# SQLite lab – SQL on non-relational data

Use Jupyter Notebook if possible. If not, you can use PyCharm or the interactive Python command line interpreter.

* We will use the DubLinked datasets relating to Dublin Bikes usage in 2018.
* The following csv file describes the bike stations in Dublin.

<https://data.smartdublin.ie/dataset/33ec9fe2-4957-4e9a-ab55-c5e917c7a9ab/resource/2dec86ed-76ed-47a3-ae28-646db5c5b965/download/dublin.csv>

* Download and run the program CreateBikeStationDB. You may use the .py or the .ipynb, depending on your environment. You do not need to download data – the program reads the data directly from the source.
* Open the program QueryBikeStationDBEmpty. Again, choose the .py or the .ipynb depending on your environment. Within that program, for the moment, comment out the last line: conn.close(). Work on the queries below before taking the comment off the close() statement and running it.

Write queries to solve the following:

* Display the first two rows of the usage table.
* Determine keys:
  + Station table:
  + Display the number of rows in the station table.
  + Display the number of distinct values for a column or set of columns you think could be act as a primary key to the station table.
  + Usage table:
  + Display the number of rows in the usage table.
  + Display the number of distinct values for a column or set of columns you think could be act as a primary key to the usage table
* Join columns:
  + Which column(s) would you pick to join the tables?
  + Are all of the values in the usage table for every join column value in the station table?
  + Are all of the values in the station table for every join column value in the usage table?
* Missing / unmatched values:
  + Write a query to list the join column(s) values in the station table that are not in the usage table, if any.
  + Write a query to list the join column(s) values in the usage table that are not in the station table, if any.
* Queries:
  + return the station name, latitude, longitude and count of times recorded for every station beginning with the letter S in the usage table.
  + return the names of stations that are in the usage table, but not in the station table
  + return the names of stations that are in the station table, but not in the usage table
  + return the count of station names that appear in both the station table and the usage table.