* **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

* + **Filter**

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\_')

* + **Sort**

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

* + **Join**

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

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

* + With statement/ Common Table Expressions
  + Case when
    - SELECT
    - \*,
    - CASE
    - WHEN e202010 >= 0
    - AND e202010 <= 1000 THEN '0-1000'
    - WHEN e202010 > 1000
    - AND e202010 <= 2000 THEN '1001-2000'
    - ELSE '>2000'
    - END AS category
    - FROM
    - dcptransportation.subwayridership
  + Spatial
    - SELECT
    - citibikestation.\*,
    - modzcta\_2010\_wgs1984\_geo.modzcta
    - FROM
    - dcptransportation.citibikestation,
    - dcptransportation.modzcta\_2010\_wgs1984\_geo
    - WHERE
    - ST\_Intersects(citibikestation.the\_geom, modzcta\_2010\_wgs1984\_geo.the\_geom)
  + **Insert**

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

* + **Delete**

DELETE FROM citibike\_skill WHERE cartodb\_id = 10

* + **Update**

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

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