# MYSQL: TASK EXAMPLES

Please note the following in reference to syntax used in the sql script when using mysql to run results, these are just a few used within sql and are not limited in order to get a result:

- ASC Sorts the result set in ascending order
- AVG The AVG() function returns the average value of a numeric column
- BETWEEN Selects values within a given range
- COUNT The COUNT() function returns the number of rows that matches a specified criterion.
- DESC Sorts the result set in descending order
- FOREIGN KEY A constraint that is a key used to link two tables together
- FROM Specifies which table to select or delete data from
- GROUP BY Groups the result set (used with aggregate functions: COUNT, MAX, MIN, SUM, AVG)
- INNER JOIN Returns rows that have matching values in both tables
- IS NOT NULL Tests for non-empty values
- LIKE Searches for a specified pattern in a column
- LIMIT Specifies the number of records to return in the result set
- NOT Only includes rows where a condition is not true
- NOT NULL A constraint that enforces a column to not accept NULL values
- OR Includes rows where either condition is true
- ORDER BY Sorts the result set in ascending or descending order
- OUTER JOIN Returns all rows when there is a match in either left table or right table
- PRIMARY KEY A constraint that uniquely identifies each record in a database table
- RIGHT JOIN Returns all rows from the right table, and the matching rows from the left table
- Left JOIN The LEFT JOIN command returns all rows from the left table, and the matching rows from the right table. The result is NULL from the right side, if there is no match
- SELECT Selects data from a database
- SUM The SUM() function returns the total sum of a numeric column
- TOP Specifies the number of records to return in the result set
- WHERE Filters a result set to include only records that fulfil a specified condition

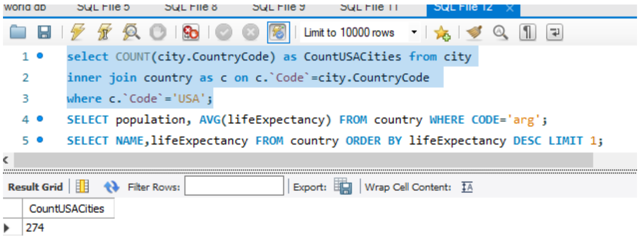
## Task 1:

#Using count, get the number of cities in the USA ?

select COUNT(city.CountryCode) as CountUSACities from city

inner join country as c on c.`Code`=city.CountryCode

where c.`Code`='USA';



```
select COUNT(city.CountryCode) as CountUSACities from city
inner join country as c on c.`Code`=city.CountryCode
where c.`Code`='USA';
SELECT population, AVG(lifeExpectancy) FROM country WHERE CODE='arg';
SELECT NAME,lifeExpectancy FROM country ORDER BY lifeExpectancy DESC LIMIT 1;
```

| CountUSACities |
|---|
| 274 |

## Task 2:

# Find out what the population and average life expectancy for people in Argentina (ARG) is?

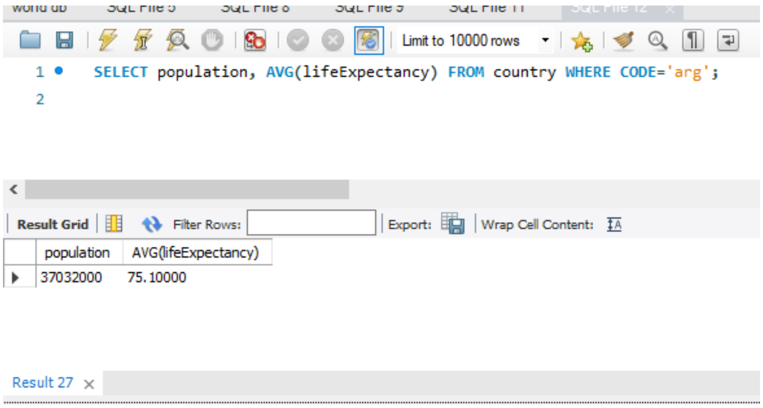
SELECT population, AVG(lifeExpectancy) FROM country WHERE CODE='arg';

(37032000, 75.10000)

In order to answer this query we need to select the country database, select population and the life expectancy rows (which we need to obtain the average number of births), also highlighting the condition of the country(where), in this case which is Argentina (ARG).

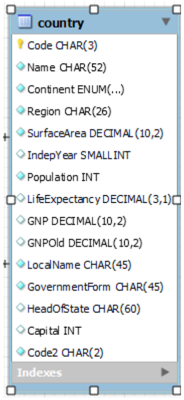
We can run: select * from country; in sql to bring up the full data base to show the column headers and rows.

Note: sql share same word features a excel, examples would be SUM, AVG, Count.



```
SELECT population, AVG(lifeExpectancy) FROM country WHERE CODE='arg';
```

| population | AVG(lifeExpectancy) |
|---|---|
| 37032000 | 75.10000 |



country

Code CHAR(3)
Name CHAR(52)
Continent ENUM(...)
Region CHAR(26)
SurfaceArea DECIMAL(10,2)
IndepYear SMALLINT
Population INT
LifeExpectancy DECIMAL(3,1)
GNP DECIMAL(10,2)
GNPOld DECIMAL(10,2)
LocalName CHAR(45)
GovernmentForm CHAR(45)
HeadOfState CHAR(60)
Capital INT
Code2 CHAR(2)

Task 3:

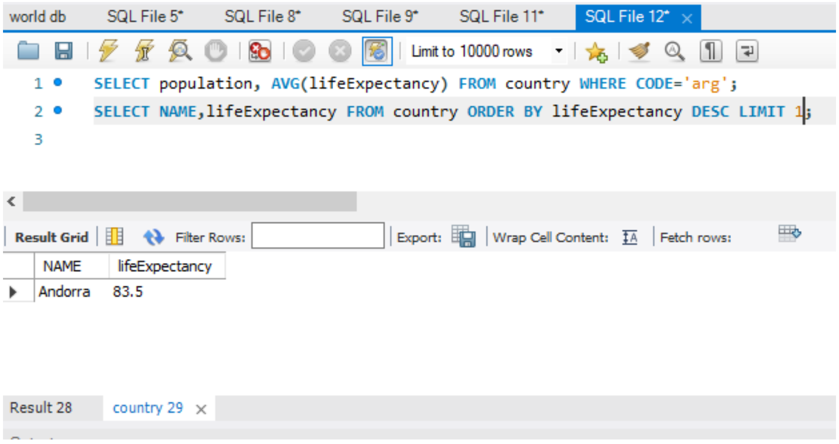
#Using ORDER BY, LIMIT, what country has the highest life expectancy?

In order to obtain this result we access the country data base again selecting the columns we need and placing the condition of “order” and “desc”, this will generate a list- we then “limit” that list to 1, in a descending order.

SELECT * FROM country;

SELECT NAME,lifeExpectancy FROM country ORDER BY lifeExpectancy DESC LIMIT 1 ;

(Andorra, 83.5)



```
SELECT population, AVG(lifeExpectancy) FROM country WHERE CODE='arg';
SELECT NAME,lifeExpectancy FROM country ORDER BY lifeExpectancy DESC LIMIT 1;
```

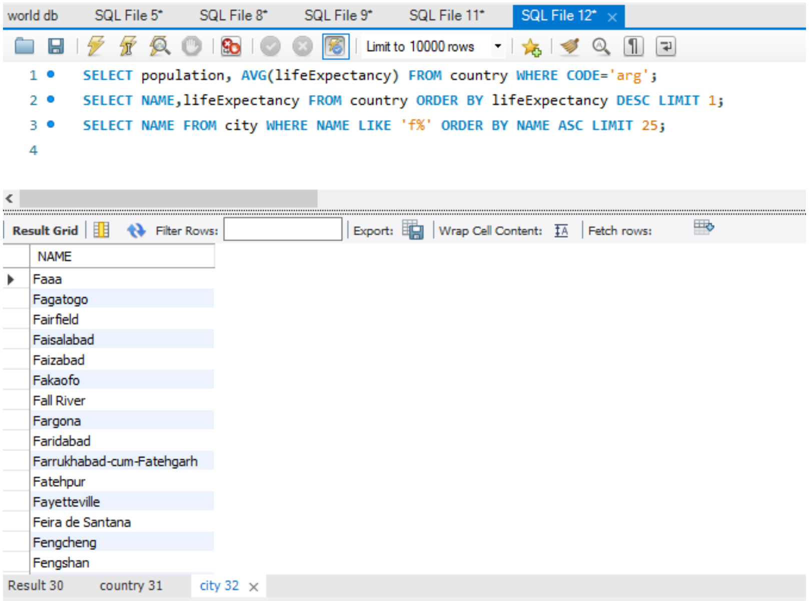
| NAME | lifeExpectancy |
|---|---|
| Andorra | 83.5 |

Task 4:

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

For this we need to select the city data base with the condition returning cites starting with F ( % allows represents zero, one, or multiple characters).

SELECT NAME FROM city WHERE NAME LIKE 'f%' ORDER BY NAME ASC LIMIT 25;



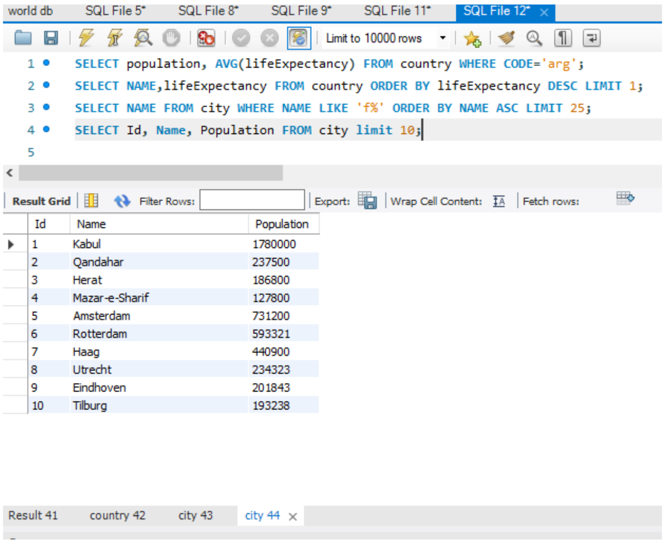
```
SELECT population, AVG(lifeExpectancy) FROM country WHERE CODE='arg';
SELECT NAME,lifeExpectancy FROM country ORDER BY lifeExpectancy DESC LIMIT 1;
SELECT NAME FROM city WHERE NAME LIKE 'f%' ORDER BY NAME ASC LIMIT 25;
```

| NAME |
|---|
| Faaa |
| Fagatogo |
| Fairfield |
| Faisalabad |
| Faizabad |
| Fakaofo |
| Fall River |
| Fargona |
| Faridabad |
| Farrukhabad-cum-Fatehgarh |
| Fatehpur |
| Fayetteville |
| Feira de Santana |
| Fengcheng |
| Fengshan |

Task 5:

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

SELECT Id, Name, Population FROM city limit 10;



```
SELECT population, AVG(lifeExpectancy) FROM country WHERE CODE='arg';
SELECT NAME,lifeExpectancy FROM country ORDER BY lifeExpectancy DESC LIMIT 1;
SELECT NAME FROM city WHERE NAME LIKE 'f%' ORDER BY NAME ASC LIMIT 25;
SELECT Id, Name, Population FROM city limit 10;
```

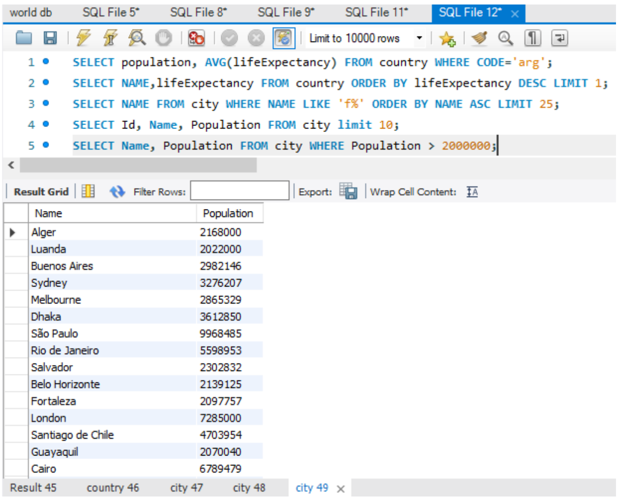
| Id | Name | Population |
|---|---|---|
| 1 | Kabul | 1780000 |
| 2 | Qandahar | 237500 |
| 3 | Herat | 186800 |
| 4 | Mazar-e-Sharif | 127800 |
| 5 | Amsterdam | 731200 |
| 6 | Rotterdam | 593321 |
| 7 | Haag | 440900 |
| 8 | Utrecht | 234323 |
| 9 | Eindhoven | 201843 |
| 10 | Tilburg | 193238 |

Task 6:

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

The ">" function allows us to return results over the required amount.

SELECT Name, Population FROM city WHERE Population > 2000000;



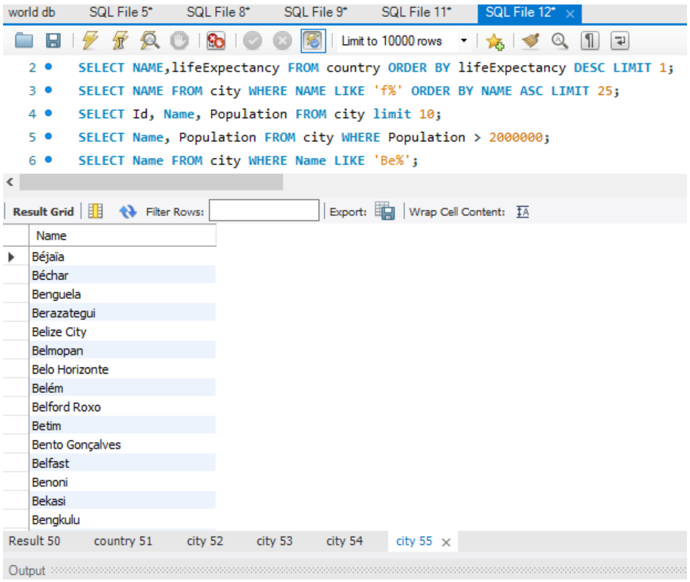
```sql
SELECT population, AVG(lifeExpectancy) FROM country WHERE CODE='arg';
SELECT NAME,lifeExpectancy FROM country ORDER BY lifeExpectancy DESC LIMIT 1;
SELECT NAME FROM city WHERE NAME LIKE 'f%' ORDER BY NAME ASC LIMIT 25;
SELECT Id, Name, Population FROM city limit 10;
SELECT Name, Population FROM city WHERE Population > 2000000;
```

| Name | Population |
|---|---|
| Alger | 2168000 |
| Luanda | 2022000 |
| Buenos Aires | 2982146 |
| Sydney | 3276207 |
| Melbourne | 2865329 |
| Dhaka | 3612850 |
| São Paulo | 9968485 |
| Rio de Janeiro | 5598953 |
| Salvador | 2302832 |
| Belo Horizonte | 2139125 |
| Fortaleza | 2097757 |
| London | 7285000 |
| Santiago de Chile | 4703954 |
| Guayaquil | 2070040 |
| Cairo | 6789479 |

## Task 7:

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

SELECT Name FROM city WHERE Name LIKE 'Be%'



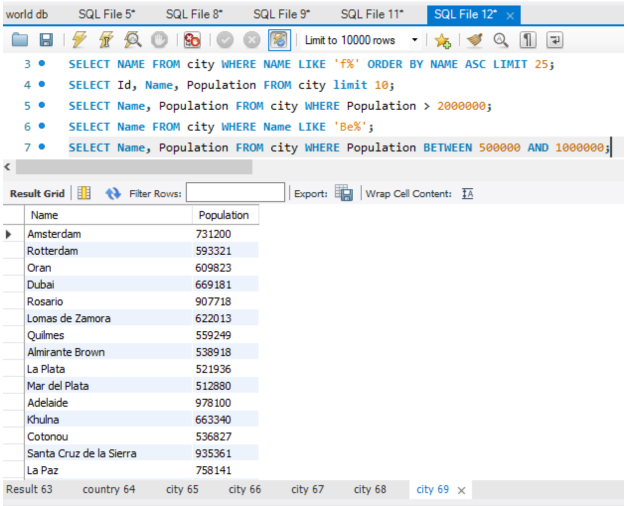
```sql
SELECT NAME,lifeExpectancy FROM country ORDER BY lifeExpectancy DESC LIMIT 1;
SELECT NAME FROM city WHERE NAME LIKE 'f%' ORDER BY NAME ASC LIMIT 25;
SELECT Id, Name, Population FROM city limit 10;
SELECT Name, Population FROM city WHERE Population > 2000000;
SELECT Name FROM city WHERE Name LIKE 'Be%';
```

| Name |
|---|
| Béjaïa |
| Béchar |
| Benguela |
| Berazategui |
| Belize City |
| Belmopan |
| Belo Horizonte |
| Belém |
| Belford Roxo |
| Betim |
| Bento Gonçalves |
| Belfast |
| Benoni |
| Bekasi |
| Bengkulu |

## Task 8:

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

SELECT Name, Population FROM city WHERE Population BETWEEN 500000 AND 1000000;



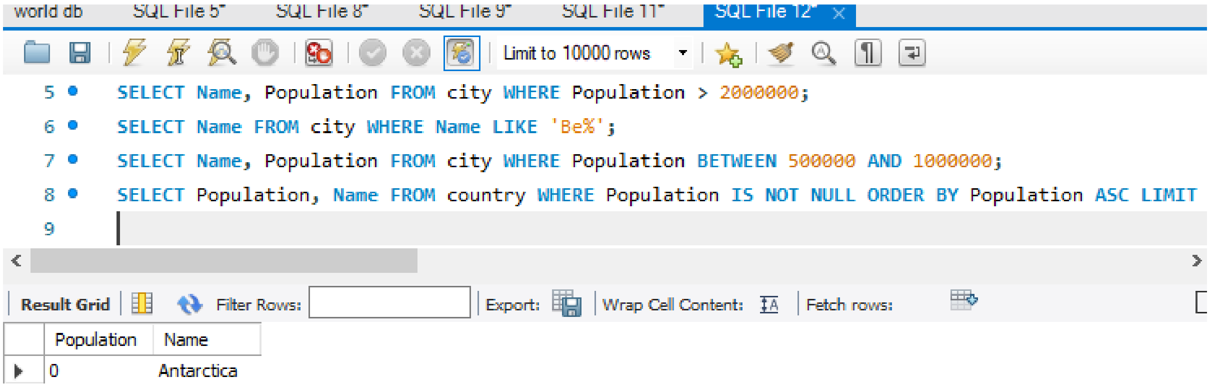
```
SELECT NAME FROM city WHERE NAME LIKE 'f%' ORDER BY NAME ASC LIMIT 25;
SELECT Id, Name, Population FROM city limit 10;
SELECT Name, Population FROM city WHERE Population > 2000000;
SELECT Name FROM city WHERE Name LIKE 'Be%';
SELECT Name, Population FROM city WHERE Population BETWEEN 500000 AND 1000000;
```

| Name | Population |
|---|---|
| Amsterdam | 731200 |
| Rotterdam | 593321 |
| Oran | 609823 |
| Dubai | 669181 |
| Rosario | 907718 |
| Lomas de Zamora | 622013 |
| Quilmes | 559249 |
| Almirante Brown | 538918 |
| La Plata | 521936 |
| Mar del Plata | 512880 |
| Adelaide | 978100 |
| Khulna | 663340 |
| Cotonou | 536827 |
| Santa Cruz de la Sierra | 935361 |
| La Paz | 758141 |

Task 9:

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

SELECT Population, Name FROM country WHERE Population IS NOT NULL ORDER BY Population ASC LIMIT 1;



```
SELECT Name, Population FROM city WHERE Population > 2000000;
SELECT Name FROM city WHERE Name LIKE 'Be%';
SELECT Name, Population FROM city WHERE Population BETWEEN 500000 AND 1000000;
SELECT Population, Name FROM country WHERE Population IS NOT NULL ORDER BY Population ASC LIMIT
```

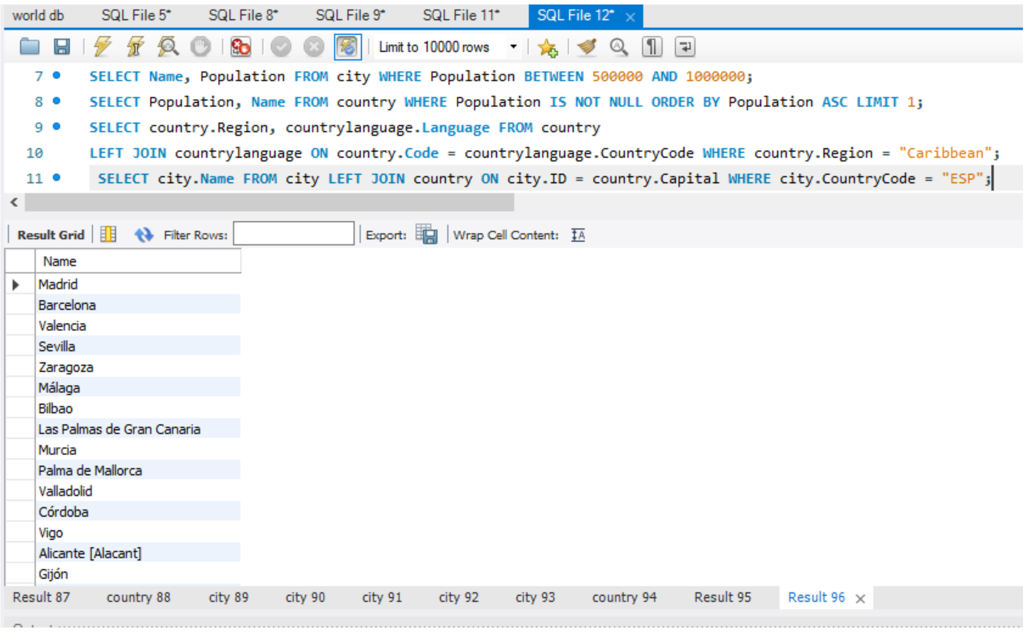
| Population | Name |
|---|---|
| 0 | Antarctica |

BONUS

Task 10:

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

SELECT city.Name FROM city LEFT JOIN country ON city.ID = country.Capital WHERE city.CountryCode = "ESP";

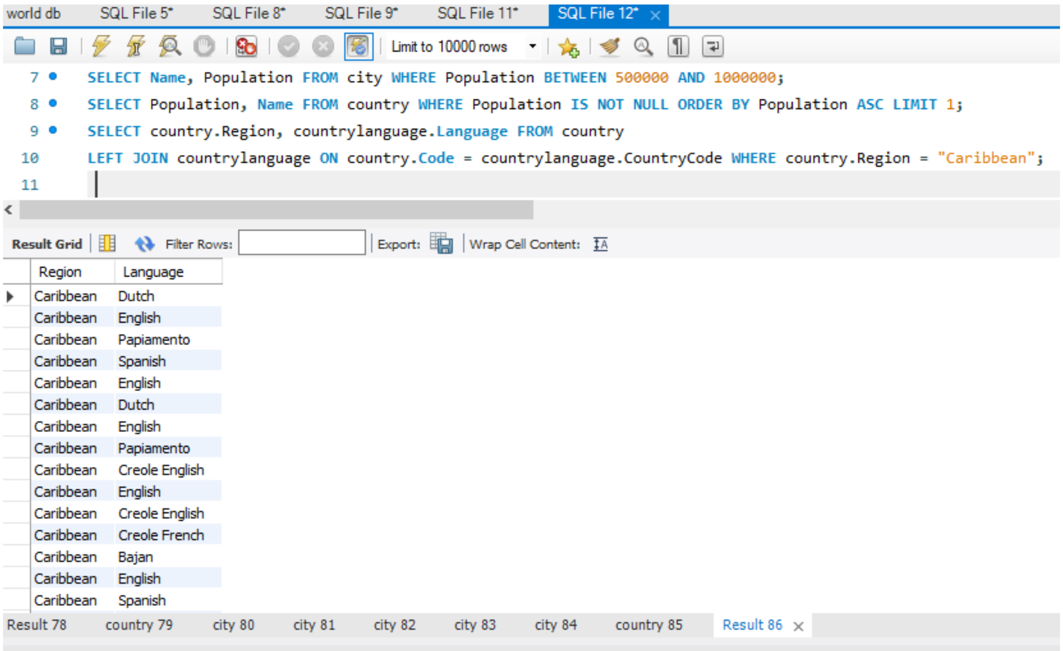


```sql
SELECT Name, Population FROM city WHERE Population BETWEEN 500000 AND 1000000;
SELECT Population, Name FROM country WHERE Population IS NOT NULL ORDER BY Population ASC LIMIT 1;
SELECT country.Region, countrylanguage.Language FROM country
LEFT JOIN countrylanguage ON country.Code = countrylanguage.CountryCode WHERE country.Region = "Caribbean";
 SELECT city.Name FROM city LEFT JOIN country ON city.ID = country.Capital WHERE city.CountryCode = "ESP";
```

| Name |
|---|
| Madrid |
| Barcelona |
| Valencia |
| Sevilla |
| Zaragoza |
| Málaga |
| Bilbao |
| Las Palmas de Gran Canaria |
| Murcia |
| Palma de Mallorca |
| Valladolid |
| Córdoba |
| Vigo |
| Alicante [Alacant] |
| Gijón |

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

SELECT country.Region, countrylanguage.Language FROM country LEFT JOIN countrylanguage ON country.Code = countrylanguage.CountryCode WHERE country.Region = "Caribbean";

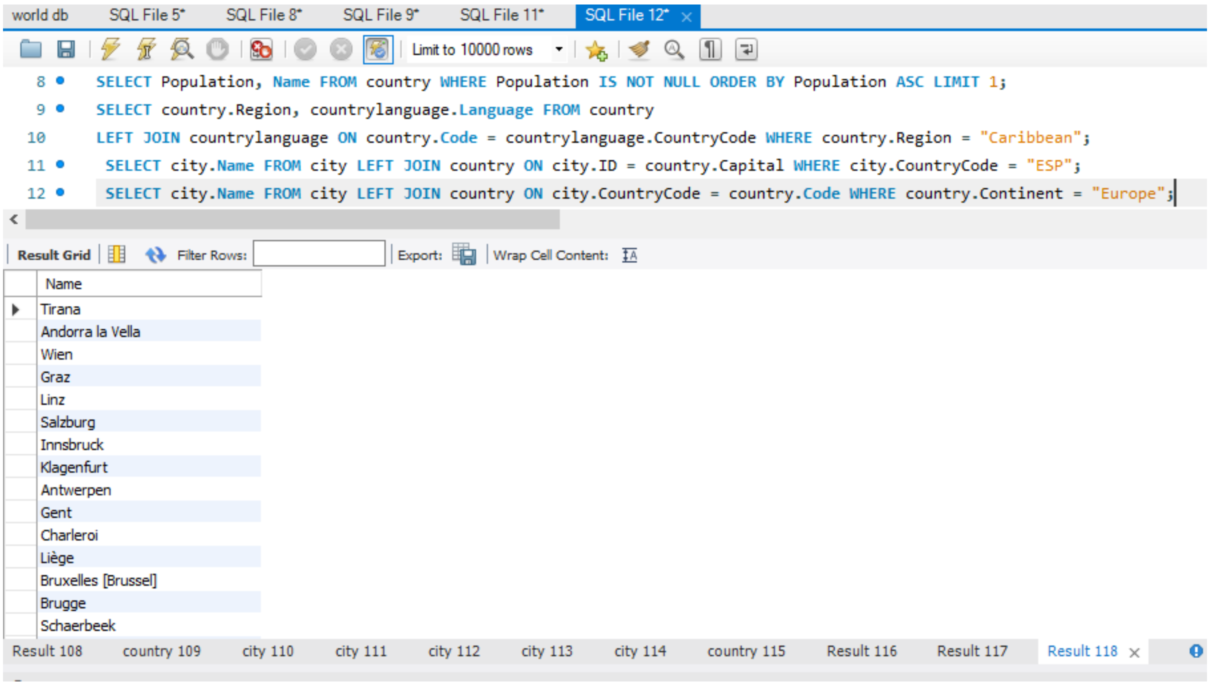


```sql
SELECT Name, Population FROM city WHERE Population BETWEEN 500000 AND 1000000;
SELECT Population, Name FROM country WHERE Population IS NOT NULL ORDER BY Population ASC LIMIT 1;
SELECT country.Region, countrylanguage.Language FROM country
LEFT JOIN countrylanguage ON country.Code = countrylanguage.CountryCode WHERE country.Region = "Caribbean";
```

| Region | Language |
|---|---|
| Caribbean | Dutch |
| Caribbean | English |
| Caribbean | Papiamento |
| Caribbean | Spanish |
| Caribbean | English |
| Caribbean | Dutch |
| Caribbean | English |
| Caribbean | Papiamento |
| Caribbean | Creole English |
| Caribbean | English |
| Caribbean | Creole English |
| Caribbean | Creole French |
| Caribbean | Bajan |
| Caribbean | English |
| Caribbean | Spanish |

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

SELECT city.Name FROM city LEFT JOIN country ON city.CountryCode = country.Code WHERE country.Continent = "Europe";

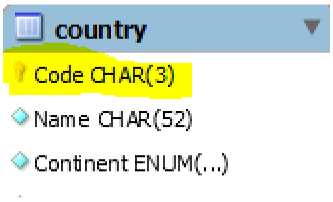


```sql
SELECT Population, Name FROM country WHERE Population IS NOT NULL ORDER BY Population ASC LIMIT 1;
SELECT country.Region, countrylanguage.Language FROM country
LEFT JOIN countrylanguage ON country.Code = countrylanguage.CountryCode WHERE country.Region = "Caribbean";
 SELECT city.Name FROM city LEFT JOIN country ON city.ID = country.Capital WHERE city.CountryCode = "ESP";
 SELECT city.Name FROM city LEFT JOIN country ON city.CountryCode = country.Code WHERE country.Continent = "Europe";
```

```
Name
Tirana
Andorra la Vella
Wien
Graz
Linz
Salzburg
Innsbruck
Klagenfurt
Antwerpen
Gent
Charleroi
Liège
Bruxelles [Brussel]
Brugge
Schaerbeek
```

Task 11:

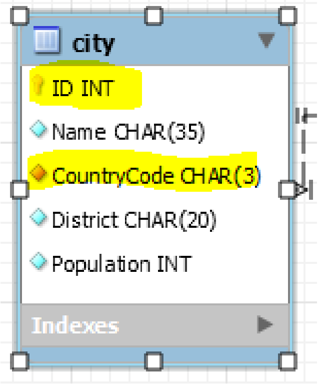
Identify the primary key in country table = **The Code CHAR(3)**



country
Code CHAR(3)
Name CHAR(52)
Continent ENUM(...)

Identify the primary key in city table = **The ID INT**

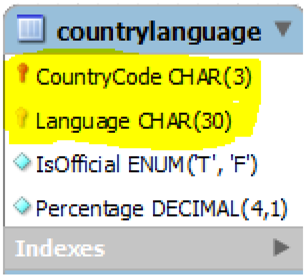
Identify the foreign key in city table.= **The CountryCode (3)**



city
ID INT
Name CHAR(35)
CountryCode CHAR(3)
District CHAR(20)
Population INT
Indexes

Identify the primary key in countrylanguage table = Lanuage CHAR(30)

Identify the foreign key in countrylanguage table = CountryCode(3)



countrylanguage

CountryCode CHAR(3)

Language CHAR(30)

IsOfficial ENUM('T', 'F')

Percentage DECIMAL(4,1)