

Subir Kumar Roy  
Data  
Yusuf Satilmis

01/2024

**World DB**

-MySQL-

Table of Contents

[#TASK 1 - Using count, get the number of cities in the USA​ 2](#_Toc156480450)

[#TASK 2 - Find out what the population and average life expectancy for people in Argentina (ARG) is.​ 2](#_Toc156480451)

[#TASK 3 - Using ORDER BY, LIMIT, what country has the highest life expectancy? 3](#_Toc156480452)

[#TASK 4 - Select 25 cities around the world that start with the letter 'F' in a single SQL query.​ 3](#_Toc156480453)

[#TASK 5 - Create a SQL statement to display columns Id, Name, Population from the city table and limit results to first 10 rows only.​ 4](#_Toc156480454)

[#TASK 6 - Create a SQL statement to find only those cities from city table whose population is larger than 2000000.​ 4](#_Toc156480455)

[#TASK 7 - Create a SQL statement to find all city names from city table whose name begins with “Be” prefix.​ 5](#_Toc156480456)

[#TASK 8 - Create a SQL statement to find only those cities from city table whose population is between 500000-1000000. 5](#_Toc156480457)

[#TASK 9 - Create a SQL statement to find a city with the lowest population in the city table. 6](#_Toc156480458)

[#BONUS 6](#_Toc156480459)

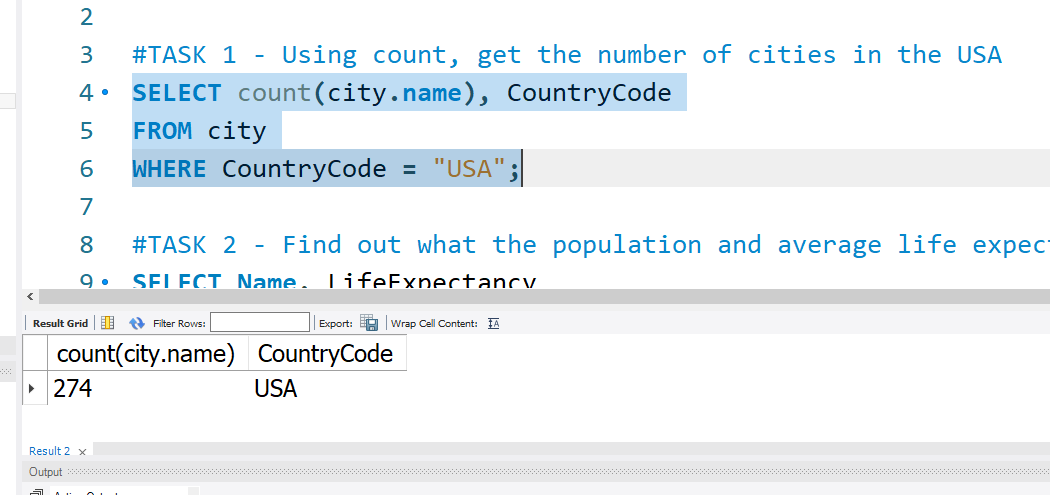
[#Create a SQL statement to find the capital of Spain (ESP).​ 6](#_Toc156480460)

[#Create a SQL statement to list all the languages spoken in the Caribbean region.​ 7](#_Toc156480461)

[#Create a SQL statement to find all cities from the Europe continent. 7](#_Toc156480462)

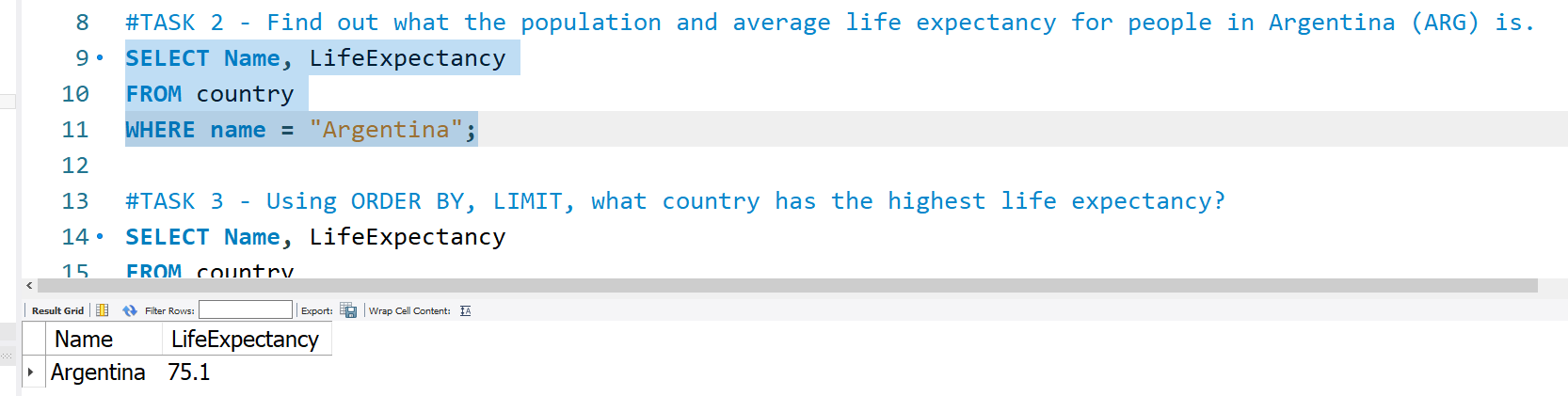
[#TASK 10 8](#_Toc156480463)

# #TASK 1 - Using count, get the number of cities in the USA​



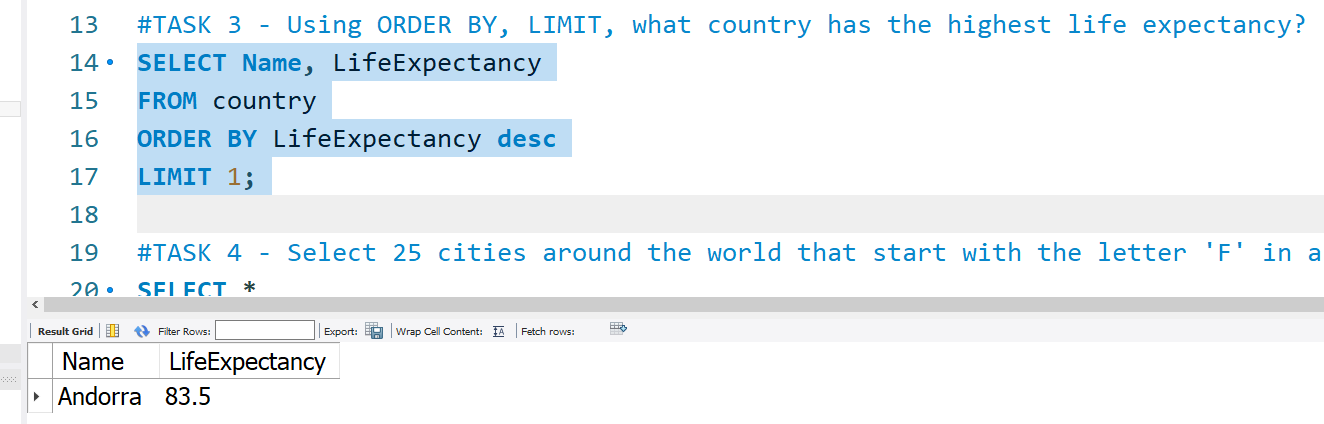
This SQL statement counts the number of cities in the "city" table with a specific CountryCode ("USA"). It selects and displays the count of cities and their corresponding CountryCode from the table, providing a summary of the cities in the United States.

# #TASK 2 - Find out what the population and average life expectancy for people in Argentina (ARG) is.​



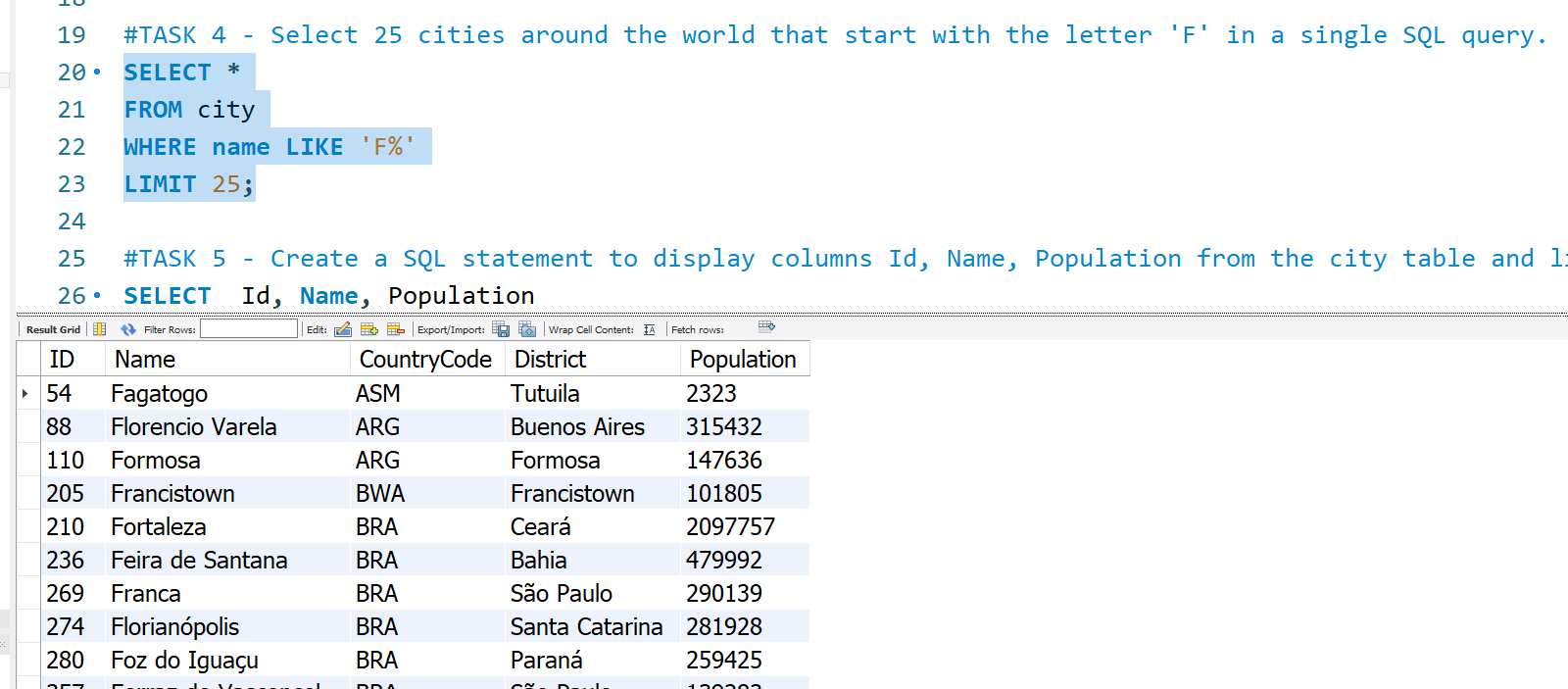
This SQL statement retrieves the "Name" and "LifeExpectancy" columns from the "country" table where the country name is "Argentina".

# #TASK 3 - Using ORDER BY, LIMIT, what country has the highest life expectancy?



The above SQL query selects the "Name" and "LifeExpectancy" columns from the "country" table, calls the result in descending order based on the "LifeExpectancy" column, and retrieves only the highest record.

# #TASK 4 - Select 25 cities around the world that start with the letter 'F' in a single SQL query.​



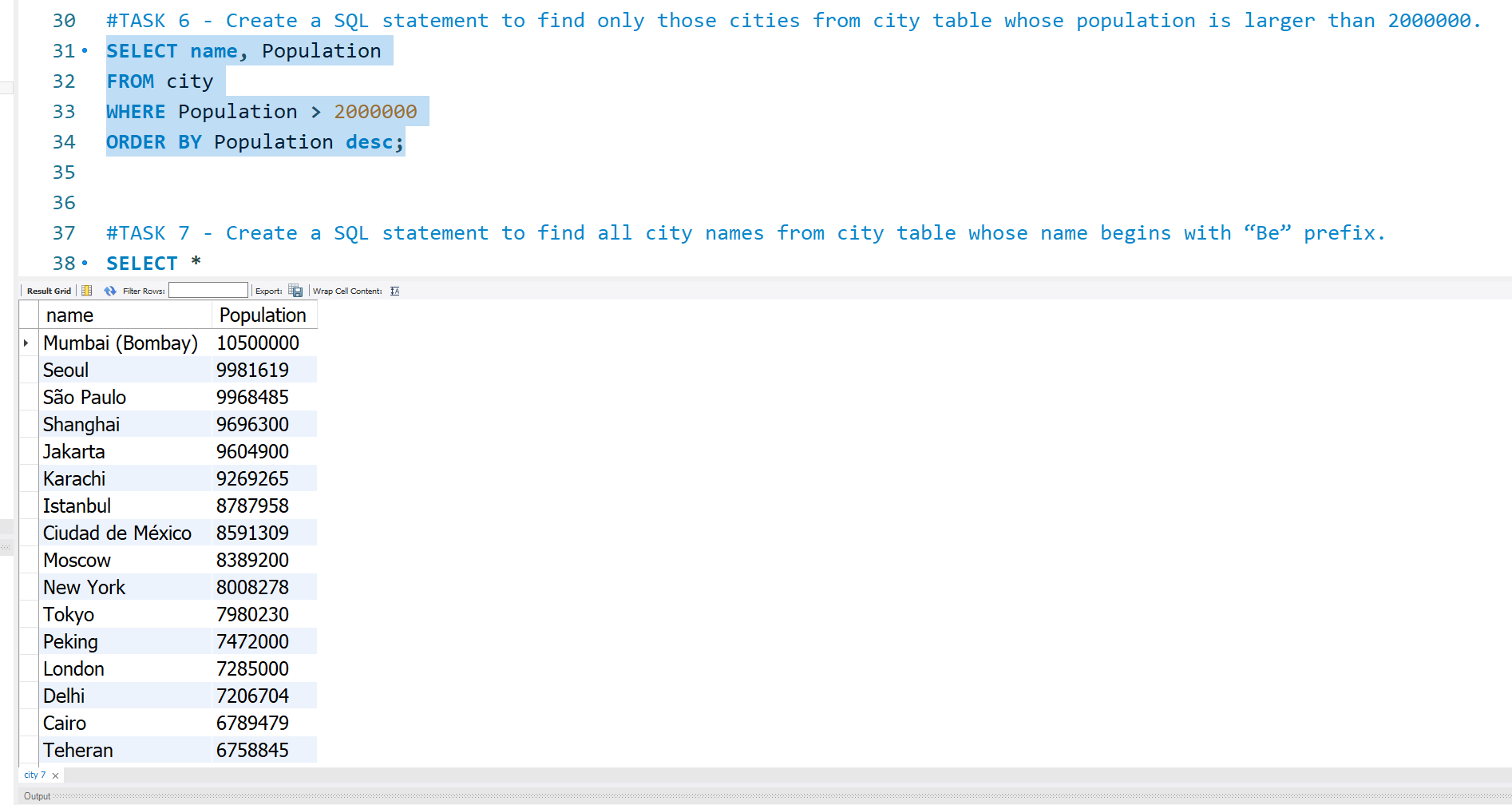
The SQL query retrieves all columns from the "city" table where the name of the city begins with the letter 'F'. The '%' is a wildcard character representing any number of characters that can follow 'F'. The LIMIT 25 ensures that only the first 25 matching rows are returned.

# #TASK 5 - Create a SQL statement to display columns Id, Name, Population from the city table and limit results to first 10 rows only.​



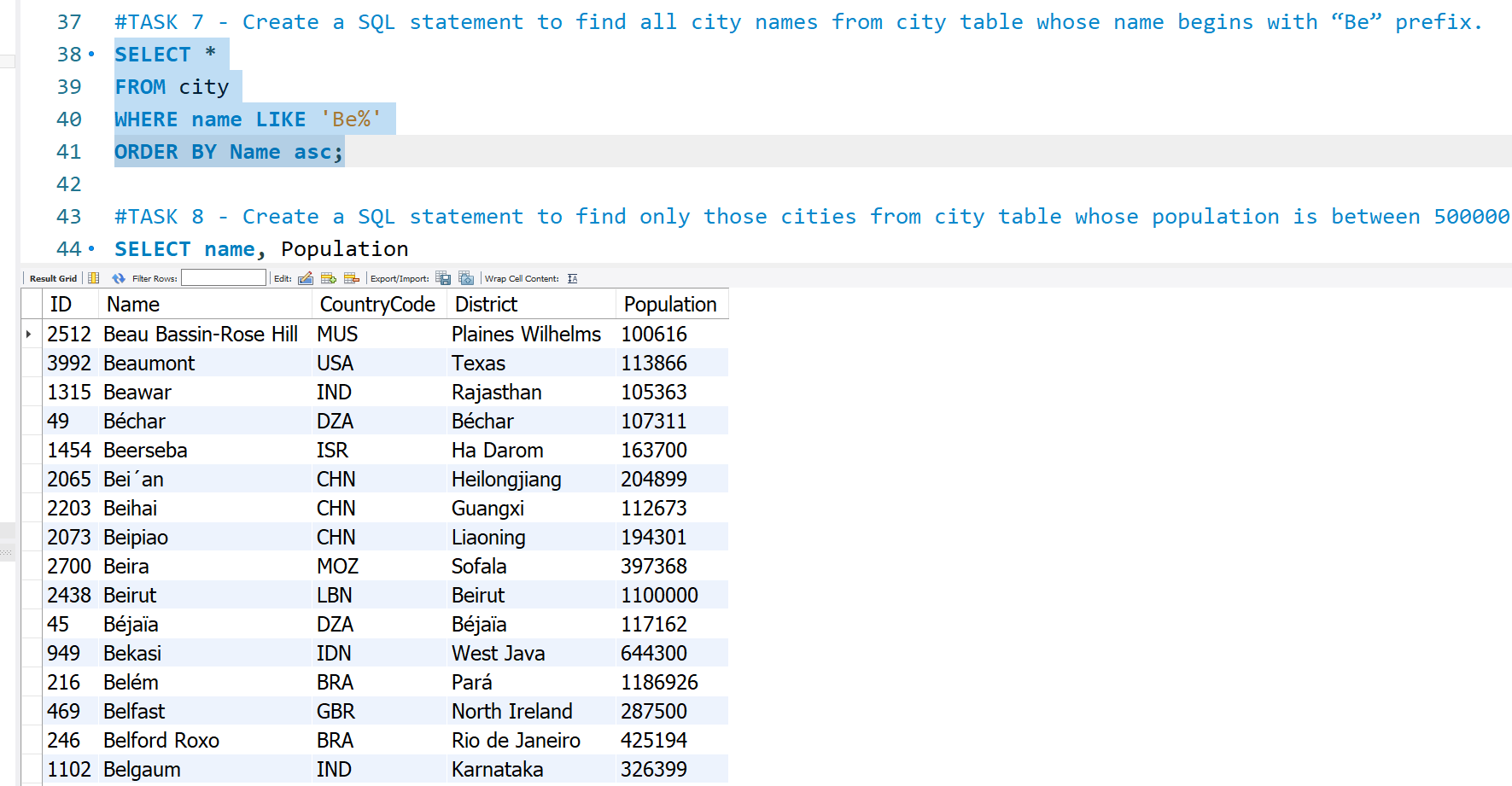
This SQL query retrieves data from the "city" table, specifically the columns "Id," "Name," and "Population" for the first 10 rows. The "LIMIT 10" clause restricts the output to only show a maximum of 10 results.

# #TASK 6 - Create a SQL statement to find only those cities from city table whose population is larger than 2000000.​



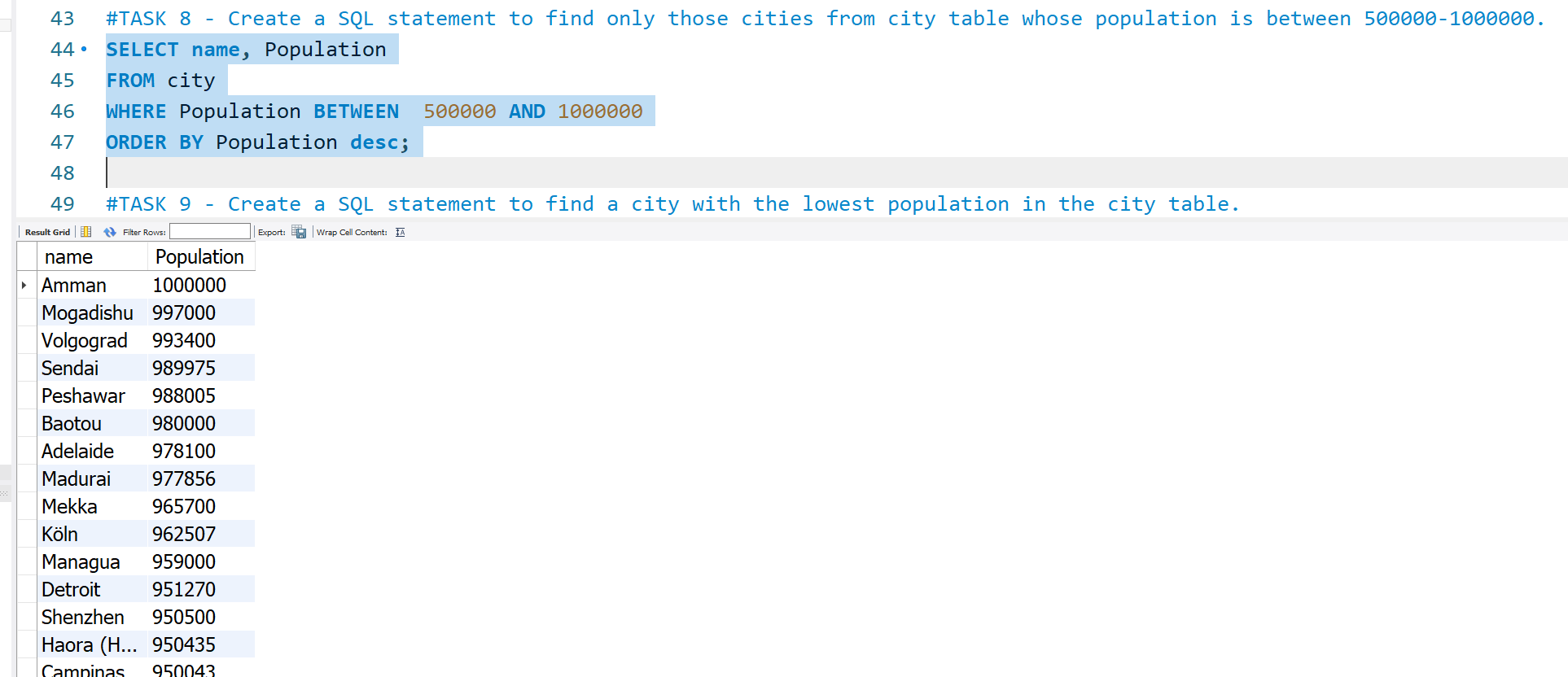
This SQL statement retrieves the names and populations of cities from a table named "city" where the population is greater than 2 million. The results are sorted in descending order based on population. The SELECT clause specifies the columns to be retrieved, the WHERE clause filters the rows based on population criteria, and the ORDER BY clause arranges the results in descending order of population.

# #TASK 7 - Create a SQL statement to find all city names from city table whose name begins with “Be” prefix.​



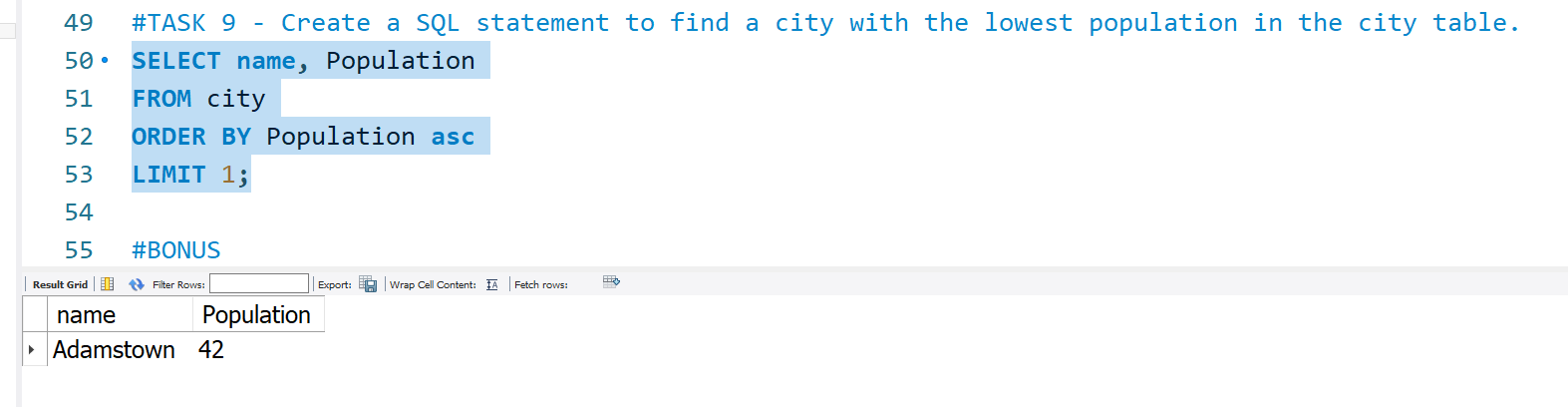
This SQL query retrieves all columns from the "city" table where the "name" column starts with 'Be'. The '%' is a wildcard representing any characters that may follow 'Be'. The results are then ordered in ascending order based on the "Name" column.

# #TASK 8 - Create a SQL statement to find only those cities from city table whose population is between 500000-1000000.



This SQL query retrieves the names and populations of cities with populations ranging from 500,000 to 1,000,000, sorted in descending order by population. It utilizes the SELECT statement to specify the columns (name, Population), the FROM clause to identify the 'city' table, the WHERE clause to filter populations within the specified range, and the ORDER BY clause to arrange the results in descending order based on population.

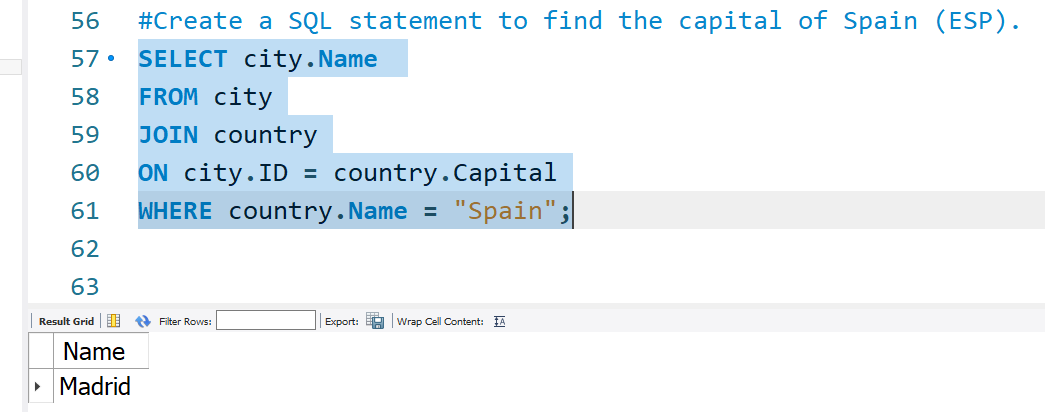
# #TASK 9 - Create a SQL statement to find a city with the lowest population in the city table.



This SQL query retrieves the "name" and "Population" columns from the "city" table, orders the results in ascending order based on the "Population" column, and limits the output to the first row.

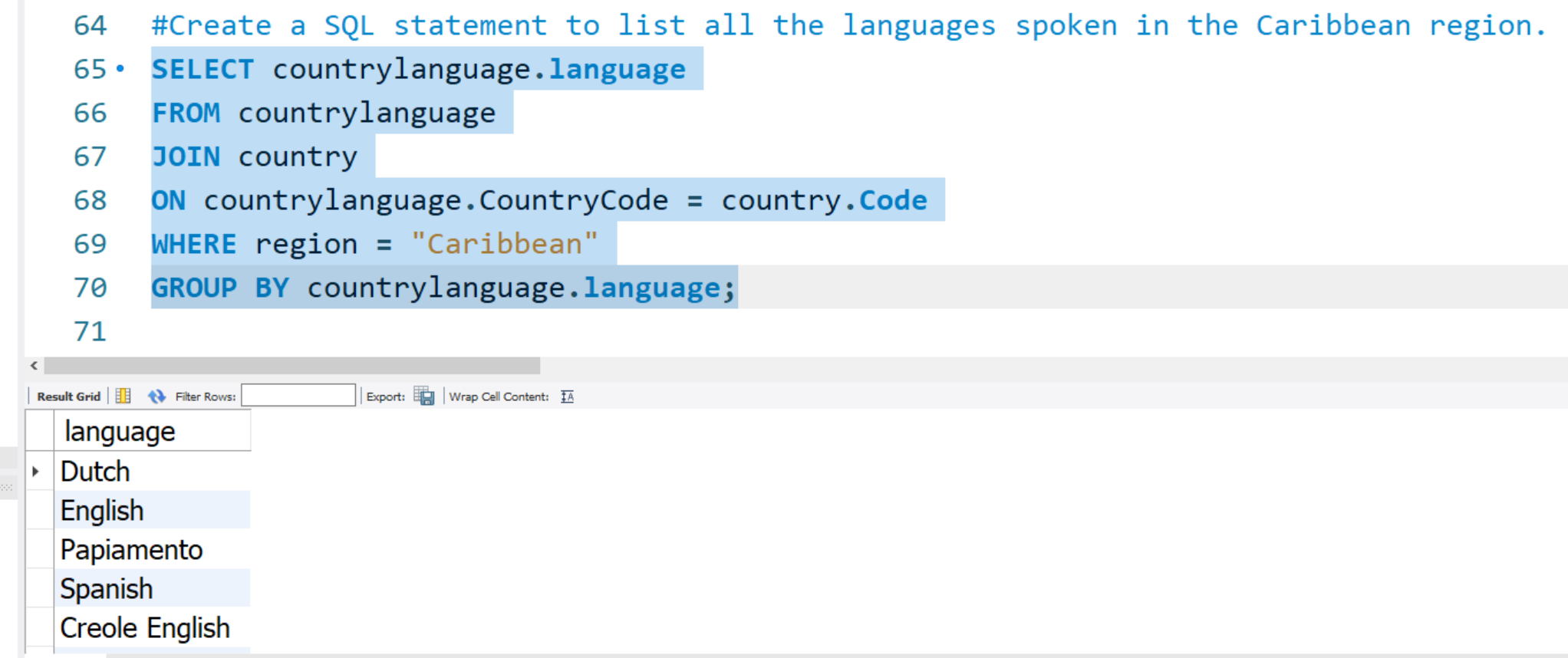
# #BONUS

## #Create a SQL statement to find the capital of Spain (ESP).​



The SQL statement retrieves the name of the capital city of Spain from the "city" table. It uses a JOIN operation with the "country" table, linking records where the city's ID matches the country's capital. The WHERE clause filters results to only include records where the country's name is "Spain." The query aims to fetch the capital city name associated with Spain from the database.

## #Create a SQL statement to list all the languages spoken in the Caribbean region.​

]

This SQL query retrieves distinct languages spoken in Caribbean countries by joining the "countrylanguage" and "country" tables based on the country code. The condition "WHERE region = 'Caribbean'" filters countries in the Caribbean region. The result is grouped by language using "GROUP BY," ensuring each language appears only once. The SELECT clause extracts the "language" column from the "countrylanguage" table, providing a list of unique languages spoken in Caribbean countries.

Another way would have been to use Distinct with Select on the first line.

## #Create a SQL statement to find all cities from the Europe continent.



The SQL statement retrieves data from the "city" table, selecting the columns "Name" and "Continent." It joins the "city" table with the "country" table based on matching country codes. The condition "WHERE Continent = 'Europe'" filters the results to include only cities from European countries. The query aims to display city names and their corresponding continents for cities located in Europe, utilizing a join operation between the "city" and "country" tables.

# #TASK 10

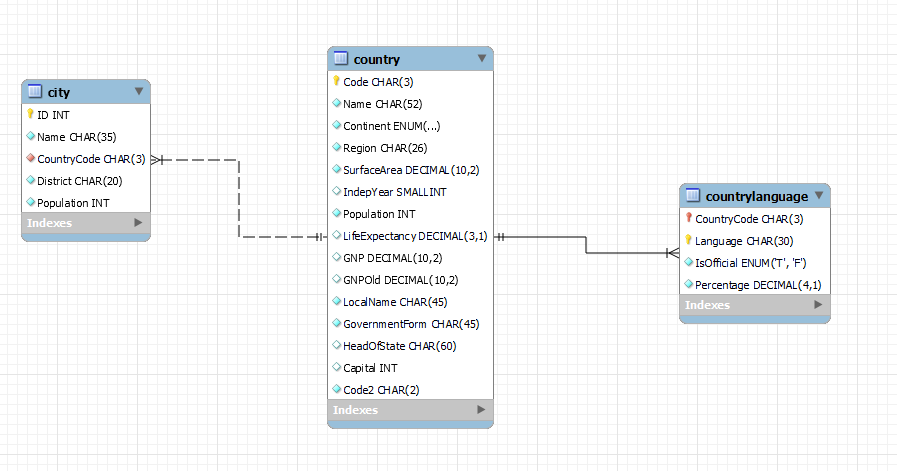
#Identify the primary key in country table.​ – Code is the Primary Key

#Identify the primary key in city table.​ – ID is the Primary Key

#Identify the primary key in countrylanguage table.​ – CountryCode is the Primary Key

#Identify the foreign key in city table.​ - CountryCode is Foreign Key

#Identify the foreign key in countrylanguage table. - CountryCode is Foreign Key

****

FOREIGN KEY

FOREIGN KEY

PRIMARY KEY

PRIMARY KEY

PRIMARY KEY