* Database type
  + Relational
    - Postgres (Open source)
    - SQL Server (Commercial)
    - SQLite (File-based)
  + Non-relational
    - MongoDB (JSON like)
* Postgres/PostGIS
  + Server
  + Carto
  + Select all
    - SELECT \* FROM dcptransportation.citibike\_skill
  + Select columns
    - SELECT cartodb\_id, the\_geom, the\_geom\_webmercator FROM dcptransportation.citibike\_skill
  + Filter
    - SELECT \* FROM dcptransportation.citibike\_skill WHERE data\_stations\_capacity >= 20 AND boro IN ('Bk','Bx') AND data\_stations\_name ILIKE 'c%'
  + Sort
    - SELECT \* FROM dcptransportation.citibike\_skill ORDER BY data\_stations\_capacity DESC, boro ASC
  + AS
  + SELECT tablea.\*, tableb.id FROM tablea, tableb WHERE
  + DATE
  + JOIN
  + GROUP BY
  + With statement
  + Case when
  + Spatial
  + Insert/Delete/Update
    - INSERT INTO table (cartodb\_id, column2) VALUES (‘10’, ‘abc’)
    - DELETE FROM Users WHERE cartodb\_id=10
    - UPDATE table SET id=’abc’ WHERE email=’abc@gmail.com’
  + Resources:
    - <https://carto.com/help/tutorials/using-sql/>
    - <https://carto.com/help/working-with-data/tips-for-geospatial-analysis/>
* SQLite/Spatialite
  + DBeaver
  + QGIS
* R