between 1 and 29
91 then 'Low' --asigna 'Low' a risk_level
92 WHEN s_musym = 'W' --cuando este campo = W,
93 then 'Water' --asigna label 'Water' a risk_level
94 end;
```

Resultados:

Name	Value
Updated Rows	631
Query	<pre>--UPDATE risk_level update g00_identity_suelos_landslides_barranquitas --actualiza esta tabla set risk_level = --en el campo risk_level... CASE WHEN g_gid is not NULL and s_slope_gradient_dominant_component > 29 --cuando g_gid no esté vacío y slope >29%, THEN 'Very High' --asigna 'Very High' a risk_level WHEN g_gid is null and s_slope_gradient_dominant_component >= 50 --cuando g_gid esté vacío y slope >=50%, then 'High' --asigna 'High' a risk_level WHEN g_gid is null and s_slope_gradient_dominant_component between 30 and 49 --cuando g_gid esté vacío y slope esté entre 30 y 49%, then 'Moderate' --asigna 'Moderate' a risk_level WHEN g_gid is null and s_slope_gradient_dominant_component between 1 and 29 --cuando g_gid esté vacío y slope esté entre 1 y 29%, then 'Low' --asigna 'Low' a risk_level WHEN g_gid is not null and s_slope_gradient_dominant_component between 1 and 29 --cuando g_gid no esté vacío y slope esté entre 1 y 29%, then 'Low' --asigna 'Low' a risk_level WHEN s_musym = 'W' --cuando este campo = W, then 'Water' --asigna label 'Water' a risk_level end</pre>
Start time	Mon Apr 08 12:05:24 BOT 2024
Finish time	Mon Apr 08 12:05:24 BOT 2024



Visualizar en QGIS

Felicidades. Usted ha mejorado el mapa de susceptibilidad a deslizamientos de terreno en relación con el mapa de 1979.

5. Evalúe cuántas personas (Censo 2020) están dentro de zonas de riesgo por deslizamientos en Municipio de Villalba. Haga el desglose por nivel de riesgo. Para calcular el número de personas deberá hacer **interpolación por proporción de área ocupada** contra el dato de **población del censo** en el **geodato de bloques censales 2020**.

Geodatos a usar: **g00_overlay_suelos_landslides_2024_villalba**
g31_censo2020_blk_villalba

Realice el cómputo a nivel municipal y por nivel de riesgo.

Refiérase a la [sección interpolación areal](#) como guía para realizar esta práctica.



GEODATOS RÁSTER EN POSTGIS

Los geodatos en formato ráster también pueden ser guardados en PostgreSQL. Hay dos maneras para registrar datos: 1) Guardar los datos **en la base de datos**, y 2) Guardar **referencias a un archivo ráster externo**. En ocasiones, cuando la imagen o ráster es muy voluminoso, en Giga bytes, por ejemplo, se prefiere la opción de registrar el archivo como referencia externa.

Vamos a realizar un ejemplo con un ráster relativamente pequeño y previamente guardado en la base de datos.

Demo: Uso de geodato ráster para calcular una columna de geodato vectorial.

En este ejemplo haremos una **comparación entre los valores de pendientes en por ciento en los geodatos** de:

- suelos-NRCS
- ráster-DEM2015 (5x5 m por píxel)

Como se vio anteriormente en las prácticas, el geodato de suelos-NRCS contiene una columna en la cual se registra la pendiente promedio en por ciento del componente dominante de una unidad cartográfica (mapping unit, dominant component). Por otro lado, el geodato ráster de pendientes nos dará el cálculo directo de pendientes en por ciento en cada píxel.

Al final presentaremos un mapa para comparar la pendiente estimada en el geodato de suelos con las pendientes directas del geodato ráster.

Crear geodato vectorial que va a contener el campo **slope_dominant_component** de la tabla de suelos y el campo **mean_slope_pct** que se extraerá del geodato ráster de pendientes.

```
drop table if exists g00_slope_comparar_villalba cascade;  
create table g00_slope_comparar_villalba  
  (gid serial primary key,  
   mean_slope_pct double precision,  
   g_unit_code varchar,  
   slope_dominant_component integer,  
   difference double precision,  
   geom geometry(multipolygon, 6566));
```

Resultado:

Statistics 1	×
Name	Value
Updated Rows	0
Execute time	0.014s
Start time	Fri Jun 27 15:12:18 BOT 2025
Finish time	Fri Jun 27 15:12:18 BOT 2025
Query	create table g00_slope_comparar_villalba (gid serial primary key, mean_slope_pct double precision, g_unit_code varchar, slope_dominant_component integer, difference double precision, geom geometry(multipolygon, 6566))



PostGIS, 3.x

Insertar filas en la tabla nueva. Este query puede tardar más que otros realizados.

```
8=insert into g00_slope_comparar_villalba --insertar filas en el geodato nuevo
9 (mean_slope_pct, slope_dominant_component,g_unit_code, difference, geom) --campos
10 -- A base de este query:
11 select (
12     st_summarystats(                                --summary statistics...
13         st_union(                                    --unir los tiles de píxeles
14             st_clip(r.rast,d.geom,true)),1)).mean --traer el promedio de pendientes%
15     as mean_slope_pct,                                --as mean_slope_pct
16     s_slope_gradient_dominant_component, --campo slope % NRCS
17     g_unit_code,                                       --código unidad geológica,
18     d.s_slope_gradient_dominant_component - --calc diferencia entre
19     (st_summarystats(                                --el campo slope%-NRCS
20         st_union(st_clip(r.rast,d.geom,true)),1)).mean --menos la pendiente DEM
21     as difference,                                    --as difference
22     d.geom as geom --traer geometrías, geodato nuevo landslides
23 from g15_dem2015_5m_slope_percent_villalba r --raster pendientes %
24 join g00_overlay_identity_landslides_2024_villalba d --landslides
25 on st_intersects(r.rast,d.geom) --Condición: que intersequen
26 group by d.gid; --agrupar por g.id
```

Resultado. El query demoró poco más de 3 minutos.

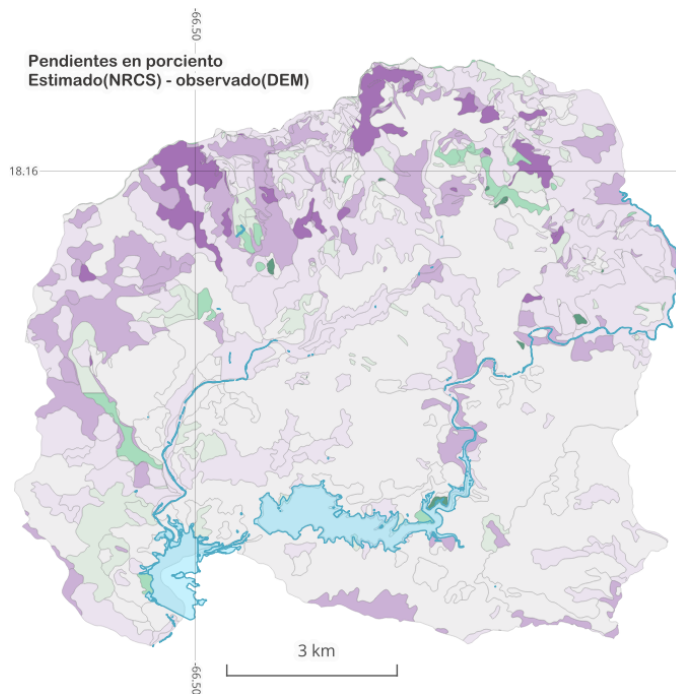
Statistics 1 ×	
Name	Value
Updated Rows	625
Query	insert into g00_slope_comparar_villalba --insertar filas en el geodato nuevo (mean_slope_pct, slope_dominant_component,g_unit_code, difference, geom) --campos -- A base de este query: select (st_summarystats(--summary statistics... st_union(--unir los tiles de píxeles st_clip(r.rast,d.geom,true)),1)).mean --traer el promedio de pendientes% as mean_slope_pct, --as mean_slope_pct s_slope_gradient_dominant_component, --campo slope % NRCS g_unit_code, --código unidad geológica, d.s_slope_gradient_dominant_component - --calc diferencia entre (st_summarystats(--el campo slope%-NRCS st_union(st_clip(r.rast,d.geom,true)),1)).mean --menos la pendiente DEM as difference, --as difference d.geom as geom --traer geometrías, geodato nuevo landslides from g15_dem2015_5m_slope_percent_villalba r --raster pendientes % join g00_overlay_identity_landslides_2024_villalba d --landslides on st_intersects(r.rast,d.geom) --Condición: que intersequen group by d.gid Start time Tue Apr 16 09:35:03 BOT 2024 Finish time Tue Apr 16 09:38:10 BOT 2024



Resultados

% Pendientes estimadas en mapping units de NRCS

- Pendientes subestimadas: -96% hasta -45%
- 45% hasta -25%
- 25% hasta -10%
- Pendientes subestimadas, poca diferencia: -10% hasta 0%
- Pendientes sobreestimadas, poca diferencia: 0% hasta 10.0%
- 10% hasta 20.0%
- Pendientes sobreestimadas: 20% hasta 41.3%
- Cuerpos de agua superficial



Las áreas en **tonos púrpuras** corresponden a pendientes subestimadas. La pendiente estimada en el **geodato de suelos-NRCS es menor al promedio de pendientes calculadas del modelo digital de elevaciones derivado de Lidar (USGS, 2015)**. Esto corresponde con las áreas más escarpadas, al norte del Municipio. Las áreas en tonos grises y verde claro son áreas donde no hay mucha diferencia entre la estimación de promedio de pendientes del NRCS y el promedio de pendientes calculado del DEM. **Las áreas en tonos verdes oscuros** corresponden a pendientes sobreestimadas. La pendiente promedio en el geodato de suelos-**NRCS fue mayor** que el **promedio de pendientes, derivado del DEM** (*pendientes sobreestimadas*). Las **áreas en azul** corresponden a los **embalses** de *Aceituna* al norte, *Guayabal* al suroeste y *Toa Vaca*.

En su mayoría, las pendientes en el norte están subestimadas en las unidades cartográficas del geodato de suelos NRCS. En el área central y sureste, la estimación no es muy diferente al promedio derivado de pendientes del modelo digital de elevaciones.



Si tomamos las áreas ocupadas por estimación, obtenemos el siguiente resultado:

```
select 'Subestimada en NRCS' as "Áreas_comparadas",
       round(sum(st_area(geom)/1000)::numeric,2) as sq_kms,
       round(sum(st_area(geom)/1000/958.8)::numeric,2) as pct
from g00_slope_comparar_villalba
where difference < 0
union
select 'Sobreestimada en NRCS' as "Áreas_comparadas",
       round(sum(st_area(geom)/1000)::numeric,2) as sq_kms,
       round(sum(st_area(geom)/1000/958.8)::numeric,2) as pct
from g00_slope_comparar_villalba
where difference > 0
order by pct desc;
```

	A-Z Áreas_comparadas	123 sq_kms	123 pct
1	Subestimada en NRCS	89,749.37	93.61
2	Sobreestimada en NRCS	6,126.83	6.39

El área total municipal es de 95.88 kilómetros cuadrados.



Descarga de datos para el ejercicio:

Los datos pueden ser descargados desde este enlace:

https://gis.otg.pr.gov/Downloads/Tutorials/tutorial_postgis/_postgis_training_gisdb_lab_geo_data_sql_files.zip



Apéndice A-1: Inventario de relaciones topológicas

Este es un inventario con una lista ilustrada de las **98 relaciones topológicas** posibles entre **objetos** geográficos *simples* de tipo **puntual**, **lineal** y de **área**/superficie.

Ya que cada intersección de la **matriz 9-IM** puede tomar **4 valores diferentes (F, 0, 1, 2)**, existen potencialmente 4^9 combinaciones posibles o **262,144 posibles matrices diferentes**. Sin embargo, **no todas se pueden realizar** por causa de restricciones relacionadas a las características topológicas de estos objetos.

Por ejemplo, un punto no tiene contorno/límite, por lo tanto los valores de intersecciones con objetos geométricos con el contorno/límite de un punto siempre será igual a F. Vea la primera línea horizontal de las matrices en la tabla A-2. Esto aplica a las relaciones de intersección entre los interiores, contornos/límites y los exteriores de los objetos.

Por ejemplo, la intersección de los exteriores de los objetos siempre será de dimensión = 2 (Área) y ningún polígono cubrirá totalmente el plano. Por tal razón, la novena celda de la matriz de intersección será siempre = 2. Otro ejemplo de restricción: si A y B son 2 objetos no puntuales, el contorno/límite de A interseca al menos una parte de B, ya sea el interior, límite o exterior y viceversa. De este modo, las matrices donde las líneas horizontales del medio horizontales o verticales solo tengan valores F no podrían existir. La lista de relaciones punto/punto (P/P), punto/polilínea (P/L), y punto/polígono (P/S) es evidente. Las relaciones que realmente existen entre polilínea/polilínea (L/L), polilínea y polígono (L/S) y polígono/polígono (S/S) han sido deducidas del conjunto de relaciones posibles entre los objetos mediante la aplicación de las restricciones topológicas descritas por Max J. Egenhofer y considerando la dimensión de las estas intersecciones (modelo DE-9IM)

Repasando:

Tabla A-1: Matriz de intersección 9-IM

	Interior	Límite/Contorno/Boundary	Exterior
Punto (Dim=0)	Dim=0	Vacío / No existe	
Poli-línea (Dim=1)	Dim=1	Dim=0	
Polígono/área (Dim=2)	Dim=2	Dim=1	

Referencia: [Predicados OGC: 2.2: Conceptos de interior, límite y exterior de objetos](#), pág. 6.
Ministerio de Ecología, desarrollo sustentable, transportación y vivienda, Gobierno de Francia.

La expresión de las restricciones relacionadas a la topología de los objetos se ilustra mediante las matrices de intersección de la tabla A-2.



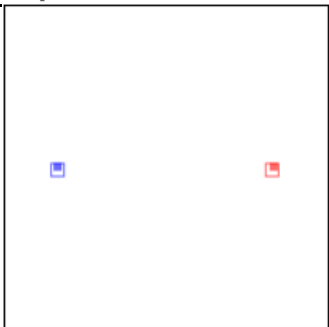
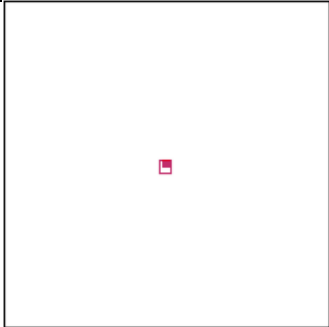
Tabla A-2: Expresión de las limitaciones relacionados a la topología de los objetos.

P/P	P/L	P/S																											
<table> <tr><td>F, 0</td><td>F</td><td>F, 0</td></tr> <tr><td>F</td><td>F</td><td>F</td></tr> <tr><td>F, 0</td><td>F</td><td>2</td></tr> </table>	F, 0	F	F, 0	F	F	F	F, 0	F	2	<table> <tr><td>F, 0</td><td>F, 0</td><td>F, 0</td></tr> <tr><td>F</td><td>F</td><td>F</td></tr> <tr><td>1</td><td>0</td><td>2</td></tr> </table>	F, 0	F, 0	F, 0	F	F	F	1	0	2	<table> <tr><td>F, 0</td><td>F, 0</td><td>F, 0</td></tr> <tr><td>F</td><td>F</td><td>F</td></tr> <tr><td>2</td><td>1</td><td>2</td></tr> </table>	F, 0	F, 0	F, 0	F	F	F	2	1	2
F, 0	F	F, 0																											
F	F	F																											
F, 0	F	2																											
F, 0	F, 0	F, 0																											
F	F	F																											
1	0	2																											
F, 0	F, 0	F, 0																											
F	F	F																											
2	1	2																											
L/P	L/L	L/S																											
<table> <tr><td>F, 0</td><td>F</td><td>1</td></tr> <tr><td>F, 0</td><td>F</td><td>0</td></tr> <tr><td>F, 0</td><td>F</td><td>2</td></tr> </table>	F, 0	F	1	F, 0	F	0	F, 0	F	2	<table> <tr><td>F, 0, 1</td><td>F, 0</td><td>F, 1</td></tr> <tr><td>F, 0</td><td>F, 0</td><td>F, 0</td></tr> <tr><td>F, 1</td><td>F, 0</td><td>2</td></tr> </table>	F, 0, 1	F, 0	F, 1	F, 0	F, 0	F, 0	F, 1	F, 0	2	<table> <tr><td>F, 1</td><td>F, 0, 1</td><td>F, 1</td></tr> <tr><td>F, 0</td><td>F, 0</td><td>F, 0</td></tr> <tr><td>2</td><td>1</td><td>2</td></tr> </table>	F, 1	F, 0, 1	F, 1	F, 0	F, 0	F, 0	2	1	2
F, 0	F	1																											
F, 0	F	0																											
F, 0	F	2																											
F, 0, 1	F, 0	F, 1																											
F, 0	F, 0	F, 0																											
F, 1	F, 0	2																											
F, 1	F, 0, 1	F, 1																											
F, 0	F, 0	F, 0																											
2	1	2																											
S/P	S/L	S/S																											
<table> <tr><td>F, 0</td><td>F</td><td>2</td></tr> <tr><td>F, 0</td><td>F</td><td>1</td></tr> <tr><td>F, 0</td><td>F</td><td>2</td></tr> </table>	F, 0	F	2	F, 0	F	1	F, 0	F	2	<table> <tr><td>F, 1</td><td>F, 0</td><td>2</td></tr> <tr><td>F, 0, 1</td><td>F, 0</td><td>1</td></tr> <tr><td>F, 1</td><td>F, 0</td><td>2</td></tr> </table>	F, 1	F, 0	2	F, 0, 1	F, 0	1	F, 1	F, 0	2	<table> <tr><td>F, 2</td><td>F, 1</td><td>F, 2</td></tr> <tr><td>F, 1</td><td>F, 0, 1</td><td>F, 1</td></tr> <tr><td>F, 2</td><td>F, 1</td><td>2</td></tr> </table>	F, 2	F, 1	F, 2	F, 1	F, 0, 1	F, 1	F, 2	F, 1	2
F, 0	F	2																											
F, 0	F	1																											
F, 0	F	2																											
F, 1	F, 0	2																											
F, 0, 1	F, 0	1																											
F, 1	F, 0	2																											
F, 2	F, 1	F, 2																											
F, 1	F, 0, 1	F, 1																											
F, 2	F, 1	2																											

Para cada matriz, las combinaciones de valores de intersección (F,0,1,2) no corresponden a relaciones topológicas reales.

Por cada relación se presentará el esquema o dibujo geométrico, la matriz de intersección, si es simétrica; es decir si la relación topológica es conmutativa y cuál es el valor devuelto por los predicados.

Punto/Punto: 2 relaciones

Representación	Matriz	Conmutativa	Predicados																																	
	Intersection Matrix AB FF0FFF0F2 BA FF0FFF0F2 B Int Body Ext Int F F 0 A Body F F F Ext 0 F 2	Sí	Binary Predicates <table><tr><td></td><td>AB</td><td>BA</td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>T</td><td>T</td></tr><tr><td>Intersects</td><td>F</td><td>F</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		AB	BA	Equals	F	F	Disjoint	T	T	Intersects	F	F	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
	AB	BA																																		
Equals	F	F																																		
Disjoint	T	T																																		
Intersects	F	F																																		
Touches	F	F																																		
Crosses	F	F																																		
Within	F	F																																		
Contains	F	F																																		
Overlaps	F	F																																		
Covers	F	F																																		
CoveredBy	F	F																																		
	Intersection Matrix AB OFFFFF2 BA OFFFFF2 B Int Body Ext Int 0 F F A Body F F F Ext F F 2	Sí	Binary Predicates <table><tr><td></td><td>AB</td><td>BA</td></tr><tr><td>Equals</td><td>T</td><td>T</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>T</td></tr><tr><td>Contains</td><td>T</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>T</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>T</td></tr></table>		AB	BA	Equals	T	T	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	T	Contains	T	T	Overlaps	F	F	Covers	T	T	CoveredBy	T	T
	AB	BA																																		
Equals	T	T																																		
Disjoint	F	F																																		
Intersects	T	T																																		
Touches	F	F																																		
Crosses	F	F																																		
Within	T	T																																		
Contains	T	T																																		
Overlaps	F	F																																		
Covers	T	T																																		
CoveredBy	T	T																																		



PostGIS, 3.x

Punto/Polilínea: 3 relaciones

Representación	Matriz	Conmutativa	Predicados																																	
	Intersection Matrix AB FF0FFF102 BA FF1FF00F2 B Int Bdy Ext Int F F 0 A Bdy F F F Ext 1 0 2	NO	Binary Predicates <table><tr><th></th><th>AB</th><th>BA</th></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>T</td><td>T</td></tr><tr><td>Intersects</td><td>F</td><td>F</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		AB	BA	Equals	F	F	Disjoint	T	T	Intersects	F	F	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
	AB	BA																																		
Equals	F	F																																		
Disjoint	T	T																																		
Intersects	F	F																																		
Touches	F	F																																		
Crosses	F	F																																		
Within	F	F																																		
Contains	F	F																																		
Overlaps	F	F																																		
Covers	F	F																																		
CoveredBy	F	F																																		
	Intersection Matrix AB F0FFFF102 BA FF10F0FF2 B Int Bdy Ext Int F 0 F A Bdy F F F Ext 1 0 2	NO	Binary Predicates <table><tr><th></th><th>AB</th><th>BA</th></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
	AB	BA																																		
Equals	F	F																																		
Disjoint	F	F																																		
Intersects	T	T																																		
Touches	T	T																																		
Crosses	F	F																																		
Within	F	F																																		
Contains	F	F																																		
Overlaps	F	F																																		
Covers	F	T																																		
CoveredBy	T	F																																		
	Intersection Matrix AB 0FFFFFF102 BA 0F1FF0FF2 B Int Bdy Ext Int 0 F F A Bdy F F F Ext 1 0 2	NO	Binary Predicates <table><tr><th></th><th>AB</th><th>BA</th></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	F	Contains	F	T	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
	AB	BA																																		
Equals	F	F																																		
Disjoint	F	F																																		
Intersects	T	T																																		
Touches	F	F																																		
Crosses	F	F																																		
Within	T	F																																		
Contains	F	T																																		
Overlaps	F	F																																		
Covers	F	T																																		
CoveredBy	T	F																																		



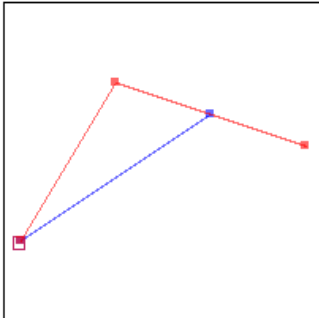
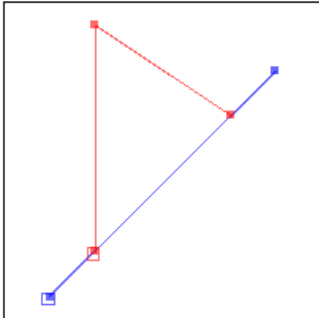
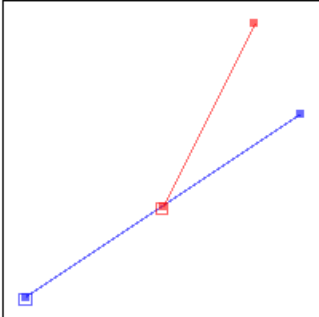
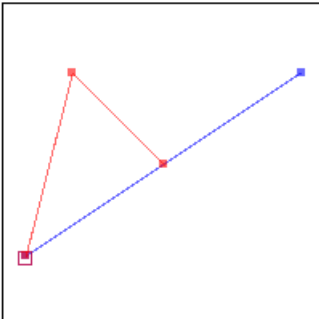
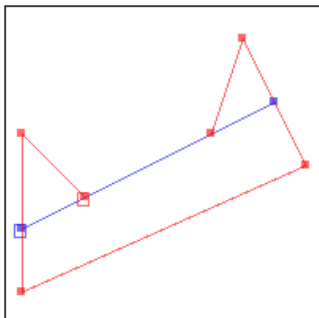
PostGIS, 3.x

Polilínea/Polilínea: 47 relaciones

Representación	Matriz	Conmutativa	Predicados																																																		
	Intersection Matrix AB FF1FF0102 BA FF1FF0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	1	A	<i>Bdy</i>	F	F	0	<i>Ext</i>	1	0	2	Sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>T</td><td>T</td></tr><tr><td>Intersects</td><td>F</td><td>F</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	T	T	Intersects	F	F	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	F	F	1																																																		
A	<i>Bdy</i>	F	F	0																																																	
<i>Ext</i>	1	0	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	T	T																																																			
Intersects	F	F																																																			
Touches	F	F																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB FF1F0F1F2 BA FF1F0F1F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	1	A	<i>Bdy</i>	F	0	F	<i>Ext</i>	1	F	2	Sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	F	F	1																																																		
A	<i>Bdy</i>	F	0	F																																																	
<i>Ext</i>	1	F	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	T	T																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB FF1F00102 BA FF1F00102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	1	A	<i>Bdy</i>	F	0	0	<i>Ext</i>	1	0	2	Sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	F	F	1																																																		
A	<i>Bdy</i>	F	0	0																																																	
<i>Ext</i>	1	0	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	T	T																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB FF10FF102 BA F01FF01F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	1	A	<i>Bdy</i>	0	F	F	<i>Ext</i>	1	0	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	F	F	1																																																		
A	<i>Bdy</i>	0	F	F																																																	
<i>Ext</i>	1	0	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	T	T																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB FF10F0102 BA F01FF0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	1	A	<i>Bdy</i>	0	F	0	<i>Ext</i>	1	0	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	F	F	1																																																		
A	<i>Bdy</i>	0	F	0																																																	
<i>Ext</i>	1	0	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	T	T																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			



PostGIS, 3.x

Representación	Matriz	Conmutativa	Predicados																																																		
	Intersection Matrix AB FF100F102 BA F01F001F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	1	A	<i>Bdy</i>	0	0	F	<i>Ext</i>	1	0	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	F	F	1																																																		
A	<i>Bdy</i>	0	0	F																																																	
<i>Ext</i>	1	0	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	T	T																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB F01FF01F2 BA FF10FF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	F	F	0	<i>Ext</i>	1	F	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	F	0	1																																																		
A	<i>Bdy</i>	F	F	0																																																	
<i>Ext</i>	1	F	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	T	T																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB F01FF0102 BA FF10F0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	F	F	0	<i>Ext</i>	1	0	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	F	0	1																																																		
A	<i>Bdy</i>	F	F	0																																																	
<i>Ext</i>	1	0	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	T	T																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB F01F001F2 BA FF100F102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	F	0	0	<i>Ext</i>	1	F	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	F	0	1																																																		
A	<i>Bdy</i>	F	0	0																																																	
<i>Ext</i>	1	F	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	T	T																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB F010FF1F2 BA F010FF1F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	0	F	F	<i>Ext</i>	1	F	2	sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	F	0	1																																																		
A	<i>Bdy</i>	0	F	F																																																	
<i>Ext</i>	1	F	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	T	T																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			



PostGIS, 3.x

Representación	Matriz	Conmutativa	Predicados																																																	
	Intersection Matrix AB F010FF102 BA F010F01F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	0	F	<i>Ext</i>	1	0	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	F	0	1																																																	
A	<i>Bdy</i>	0	F																																																	
<i>Ext</i>	1	0	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	T	T																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB F010F01F2 BA F010FF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	0	F	<i>Ext</i>	1	F	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	F	0	1																																																	
A	<i>Bdy</i>	0	F																																																	
<i>Ext</i>	1	F	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	T	T																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB F010F0102 BA F010F0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	0	F	<i>Ext</i>	1	0	2	sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	F	0	1																																																	
A	<i>Bdy</i>	0	F																																																	
<i>Ext</i>	1	0	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	T	T																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB F0100F1F2 BA F0100F1F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	0	F	<i>Ext</i>	1	F	2	sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	F	0	1																																																	
A	<i>Bdy</i>	0	F																																																	
<i>Ext</i>	1	F	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	T	T																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 1FFF0FFF2 BA 1FFF0FFF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Bdy</i>	F	F	<i>Ext</i>	F	F	2	sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>T</td><td>T</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>T</td></tr><tr><td>Contains</td><td>T</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>T</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>T</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	T	T	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	T	Contains	T	T	Overlaps	F	F	Covers	T	T	CoveredBy	T	T
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	F	F																																																	
A	<i>Bdy</i>	F	F																																																	
<i>Ext</i>	F	F	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	T	T																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	T	T																																																		
Contains	T	T																																																		
Overlaps	F	F																																																		
Covers	T	T																																																		
CoveredBy	T	T																																																		

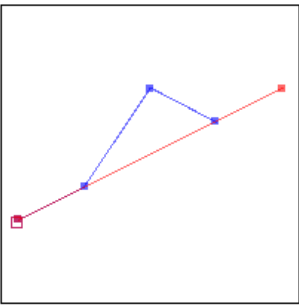
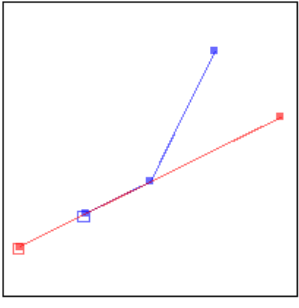
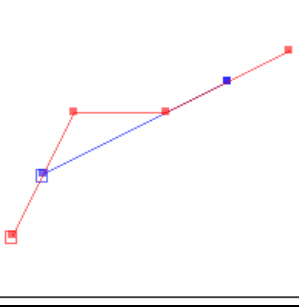
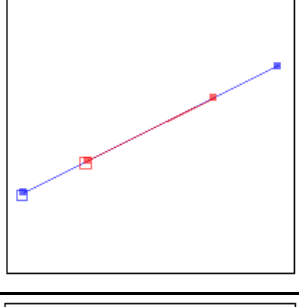
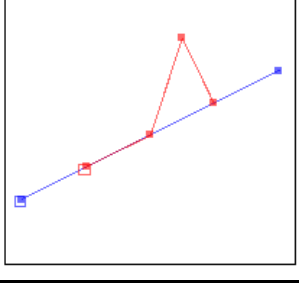


PostGIS, 3.x

Representación	Matriz	Conmutativa	Predicados																																																	
	Intersection Matrix AB 1FF0FF102 BA 101FF0FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Bdy</i>	0	F	<i>Ext</i>	1	0	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	F	Contains	F	T	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	F	F																																																	
A	<i>Bdy</i>	0	F																																																	
<i>Ext</i>	1	0	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	T	F																																																		
Contains	F	T																																																		
Overlaps	F	F																																																		
Covers	F	T																																																		
CoveredBy	T	F																																																		
	Intersection Matrix AB 1FF00F102 BA 101F00FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Bdy</i>	0	0	<i>Ext</i>	1	0	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	F	Contains	F	T	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	F	F																																																	
A	<i>Bdy</i>	0	0																																																	
<i>Ext</i>	1	0	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	T	F																																																		
Contains	F	T																																																		
Overlaps	F	F																																																		
Covers	F	T																																																		
CoveredBy	T	F																																																		
	Intersection Matrix AB 1F1FF0102 BA 1F1FF0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	1	0	2	Sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	F	1																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	1	0	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	T	T																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 1F1F0F1F2 BA 1F1F0F1F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	1	F	2	sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	F	1																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	1	F	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	T	T																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 1F1F00102 BA 1F1F00102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	1	0	2	sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	F	1																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	1	0	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	T	T																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		



PostGIS, 3.x

Representación	Matriz	Conmutativa	Predicados																																																	
	Intersection Matrix AB 1F100F102 BA 101F001F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	1	A	<i>Bdy</i>	0	0	<i>Ext</i>	1	0	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	F	1																																																	
A	<i>Bdy</i>	0	0																																																	
<i>Ext</i>	1	0	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	T	T																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 1F10F0102 BA 101FF0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	1	A	<i>Bdy</i>	0	0	<i>Ext</i>	1	0	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	F	1																																																	
A	<i>Bdy</i>	0	0																																																	
<i>Ext</i>	1	0	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	T	T																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 1F10FF102 BA 101FF01F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	1	A	<i>Bdy</i>	0	F	<i>Ext</i>	1	0	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	F	1																																																	
A	<i>Bdy</i>	0	F																																																	
<i>Ext</i>	1	0	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	T	T																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 101FF0FF2 BA 1FF0FF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	F	F	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>T</td></tr><tr><td>Contains</td><td>T</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>T</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>T</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	T	Contains	T	F	Overlaps	F	F	Covers	T	F	CoveredBy	F	T
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	0	1																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	F	F	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	F	T																																																		
Contains	T	F																																																		
Overlaps	F	F																																																		
Covers	T	F																																																		
CoveredBy	F	T																																																		
	Intersection Matrix AB 101FF01F2 BA 1F10FF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	1	F	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	0	1																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	1	F	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	T	T																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		



PostGIS, 3.x

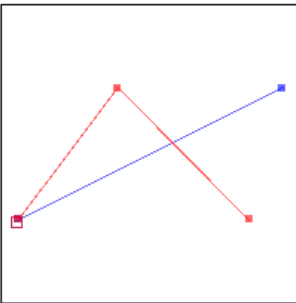
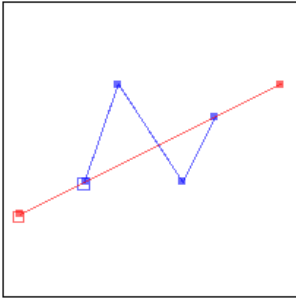
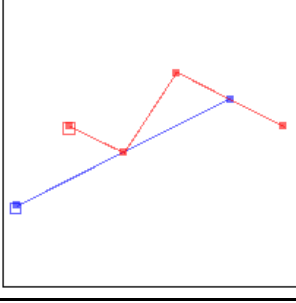
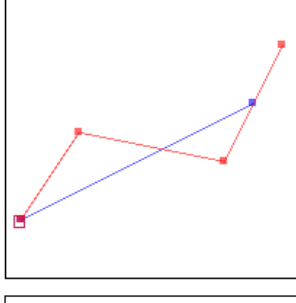
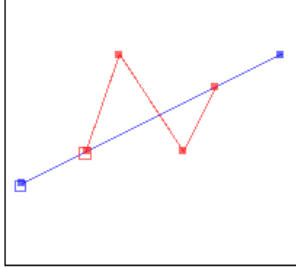
Representación	Matriz	Conmutativa	Predicados																																																		
	Intersection Matrix AB 101FF0102 BA 1F10F0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	F	F	0	<i>Ext</i>	1	0	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	1	0	1																																																		
A	<i>Bdy</i>	F	F	0																																																	
<i>Ext</i>	1	0	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	T	T																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 101F00FF2 BA 1FF00F102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	F	0	0	<i>Ext</i>	F	F	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>T</td></tr><tr><td>Contains</td><td>T</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>T</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>T</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	T	Contains	T	F	Overlaps	F	F	Covers	T	F	CoveredBy	F	T
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	1	0	1																																																		
A	<i>Bdy</i>	F	0	0																																																	
<i>Ext</i>	F	F	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	F	F																																																			
Within	F	T																																																			
Contains	T	F																																																			
Overlaps	F	F																																																			
Covers	T	F																																																			
CoveredBy	F	T																																																			
	Intersection Matrix AB 101F001F2 BA 1F100F102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	F	0	0	<i>Ext</i>	1	F	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	1	0	1																																																		
A	<i>Bdy</i>	F	0	0																																																	
<i>Ext</i>	1	F	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	T	T																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 1010FF1F2 BA 1010FF1F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	0	F	F	<i>Ext</i>	1	F	2	Sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	1	0	1																																																		
A	<i>Bdy</i>	0	F	F																																																	
<i>Ext</i>	1	F	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	T	T																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 1010FF102 BA 1010F01F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	0	F	F	<i>Ext</i>	1	0	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	1	0	1																																																		
A	<i>Bdy</i>	0	F	F																																																	
<i>Ext</i>	1	0	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	T	T																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			



Representación	Matriz	Conmutativa	Predicados																																																		
	Intersection Matrix AB 1010F01F2 BA 1010FF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	0	F	0	<i>Ext</i>	1	F	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	1	0	1																																																		
A	<i>Bdy</i>	0	F	0																																																	
<i>Ext</i>	1	F	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	T	T																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 1010F0102 BA 1010F0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	0	F	0	<i>Ext</i>	1	0	2	sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	1	0	1																																																		
A	<i>Bdy</i>	0	F	0																																																	
<i>Ext</i>	1	0	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	T	T																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 10100F1F2 BA 10100F1F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	0	0	F	<i>Ext</i>	1	F	2	sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	1	0	1																																																		
A	<i>Bdy</i>	0	0	F																																																	
<i>Ext</i>	1	F	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	T	T																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 0F1FF0102 BA 0F1FF0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	F	1	A	<i>Bdy</i>	F	F	0	<i>Ext</i>	1	0	2	sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	0	F	1																																																		
A	<i>Bdy</i>	F	F	0																																																	
<i>Ext</i>	1	0	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	T	T																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 0F1F0F1F2 BA 0F1F0F1F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	F	1	A	<i>Bdy</i>	F	0	F	<i>Ext</i>	1	F	2	sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	0	F	1																																																		
A	<i>Bdy</i>	F	0	F																																																	
<i>Ext</i>	1	F	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	T	T																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			



PostGIS, 3.x

Representación	Matriz	Conmutativa	Predicados																																	
	Intersection Matrix AB0F1F00102 BA0F1F00102 BIntBdyExt Int0F1 ABdyF00 Ext102	Sí	Binary Predicates <table><tr><td></td><td>AB</td><td>BA</td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
	AB	BA																																		
Equals	F	F																																		
Disjoint	F	F																																		
Intersects	T	T																																		
Touches	F	F																																		
Crosses	T	T																																		
Within	F	F																																		
Contains	F	F																																		
Overlaps	F	F																																		
Covers	F	F																																		
CoveredBy	F	F																																		
	Intersection Matrix AB0F10FF102 BA001FF01F2 BIntBdyExt Int0F1 ABdy0F0 Ext102	NO	Binary Predicates <table><tr><td></td><td>AB</td><td>BA</td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
	AB	BA																																		
Equals	F	F																																		
Disjoint	F	F																																		
Intersects	T	T																																		
Touches	F	F																																		
Crosses	T	T																																		
Within	F	F																																		
Contains	F	F																																		
Overlaps	F	F																																		
Covers	F	F																																		
CoveredBy	F	F																																		
	Intersection Matrix AB0F10F0102 BA001FF0102 BIntBdyExt Int0F1 ABdy0F0 Ext102	NO	Binary Predicates <table><tr><td></td><td>AB</td><td>BA</td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
	AB	BA																																		
Equals	F	F																																		
Disjoint	F	F																																		
Intersects	T	T																																		
Touches	F	F																																		
Crosses	T	T																																		
Within	F	F																																		
Contains	F	F																																		
Overlaps	F	F																																		
Covers	F	F																																		
CoveredBy	F	F																																		
	Intersection Matrix AB0F100F102 BA001F001F2 BIntBdyExt Int0F1 ABdy00F Ext102	NO	Binary Predicates <table><tr><td></td><td>AB</td><td>BA</td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
	AB	BA																																		
Equals	F	F																																		
Disjoint	F	F																																		
Intersects	T	T																																		
Touches	F	F																																		
Crosses	T	T																																		
Within	F	F																																		
Contains	F	F																																		
Overlaps	F	F																																		
Covers	F	F																																		
CoveredBy	F	F																																		
	Intersection Matrix AB001FF01F2 BA0F10FF102 BIntBdyExt Int001 ABdyF00 Ext1F2	NO	Binary Predicates <table><tr><td></td><td>AB</td><td>BA</td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
	AB	BA																																		
Equals	F	F																																		
Disjoint	F	F																																		
Intersects	T	T																																		
Touches	F	F																																		
Crosses	T	T																																		
Within	F	F																																		
Contains	F	F																																		
Overlaps	F	F																																		
Covers	F	F																																		
CoveredBy	F	F																																		



PostGIS, 3.x

Representación	Matriz	Conmutativa	Predicados																																																		
	Intersection Matrix AB 001FF0102 BA 0F10F0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Bdy</i>	F	F	0	<i>Ext</i>	1	0	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	0	0	1																																																		
A	<i>Bdy</i>	F	F	0																																																	
<i>Ext</i>	1	0	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	T	T																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 001F001F2 BA 0F100F102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Bdy</i>	F	0	0	<i>Ext</i>	1	F	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	0	0	1																																																		
A	<i>Bdy</i>	F	0	0																																																	
<i>Ext</i>	1	F	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	T	T																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 0010FF1F2 BA 0010FF1F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Bdy</i>	0	F	F	<i>Ext</i>	1	F	2	sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	0	0	1																																																		
A	<i>Bdy</i>	0	F	F																																																	
<i>Ext</i>	1	F	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	T	T																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 0010FF102 BA 0010F01F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Bdy</i>	0	F	F	<i>Ext</i>	1	0	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	0	0	1																																																		
A	<i>Bdy</i>	0	F	F																																																	
<i>Ext</i>	1	0	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	T	T																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 0010F01F2 BA 0010FF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Bdy</i>	0	F	0	<i>Ext</i>	1	F	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	0	0	1																																																		
A	<i>Bdy</i>	0	F	0																																																	
<i>Ext</i>	1	F	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	T	T																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			



PostGIS, 3.x

Representación	Matriz	Conmutativa	Predicados																																																		
	Intersection Matrix AB 0010F0102 BA 0010F0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Bdy</i>	0	F	0	<i>Ext</i>	1	0	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	0	0	1																																																		
A	<i>Bdy</i>	0	F	0																																																	
<i>Ext</i>	1	0	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	T	T																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 0010F0102 BA 0010F0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Bdy</i>	0	F	0	<i>Ext</i>	1	0	2	Sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	0	0	1																																																		
A	<i>Bdy</i>	0	F	0																																																	
<i>Ext</i>	1	0	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	T	T																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 00100F1F2 BA 00100F1F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Bdy</i>	0	0	F	<i>Ext</i>	1	F	2	Sí	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	0	0	1																																																		
A	<i>Bdy</i>	0	0	F																																																	
<i>Ext</i>	1	F	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	T	T																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			



PostGIS, 3.x

Polígono(S)/Línea: 31 relaciones

Representación	Matriz	Conmutativa	Predicados																																																	
	Intersection Matrix AB FF1FF0212 BA FF2FF1102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	1	A	<i>Bdy</i>	F	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>F</td><td>F</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	F	F	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	F	F	1																																																	
A	<i>Bdy</i>	F	F																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	F	F																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 1FF0FF212 BA 102FF1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Bdy</i>	0	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	F	Contains	F	T	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	F	F																																																	
A	<i>Bdy</i>	0	F																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	T	F																																																		
Contains	F	T																																																		
Overlaps	F	F																																																		
Covers	F	T																																																		
CoveredBy	T	F																																																		
	Intersection Matrix AB F01FF0212 BA FF20F1102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	F	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	F	0	1																																																	
A	<i>Bdy</i>	F	F																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	T	T																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 101FF0212 BA 1F20F1102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	F	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	0	1																																																	
A	<i>Bdy</i>	F	F																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	T	T																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 1010F0212 BA 1020F1102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	0	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	0	1																																																	
A	<i>Bdy</i>	0	F																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	T	T																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		



PostGIS, 3.x

Representación	Matriz	Conmutativa	Predicados																																																	
	Intersection Matrix AB 10F0FF212 BA 1020F1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	F	A	<i>Bdy</i>	0	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	F	Contains	F	T	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	0	F																																																	
A	<i>Bdy</i>	0	F																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	T	F																																																		
Contains	F	T																																																		
Overlaps	F	F																																																		
Covers	F	T																																																		
CoveredBy	T	F																																																		
	Intersection Matrix AB 1010FF212 BA 1020F11F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	0	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	0	1																																																	
A	<i>Bdy</i>	0	F																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	T	T																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB F11FF0212 BA FF21F1102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	1	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	F	1	1																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	T	T																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 111FF0212 BA 1F21F1102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	1	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	1	1																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	T	T																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 11F0FF212 BA 1021F1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	1	F	A	<i>Bdy</i>	0	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	F	Contains	F	T	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	1	F																																																	
A	<i>Bdy</i>	0	F																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	T	F																																																		
Contains	F	T																																																		
Overlaps	F	F																																																		
Covers	F	T																																																		
CoveredBy	T	F																																																		



PostGIS, 3.x

Representación	Matriz	Conmutativa	Predicados																																																	
	Intersection Matrix AB 1110FF212 BA 1021F11F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	1	1	A	<i>Bdy</i>	0	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	1	1																																																	
A	<i>Bdy</i>	0	F																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	T	T																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 1110F0212 BA 1021F1102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	1	1	A	<i>Bdy</i>	0	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	1	1																																																	
A	<i>Bdy</i>	0	F																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	T	T																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB FF1F0F212 BA FF2F011F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	F	F	1																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	T	T																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB FF1F00212 BA FF2F01102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	F	F	1																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	T	T																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 1FFF0F212 BA 1F2F01FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	F	Contains	F	T	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	F	F																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	T	F																																																		
Contains	F	T																																																		
Overlaps	F	F																																																		
Covers	F	T																																																		
CoveredBy	T	F																																																		



PostGIS, 3.x

Representación	Matriz	Conmutativa	Predicados																																																	
	Intersection Matrix AB 1FF00F212 BA 102F01FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Bdy</i>	0	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	F	Contains	F	T	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	F	F																																																	
A	<i>Bdy</i>	0	0																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	T	F																																																		
Contains	F	T																																																		
Overlaps	F	F																																																		
Covers	F	T																																																		
CoveredBy	T	F																																																		
	Intersection Matrix AB F01F0F212 BA FF20011F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	F	0	1																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	T	T																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB F01F00212 BA FF2001102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	F	0	1																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	T	T																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 10FF0F212 BA 1F2001FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	F	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	F	Contains	F	T	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	0	F																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	T	F																																																		
Contains	F	T																																																		
Overlaps	F	F																																																		
Covers	F	T																																																		
CoveredBy	T	F																																																		
	Intersection Matrix AB 101F0F212 BA 1F20011F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	0	1																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	T	T																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		



PostGIS, 3.x

Representación	Matriz	Conmutativa	Predicados																																																	
	Intersection Matrix AB 101F00212 BA 1F2001102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td>AB</td><td>BA</td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	0	1																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	2	1	2																																																	
	AB	BA																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	T	T																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 10F00F212 BA 102001FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	F	A	<i>Bdy</i>	0	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td>AB</td><td>BA</td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	F	Contains	F	T	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	0	F																																																	
A	<i>Bdy</i>	0	0																																																	
<i>Ext</i>	2	1	2																																																	
	AB	BA																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	T	F																																																		
Contains	F	T																																																		
Overlaps	F	F																																																		
Covers	F	T																																																		
CoveredBy	T	F																																																		
	Intersection Matrix AB 10100F212 BA 1020011F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	0	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td>AB</td><td>BA</td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	1	0	1																																																	
A	<i>Bdy</i>	0	0																																																	
<i>Ext</i>	2	1	2																																																	
	AB	BA																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	T	T																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB F1FF0F212 BA FF2101FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>1</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	1	F	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td>AB</td><td>BA</td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	F	1	F																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	2	1	2																																																	
	AB	BA																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	T	T																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	T																																																		
CoveredBy	T	F																																																		
	Intersection Matrix AB F11F0F212 BA FF21011F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	1	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td>AB</td><td>BA</td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	F	1	1																																																	
A	<i>Bdy</i>	F	0																																																	
<i>Ext</i>	2	1	2																																																	
	AB	BA																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	T	T																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	F	F																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		



PostGIS, 3.x

Representación	Matriz	Conmutativa	Predicados																																																		
	Intersection Matrix AB F11F00212 BA FF2101102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	1	1	A	<i>Bdy</i>	F	0	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	F	1	1																																																		
A	<i>Bdy</i>	F	0	0																																																	
<i>Ext</i>	2	1	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	T	T																																																			
Crosses	F	F																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 11FF0F212 BA 1F2101FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	1	F	A	<i>Bdy</i>	F	0	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	F	Contains	F	T	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	1	1	F																																																		
A	<i>Bdy</i>	F	0	F																																																	
<i>Ext</i>	2	1	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	F	F																																																			
Within	T	F																																																			
Contains	F	T																																																			
Overlaps	F	F																																																			
Covers	F	T																																																			
CoveredBy	T	F																																																			
	Intersection Matrix AB 111F0F212 BA 1F21011F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	1	1	A	<i>Bdy</i>	F	0	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	1	1	1																																																		
A	<i>Bdy</i>	F	0	F																																																	
<i>Ext</i>	2	1	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	T	T																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 111F00212 BA 1F2101102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	1	1	A	<i>Bdy</i>	F	0	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	1	1	1																																																		
A	<i>Bdy</i>	F	0	0																																																	
<i>Ext</i>	2	1	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	T	T																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			
	Intersection Matrix AB 11F00F212 BA 102101FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	1	F	A	<i>Bdy</i>	0	0	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	F	Contains	F	T	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	1	1	F																																																		
A	<i>Bdy</i>	0	0	F																																																	
<i>Ext</i>	2	1	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	F	F																																																			
Within	T	F																																																			
Contains	F	T																																																			
Overlaps	F	F																																																			
Covers	F	T																																																			
CoveredBy	T	F																																																			
	Intersection Matrix AB 11100F212 BA 1021011F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	1	1	A	<i>Bdy</i>	0	0	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>T</td><td>T</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	T	T	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																		
<i>Int</i>	1	1	1																																																		
A	<i>Bdy</i>	0	0	F																																																	
<i>Ext</i>	2	1	2																																																		
	<i>AB</i>	<i>BA</i>																																																			
Equals	F	F																																																			
Disjoint	F	F																																																			
Intersects	T	T																																																			
Touches	F	F																																																			
Crosses	T	T																																																			
Within	F	F																																																			
Contains	F	F																																																			
Overlaps	F	F																																																			
Covers	F	F																																																			
CoveredBy	F	F																																																			



Polígono(S)/Polígono(S): 12 relaciones

Representación	Matriz	Conmutativa	Predicados																																	
	Intersection Matrix AB 2FFF1FFF2 BA 2FFF1FFF2 B Int Bdy Ext Int 2 F F A Bdy F 1 F Ext F F 2	SÍ	Binary Predicates <table><thead><tr><th></th><th>AB</th><th>BA</th></tr></thead><tbody><tr><td>Equals</td><td>T</td><td>T</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>T</td></tr><tr><td>Contains</td><td>T</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>T</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>T</td></tr></tbody></table>		AB	BA	Equals	T	T	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	T	Contains	T	T	Overlaps	F	F	Covers	T	T	CoveredBy	T	T
	AB	BA																																		
Equals	T	T																																		
Disjoint	F	F																																		
Intersects	T	T																																		
Touches	F	F																																		
Crosses	F	F																																		
Within	T	T																																		
Contains	T	T																																		
Overlaps	F	F																																		
Covers	T	T																																		
CoveredBy	T	T																																		
	Intersection Matrix AB FF2FF1212 BA FF2FF1212 B Int Bdy Ext Int F F 2 A Bdy F F 1 Ext 2 1 2	SÍ	Binary Predicates <table><thead><tr><th></th><th>AB</th><th>BA</th></tr></thead><tbody><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>T</td><td>T</td></tr><tr><td>Intersects</td><td>F</td><td>F</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></tbody></table>		AB	BA	Equals	F	F	Disjoint	T	T	Intersects	F	F	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
	AB	BA																																		
Equals	F	F																																		
Disjoint	T	T																																		
Intersects	F	F																																		
Touches	F	F																																		
Crosses	F	F																																		
Within	F	F																																		
Contains	F	F																																		
Overlaps	F	F																																		
Covers	F	F																																		
CoveredBy	F	F																																		
	Intersection Matrix AB FF2F01212 BA FF2F01212 B Int Bdy Ext Int F F 2 A Bdy F 0 1 Ext 2 1 2	SÍ	Binary Predicates <table><thead><tr><th></th><th>AB</th><th>BA</th></tr></thead><tbody><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></tbody></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
	AB	BA																																		
Equals	F	F																																		
Disjoint	F	F																																		
Intersects	T	T																																		
Touches	T	T																																		
Crosses	F	F																																		
Within	F	F																																		
Contains	F	F																																		
Overlaps	F	F																																		
Covers	F	F																																		
CoveredBy	F	F																																		
	Intersection Matrix AB FF2F11212 BA FF2F11212 B Int Bdy Ext Int F F 2 A Bdy F 1 1 Ext 2 1 2	SÍ	Binary Predicates <table><thead><tr><th></th><th>AB</th><th>BA</th></tr></thead><tbody><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>T</td><td>T</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></tbody></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	T	T	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	F	F	Covers	F	F	CoveredBy	F	F
	AB	BA																																		
Equals	F	F																																		
Disjoint	F	F																																		
Intersects	T	T																																		
Touches	T	T																																		
Crosses	F	F																																		
Within	F	F																																		
Contains	F	F																																		
Overlaps	F	F																																		
Covers	F	F																																		
CoveredBy	F	F																																		
	Intersection Matrix AB 212FF1FF2 BA 2FF1FF212 B Int Bdy Ext Int 2 1 2 A Bdy F F 1 Ext F F 2	NO	Binary Predicates <table><thead><tr><th></th><th>AB</th><th>BA</th></tr></thead><tbody><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>T</td></tr><tr><td>Contains</td><td>T</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>T</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>T</td></tr></tbody></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	T	Contains	T	F	Overlaps	F	F	Covers	T	F	CoveredBy	F	T
	AB	BA																																		
Equals	F	F																																		
Disjoint	F	F																																		
Intersects	T	T																																		
Touches	F	F																																		
Crosses	F	F																																		
Within	F	T																																		
Contains	T	F																																		
Overlaps	F	F																																		
Covers	T	F																																		
CoveredBy	F	T																																		
	Intersection Matrix AB 212F01FF2 BA 2FF10F212 B Int Bdy Ext Int 2 1 2 A Bdy F 0 1 Ext F F 2	NO	Binary Predicates <table><thead><tr><th></th><th>AB</th><th>BA</th></tr></thead><tbody><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>T</td></tr><tr><td>Contains</td><td>T</td><td>F</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>T</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>T</td></tr></tbody></table>		AB	BA	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	T	Contains	T	F	Overlaps	F	F	Covers	T	F	CoveredBy	F	T
	AB	BA																																		
Equals	F	F																																		
Disjoint	F	F																																		
Intersects	T	T																																		
Touches	F	F																																		
Crosses	F	F																																		
Within	F	T																																		
Contains	T	F																																		
Overlaps	F	F																																		
Covers	T	F																																		
CoveredBy	F	T																																		



PostGIS, 3.x

Representación	Matriz	Conmutativa	Predicados																																																	
	Intersection Matrix AB 212111212 BA 212111212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>1</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>1</td><td>1</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	2	1	2	A	<i>Bdy</i>	1	1	<i>Ext</i>	2	1	2	SÍ	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	2	1	2																																																	
A	<i>Bdy</i>	1	1																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	T	T																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 212101212 BA 212101212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>1</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>1</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	2	1	2	A	<i>Bdy</i>	1	0	<i>Ext</i>	2	1	2	SÍ	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>F</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>F</td></tr><tr><td>Overlaps</td><td>T</td><td>T</td></tr><tr><td>Covers</td><td>F</td><td>F</td></tr><tr><td>CoveredBy</td><td>F</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	F	F	Contains	F	F	Overlaps	T	T	Covers	F	F	CoveredBy	F	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	2	1	2																																																	
A	<i>Bdy</i>	1	0																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	F	F																																																		
Contains	F	F																																																		
Overlaps	T	T																																																		
Covers	F	F																																																		
CoveredBy	F	F																																																		
	Intersection Matrix AB 2FF1FF212 BA 212FF1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>1</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	2	F	F	A	<i>Bdy</i>	1	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	F	Contains	F	T	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	2	F	F																																																	
A	<i>Bdy</i>	1	F																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	T	F																																																		
Contains	F	T																																																		
Overlaps	F	F																																																		
Covers	F	T																																																		
CoveredBy	T	F																																																		
	Intersection Matrix AB 2FF11F212 BA 212F11FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>1</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	2	F	F	A	<i>Bdy</i>	1	F	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	F	Contains	F	T	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	2	F	F																																																	
A	<i>Bdy</i>	1	F																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	T	F																																																		
Contains	F	T																																																		
Overlaps	F	F																																																		
Covers	F	T																																																		
CoveredBy	T	F																																																		
	Intersection Matrix AB 2FF10F212 BA 212F01FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>1</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	2	F	F	A	<i>Bdy</i>	1	0	<i>Ext</i>	2	1	2	NO	Binary Predicates <table><tr><td></td><td><i>AB</i></td><td><i>BA</i></td></tr><tr><td>Equals</td><td>F</td><td>F</td></tr><tr><td>Disjoint</td><td>F</td><td>F</td></tr><tr><td>Intersects</td><td>T</td><td>T</td></tr><tr><td>Touches</td><td>F</td><td>F</td></tr><tr><td>Crosses</td><td>F</td><td>F</td></tr><tr><td>Within</td><td>T</td><td>F</td></tr><tr><td>Contains</td><td>F</td><td>T</td></tr><tr><td>Overlaps</td><td>F</td><td>F</td></tr><tr><td>Covers</td><td>F</td><td>T</td></tr><tr><td>CoveredBy</td><td>T</td><td>F</td></tr></table>		<i>AB</i>	<i>BA</i>	Equals	F	F	Disjoint	F	F	Intersects	T	T	Touches	F	F	Crosses	F	F	Within	T	F	Contains	F	T	Overlaps	F	F	Covers	F	T	CoveredBy	T	F
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																																																	
<i>Int</i>	2	F	F																																																	
A	<i>Bdy</i>	1	0																																																	
<i>Ext</i>	2	1	2																																																	
	<i>AB</i>	<i>BA</i>																																																		
Equals	F	F																																																		
Disjoint	F	F																																																		
Intersects	T	T																																																		
Touches	F	F																																																		
Crosses	F	F																																																		
Within	T	F																																																		
Contains	F	T																																																		
Overlaps	F	F																																																		
Covers	F	T																																																		
CoveredBy	T	F																																																		



Apéndice A-2: Relaciones topológicas clasificadas por predicado

Servirá para ilustrar de manera concreta los efectos de cada predicado.

PATRÓN DE 9 CARACTERES PARA MATRIZ DE-9IM

Las variantes de la matriz DE-9IM se pueden representar por un patrón de 9 caracteres de texto mediante los símbolos F, 0, 1, 2, por ejemplo 'FF1FF0102'. Un tipo específico de relación espacial puede expresarse al parear la matriz de intersección con el patrón escogido. Los patrones pueden tener además los símbolos:

Símbolo	Significado
T	"la intersección es cierta " (NOT-NULL)
*	"cualquier valor"

Por ejemplo, la relación "**Intersección-de-interiores**" tendrá el patrón '**T*******' el cual no está evaluado en ninguno de los predicados nombrados anteriormente.

Repasando:

Tabla A-1: Matriz de intersección 9-IM

	Interior	Límite/Contorno/Boundary	Exterior
Punto (Dim=0) 	Dim=0 	Vacío / No existe	
Poli-línea (Dim=1) 	Dim=1 	Dim=0  	
Polígono/área (Dim=2) 	Dim=2 	Dim=1 	

Referencia: [Predicados OGC: 2.2: Conceptos de interior, límite y exterior de objetos](#), pág. 6.
Ministerio de Ecología, desarrollo sustentable, transportación y vivienda, Gobierno de Francia.



ST_EQUALS

Equivalente para usar en ST_Relate: [T*F***FF*].

Citado de https://postgis.net/docs/ST_Equals.html

Devuelve TRUE cuando las geometrías evaluadas son “topológicamente iguales”. Use esto como una mejor respuesta que decir que son iguales (=). Igualdad topológica significa que las geometrías tienen:

- La misma dimensión
- Los puntos ocupan el mismo espacio

Esto quiere decir que el orden de los vértices puede ser diferente entre geometrías topológicamente iguales.

La siguiente relación es cierta: **ST_Equals(a,b) <-> ST_Within(a,b) ^ ST_Within(b,a).**

Relación	Diagrama	Matriz
P/P		Intersection Matrix AB 0FFFFFFF2 BA 0FFFFFFF2 B <i>Int Bdy Ext</i> <i>Int</i> 0 F F A <i>Bdy Ext</i> <i>Bdy</i> F F F <i>Ext</i> F F 2
L/L		Intersection Matrix AB 1FFF0FFF2 BA 1FFF0FFF2 B <i>Int Bdy Ext</i> <i>Int</i> 1 F F A <i>Bdy Ext</i> <i>Bdy</i> F 0 F <i>Ext</i> F F 2
S/S		Intersection Matrix AB 2FFF1FFF2 BA 2FFF1FFF2 B <i>Int Bdy Ext</i> <i>Int</i> 2 F F A <i>Bdy Ext</i> <i>Bdy</i> F 1 F <i>Ext</i> F F 2



ST_Disjoint

Equivalente para usar en ST_Relate: [FF*FF****].

Citado de https://postgis.net/docs/ST_Disjoint.html

Devuelve cierto si las dos geometrías evaluadas están desjuntas. Las geometrías están desjuntas si no tienen ningún punto en común. Si alguna otra relación espacial es cierta para un par de geometrías, entonces no están desjuntas. Disjoint implica que ST_Intersects devolverá falso.

Relación	Diagrama	Matriz
P/P		Intersection Matrix AB FF0FFF0F2 BA FF0FFF0F2 B Int Body Ext Int F F 0 A Body F F F Ext 0 F 2



ST_Disjoint (cont.)

Relación	Diagrama	Matriz																
S/P		Intersection Matrix AB FF2FF10F2 BA FF0FFF212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>0</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	F	F	<i>Ext</i>	0	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	F	F															
<i>Ext</i>	0	F	2															
S/L		Intersection Matrix AB FF2FF1102 BA FF1FF0212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	F	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	F	F															
<i>Ext</i>	1	0	2															
S/S		Intersection Matrix AB FF2FF1212 BA FF2FF1212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	F	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	F	F															
<i>Ext</i>	2	1	2															



ST_Touches

Citado de https://postgis.net/docs/ST_Touches.html

Devuelve TRUE si las geometrías a y b se intersectan, pero sus interiores no intersectan. De igual manera, a y b deben tener al menos un punto en común y estos puntos en común deben estar al menos en un límite/contorno. Al evaluar esta función con geometrías Punto/Punto, la relación es siempre FALSE, ya que los puntos no tienen contorno.

Equivalentes para usar en ST_Relate: [FT***], [F***T****], [F**T*****].**

Esta función utiliza los comparadores de cajas de extensión (bounding boxes) que hacen uso de cualquier índice espacial disponible para las geometrías. Si no desea usar el índice disponible, sustituya por la función _ST_Touches.

Relación	Diagrama	Matriz
P/L		Intersection Matrix AB FF0FFF102 BA FF10F0FF2 B Int Bdy Ext Int F 0 F A Bdy F F F Ext 1 0 2
P/S		Intersection Matrix AB FF0FFF212 BA FF2FF10F2 B Int Bdy Ext Int F F 0 A Bdy F F F Ext 2 1 2
L/P		Intersection Matrix AB FF10F0FF2 BA FF0FFF102 B Int Bdy Ext Int F F 1 A Bdy 0 F 0 Ext F F 2
L/L		Intersection Matrix AB FF1F0F1F2 BA FF1F0F1F2 B Int Bdy Ext Int F F 1 A Bdy F 0 F Ext 1 F 2
L/L		Intersection Matrix AB FF1F00102 BA FF1F00102 B Int Bdy Ext Int F F 1 A Bdy F 0 0 Ext 1 0 2
L/L		Intersection Matrix AB FF10FF102 BA F01FF01F2 B Int Bdy Ext Int F F 1 A Bdy 0 F F Ext 1 0 2



ST_Touches (cont.)

Relación	Diagrama	Matriz																	
L/L		Intersection Matrix AB FF10F0102 BA F01FF0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	1	A	<i>Bdy</i>	0	F	0	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	F	F	1																
A	<i>Bdy</i>	0	F	0															
<i>Ext</i>	1	0	2																
L/L		Intersection Matrix AB FF100F102 BA F01F001F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	1	A	<i>Bdy</i>	0	0	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	F	F	1																
A	<i>Bdy</i>	0	0	F															
<i>Ext</i>	1	0	2																
L/L		Intersection Matrix AB F01FF01F2 BA FF10FF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	F	F	0	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	F	0	1																
A	<i>Bdy</i>	F	F	0															
<i>Ext</i>	1	F	2																
L/L		Intersection Matrix AB F01FF0102 BA FF10F0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	F	F	0	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	F	0	1																
A	<i>Bdy</i>	F	F	0															
<i>Ext</i>	1	0	2																
L/L		Intersection Matrix AB F01F001F2 BA FF100F102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	F	0	0	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	F	0	1																
A	<i>Bdy</i>	F	0	0															
<i>Ext</i>	1	F	2																
L/L		Intersection Matrix AB F010FF1F2 BA F010FF1F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	0	F	F	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	F	0	1																
A	<i>Bdy</i>	0	F	F															
<i>Ext</i>	1	F	2																
L/L		Intersection Matrix AB F010FF102 BA F010F01F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	0	F	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	F	0	1																
A	<i>Bdy</i>	0	F	F															
<i>Ext</i>	1	0	2																



ST_Touches (cont.)

Relación	Diagrama	Matriz																	
L/L		Intersection Matrix AB F010F01F2 BA F010FF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	0	F	0	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	F	0	1																
A	<i>Bdy</i>	0	F	0															
<i>Ext</i>	1	F	2																
L/L		Intersection Matrix AB F010F01F2 BA F010FF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	0	F	0	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	F	0	1																
A	<i>Bdy</i>	0	F	0															
<i>Ext</i>	1	F	2																
L/L		Intersection Matrix AB F010F0102 BA F010F0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	0	F	0	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	F	0	1																
A	<i>Bdy</i>	0	F	0															
<i>Ext</i>	1	0	2																
L/L		Intersection Matrix AB F0100F1F2 BA F0100F1F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	0	0	F	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	F	0	1																
A	<i>Bdy</i>	0	0	F															
<i>Ext</i>	1	F	2																
L/S		Intersection Matrix AB F01FF0212 BA FF20F1102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	F	F	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	F	0	1																
A	<i>Bdy</i>	F	F	0															
<i>Ext</i>	2	1	2																
L/S		Intersection Matrix AB F11FF0212 BA FF21F1102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	1	1	A	<i>Bdy</i>	F	F	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	F	1	1																
A	<i>Bdy</i>	F	F	0															
<i>Ext</i>	2	1	2																
L/S		Intersection Matrix AB FF1F0F212 BA FF2F011F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	1	A	<i>Bdy</i>	F	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	F	F	1																
A	<i>Bdy</i>	F	0	F															
<i>Ext</i>	2	1	2																



ST_Touches (cont.)

Relación	Diagrama	Matriz																
L/S		Intersection Matrix AB FF1F00212 BA FF2F01102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	1															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB F01F0F212 BA FF20011F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	0	1															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB F01F00212 BA FF2001102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	0	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	0	1															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB F1FF0F212 BA FF2101FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>1</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	1	F	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	1	F															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB F11F0F212 BA FF21011F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	1	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	1	1															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB F11F00212 BA FF2101102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	1	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	1	1															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	2	1	2															
S/P		Intersection Matrix AB FF20F1FF2 BA F0FFFF212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	0	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	0	F															
<i>Ext</i>	F	F	2															



ST_Touches (cont.)

Relación	Diagrama	Matriz																
S/L		Intersection Matrix AB FF20F1102 BA F01FF0212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	0	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	0	F															
<i>Ext</i>	1	0	2															
S/L		Intersection Matrix AB FF21F1102 BA F11FF0212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>1</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	1	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	1	F															
<i>Ext</i>	1	0	2															
S/L		Intersection Matrix AB FF2F011F2 BA FF1F0F212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	F	0	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	1	F	2															
S/L		Intersection Matrix AB FF2F01102 BA FF1F00212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	F	0	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	1	0	2															
S/L		Intersection Matrix AB FF20011F2 BA F01F0F212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	0	0	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	0	0															
<i>Ext</i>	1	F	2															
S/L		Intersection Matrix AB FF2001102 BA F01F00212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	0	0	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	0	0															
<i>Ext</i>	1	0	2															
S/L		Intersection Matrix AB FF2101FF2 BA F1FF0F212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>1</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	1	0	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	1	0															
<i>Ext</i>	F	F	2															



ST_Touches (cont.)

Relación	Diagrama	Matriz																
S/L		Intersection Matrix AB FF21011F2 BA F11F0F212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>1</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	1	0	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	1	0															
<i>Ext</i>	1	F	2															
S/L		Intersection Matrix AB FF2101102 BA F11F00212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>1</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	1	0	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	1	0															
<i>Ext</i>	1	0	2															
S/S		Intersection Matrix AB FF2F01212 BA FF2F01212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	2	1	2															
S/S		Intersection Matrix AB FF2F11212 BA FF2F11212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>1</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	F	1	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	F	1															
<i>Ext</i>	2	1	2															



ST_CROSSES

Citado de https://postgis.net/docs/ST_Crosses.html

Compara dos objetos geométricos y devuelve cierto si su intersección “cruza espacialmente”; esto es:

- que las geometrías tienen uno o más pero no todos sus puntos interiores en común.
- La intersección de los interiores debe ser no-vacío y su dimensión debe ser menor que la dimensión máxima entre las dos geometrías,
- Además, la intersección de las dos geometrías no puede ser igual que cualquiera de las geometrías. Dicho de otro modo, devolverá falso.

La relación “crosses” es simétrica e irreflexiva.

Las geometrías cruzan si la matriz de intersección DE-9IM tiene estos pareos en ST_Relate:

- T*T***** al evaluar P/L, P/S y L/S
- T*****T** al evaluar L/P, S/P y S/L
- 0***** al evaluar L/L

El resultado será falso al evaluar geometrías P/P y S/S.

Esta función utiliza los comparadores de cajas de extensión (bounding boxes) que hacen uso de cualquier índice espacial disponible para las geometrías. Si no desea usar el índice disponible, sustituya por la función `_ST_Crosses`.

Relación	Diagrama	Matriz
L/L		Intersection Matrix AB 0F1FF0102 BA 0F1FF0102 B Int Bdy Ext Int 0 F 1 A Bdy F F 0 Ext 1 0 2
L/L		Intersection Matrix AB 0F1F0F1F2 BA 0F1F0F1F2 B Int Bdy Ext Int 0 F 1 A Bdy F 0 F Ext 1 F 2
L/L		Intersection Matrix AB 0F1F00102 BA 0F1F00102 B Int Bdy Ext Int 0 F 1 A Bdy F 0 0 Ext 1 0 2
L/L		Intersection Matrix AB 0F10FF102 BA 001FF01F2 B Int Bdy Ext Int 0 F 1 A Bdy 0 F F Ext 1 0 2



ST_Crosses (cont.)

Relación	Diagrama	Matriz																	
L/L		Intersection Matrix AB 0F10F0102 BA 001FF0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	F	1	A	<i>Bdy</i>	0	F	0	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	0	F	1																
A	<i>Bdy</i>	0	F	0															
<i>Ext</i>	1	0	2																
L/L		Intersection Matrix AB 0F10F0102 BA 001FF0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	F	1	A	<i>Bdy</i>	0	F	0	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	0	F	1																
A	<i>Bdy</i>	0	F	0															
<i>Ext</i>	1	0	2																
L/L		Intersection Matrix AB 0F100F102 BA 001F001F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	F	1	A	<i>Bdy</i>	0	0	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	0	F	1																
A	<i>Bdy</i>	0	0	F															
<i>Ext</i>	1	0	2																
L/L		Intersection Matrix AB 001FF01F2 BA 0F10FF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Bdy</i>	F	F	0	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	0	0	1																
A	<i>Bdy</i>	F	F	0															
<i>Ext</i>	1	F	2																
L/L		Intersection Matrix AB 001FF0102 BA 0F10F0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Bdy</i>	F	F	0	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	0	0	1																
A	<i>Bdy</i>	F	F	0															
<i>Ext</i>	1	0	2																
L/L		Intersection Matrix AB 001F001F2 BA 0F100F102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Bdy</i>	F	0	0	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	0	0	1																
A	<i>Bdy</i>	F	0	0															
<i>Ext</i>	1	F	2																
L/L		Intersection Matrix AB 0010FF1F2 BA 0010FF1F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Bdy</i>	0	F	F	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																
<i>Int</i>	0	0	1																
A	<i>Bdy</i>	0	F	F															
<i>Ext</i>	1	F	2																



ST_Crosses (cont.)

Relación	Diagrama	Matriz																
L/L		Intersection Matrix AB 0010FF102 BA 0010F01F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Body</i>	0	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	0	0	1															
A	<i>Body</i>	0	F															
<i>Ext</i>	1	0	2															
L/L		Intersection Matrix AB 0010F01F2 BA 0010FF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Body</i>	0	F	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	0	0	1															
A	<i>Body</i>	0	F															
<i>Ext</i>	1	F	2															
L/L		Intersection Matrix AB 0010F0102 BA 0010F0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Body</i>	0	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	0	0	1															
A	<i>Body</i>	0	F															
<i>Ext</i>	1	0	2															
L/L		Intersection Matrix AB 00100F1F2 BA 00100F1F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	0	0	1	A	<i>Body</i>	0	F	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	0	0	1															
A	<i>Body</i>	0	F															
<i>Ext</i>	1	F	2															
L/S		Intersection Matrix AB 101FF0212 BA 1F20F1102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Body</i>	F	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	1															
A	<i>Body</i>	F	F															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 1010F0212 BA 1020F1102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Body</i>	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	1															
A	<i>Body</i>	0	F															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 1010FF212 BA 1020F11F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Body</i>	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	1															
A	<i>Body</i>	0	F															
<i>Ext</i>	2	1	2															



ST_Crosses (cont.)

Relación	Diagrama	Matriz																		
L/S		Intersection Matrix AB 111FF0212 BA 1F21F1102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td><td></td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	1	1	A	<i>Bdy</i>	F	F	0	<i>Ext</i>	2	1	2	
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																	
<i>Int</i>	1	1	1																	
A	<i>Bdy</i>	F	F	0																
<i>Ext</i>	2	1	2																	
L/S		Intersection Matrix AB 1110FF212 BA 1021F11F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td><td></td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	1	1	A	<i>Bdy</i>	0	F	F	<i>Ext</i>	2	1	2	
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																	
<i>Int</i>	1	1	1																	
A	<i>Bdy</i>	0	F	F																
<i>Ext</i>	2	1	2																	
L/L		Intersection Matrix AB 1110F0212 BA 1021F1102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td><td></td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	1	1	A	<i>Bdy</i>	0	F	0	<i>Ext</i>	2	1	2	
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																	
<i>Int</i>	1	1	1																	
A	<i>Bdy</i>	0	F	0																
<i>Ext</i>	2	1	2																	
L/S		Intersection Matrix AB 101F0F212 BA 1F20011F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td><td></td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	F	0	F	<i>Ext</i>	2	1	2	
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																	
<i>Int</i>	1	0	1																	
A	<i>Bdy</i>	F	0	F																
<i>Ext</i>	2	1	2																	
L/S		Intersection Matrix AB 101F00212 BA 1F2001102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td><td></td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	F	0	0	<i>Ext</i>	2	1	2	
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																	
<i>Int</i>	1	0	1																	
A	<i>Bdy</i>	F	0	0																
<i>Ext</i>	2	1	2																	
L/S		Intersection Matrix AB 10100F212 BA 1020011F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td><td></td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	0	0	F	<i>Ext</i>	2	1	2	
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																	
<i>Int</i>	1	0	1																	
A	<i>Bdy</i>	0	0	F																
<i>Ext</i>	2	1	2																	
L/S		Intersection Matrix AB 1010FF212 BA 1020F11F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td><td></td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	0	F	F	<i>Ext</i>	2	1	2	
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>																	
<i>Int</i>	1	0	1																	
A	<i>Bdy</i>	0	F	F																
<i>Ext</i>	2	1	2																	



ST_Crosses (cont.)

Relación	Diagrama	Matriz																
L/S		Intersection Matrix AB 111F0F212 BA 1F21011F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	1	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	1	1															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 111F00212 BA 1F2101102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	1	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	1	1															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 11100F212 BA 1021011F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	1	1	A	<i>Bdy</i>	0	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	1	1															
A	<i>Bdy</i>	0	0															
<i>Ext</i>	2	1	2															
S/L		Intersection Matrix AB 1F20F1102 BA 101FF0212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	2	A	<i>Bdy</i>	0	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	F	2															
A	<i>Bdy</i>	0	F															
<i>Ext</i>	1	0	2															
S/L		Intersection Matrix AB 1020F1102 BA 1010F0212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	2	A	<i>Bdy</i>	0	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	0	2															
A	<i>Bdy</i>	0	F															
<i>Ext</i>	1	0	2															
S/L		Intersection Matrix AB 1020F11F2 BA 1010FF212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	2	A	<i>Bdy</i>	0	F	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	0	2															
A	<i>Bdy</i>	0	F															
<i>Ext</i>	1	F	2															
S/L		Intersection Matrix AB 1020F11F2 BA 1010FF212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	2	A	<i>Bdy</i>	0	F	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	0	2															
A	<i>Bdy</i>	0	F															
<i>Ext</i>	1	F	2															



PostGIS, 3.x

ST_Crosses (cont.)

Relación	Diagrama	Matriz																
S/L		Intersection Matrix AB1F21F1102 BA111FF0212 <table><tr><td>B</td><td>Int</td><td>Body</td><td>Ext</td></tr><tr><td>Int</td><td>1</td><td>F</td><td>2</td></tr><tr><td>A</td><td>Body</td><td>1</td><td>F</td></tr><tr><td>Ext</td><td>1</td><td>0</td><td>2</td></tr></table>	B	Int	Body	Ext	Int	1	F	2	A	Body	1	F	Ext	1	0	2
B	Int	Body	Ext															
Int	1	F	2															
A	Body	1	F															
Ext	1	0	2															
S/L		Intersection Matrix AB1021F11F2 BA1110FF212 <table><tr><td>B</td><td>Int</td><td>Body</td><td>Ext</td></tr><tr><td>Int</td><td>1</td><td>0</td><td>2</td></tr><tr><td>A</td><td>Body</td><td>1</td><td>F</td></tr><tr><td>Ext</td><td>1</td><td>F</td><td>2</td></tr></table>	B	Int	Body	Ext	Int	1	0	2	A	Body	1	F	Ext	1	F	2
B	Int	Body	Ext															
Int	1	0	2															
A	Body	1	F															
Ext	1	F	2															
S/L		Intersection Matrix AB1021F1102 BA1110F0212 <table><tr><td>B</td><td>Int</td><td>Body</td><td>Ext</td></tr><tr><td>Int</td><td>1</td><td>0</td><td>2</td></tr><tr><td>A</td><td>Body</td><td>1</td><td>F</td></tr><tr><td>Ext</td><td>1</td><td>0</td><td>2</td></tr></table>	B	Int	Body	Ext	Int	1	0	2	A	Body	1	F	Ext	1	0	2
B	Int	Body	Ext															
Int	1	0	2															
A	Body	1	F															
Ext	1	0	2															
S/L		Intersection Matrix AB1F20011F2 BA101F0F212 <table><tr><td>B</td><td>Int</td><td>Body</td><td>Ext</td></tr><tr><td>Int</td><td>1</td><td>F</td><td>2</td></tr><tr><td>A</td><td>Body</td><td>0</td><td>0</td></tr><tr><td>Ext</td><td>1</td><td>F</td><td>2</td></tr></table>	B	Int	Body	Ext	Int	1	F	2	A	Body	0	0	Ext	1	F	2
B	Int	Body	Ext															
Int	1	F	2															
A	Body	0	0															
Ext	1	F	2															
S/L		Intersection Matrix AB1F2001102 BA101F00212 <table><tr><td>B</td><td>Int</td><td>Body</td><td>Ext</td></tr><tr><td>Int</td><td>1</td><td>F</td><td>2</td></tr><tr><td>A</td><td>Body</td><td>0</td><td>0</td></tr><tr><td>Ext</td><td>1</td><td>0</td><td>2</td></tr></table>	B	Int	Body	Ext	Int	1	F	2	A	Body	0	0	Ext	1	0	2
B	Int	Body	Ext															
Int	1	F	2															
A	Body	0	0															
Ext	1	0	2															
S/L		Intersection Matrix AB1020011F2 BA10100F212 <table><tr><td>B</td><td>Int</td><td>Body</td><td>Ext</td></tr><tr><td>Int</td><td>1</td><td>0</td><td>2</td></tr><tr><td>A</td><td>Body</td><td>0</td><td>0</td></tr><tr><td>Ext</td><td>1</td><td>F</td><td>2</td></tr></table>	B	Int	Body	Ext	Int	1	0	2	A	Body	0	0	Ext	1	F	2
B	Int	Body	Ext															
Int	1	0	2															
A	Body	0	0															
Ext	1	F	2															
S/L		Intersection Matrix AB1F21011F2 BA111F0F212 <table><tr><td>B</td><td>Int</td><td>Body</td><td>Ext</td></tr><tr><td>Int</td><td>1</td><td>F</td><td>2</td></tr><tr><td>A</td><td>Body</td><td>1</td><td>0</td></tr><tr><td>Ext</td><td>1</td><td>F</td><td>2</td></tr></table>	B	Int	Body	Ext	Int	1	F	2	A	Body	1	0	Ext	1	F	2
B	Int	Body	Ext															
Int	1	F	2															
A	Body	1	0															
Ext	1	F	2															



ST_Crosses (cont.)

Relación	Diagrama	Matriz																
S/L		Intersection Matrix AB 1F2101102 BA 111F00212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>1</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	2	A	<i>Body</i>	1	0	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	2															
A	<i>Body</i>	1	0															
<i>Ext</i>	1	0	2															
S/L		Intersection Matrix AB 1021011F2 BA 11100F212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>1</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	2	A	<i>Body</i>	1	0	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	2															
A	<i>Body</i>	1	0															
<i>Ext</i>	1	F	2															

ST_Within

Citado de https://postgis.net/docs/ST_Within.html

Devolverá TRUE si la geometría A está dentro de la geometría B. La geometría A está dentro de la geometría B si y solo si:

- Todos los puntos que componen A están dentro (ya sea en el interior o límite/contorno) de B
- Dicho de otro modo, ningún punto de A está en el exterior de B
- Además, los interiores de A y B al menos tienen un punto en común.

Para que esta función tenga sentido, las geometrías que se van a comparar deben estar en el mismo sistema de coordenadas; deben tener el mismo SRID.

La relación within es reflexiva: cada geometría está dentro de ella misma. La relación es antisimétrica: si $ST_Within(A,B) = true$ and $ST_Within(B,A) = true$, entonces las dos geometrías deben ser topológicamente iguales ($ST_Equals(A,B) = true$).

ST_Within es el reverso de $ST_Contains$, por lo tanto, $ST_Within(A,B) = ST_Contains(B,A)$.

Expresión equivalente en matriz DE-9IM para uso con $ST_Relate(geomA,geomB,'patrón')$ [T*F**F***].

Nota:

Ya que los interiores deben tener al menos un punto en común, un detalle de la definición de ST_Within es que las líneas y puntos que estén completamente en el contorno de polígonos o líneas NO están dentro de la geometría de estos polígonos. Para más detalles, refiérase a las [definiciones/sutilezas de Covers, Contains y de Within](#). El predicado [ST_CoveredBy](#) provee una relación más inclusiva.



PostGIS, 3.x

Esta función utiliza los comparadores de cajas de extensión (bounding boxes) que hacen uso de cualquier índice espacial disponible para las geometrías. Si no desea usar el índice disponible, sustituya por la función `_ST_Within`.

`ST_Within` (cont.)

Relación	Diagrama	Matriz																
P/P		Intersection Matrix AB 0FFFFFFF2 BA 0FFFFFFF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	F	F	A	<i>Bdy</i>	F	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	0	F	F															
A	<i>Bdy</i>	F	F															
<i>Ext</i>	F	F	2															
P/L		Intersection Matrix AB 0FFFFFF102 BA 0F1FF0FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	F	F	A	<i>Bdy</i>	F	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	0	F	F															
A	<i>Bdy</i>	F	F															
<i>Ext</i>	1	0	2															
P/S		Intersection Matrix AB 0FFFFFF212 BA 0F2FF1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	F	F	A	<i>Bdy</i>	F	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	0	F	F															
A	<i>Bdy</i>	F	F															
<i>Ext</i>	2	1	2															
L/L		Intersection Matrix AB 1FFF0FFF2 BA 1FFF0FFF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Bdy</i>	F	0	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	F	F															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	F	F	2															
L/L		Intersection Matrix AB 1FF0FF102 BA 101FF0FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Bdy</i>	0	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	F	F															
A	<i>Bdy</i>	0	F															
<i>Ext</i>	1	0	2															
L/L		Intersection Matrix AB 1FF00F102 BA 101F00FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Bdy</i>	0	0	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	F	F															
A	<i>Bdy</i>	0	0															
<i>Ext</i>	1	0	2															
L/S		Intersection Matrix AB 1FF0FF212 BA 102FF1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Bdy</i>	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	F	F															
A	<i>Bdy</i>	0	F															
<i>Ext</i>	2	1	2															



PostGIS, 3.x

ST_Within (cont.)

Relación	Diagrama	Matriz																
L/S		Intersection Matrix AB 10F0FF212 BA 1020F1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	F	A	<i>Body</i>	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	F															
A	<i>Body</i>	0	F															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 11F0FF212 BA 1021F1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	1	F	A	<i>Body</i>	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	1	F															
A	<i>Body</i>	0	F															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 1FFF0F212 BA 1F2F01FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Body</i>	F	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	F															
A	<i>Body</i>	F	0															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 1FF00F212 BA 102F01FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Body</i>	0	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	F															
A	<i>Body</i>	0	0															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 1FF00F212 BA 102F01FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Body</i>	0	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	F															
A	<i>Body</i>	0	0															
<i>Ext</i>	2	1	2															
L/L		Intersection Matrix AB 10FF0F212 BA 1F2001FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	F	A	<i>Body</i>	F	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	F															
A	<i>Body</i>	F	0															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 10F00F212 BA 102001FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	F	A	<i>Body</i>	0	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	F															
A	<i>Body</i>	0	0															
<i>Ext</i>	2	1	2															



PostGIS, 3.x

ST_Within (cont.)

Relación	Diagrama	Matriz																	
L/S		Intersection Matrix AB 11FF0F212 BA 1F2101FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	1	F	A	<i>Body</i>	F	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	1	1	F																
A	<i>Body</i>	F	0	F															
<i>Ext</i>	2	1	2																
L/S		Intersection Matrix AB 11F00F212 BA 102101FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	1	F	A	<i>Body</i>	0	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	1	1	F																
A	<i>Body</i>	0	0	F															
<i>Ext</i>	2	1	2																
S/S		Intersection Matrix AB 2FFF1FFF2 BA 2FFF1FFF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>1</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	2	F	F	A	<i>Body</i>	F	1	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	2	F	F																
A	<i>Body</i>	F	1	F															
<i>Ext</i>	F	F	2																
S/S		Intersection Matrix AB 2FF1FF212 BA 212FF1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	2	F	F	A	<i>Body</i>	1	F	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	2	F	F																
A	<i>Body</i>	1	F	F															
<i>Ext</i>	2	1	2																
S/S		Intersection Matrix AB 2FF11F212 BA 212F11FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>1</td><td>1</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	2	F	F	A	<i>Body</i>	1	1	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	2	F	F																
A	<i>Body</i>	1	1	F															
<i>Ext</i>	2	1	2																
S/S		Intersection Matrix AB 2FF10F212 BA 212F01FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>1</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	2	F	F	A	<i>Body</i>	1	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	2	F	F																
A	<i>Body</i>	1	0	F															
<i>Ext</i>	2	1	2																



ST_CONTAINS

Citado de https://postgis.net/docs/ST_Contains.html

Devuelve TRUE si la geometría A contiene a la geometría B. La geometría A contiene a B si y solo si:

- Todos los puntos de B están dentro (en el interior o límite de) A o igualmente, ningún punto de B está en el exterior de A)
- Y que los interiores de A y B tengan al menos un punto en común.

La relación Contains es reflexiva: cada geometría se contiene a sí misma. En contraste, en el predicado ST_ContainsProperly la geometría no está contenida propiamente en ella. Esta relación es antisimétrica: si $ST_Contains(A,B) = true$ y $ST_Contains(B,A) = true$, entonces las dos geometrías deben ser topológicamente iguales ($ST_Equals(A,B) = true$).

ST_Contains es el inverso de ST_Within. Entonces, $ST_Contains(A,B) = ST_Within(B,A)$.

Expresión equivalente en matriz DE-9IM para uso con ST_Relate(geomA,geomB,'patrón') [T*T***T**].

Nota:

Debido a que los interiores deben tener un punto en común, una de las sutilezas de esta definición es que tanto polígonos como líneas no contienen líneas y puntos completamente en su contorno. Para más detalles, refiérase a las [definiciones/sutilezas de Covers, Contains y de Within](#). El predicado [ST_Covers](#) provee una relación más inclusiva.

Esta función utiliza los comparadores de cajas de extensión (bounding boxes) que hacen uso de cualquier índice espacial disponible para las geometrías. Si no desea usar el índice disponible, sustituya por la función ST_Contains.

Relación	Diagrama	Matriz
P/P		Intersection Matrix AB 0FFFFFF2 BA 0FFFFFF2 B Int Body Ext Int 0 F F A Body F F F Ext F F 2



ST_Contains (cont.)

Relación	Diagrama	Matriz																
L/L		Intersection Matrix AB 101FF0FF2 BA 1FF0FF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	F	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	0	1															
A	<i>Bdy</i>	F	F															
<i>Ext</i>	F	F	2															
L/L		Intersection Matrix AB 101F00FF2 BA 1FF00F102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	0	1															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	F	F	2															
S/P		Intersection Matrix AB 0F2FF1FF2 BA 0FFFFFF212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	F	2	A	<i>Bdy</i>	F	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	0	F	2															
A	<i>Bdy</i>	F	F															
<i>Ext</i>	F	F	2															
S/S		Intersection Matrix AB 2FFF1FFF2 BA 2FFF1FFF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>1</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	2	F	F	A	<i>Bdy</i>	F	1	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	2	F	F															
A	<i>Bdy</i>	F	1															
<i>Ext</i>	F	F	2															
S/S		Intersection Matrix AB 212FF1FF2 BA 2FF1FF212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>1</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	2	1	2	A	<i>Bdy</i>	F	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	2	1	2															
A	<i>Bdy</i>	F	F															
<i>Ext</i>	F	F	2															
S/S		Intersection Matrix AB 212F01FF2 BA 2FF10F212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>1</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	2	1	2	A	<i>Bdy</i>	F	0	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	2	1	2															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	F	F	2															
S/S		Intersection Matrix AB 212F11FF2 BA 2FF11F212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>1</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>1</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	2	1	2	A	<i>Bdy</i>	F	1	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	2	1	2															
A	<i>Bdy</i>	F	1															
<i>Ext</i>	F	F	2															



ST_OVERLAPS

Citado de https://postgis.net/docs/ST_Overlaps.html

Devuelve TRUE si la geometría A y la B se solapan espacialmente. Dos geometrías se solapan si:

- Tienen la misma dimensión (P/P, L/L, S/S)
- Sus interiores se deben intersectar en esa dimensión.
- Cada geometría tiene al menos un punto dentro de la otra geometría, dicho de otra manera, ninguna debe cubrir (Covers) la otra.

La relación Overlaps es simétrica e irreflexiva.

Esta función incluye de manera automática el uso de cajas de extensión (bounding boxes), lo cual hará uso de cualquier índice espacial existente. Si no desea usar índices, use la función `_ST_Overlaps`.

Expresión en matrices DE-9IM para uso con `ST_Relate(geomA,geomB,'patrón')`

[T*T***T**] (P/P, S/S)

[1*T***T**] (L/L)

Relación	Diagrama	Matriz
P/P		Intersection Matrix AB 0FFFFFFF2 BA 0FFFFFFF2 B Int Bdy Ext Int 0 F F A Bdy F F F Ext F F 2
L/L		Intersection Matrix AB 1F1FF0102 BA 1F1FF0102 B Int Bdy Ext Int 1 F 1 A Bdy F F 0 Ext 1 0 2
L/L		Intersection Matrix AB 1F1F0F1F2 BA 1F1F0F1F2 B Int Bdy Ext Int 1 F 1 A Bdy F 0 F Ext 1 F 2
L/L		Intersection Matrix AB 1F1F00102 BA 1F1F00102 B Int Bdy Ext Int 1 F 1 A Bdy F 0 0 Ext 1 0 2



ST_Overlaps (cont.)

Relación	Diagrama	Matriz																
L/L		Intersection Matrix AB 1F100F102 BA 101F001F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	1	A	<i>Body</i>	0	0	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	1															
A	<i>Body</i>	0	0															
<i>Ext</i>	1	0	2															
L/L		Intersection Matrix AB 1F10F0102 BA 101FF0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	1	A	<i>Body</i>	0	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	1															
A	<i>Body</i>	0	F															
<i>Ext</i>	1	0	2															
L/L		Intersection Matrix AB 101FF01F2 BA 1F10FF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Body</i>	F	F	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	1															
A	<i>Body</i>	F	F															
<i>Ext</i>	1	F	2															
L/L		Intersection Matrix AB 101FF0102 BA 1F10F0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Body</i>	F	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	1															
A	<i>Body</i>	F	F															
<i>Ext</i>	1	0	2															
L/L		Intersection Matrix AB 101F001F2 BA 1F100F102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Body</i>	F	0	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	1															
A	<i>Body</i>	F	0															
<i>Ext</i>	1	F	2															
L/L		Intersection Matrix AB 1010FF1F2 BA 1010FF1F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Body</i>	0	F	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	1															
A	<i>Body</i>	0	F															
<i>Ext</i>	1	F	2															
L/L		Intersection Matrix AB 1010FF102 BA 1010F01F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Body</i>	0	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	1															
A	<i>Body</i>	0	F															
<i>Ext</i>	1	0	2															



ST_Overlaps (cont.)

Relación	Diagrama	Matriz																	
L/L		Intersection Matrix AB 1010F01F2 BA 1010FF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Body</i>	0	F	0	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	1	0	1																
A	<i>Body</i>	0	F	0															
<i>Ext</i>	1	F	2																
L/L		Intersection Matrix AB 1010F0102 BA 1010F0102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Body</i>	0	F	0	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	1	0	1																
A	<i>Body</i>	0	F	0															
<i>Ext</i>	1	0	2																
L/L		Intersection Matrix AB 10100F1F2 BA 10100F1F2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Body</i>	0	0	F	<i>Ext</i>	1	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	1	0	1																
A	<i>Body</i>	0	0	F															
<i>Ext</i>	1	F	2																
S/S		Intersection Matrix AB 212111212 BA 212111212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>1</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>1</td><td>1</td><td>1</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	2	1	2	A	<i>Body</i>	1	1	1	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	2	1	2																
A	<i>Body</i>	1	1	1															
<i>Ext</i>	2	1	2																
S/S		Intersection Matrix AB 212101212 BA 212101212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>1</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	2	1	2	A	<i>Body</i>	1	0	1	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	2	1	2																
A	<i>Body</i>	1	0	1															
<i>Ext</i>	2	1	2																



ST_INTERSECTS

El predicado ST_Intersects aplica a todas las relaciones, con excepción de ST_Disjoint, ya que que las geometrías tendrían al menos un punto en común, ya sea en el interior o el límite/contorno.

Ejemplos de patrones en matrices DE-9IM para uso con ST_Relate(geomA,geomB,'patrón')

InteriorA/InteriorB	[T*****]
InteriorA/ContornoB	[*T*****]
InteriorB/InteriorA	[***T*****]
InteriorB/ContornoA	[***T****]

Esta función incluye de manera automática el uso de cajas de extensión (bounding boxes), lo cual hará uso de cualquier índice espacial existente.

OPERADOR &&

Citado de https://postgis.net/docs/geometry_overlaps.html

El operador de **doble ampersand &&** ([en español literalmente: "y por sí mismo"](#)), devuelve TRUE si la caja de extensión (bounding box) 2D de la geometría A interseca con la caja de extensión 2D de la geometría B.

Este operador hace uso de índices existentes en las geometrías. Úselo para combinarlo con ST_Relate(geomA,geomB,'patrón'), ya que ST_Relate no usa índices automáticamente.

Es preferible usar relaciones con nombre, en vez de ST_Relate debido al hecho que esos utilizan el comparador de cajas de extensión o índices de geometrías, además de mejoras en desempeño que no están en ST_Relate. Solo debe usarse ST_Relate para casos especiales

ST_COVERS

Citado de https://postgis.net/docs/ST_Covers.html

** NO es un estándar del OGC **

Devuelve TRUE si cada punto en la geometría/geografía B está dentro del interior o el contorno de la geometría/geografía de A. De forma equivalente, prueba que ningún punto de B está fuera o en el exterior de A.

ST_Covers es el reverso de ST_CoveredBy, por lo tanto, **ST_Covers(A,B) = ST_CoveredBy(B,A)**.

Por lo general, la función ST_Covers debe ser usada en lugar de ST_Contains, ya que su definición no tiene la restricción "las geometrías no contienen su contorno".

Esta función incluye de manera automática el uso de cajas de extensión (bounding boxes), lo cual hará uso de cualquier índice espacial existente. Si no desea usar índices, use la función ST_Covers.



PostGIS, 3.x

Ejemplos de patrones en matrices DE-9IM para uso con ST_Relate(geomA,geomB,'patrón')

InteriorA/InteriorB	[T*****FF*]
InteriorA/ContornoB	[*T*****FF*]
InteriorB/InteriorA	[***T**FF*]
InteriorB/ContornoA	[****T*FF*]

ST_Covers (cont.)

Relación	Diagrama	Matriz																
P/P		Intersection Matrix AB 0FFFFFFF2 BA 0FFFFFFF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	F	F	A	<i>Bdy</i>	F	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	0	F	F															
A	<i>Bdy</i>	F	F															
<i>Ext</i>	F	F	2															
L/P		Intersection Matrix AB FF10F0FF2 BA F0FFFF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	1	A	<i>Bdy</i>	0	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	1															
A	<i>Bdy</i>	0	F															
<i>Ext</i>	F	F	2															
L/P		Intersection Matrix AB 0F1FF0FF2 BA 0FFFFFF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	F	1	A	<i>Bdy</i>	F	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	0	F	1															
A	<i>Bdy</i>	F	F															
<i>Ext</i>	F	F	2															
L/L		Intersection Matrix AB 1FFF0FFF2 BA 1FFF0FFF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Bdy</i>	F	0	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	F	F															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	F	F	2															
L/L		Intersection Matrix AB 101FF0FF2 BA 1FF0FF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	F	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	0	1															
A	<i>Bdy</i>	F	F															
<i>Ext</i>	F	F	2															
L/L		Intersection Matrix AB 101F00FF2 BA 1FF00F102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	1	0	1	A	<i>Bdy</i>	F	0	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	1	0	1															
A	<i>Bdy</i>	F	0															
<i>Ext</i>	F	F	2															
S/P		Intersection Matrix AB FF20F1FF2 BA F0FFFF212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Bdy</i>	0	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Bdy</i>	0	F															
<i>Ext</i>	F	F	2															



ST_Covers (cont.)

Relación	Diagrama	Matriz																
S/P		Intersection Matrix AB 0F2FF1FF2 BA 0FFFFFF12 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	0	F	2	A	<i>Body</i>	F	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	0	F	2															
A	<i>Body</i>	F	F															
<i>Ext</i>	F	F	2															
S/L		Intersection Matrix AB 102FF1FF2 BA 1FF0FF212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	2	A	<i>Body</i>	F	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	2															
A	<i>Body</i>	F	F															
<i>Ext</i>	F	F	2															
S/L		Intersection Matrix AB 1020F1FF2 BA 10F0FF212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	2	A	<i>Body</i>	0	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	2															
A	<i>Body</i>	0	F															
<i>Ext</i>	F	F	2															
S/L		Intersection Matrix AB 1021F1FF2 BA 11F0FF212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>1</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	2	A	<i>Body</i>	1	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	2															
A	<i>Body</i>	1	F															
<i>Ext</i>	F	F	2															
S/L		Intersection Matrix AB 1F2F01FF2 BA 1FFF0F212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	2	A	<i>Body</i>	F	0	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	2															
A	<i>Body</i>	F	0															
<i>Ext</i>	F	F	2															
S/L		Intersection Matrix AB 102F01FF2 BA 1FF00F212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	2	A	<i>Body</i>	F	0	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	2															
A	<i>Body</i>	F	0															
<i>Ext</i>	F	F	2															
S/L		Intersection Matrix AB 1F2001FF2 BA 10FF0F212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	2	A	<i>Body</i>	0	0	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	2															
A	<i>Body</i>	0	0															
<i>Ext</i>	F	F	2															



ST_Covers (cont.)

Relación	Diagrama	Matriz																
S/L		Intersection Matrix AB 102001FF2 BA 10F00F212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	2	A	<i>Body</i>	0	0	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	2															
A	<i>Body</i>	0	0															
<i>Ext</i>	F	F	2															
S/L		Intersection Matrix AB FF2101FF2 BA F1FF0F212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>1</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	F	F	2	A	<i>Body</i>	1	0	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	F	F	2															
A	<i>Body</i>	1	0															
<i>Ext</i>	F	F	2															
S/L		Intersection Matrix AB 1F2101FF2 BA 11FF0F212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>1</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	2	A	<i>Body</i>	1	0	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	2															
A	<i>Body</i>	1	0															
<i>Ext</i>	F	F	2															
S/S		Intersection Matrix AB 2FFF1FFF2 BA 2FFF1FFF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>1</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	2	F	F	A	<i>Body</i>	F	1	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	2	F	F															
A	<i>Body</i>	F	1															
<i>Ext</i>	F	F	2															
S/S		Intersection Matrix AB 212FF1FF2 BA 2FF1FF212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>1</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	2	1	2	A	<i>Body</i>	F	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	2	1	2															
A	<i>Body</i>	F	F															
<i>Ext</i>	F	F	2															
S/S		Intersection Matrix AB 212F01FF2 BA 2FF10F212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>1</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	2	1	2	A	<i>Body</i>	F	0	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	2	1	2															
A	<i>Body</i>	F	0															
<i>Ext</i>	F	F	2															
S/S		Intersection Matrix AB 212F11FF2 BA 2FF11F212 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>1</td><td>2</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>1</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	2	1	2	A	<i>Body</i>	F	1	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	2	1	2															
A	<i>Body</i>	F	1															
<i>Ext</i>	F	F	2															



ST_COVEREDBY

Citado de https://postgis.net/docs/ST_CoveredBy.html

** NO es un estándar del OGC **

Devuelve TRUE si cada punto en la geometría/geografía A está dentro del interior o el contorno de la geometría/geografía de B. De forma equivalente, prueba que ningún punto de A está fuera o en el exterior de B.

ST_CoveredBy es el reverso de ST_Covers, por lo tanto, **ST_CoveredBy(A,B) = ST_Covers(B,A).**

Por lo general, la función ST_CoveredBy debe ser usada en lugar de ST_Within, ya que su definición no tiene la restricción “las geometrías no contienen su contorno”.

Esta función incluye de manera automática el uso de cajas de extensión (bounding boxes), lo cual hará uso de cualquier índice espacial existente. Si no desea usar índices, use la función ST_CoveredBy.

Ejemplos de patrones en matrices DE-9IM para uso con ST_Relate(geomA,geomB,'patrón')

InteriorA/InteriorB	[T*F**F***]
InteriorA/ContornoB	[*TF**F***]
InteriorB/InteriorA	[**FT*F***]
InteriorB/ContornoA	[**F*TF***]

Relación	Diagrama	Matriz																
P/P		Intersection Matrix AB 0FFFFFFF2 BA 0FFFFFFF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	F	F	A	<i>Bdy</i>	F	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	0	F	F															
A	<i>Bdy</i>	F	F															
<i>Ext</i>	F	F	2															
P/L		Intersection Matrix AB FF10F0FF2 BA F0FFFF102 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>F</td><td>1</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	F	F	1	A	<i>Bdy</i>	0	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	F	F	1															
A	<i>Bdy</i>	0	F															
<i>Ext</i>	F	F	2															
P/L		Intersection Matrix AB 0FFFFFF102 BA 0F1FF0FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Bdy</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Bdy</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>	<i>Int</i>	0	F	F	A	<i>Bdy</i>	F	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Bdy</i>	<i>Ext</i>															
<i>Int</i>	0	F	F															
A	<i>Bdy</i>	F	F															
<i>Ext</i>	1	0	2															



ST_CoveredBy (cont.)

Relación	Diagrama	Matriz																
P/S		Intersection Matrix AB F0FFFF212 BA FF20F1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>F</td><td>0</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	F	0	F	A	<i>Body</i>	F	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	F	0	F															
A	<i>Body</i>	F	F															
<i>Ext</i>	2	1	2															
P/S		Intersection Matrix AB 0FFFFFF212 BA 0F2FF1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>0</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	0	F	F	A	<i>Body</i>	F	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	0	F	F															
A	<i>Body</i>	F	F															
<i>Ext</i>	2	1	2															
L/L		Intersection Matrix AB 1FFF0FFF2 BA 1FFF0FFF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Body</i>	F	0	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	F															
A	<i>Body</i>	F	0															
<i>Ext</i>	F	F	2															
L/L		Intersection Matrix AB 1FF0FF102 BA 101FF0FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Body</i>	0	F	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	F															
A	<i>Body</i>	0	F															
<i>Ext</i>	1	0	2															
L/L		Intersection Matrix AB 1FF00F102 BA 101F00FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>1</td><td>0</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Body</i>	0	0	<i>Ext</i>	1	0	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	F															
A	<i>Body</i>	0	0															
<i>Ext</i>	1	0	2															
L/S		Intersection Matrix AB 1FF0FF212 BA 102FF1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Body</i>	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	F															
A	<i>Body</i>	0	F															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 10F0FF212 BA 1020F1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	F	A	<i>Body</i>	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	F															
A	<i>Body</i>	0	F															
<i>Ext</i>	2	1	2															



ST_CoveredBy (cont.)

Relación	Diagrama	Matriz																
L/S		Intersection Matrix AB 11F0FF212 BA 1021F1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	1	F	A	<i>Body</i>	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	1	F															
A	<i>Body</i>	0	F															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 1FF0F0F212 BA 1F2F0F1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Body</i>	F	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	F															
A	<i>Body</i>	F	0															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 1FF00F212 BA 102F0F1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Body</i>	0	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	F															
A	<i>Body</i>	0	0															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 10FF0F212 BA 1F200F1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	F	A	<i>Body</i>	F	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	F															
A	<i>Body</i>	F	0															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 10F00F212 BA 10200F1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>0</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	F	A	<i>Body</i>	0	0	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	F															
A	<i>Body</i>	0	0															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 1FF0FF212 BA 102FF1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	F	F	A	<i>Body</i>	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	F	F															
A	<i>Body</i>	0	F															
<i>Ext</i>	2	1	2															
L/S		Intersection Matrix AB 10F0FF212 BA 1020F1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>0</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	0	F	A	<i>Body</i>	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>															
<i>Int</i>	1	0	F															
A	<i>Body</i>	0	F															
<i>Ext</i>	2	1	2															



PostGIS, 3.x

ST_CoveredBy (cont.)

Relación	Diagrama	Matriz																	
L/S		Intersection Matrix AB 11F00F212 BA 102101FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>1</td><td>1</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>0</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	1	1	F	A	<i>Body</i>	0	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	1	1	F																
A	<i>Body</i>	0	0	F															
<i>Ext</i>	2	1	2																
S/S		Intersection Matrix AB 2FFF1FFF2 BA 2FFF1FFF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>F</td><td>1</td><td>F</td></tr><tr><td><i>Ext</i></td><td>F</td><td>F</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	2	F	F	A	<i>Body</i>	F	1	F	<i>Ext</i>	F	F	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	2	F	F																
A	<i>Body</i>	F	1	F															
<i>Ext</i>	F	F	2																
S/S		Intersection Matrix AB 2FF1FF212 BA 212FF1FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>1</td><td>F</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	2	F	F	A	<i>Body</i>	1	F	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	2	F	F																
A	<i>Body</i>	1	F	F															
<i>Ext</i>	2	1	2																
S/S		Intersection Matrix AB 2FF11F212 BA 212F11FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>1</td><td>1</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	2	F	F	A	<i>Body</i>	1	1	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	2	F	F																
A	<i>Body</i>	1	1	F															
<i>Ext</i>	2	1	2																
S/S		Intersection Matrix AB 2FF10F212 BA 212F01FF2 <table><tr><td>B</td><td><i>Int</i></td><td><i>Body</i></td><td><i>Ext</i></td></tr><tr><td><i>Int</i></td><td>2</td><td>F</td><td>F</td></tr><tr><td>A</td><td><i>Body</i></td><td>1</td><td>0</td><td>F</td></tr><tr><td><i>Ext</i></td><td>2</td><td>1</td><td>2</td></tr></table>	B	<i>Int</i>	<i>Body</i>	<i>Ext</i>	<i>Int</i>	2	F	F	A	<i>Body</i>	1	0	F	<i>Ext</i>	2	1	2
B	<i>Int</i>	<i>Body</i>	<i>Ext</i>																
<i>Int</i>	2	F	F																
A	<i>Body</i>	1	0	F															
<i>Ext</i>	2	1	2																



Referencias

Interpolación areal

Flowerdew, R., & Green, M. (1994). Areal interpolation and types of data. *Spatial analysis and GIS*, 121, 145.

Predicados topológicos OGC :

PROTEAU, J. C., (2021) *Matrices de Clementini et Prédicats spatiaux de l'OGC*. Ministère de l'Écologie, du Développement durable, des Transports et du Logement.

Copia original en francés disponible en:

https://gis.pr.gov/seccioneducativa/Documents/Tutoriales/PostGIS/Predicats_OGC_V3_cle119417.pdf

Open Geospatial Consortium Inc. (2011, May 28). *OpenGIS® Implementation Standard for Geographic information—Simple feature access—Part 1: Common architecture. Version 1.2.1*. Open Geospatial Consortium Inc. http://portal.opengeospatial.org/files/?artifact_id=25355

Referencias, PostGIS:

"PostGIS Documentation", *PostGIS*, fecha de consulta 10 abril 2024, en <https://postgis.net/>.

Martínez Llario, J. C. (2020). *PostGIS 2: Análisis espacial avanzado* (2ª ed.). CreateSpace Independent.

Obe, R. O. (with Hsu, L. S.). (2011). *PostGIS in action*. Manning.

Referencia, SQL:

Fabien Celaia. (2014, March 2). *La généalogie des SGBD*. Developpez.com.

<https://fadace.developpez.com/sqbdcmp/story/>

Damien POULAIN. (2013, September 20). *Petite histoire des bases relationnelles et du langage SQL*. https://www.decideo.fr/Petite-histoire-des-bases-relationnelles-et-du-langage-SQL_a6377.html

Viescas, J. L., & Hernández, M. J. (2014). *SQL queries for mere mortals: A hands-on guide to data manipulation in SQL* (3. ed). Addison-Wesley.