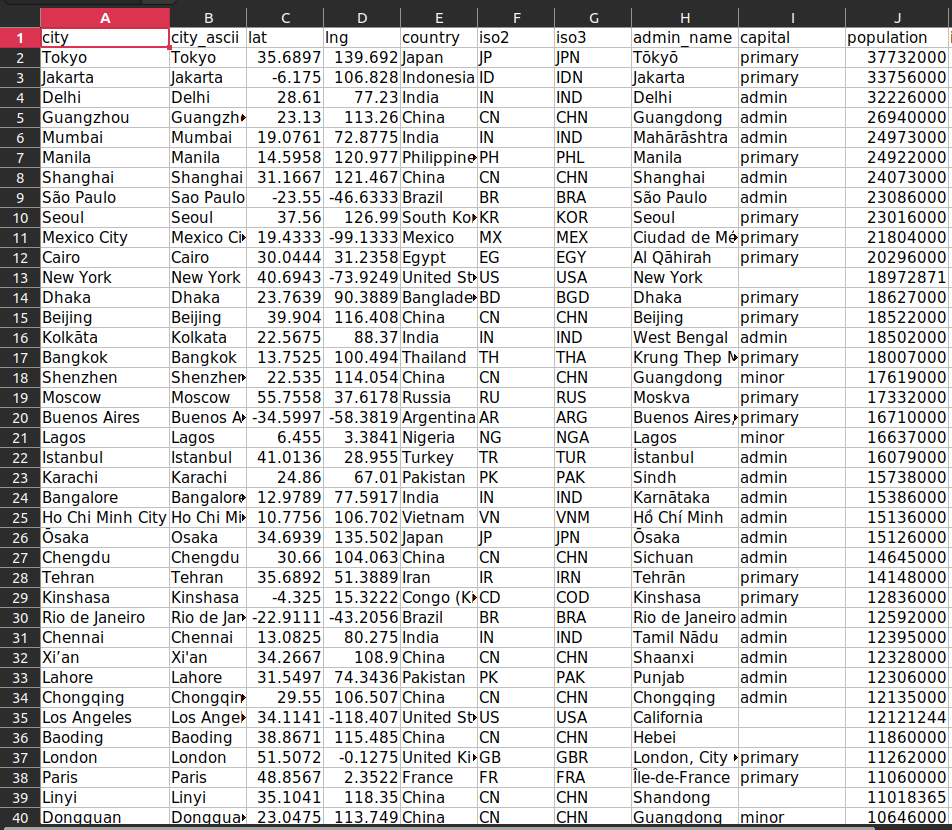
The aim of the project is to create the Geographic Information System (GIS) analysis. The database used in this project is a sql database postgreSQL. PostGIS package for the postgreSQL database was used. PostGIS is an open source software package that increases the capabilities of the database management system by adding support for spatial data. Here, spatial data refers to any kind of data that contains geographic or location information, i.e latitude, longitude, zip code, PostGIS allows you to store, manipulate and query the spatial data such as points, lines, and other dimensions within a database.

**Execution of practical part**

Dataset – The dataset required for the analysis of the spatial data was downloaded from https://simplemaps.com/data/world-cities



Dataset size – 44,690 tuples of world cities along with their informations like latitude, longitude and other information

**Database creation and importing the data into table**

create extension postgis;

select \* from pg\_catalog.pg\_extension;

CREATE TABLE gis\_location (

id SERIAL PRIMARY KEY,

country VARCHAR(50),

city VARCHAR(50),

lat NUMERIC(9, 6),

lon NUMERIC(9, 6),

population INT

);

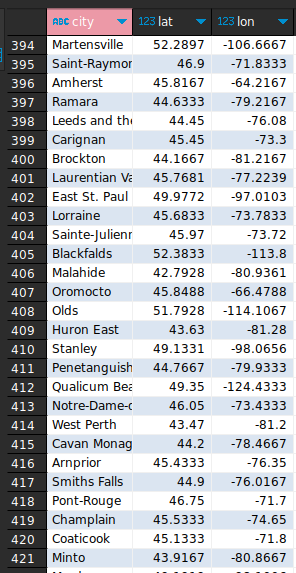
select table\_name, column\_name, data\_type from information\_schema.columns where table\_name='gis\_location';

copy gis\_location(city, lat, lon, country, population) from '/tmp/worldcities.csv' delimiter ',' csv header;

**1. Retrieve Locations of specific features**

-- Retrive cities in a specific country

SELECT city, lat, lon FROM gis\_location WHERE country = 'Canada';



-- Retrieve the city with the highest population in each country

SELECT country, city, lat, lon, population

FROM gis\_location

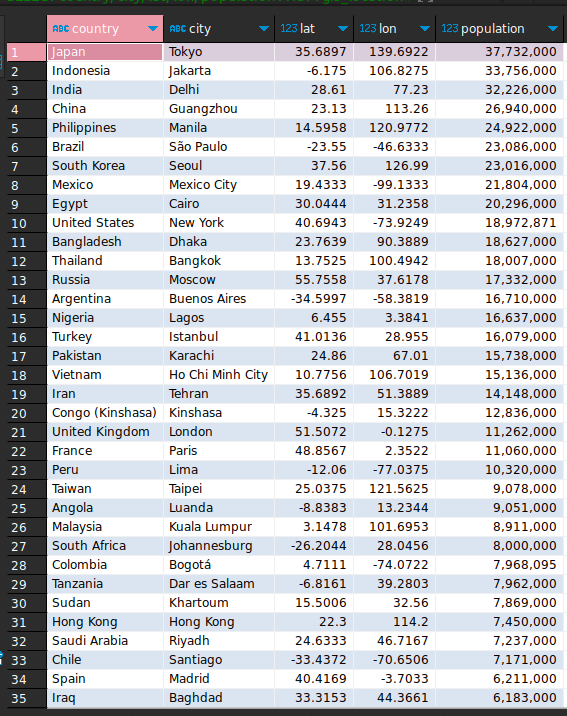
WHERE (country, population) IN (

SELECT country, MAX(population)

FROM gis\_location

GROUP BY country

);

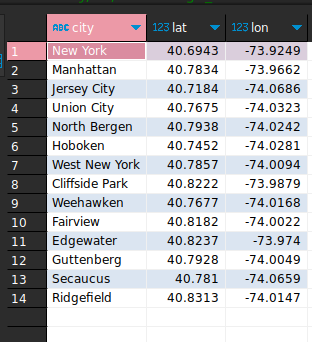


-- Retrieve all cities within a certain distance of a specific point

SELECT city, lat, lon

FROM gis\_location

WHERE ST\_DistanceSphere(ST\_SetSRID(ST\_MakePoint(-73.9857, 40.7484), 4326), ST\_SetSRID(ST\_MakePoint(lon, lat), 4326)) < 10000;



**2. Calculate Distance between points**

-- distance of all the cities from a given points

SELECT city, ST\_DistanceSphere(

ST\_SetSRID(ST\_MakePoint(-73.9857, 40.7484), 4326),

ST\_SetSRID(ST\_MakePoint(lon, lat), 4326)

) AS distance FROM gis\_location;

A picture containing text, screenshot, software, number

Description automatically generated

-- Distance between all cities in the gis\_location table and a specific city (e.g. "New York") using

the ST\_DistanceSphere function:

SELECT city, ST\_DistanceSphere(

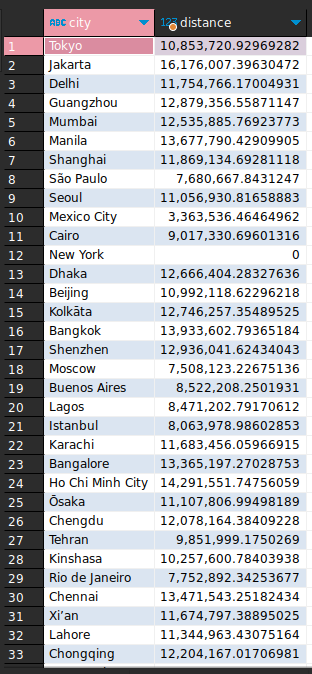
ST\_SetSRID(ST\_MakePoint(lon, lat), 4326),

(SELECT ST\_SetSRID(ST\_MakePoint(lon, lat), 4326)

FROM gis\_location WHERE city = 'New York')

) AS distance

FROM gis\_location;



-- the distance between all cities in the gis\_location table that have a population greater than a certain value (e.g. 1 million) using the ST\_DistanceSpheroid function:

SELECT city, ST\_DistanceSpheroid(

ST\_SetSRID(ST\_MakePoint(lon, lat), 4326),

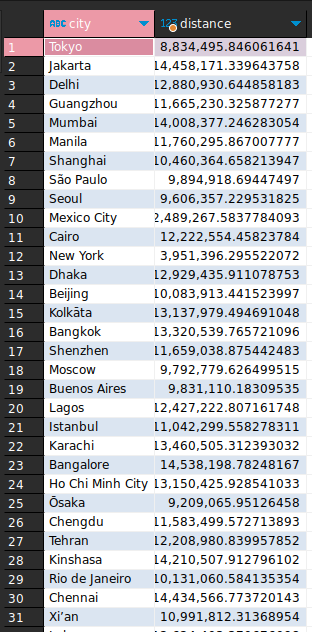
ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326),

' SPHEROID["WGS 84",6378137,298.257223563]'

) AS distance

FROM gis\_location

WHERE population > 1000000;



**3. Calculate Areas of Interest**

-- Find all cities within a certain distance of a given point:

SELECT city, ST\_Distance(

ST\_SetSRID(ST\_MakePoint(lon, lat), 4326),

ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326)

) AS distance

FROM gis\_location

WHERE ST\_DWithin(

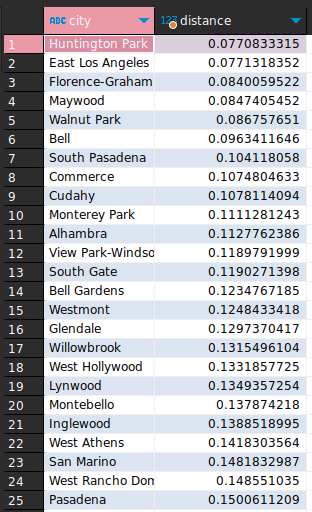
ST\_SetSRID(ST\_MakePoint(lon, lat), 4326),

ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326),

50000

)

ORDER BY distance;



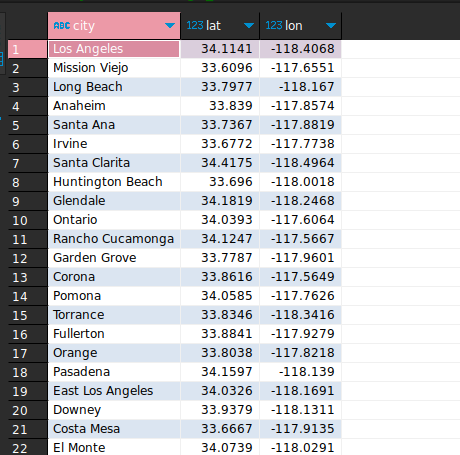
-- Find all cities within a given bounding box:

SELECT city, lat, lon

FROM gis\_location

WHERE lon BETWEEN -118.5 AND -117.5

AND lat BETWEEN 33.5 AND 34.5;



-- Find all cities within a given polygon:

SELECT city, lat, lon

FROM gis\_location

WHERE ST\_Contains(

ST\_SetSRID(

ST\_MakePolygon(

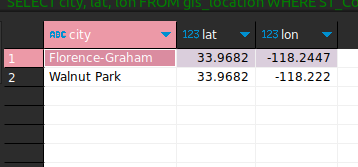
ST\_GeomFromText('LINESTRING(-118.2437 34.0522, -118.3 34.05, -118.2 33.9, -118.2437 34.0522)')

),

4326

),

ST\_SetSRID(ST\_MakePoint(lon, lat), 4326));



-- Find all cities with a population above a certain threshold within a certain distance of a given point:

SELECT city, population, ST\_Distance(

ST\_SetSRID(ST\_MakePoint(lon, lat), 4326),

ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326)

) AS distance

FROM gis\_location

WHERE population > 100000

AND ST\_DWithin(

ST\_SetSRID(ST\_MakePoint(lon, lat), 4326),

ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326),

50000

)ORDER BY distance;

A screenshot of a computer

Description automatically generated with medium confidence

**4. Sorting and Limit Executions**

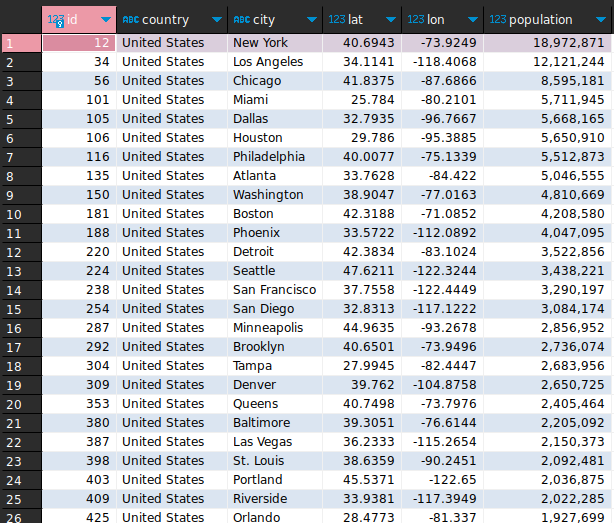
-- Retrieving GIS locations for specific features and sorting by population in descending order:

SELECT \*

FROM gis\_location

WHERE country = 'United States'

ORDER BY population DESC;



-- Retrieving the top 10 most populous cities in the USA:

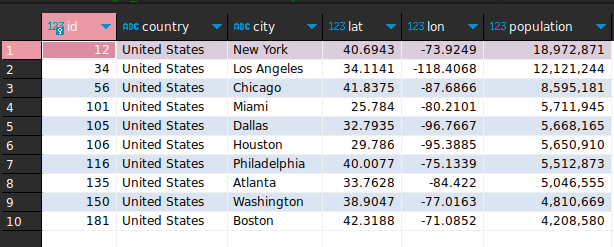
SELECT \*

FROM gis\_location

WHERE country = 'United States'

ORDER BY population DESC

LIMIT 10;



-- Retrieving GIS locations within a certain distance from a point, and sorting by distance in ascending order:

SELECT city, ST\_Distance(

ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326),

ST\_SetSRID(ST\_MakePoint(gis\_location.lon, gis\_location.lat), 4326)

) AS distance

FROM gis\_location

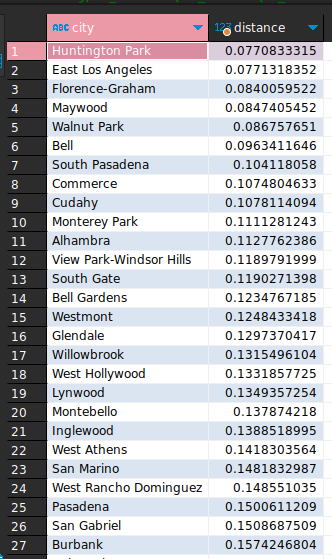
WHERE country = 'United States' AND ST\_Distance(

ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326),

ST\_SetSRID(ST\_MakePoint(gis\_location.lon, gis\_location.lat), 4326)

) < 500000

ORDER BY distance ASC;



-- Retrieving the 5 closest cities to a specific point:

SELECT city, ST\_Distance(

ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326),

ST\_SetSRID(ST\_MakePoint(gis\_location.lon, gis\_location.lat), 4326)

) AS distance

FROM gis\_location

WHERE country = 'United States' AND ST\_Distance(

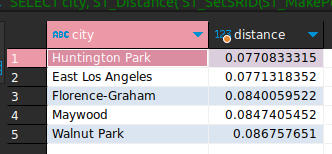
ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326),

ST\_SetSRID(ST\_MakePoint(gis\_location.lon, gis\_location.lat), 4326)

) < 500000

ORDER BY distance ASC

LIMIT 5;



**5. Optimize the queries to speed up execution time**

-- creating spatial index on lon and lat

CREATE INDEX gis\_location\_geom\_idx ON gis\_location USING GIST (ST\_SetSRID(ST\_MakePoint(lon, lat), 4326));

SELECT city, ST\_Distance(

ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326),

ST\_SetSRID(ST\_MakePoint(gis\_location.lon, gis\_location.lat), 4326)

) AS distance

FROM gis\_location

WHERE country = 'United States' AND

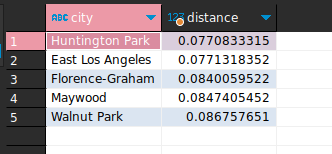
ST\_DWithin(

ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326),

ST\_SetSRID(ST\_MakePoint(gis\_location.lon, gis\_location.lat), 4326),

500000

)ORDER BY distance ASC LIMIT 5;



-- Use a smaller bounding box to limit the search space:

SELECT city, ST\_Distance(

ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326),

ST\_SetSRID(ST\_MakePoint(gis\_location.lon, gis\_location.lat), 4326)

) AS distance

FROM gis\_location

WHERE country = 'United States' AND

lon BETWEEN -77 AND -76 AND

lat BETWEEN 38 AND 39 AND

ST\_DWithin(

ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326),

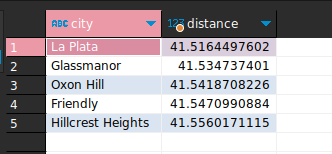
ST\_SetSRID(ST\_MakePoint(gis\_location.lon, gis\_location.lat), 4326),

500000

)

ORDER BY distance ASC

LIMIT 5;



-- Use the ST\_DWithin function instead of ST\_Distance:

SELECT city

FROM gis\_location

WHERE ST\_DWithin(

ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326),

ST\_SetSRID(ST\_MakePoint(gis\_location.lon, gis\_location.lat), 4326),

500000

) AND country = 'United States'

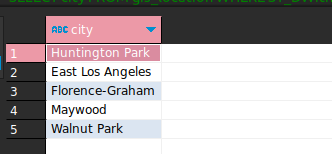
ORDER BY ST\_Distance(

ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326),

ST\_SetSRID(ST\_MakePoint(gis\_location.lon, gis\_location.lat), 4326)

) ASC

LIMIT 5;



**6. N-Optimization of queries**

-- create spatial index

CREATE INDEX gis\_location\_geom\_idx ON gis\_location USING GIST (ST\_SetSRID(ST\_MakePoint(lon, lat), 4326));

-- Use filters to narrow down the search space:

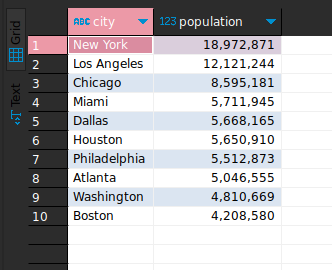
SELECT city, population

FROM gis\_location

WHERE country = 'United States' AND population > 1000000

ORDER BY population DESC

LIMIT 10;



-- Use subqueries to break down complex queries:

SELECT city, population

FROM (

SELECT city, population,

ST\_Distance(

ST\_SetSRID(ST\_MakePoint(lon, lat), 4326),

ST\_SetSRID(ST\_MakePoint(-118.2437, 34.0522), 4326)

) AS distance

FROM gis\_location

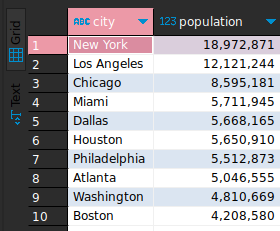
WHERE country = 'United States'

) AS subquery

WHERE distance < 50000

ORDER BY population DESC

LIMIT 10;



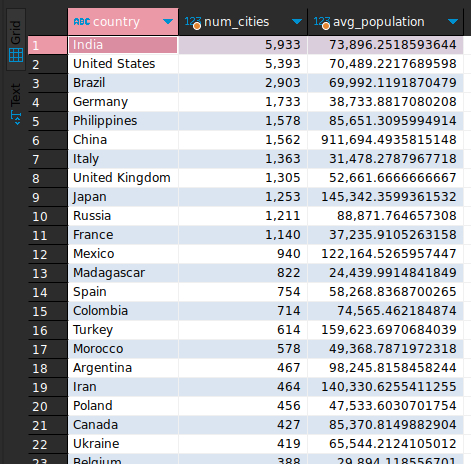
-- Use aggregates to summarize data:

SELECT country, COUNT(\*) AS num\_cities, AVG(population) AS avg\_population

FROM gis\_location

GROUP BY country

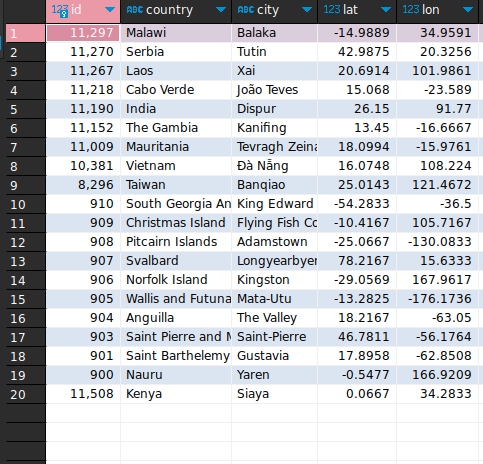
ORDER BY num\_cities DESC;



--

Use LIMIT and OFFSET to paginate results:

select \* from gis\_location order by population desc limit 20 offset 40;



-- Use EXPLAIN to analyze query performance:

EXPLAIN ANALYZE SELECT city, population

FROM gis\_location

WHERE country = 'USA' AND population > 1000000

ORDER BY population DESC LIMIT 10;

