

Spatial SQL I

CASA0025: Building Spatial Applications
with Big Data



Ollie Ballinger

Outline

1. SQL Recap
2. Geometries
3. Spatial Relationships

What is a database?

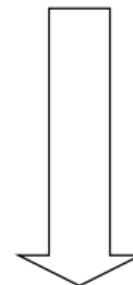
1. A SQL database or relational database is a collection of highly structured tables, wherein each row reflects a data entity, and every column defines a specific information field.
2. Relational databases are built using the structured query language (SQL) to create, store, update, and retrieve data.
3. Therefore, SQL is the underlying programming language for all relational database management systems (RDBMS) such as MySQL, Oracle, and Sybase, among others.

Simple SQL Query: Selection

Selection is the operation of filtering a relation's tuples on some condition

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

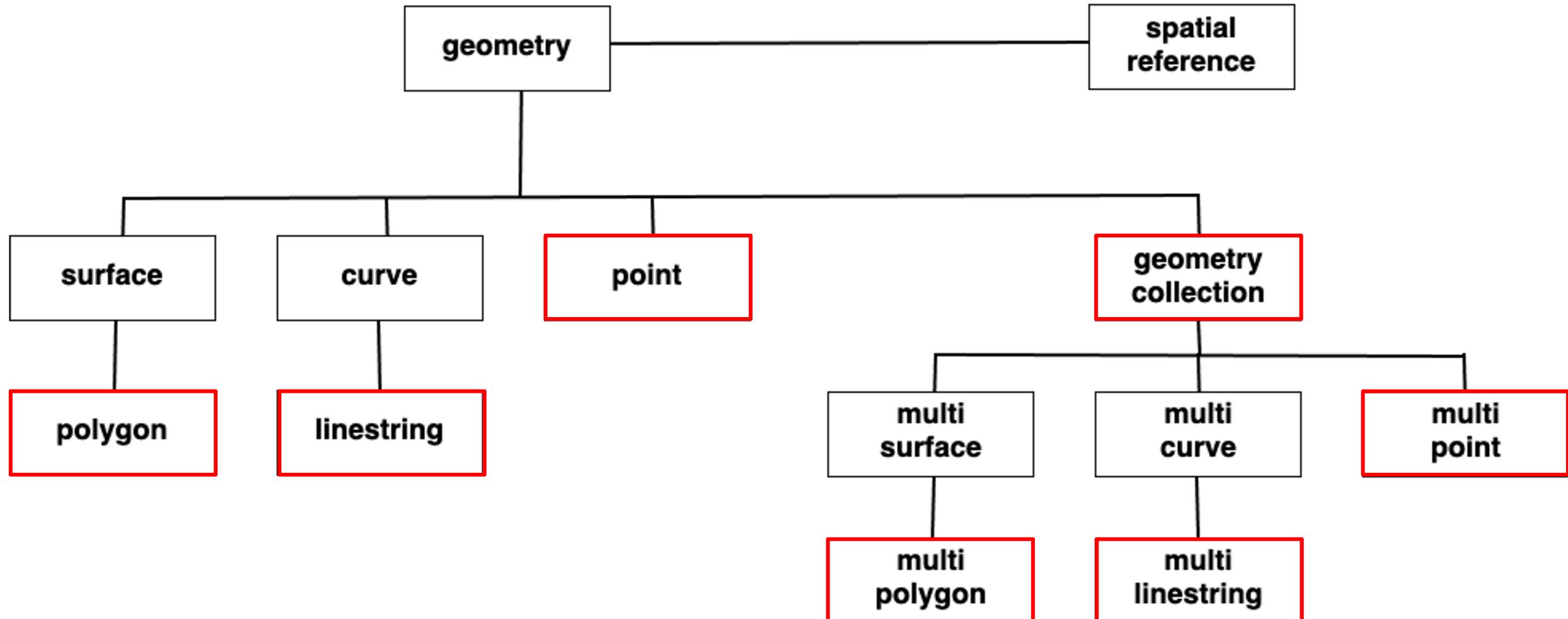
```
SELECT *
FROM Product
WHERE Category = 'Gadgets'
```



PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks

Geometries

Spatial Types



Creating a table with geometry

```
CREATE TABLE geometries  
(  
    name varchar,  
    geom geometry  
);
```

Creating a table with geometry

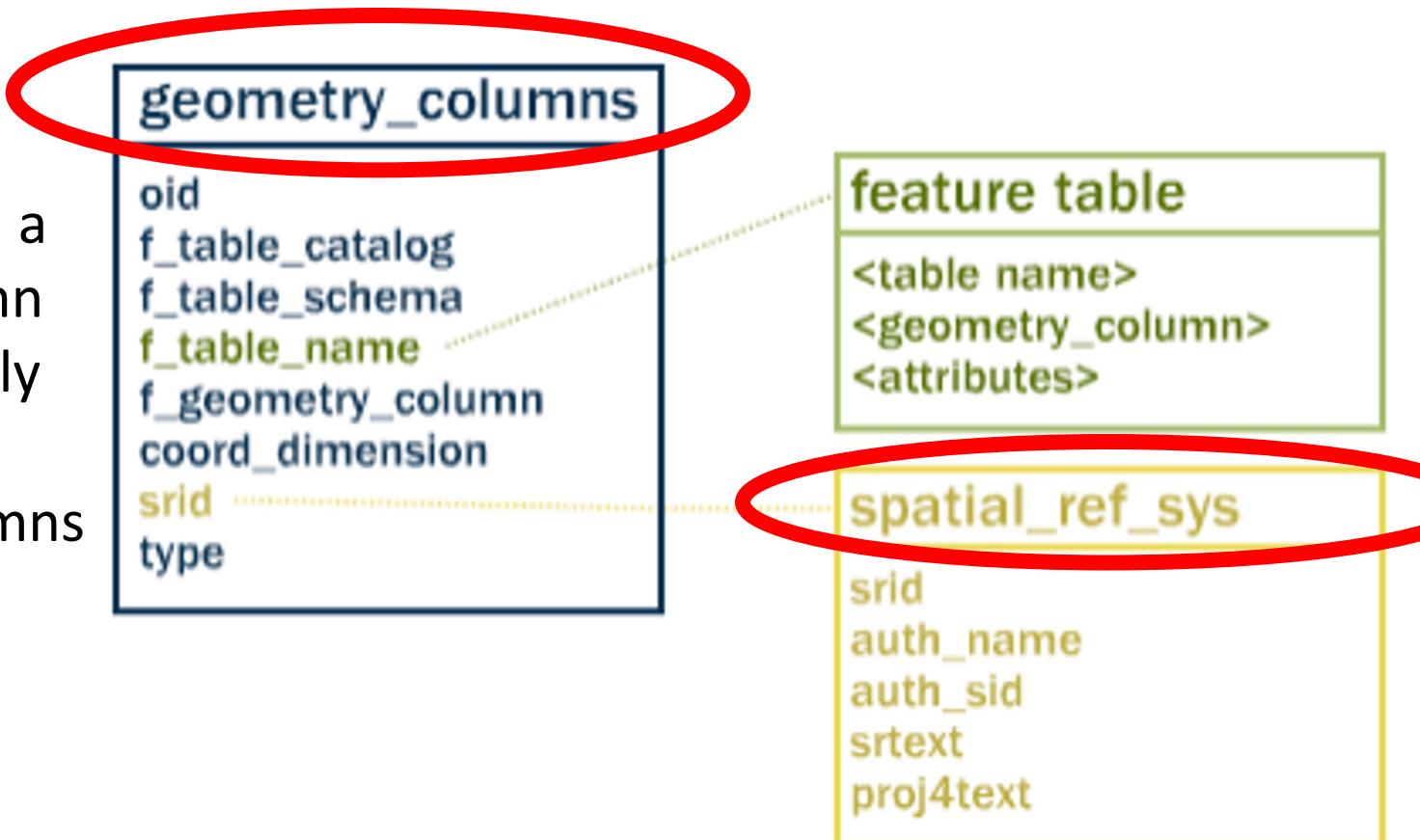
```
INSERT INTO geometries (name, geom) VALUES
('Point', 'POINT(0 0)'),
('Linestring', 'LINESTRING(0 0, 1 1, 2 1, 2 2)'),
('Polygon', 'POLYGON((0 0, 1 0, 1 1, 0 1, 0 0)))'),
('PolygonWithHole', 'POLYGON((...))'),
('Collection', 'GEOMETRYCOLLECTION(...)');
```

Creating a table with geometry

```
SELECT name, ST_AsText(geom)  
FROM geometries;
```

Table Relationships

Every table with a geometry column will automatically appear in the geometry_columns view.



The SRID numbers on geometries and columns are converted into coordinate transforms using data from the spatial_ref_sys table.

geometry_columns

```
SELECT *  
FROM geometry_columns
```

geometry_columns

nyc/postgres@localhost ▾

Query Editor Query History Scratch Pad

1 `SELECT * FROM geometry_columns;`

Data Output Explain Messages Notifications

	f_table_catalog character varying (256)	f_table_schema name	f_table_name name	f_geometry_column name	coord_dimension integer	srid integer	type character varying (30)
1	nyc	public	nyc_census_blocks	geom	2	26918	MULTIPOLYGON
2	nyc	public	nyc_homicides	geom	2	26918	POINT
3	nyc	public	nyc_neighborhoods	geom	2	26918	MULTIPOLYGON
4	nyc	public	nyc_streets	geom	2	26918	MULTILINESTRING
5	nyc	public	nyc_subway_stati...	geom	2	26918	POINT

Metadata functions

SELECT

```
name,  
ST_GeometryType(geom),  
ST_NDims(geom),  
ST_SRID(geom)
```

FROM geometries;

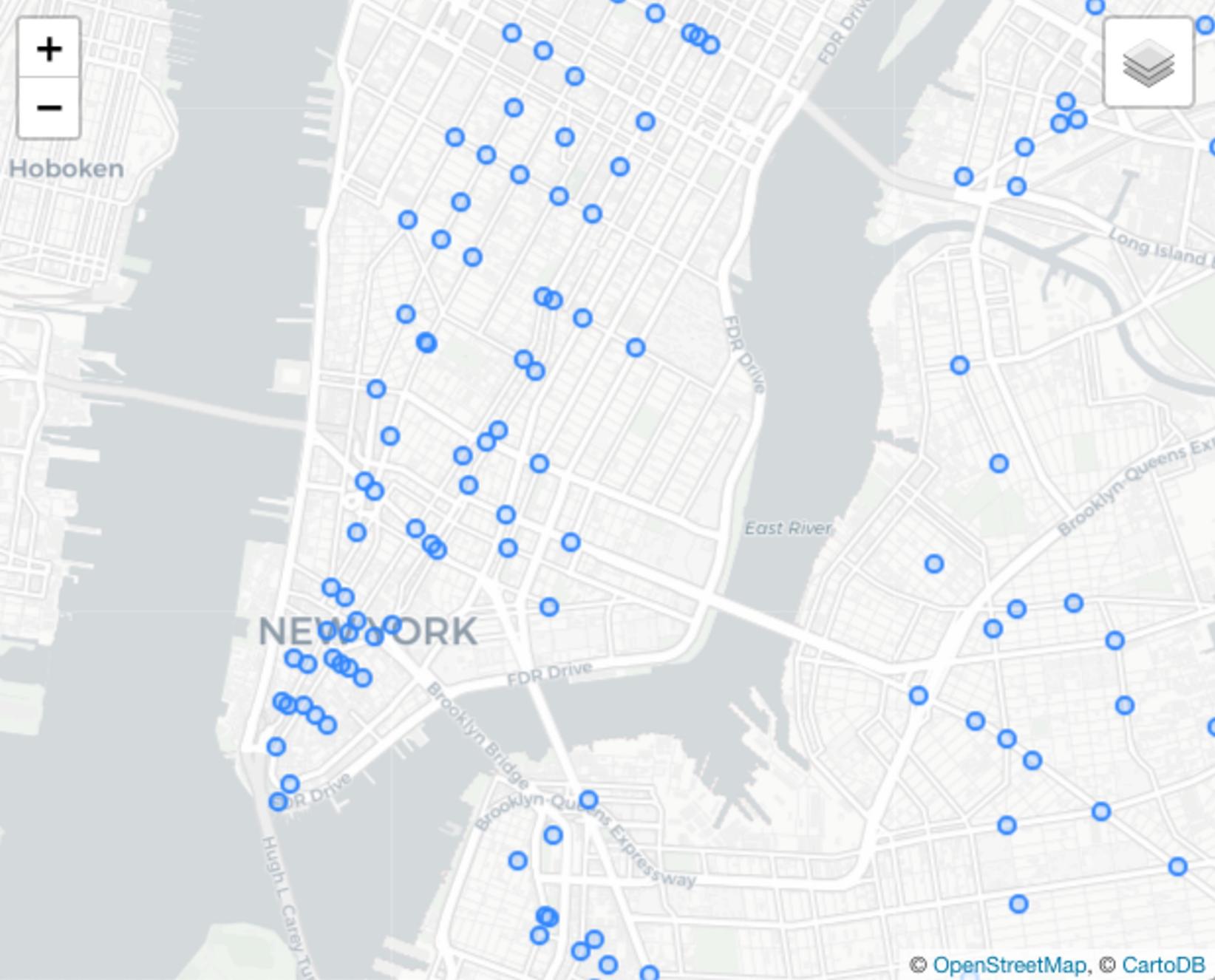
Metadata functions

<code>name</code>	<code>st_geometrytype</code>	<code>st_ndims</code>	<code>st_srid</code>
<code>Point</code>	<code>ST_Point</code>	2	0
<code>Linestring</code>	<code>ST_LineString</code>	2	0
<code>Polygon</code>	<code>ST_Polygon</code>	2	0
<code>PolygonWithHole</code>	<code>ST_Polygon</code>	2	0
<code>Collection</code>	<code>ST_GeometryCollection</code>	2	0

Points

“Point” or “MultiPoint”, representing one or more 0-dimensional locations.

New York city subway stations, stop signs, man holes, address points, current locations of vehicles, might all use a “Point” geometry type.



Points

```
SELECT ST_AsText(geom)  
FROM geometries  
WHERE name = 'Point';
```

POINT(0 0)

Points

```
SELECT  
    ST_X(geom),  
    ST_Y(geom)  
FROM geometries  
WHERE name = 'Point'
```

0 0

Points

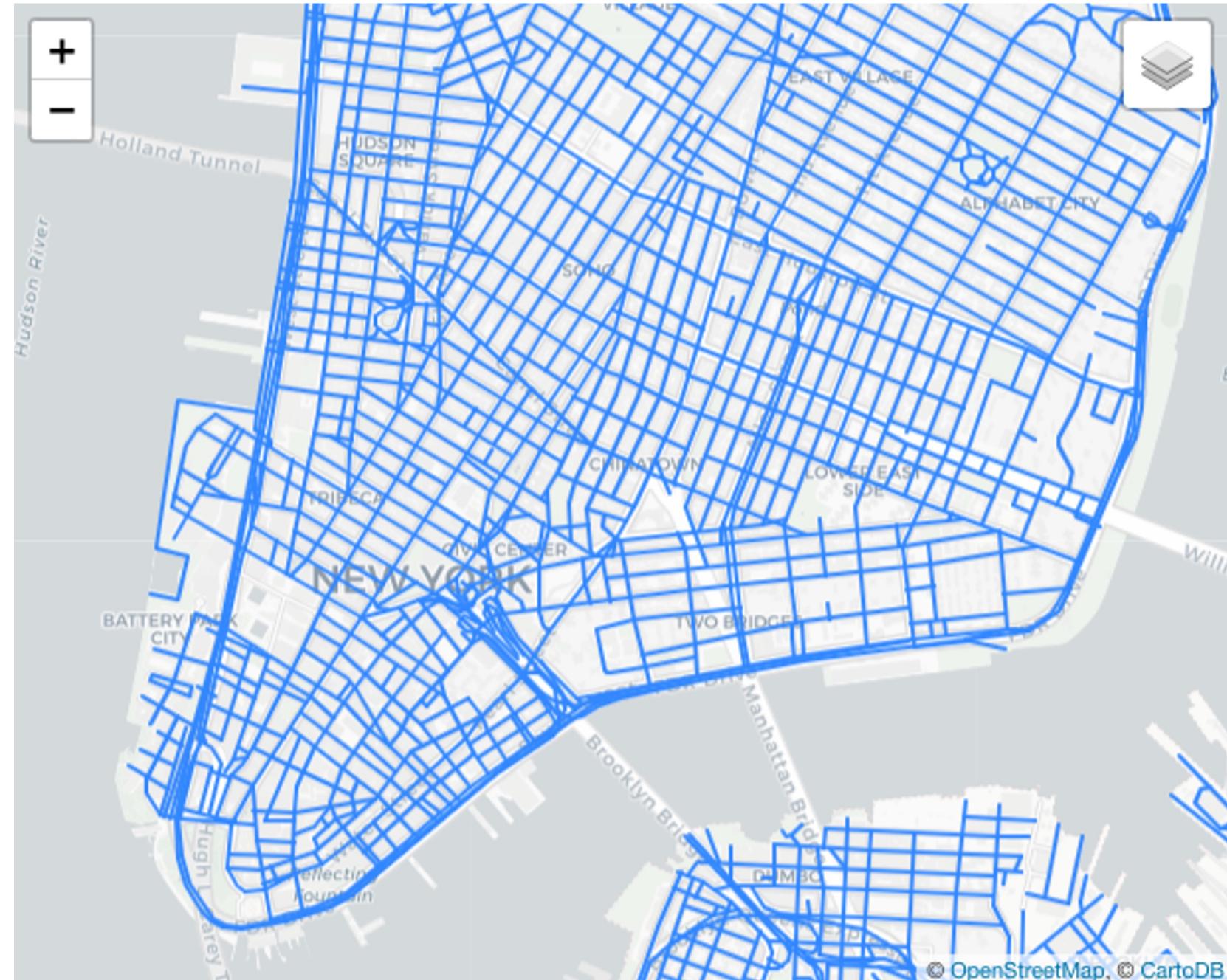
```
SELECT  
    name,  
    ST_AsText(geom)  
FROM nyc_subway_stations  
LIMIT 1;
```

Cortlandt St | POINT(583521 4507077)

LineStrings

“LineString” or
“MultiLineString”,
representing one or more 1-dimensional objects.

Streets, streams, bus routes, power lines, driven routes, highways, might all use a “LineString” geometry type.



LineStrings

```
SELECT ST_AsText(geom)  
FROM geometries  
WHERE name = 'Linestring';
```

LINESTRING(0 0, 1 1, 2 2)

LineStrings

```
SELECT ST_Length(geom)  
FROM geometries  
WHERE name = 'Linestring';
```

3.41421356237309

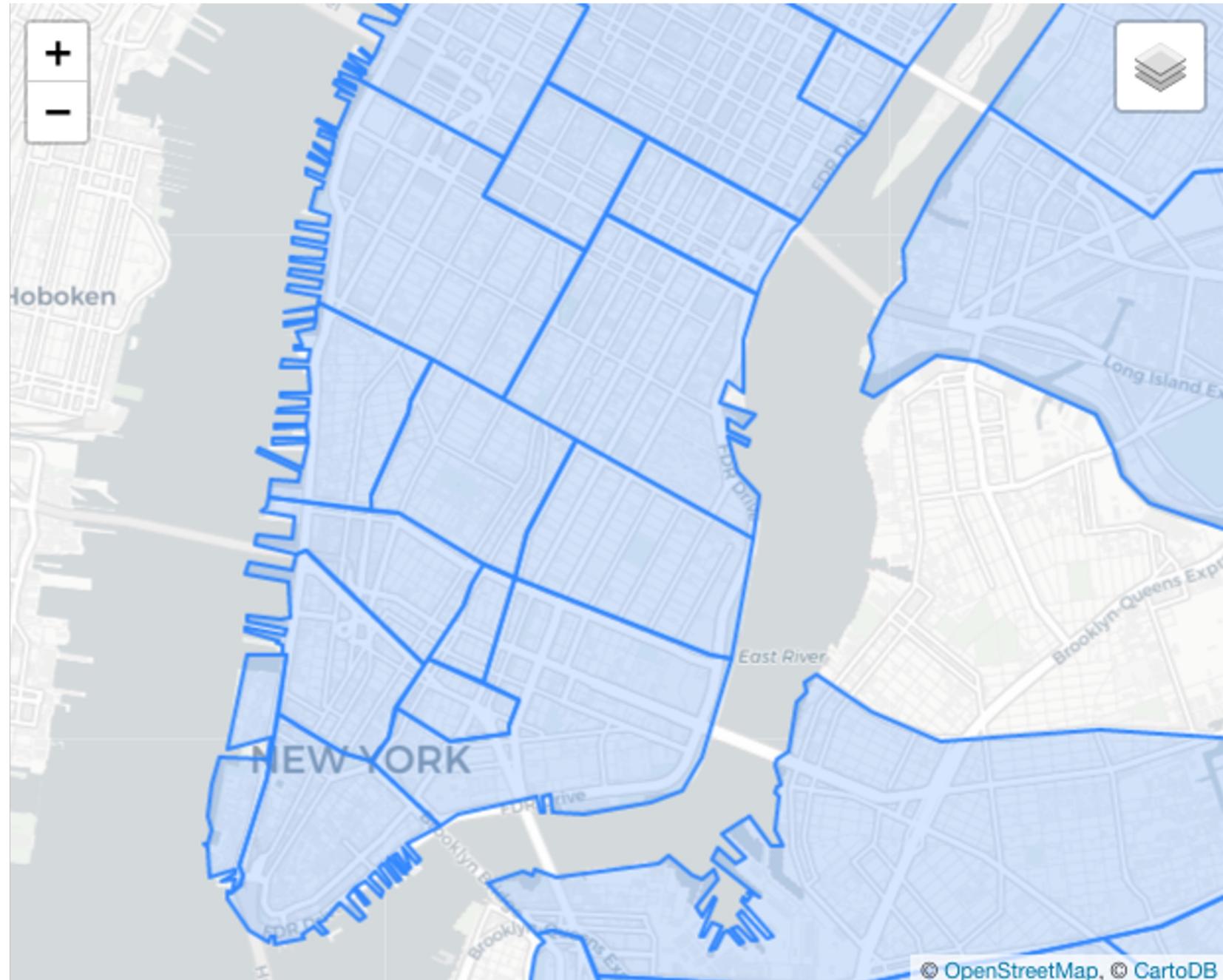
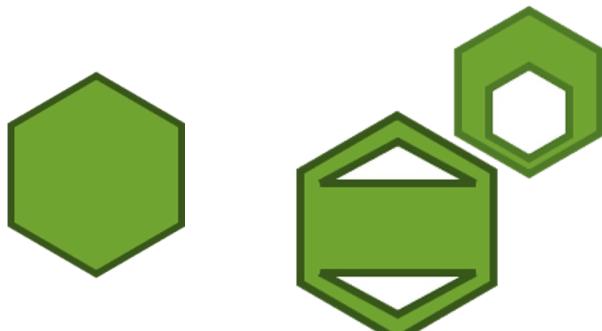
LineStrings

- `ST_Length(linestring)`
- `ST_StartPoint(linestring)`
- `ST_EndPoint(linestring)`
- `ST_NumPoints(linestring)`

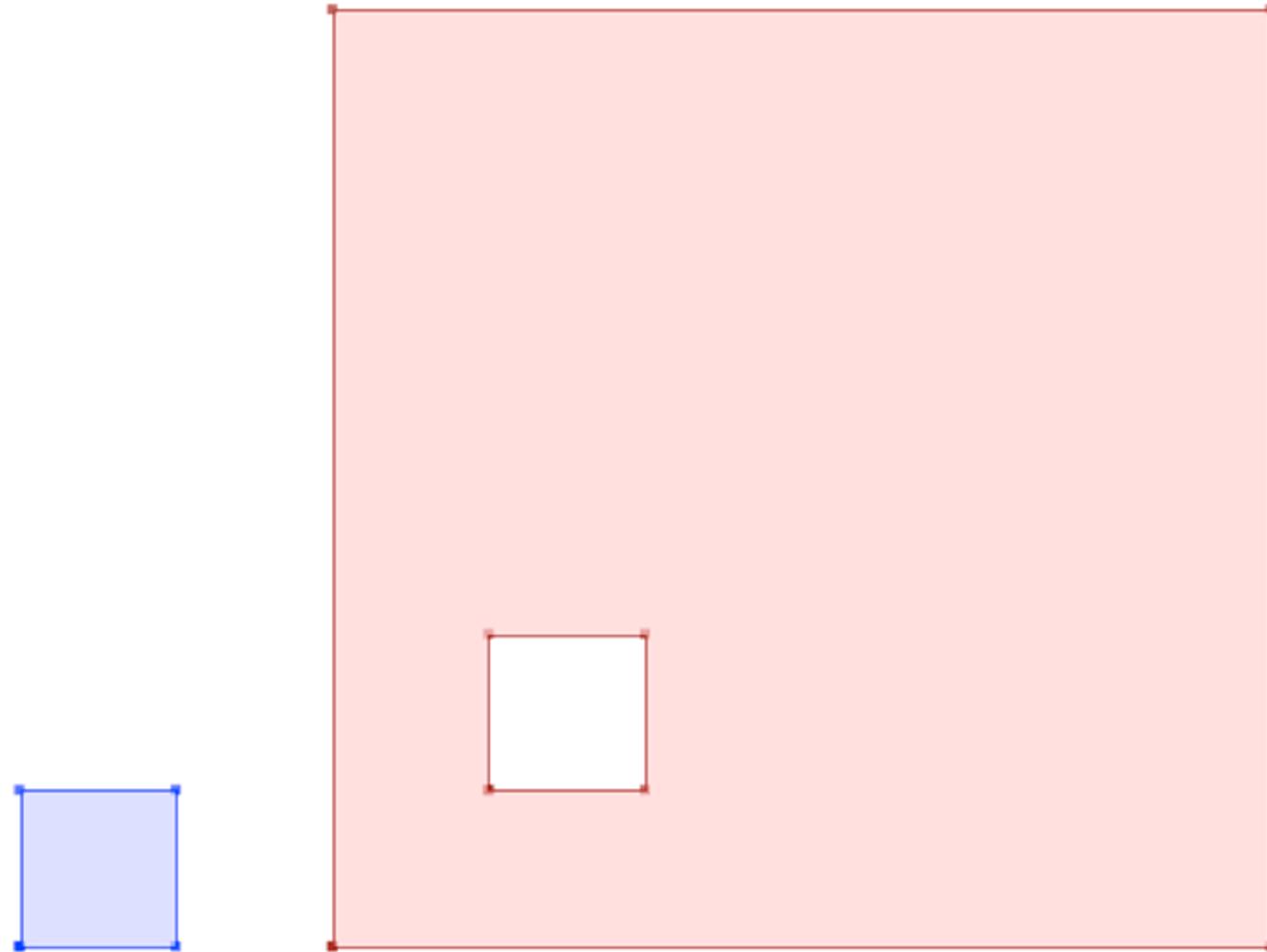
Polygons

“Polygon” or
“MultiPolygon”,
representing one or more 2-dimensional objects.

Census areas, parcels,
counties, countries,
neighborhoods, zoning
areas, watersheds, and
more.



Polygons



Polygons

```
SELECT ST_AsText(geom)  
FROM geometries  
WHERE name LIKE 'Polygon%' ;
```

```
POLYGON((0 0, 1 0, 1 1, 0 1, 0 0))  
POLYGON((0 0, 10 0, 10 10, 0 10, 0 0),  
        (1 1, 1 2, 2 2, 2 1, 1 1))
```

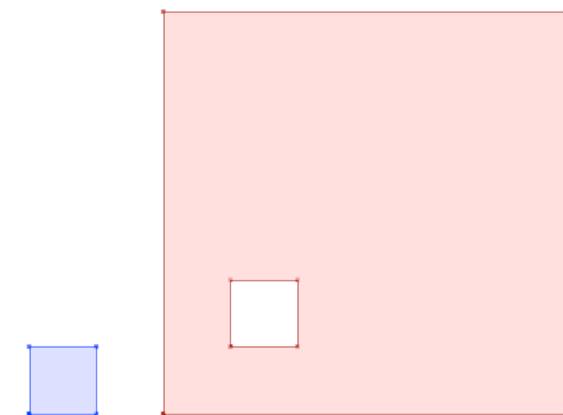
Polygons

- `ST_Area(polygon)`
- `ST_NumInteriorRings(polygon)`
- `ST_ExteriorRing(polygon)`
- `ST_InteriorRing(polygon,n)`
- `ST_Perimeter(polygon)`

Polygons

```
SELECT name, ST_Area(geom)  
FROM geometries  
WHERE name LIKE 'Polygon%' ;
```

Polygon		1
PolygonWithHole		99



Geometry construction

```
SELECT ST_AsEWKT(  
    ST_GeomFromText('POINT(1 1)', 4326)  
);
```

SRID=4326;POINT(1 1)

Geometry construction

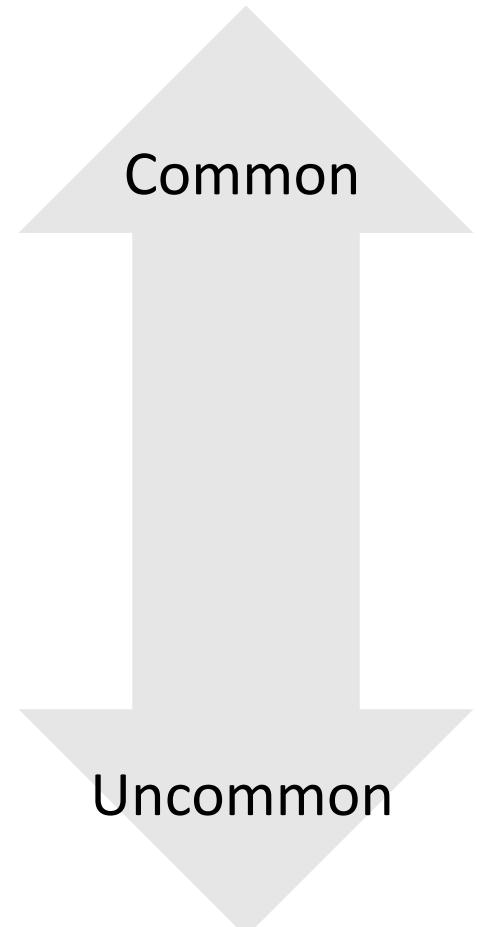
```
SELECT ST_AsEWKT(  
    'SRID=4326;POINT(1 1)' ::geometry  
) ;
```

SRID=4326;POINT(1 1)

Spatial Relationships

Spatial Relationship Functions

- `ST_Intersects(A, B)`
- `ST_DWithin(A, B, d)`
- `ST_Distance(A, B)`
- `ST_Within, ST_Contains(A, B)`
- `ST_Equals(A, B)`
- `ST_Touches(A, B)`
- `ST_Disjoint, ST_Crosses, ST_Overlaps(A, B)`



ST_Equals(A, B)

ST_OrderingEquals(A, B)

Equals tests that A and B cover the same space, regardless of representation differences (extra vertices, order of vertices).

OrderingEquals insists on structural identity.

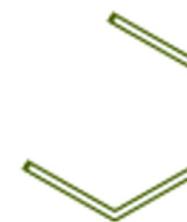
Question: Two Linestrings for the same bus route, but going in opposite directions. What does ST_Equals return. ST_OrderingEquals?



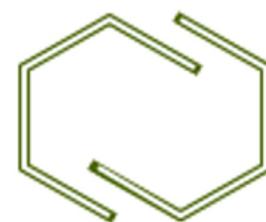
Point & Multipoint



Multipoint & Multipoint



Linestring & Linestring



Multilinestring & Multilinestring



Polygon & Polygon



Multipolygon & Multipolygon

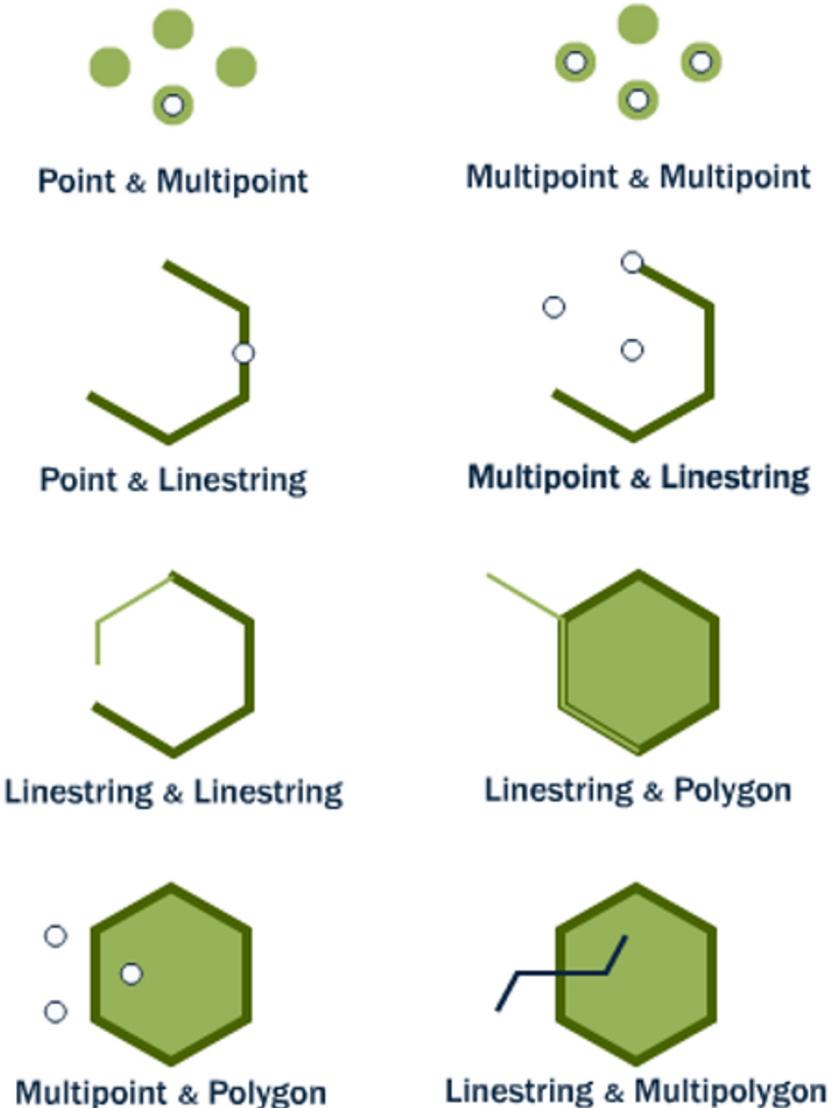
ST_Intersects(A, B)

ST_Disjoint(A, B)

Intersects and disjoint are opposites.
Any kind of interactions between two shapes is an intersection, and implies the pair are not disjoint, and vice versa.

A intersects B \Rightarrow A not disjoint B

A disjoint B \Rightarrow A not intersects B



What is the well-known text (WKT) of Broad Street subway station?

```
SELECT name, ST_AsText(geom, 0)
  FROM nyc_subway_stations
 WHERE name = 'Broad St';
```

POINT(583571 4506714)

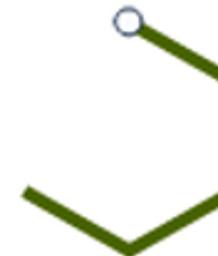
What neighborhood intersects that subway station?

```
SELECT name, boroname  
FROM nyc_neighborhoods  
WHERE ST_Intersects(  
    geom,  
    ST_GeomFromText(  
        'POINT(583571 4506714)',  
        26918));
```

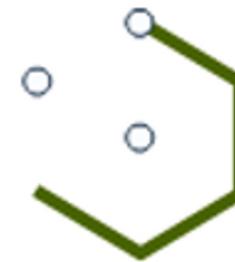
Financial District | Manhattan

ST_Touches(A, B)

Shapes touch when their boundaries interact but their interiors do not. End points for lines, exterior rings for polygons. Usually used for testing that polygons have ring-touching only.



Point & Linestring



Multipoint & Linestring



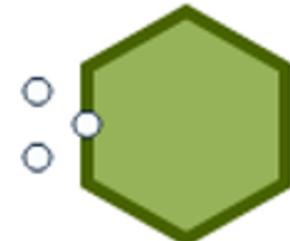
Linestring & Linestring



Linestring & Polygon



Point & Polygon



Multipoint & Polygon

ST_Within(A, B)

ST_Contains(B, A)

Within and contains are about objects being fully inside. One important caveat, for both functions an object on the **boundary** is not considered within. So a point on the outer ring of a polygon is not within the polygon.



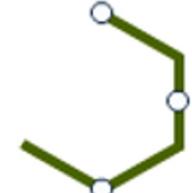
Point & Multipoint



Multipoint & Multipoint



Point & Linestring



Multipoint & Linestring



Linestring & Linestring



Linestring & Polygon



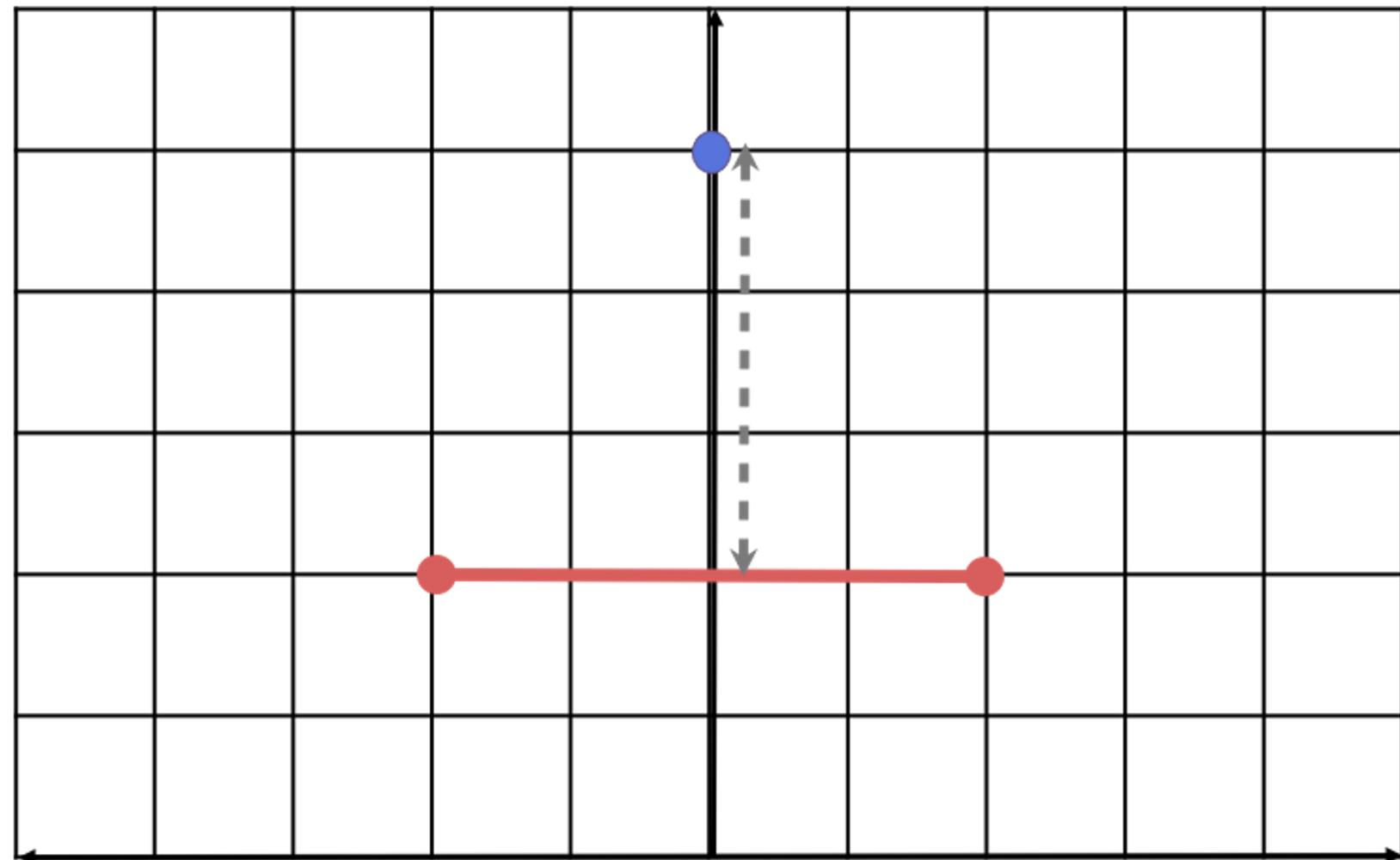
Point & Polygon



Multipoint & Polygon

ST_Distance(A, B)

Returns the **shortest** distance between the two geometries, in this case the distance from the point to the line mid-point.



ST_Distance(A, B)

```
SELECT ST_Distance(  
    'POINT(0 5)'::geometry,  
    'LINESTRING(-2 2, 2 2)'::geometry  
)
```

3

ST_DWithin(A, B, R)

Index-enabled radius search function. True when the distance from geometry A to geometry B is less than radius R. False otherwise.

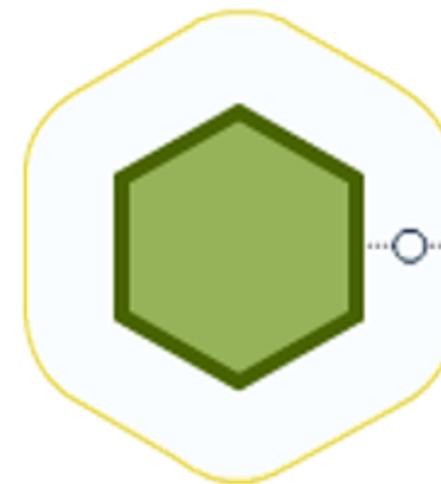
Use instead of
ST_Distance(A, B) < R, in
order to get benefit of spatial
index.



Point & Point (True)



Point & Point (False)



Polygon & Point (True)



Polygon & Point (False)

What streets are within 10 meters of Broad Street subway station?

```
SELECT name
FROM nyc_streets
WHERE ST_DWithin(
    geom,
    ST_GeomFromText('POINT(583571 4506714)', 26918),
    10
);
```

