**Laboration 1**

To start of, we decided to use SQLitestudio which is a easier platform to work on than the terminal.

1.

SELECT Name

FROM Country

WHERE Population < 1000

2.

SELECT Name AS CountryName, Area, Population

FROM Country

WHERE Population < 1000

Using Name AS CountryName to display Name as CountryName instead of altering the whole table

3.

SELECT Name, Population, Area

FROM Country

WHERE 2000 > Population AND Population > 1000 AND Area >= 1

4.

SELECT Code

FROM Country

WHERE Name = 'Norway'

5.

SELECT Name

FROM City

WHERE Country = ‘S’ AND Population > 500000

Found that Sweden had country code “S” by looking through the database

6.

SELECT Name, Population, Elevation

FROM City

WHERE Elevation < 0

7.

SELECT SUM(Population)AS total, AVG(Population) AS average, MIN(Population) AS minimum, MAX(Population) AS maximum

FROM City

WHERE Elevation < 0

8.

**Using the except statement:**

SELECT Name

FROM City

WHERE (Name LIKE 'Los%' OR Name LIKE '%holm')

EXCEPT

SELECT Name

FROM City

WHERE Name LIKE '%is'

**Not using the except statement:**

SELECT Name

FROM City

WHERE (Name LIKE 'Los%' OR Name LIKE '%holm') AND Name NOT LIKE '%is'

**Brackets:**

The brackets matter in this scenario since the “OR” operator is prioritized over the “AND” operator. If we skip the brackets in this case, we don’t get the right answer.

9.

SELECT Name, Population

FROM Country

ORDER BY Population DESC

LIMIT 5

(DESC) means descending

10.

SELECT Country.Name, City.Name, City.Elevation

FROM City

FULL OUTER JOIN Country ON City.Country = Country.Code

WHERE City.Elevation IS NOT NULL

ORDER BY City.Elevation

LIMIT 5

We tested INNERJOIN aswell, both codes are sufficient, FULL OUTERJOIN gives us all the attributes from Country and City where the Country.Code is same as City.Country

11.

SELECT Name

FROM Country

WHERE Name LIKE 'Y%'

UNION

SELECT Name

FROM City

WHERE Name LIKE '%x'

Writing two queries to give us the result