**Tasks**

1. **Using count, get the number of cities in the USA?**

This query will count the number of rows in the cities table where the country column is equal to 'USA', giving you the number of cities in the USA.

SELECT COUNT(Countrycode) FROM city WHERE Countrycode = 'USA';

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

This query will order the countries by life expectancy in descending order and then limit the result to only one row, which corresponds to the country with the highest life expectancy.

SELECT name, LifeExpectancy FROM country ORDER BY LifeExpectancy DESC

LIMIT 1;

1. **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.

SELECT \* FROM city WHERE Name LIKE 'F%'

LIMIT 25;

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

This query selects specific columns (Id, Name, Population) from the city table and restricts the result to the first 10 rows.

SELECT Id, Name, Population

FROM city

LIMIT 10;

1. **Create a SQL statement to display columns Id, Name, Population from the city table and limit results to rows 31-40.**

This query will return the columns Id, Name, and Population from the "city" table, limited to rows 31-40 based on the Id column.

**LIMIT 30, 10**: Specifies that we want to retrieve 10 rows starting from the 31st row. The first number (30) represents the offset (number of rows to skip), and the second number (10) represents the number of rows to retrieve.

SELECT Id, Name, Population FROM city ORDER BY Id LIMIT 30, 10;

1. **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 and filters the result to only include cities where the Population column is larger than 2,000,000.

SELECT \* FROM city WHERE Population >2000000;

1. **Create a SQL statement to find all city names from city table whose name begins with Be prefix.**

This query selects the Name column from the city table and filters the result to only include city names that begin with the prefix "Be" using the LIKE operator with the % wildcard character, which matches any sequence of characters.

SELECT Name FROMcity WHERE Name LIKE 'Be%';

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

This query selects all columns (\*) from the city table and filters the result to only include cities where the Population column falls within the range of 500,000 to 1,000,000 using the BETWEEN operator. The BETWEEN operator is inclusive of both endpoints, so it will include cities with a population of exactly 500,000 or 1,000,000 in the result set.

SELECT \* FROM city WHERE Population BETWEEN 500000 AND 1000000;

1. **Create a SQL statement to display all cities from the city table sorted by Name in ascending order.**

This query will return all cities from the "city" table, sorted alphabetically by name in ascending order.

SELECT \* FROM city ORDER BY Name ASC;

1. **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 by population in ascending order (ASC), which will put the cities with the lowest population at the top, and limits the result to only one row using LIMIT 1. Therefore, it will return the city with the lowest population.

SELECT \* FROM city ORDER BY Population ASC

LIMIT1;

1. **Create a SQL statement to find a country with the largest population in the country table.**

This SQL query efficiently finds and retrieves the country with the highest population from the **country** table by sorting the rows based on the population column in descending order and then limiting the result to only the first row.

SELECT \* FROM country ORDER BY Population DESC LIMIT 1;