Estefany Caicedo

1st December 2023

**World**

**Assignment 2 – SQL**

**Contents**

**Database……………………………..1**

**MySQL – SQLiteonline.com………..2**

**Database:**

*MySQL – SQLiteonline*

The data set is available as a set of three tables:

* country: Information about countries of the world.
* city: Information about some of the cities in those countries.
* countrylanguage: Languages spoken in each country.

A screen shot of a computer

Description automatically generated

By using Visual Studio Code, I was able to copy the script and save it into a sql file. From here I imported it and run it to create the data base and the data.

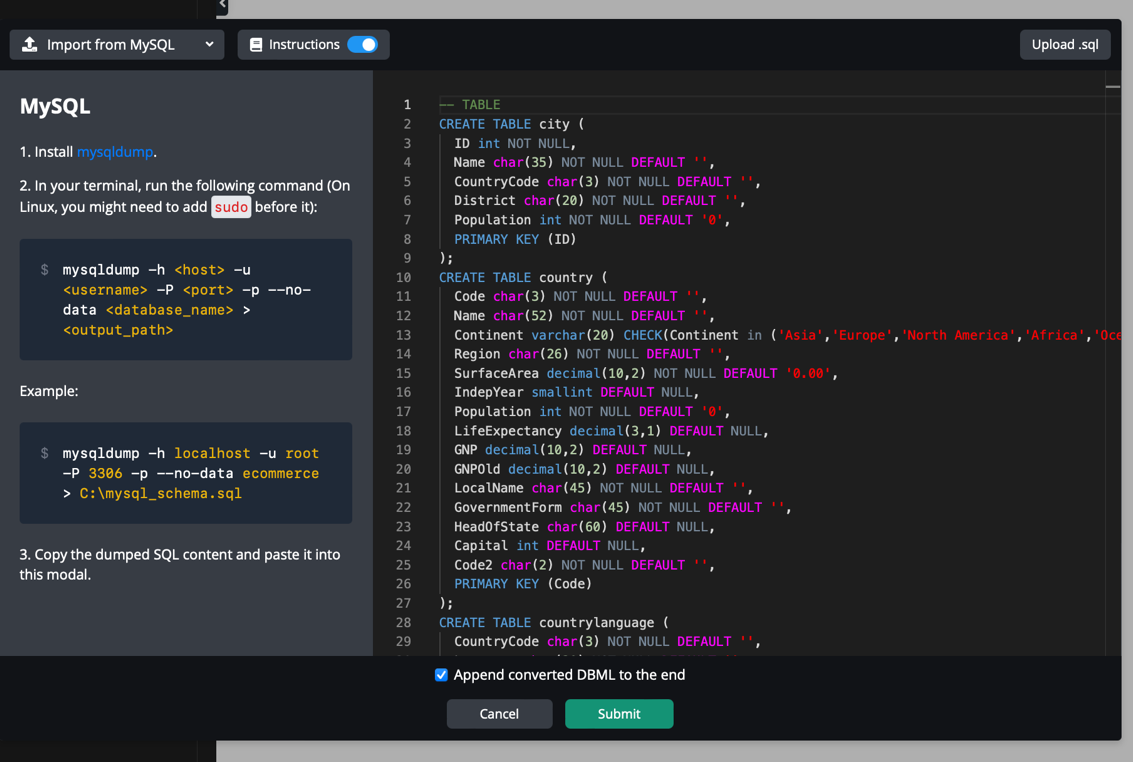
A screenshot of a computer

Description automatically generated

**MySQL – SQLiteonline.com**

*Tasks:*

**1. Produce schema diagram**: To create the diagram I used the dbdiagram.io and uploaded the data by importing from MySQL.



The diagram below shows that these relationships are based on the foreign key constraints and column associations present in the tables. The relationships can be:

one country can have many cities, one country can have many languages, one city belongs to one country, one language is associated with one country and finally multiple countries may share the same capital city.

A screenshot of a computer

Description automatically generated

**2.** Using **count**, get the **number of cities in the USA**:

This syntax selects the count of rows from the city table where the countrycode is 'USA'. The result will give me the number of cities in the USA.

A screenshot of a computer program

Description automatically generated

**3.** Find out what **the population and average life expectancy** for people **in Argentina (ARG)** is:

This query calculates the total population and average life expectancy for the country with the code 'ARG'. The SUM function is used for the Population column, and the AVG function is used for the LifeExpectancy column. The WHERE clause filters the results to include only data for Argentina.

A screenshot of a computer program

Description automatically generated

**4.** Using **ORDER BY**, **LIMIT**, what country has the **highest life expectancy**?

This syntax selects the country name (Name) and life expectancy (LifeExpectancy) from the country table, orders the results in descending order based on life expectancy, and then limits the result to only one row using LIMIT 1. This way, I would get the country with the highest life expectancy.

A screenshot of a computer program

Description automatically generated

**5.** Select **25 cities** around the world **that start with the letter 'F'** in a single SQL query:

This query selects all columns from the city table where the Name column starts with the letter 'F' and limits the result to 25 rows using the LIMIT clause.

A screenshot of a computer program

Description automatically generated

**6.** Create a SQL statement to **display columns Id, Name, Population** from the **city table** and **limit** results to first **10 rows only**.

A screenshot of a computer program

Description automatically generated

**7.** Create a SQL statement to find **only those cities** from city table whose **population is larger than 2000000**:

This query selects all columns (\*) from the city table where the Population column is greater than 2,000,000. I can customize the columns I would like to retrieve by replacing \* with the specific column names if needed.

A screenshot of a computer

Description automatically generated

**8.** To Create a SQL statement **to find all city names** from **city table** whose **name begins with “Be” prefix:**

This query uses the LIKE operator with the pattern 'Be%', where '%' is a wildcard that matches any sequence of characters. So, the query will retrieve all city names from the city table where the name starts with "Be".

A screenshot of a computer

Description automatically generated

**9.** Create a SQL statement to **find only those cities** from city table whose **population is between 500000-1000000:**

This syntax uses the SELECT statement to retrieve all columns (\*) from the city table. The WHERE clause is then used to filter the results, only selecting rows where the Population column is between 500,000 and 1,000,000.

A screenshot of a computer

Description automatically generated

**10.** Using **LEFT JOIN**, **ON**, what is the capital of Spain **(ESP)**?

This query retrieves the name of the capital city of Spain (ESP) by first selecting the "Capital" column value from the "country" table where the country code is 'ESP', and then using that value to look up the corresponding city name in the "city" table.

A screenshot of a computer

Description automatically generated

**11.** Using **LEFT JOIN**, **ON**, list all **the languages** spoken in the **'Southeast Asia' region:**

**A screenshot of a computer

Description automatically generated**

**12.** Create a SQL statement **to find a city** with the **lowest population** in the city table.

This query selects all columns from the city table, orders the result set by the population column in ascending order (ASC), and then limits the result set to only one row using LIMIT 1.

**A screenshot of a computer

Description automatically generated**