* **Database type**
  + **Relational**
    - Postgres (Open source)
    - SQL Server (Commercial)
    - SQLite (File-based)
  + **Non-relational**
    - MongoDB (JSON like)
* **Postgres/PostGIS**
  + **Server Configurations**
  + **Carto**
  + **Select All**

SELECT \* FROM dcptransportation.citibike\_skill

* + **Select Columns**

SELECT cartodb\_id, the\_geom, the\_geom\_webmercator, data\_stations\_lon, TO\_CHAR(data\_stations\_lon, '99D9') AS lon1, ROUND(data\_stations\_lon) AS lon2, EXTRACT(DOW FROM time) AS dow FROM citibike\_skill

* + **Filtering**

SELECT \* FROM citibike\_skill WHERE data\_stations\_capacity >= 20 AND boro IN ('Bk','Bx') AND (data\_stations\_name ILIKE '%ave%' OR data\_stations\_name LIKE 'Old Fulton S\_')

* + **Sorting**

SELECT \* FROM citibike\_skill ORDER BY data\_stations\_capacity DESC, boro ASC

* + **Join**
    - **WHERE**

SELECT station.\*, ridership.\_07\_06\_07\_1 AS citibikeridership FROM citibike\_skill AS station, citibike\_ridership AS ridership WHERE station.data\_stations\_station\_id = ridership.id

* + - **JOIN**

SELECT station.\*, ridership.\_07\_06\_07\_1 AS citibikeridership FROM citibike\_skill station LEFT JOIN citibike\_ridership ridership ON data\_stations\_station\_id = id

* + **Aggregation**

SELECT boro, MIN(cartodb\_id), AVG(data\_stations\_station\_id), SUM(data\_stations\_capacity), COUNT(data\_stations\_name) FROM citibike\_skill GROUP BY boro

* + **Case Statement**

SELECT \*, CASE WHEN cartodb\_id <= 500 THEN 'a'

WHEN cartodb\_id <= 1000 THEN 'b'

ELSE 'c' END AS category FROM citibike\_skill

* + **Spatial**
    - **Spatial join**

SELECT station.\*, modzcta.modzcta AS zcta FROM citibike\_skill AS station, modzcta\_2010\_wgs1984\_geo AS modzcta WHERE ST\_Intersects(station.the\_geom, modzcta.the\_geom)

* + - **Buffer**

SELECT cartodb\_id, ST\_Transform(ST\_Buffer(ST\_Transform(the\_geom, 6539), 1320), 4326) AS the\_geom, ST\_Transform(ST\_Buffer(ST\_Transform(the\_geom, 6539), 1320), 3857) As the\_geom\_webmercator FROM citibike\_skill

* + - **Centroid**

SELECT modzcta.cartodb\_id, modzcta.modzcta, ST\_Centroid(the\_geom) the\_geom, ST\_Centroid(the\_geom\_webmercator) the\_geom\_webmercator FROM modzcta\_2010\_wgs1984\_geo modzcta

* + **With Clause / Common Table Expressions**

WITH zcta AS (SELECT cartodb\_id, the\_geom, the\_geom\_webmercator, modzcta FROM modzcta\_2010\_wgs1984\_geo),

citibikezcta AS (SELECT station.\*, zcta.modzcta FROM citibike\_skill AS station, zcta WHERE ST\_Intersects(station.the\_geom, zcta.the\_geom)),

zctasumcap AS (SELECT modzcta, SUM(data\_stations\_capacity) AS sumcap FROM citibikezcta GROUP BY modzcta)

SELECT zcta.\*, zctasumcap.sumcap FROM zcta, zctasumcap WHERE zcta.modzcta = zctasumcap.modzcta

* + **Insert**

INSERT INTO citibike\_skill (cartodb\_id, data\_stations\_short\_name) VALUES (10, 'abcd')

* + **Delete**

DELETE FROM citibike\_skill WHERE cartodb\_id = 10

* + **Update**
    - **General**

UPDATE citibike\_skill SET data\_stations\_short\_name = 'abc' WHERE cartodb\_id = 1

* + - **Join**

UPDATE citibike\_skill SET citibikeridership = ridership.\_07\_06\_07\_1 FROM citibike\_ridership AS ridership WHERE data\_stations\_station\_id = id

* + **Resources:**
    - <https://carto.com/help/tutorials/using-sql/>
    - <https://carto.com/help/working-with-data/tips-for-geospatial-analysis/>
    - <https://www.postgresqltutorial.com/postgresql-extract/>
    - <https://www.postgresqltutorial.com/postgresql-joins/>
* **SQLite/Spatialite**
  + DBeaver
  + QGIS
* **R**