MySQL Assignment

Alex Thompson

04/11/2024

JustIT

Data Technician Bootcamp

Table of Contents

Task 1: List the different type of relationships in relational databases and provide examples.

1. **One-to-one relationship**: A relationship where each record in one table is linked to one and only one record in another table. For example, a person can have only one passport, and a passport belongs to only one person.

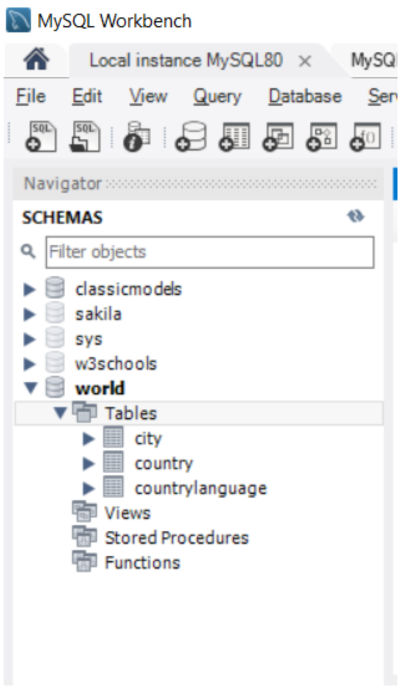
2. **One-to-many relationship**: A relationship where each record in one table is linked to one or more records in another table. For example, a customer can place many orders, and an order belongs to only one customer.

3. **Many-to-many relationship**: A relationship where each record in one table is linked to one or more records in another table, and vice versa. For example, a student can enrol in many courses, and a course can have many students. This type of relationship requires a bridge table to store the links between the two tables.

Task 2: What is Normalization and why is it important to database development?

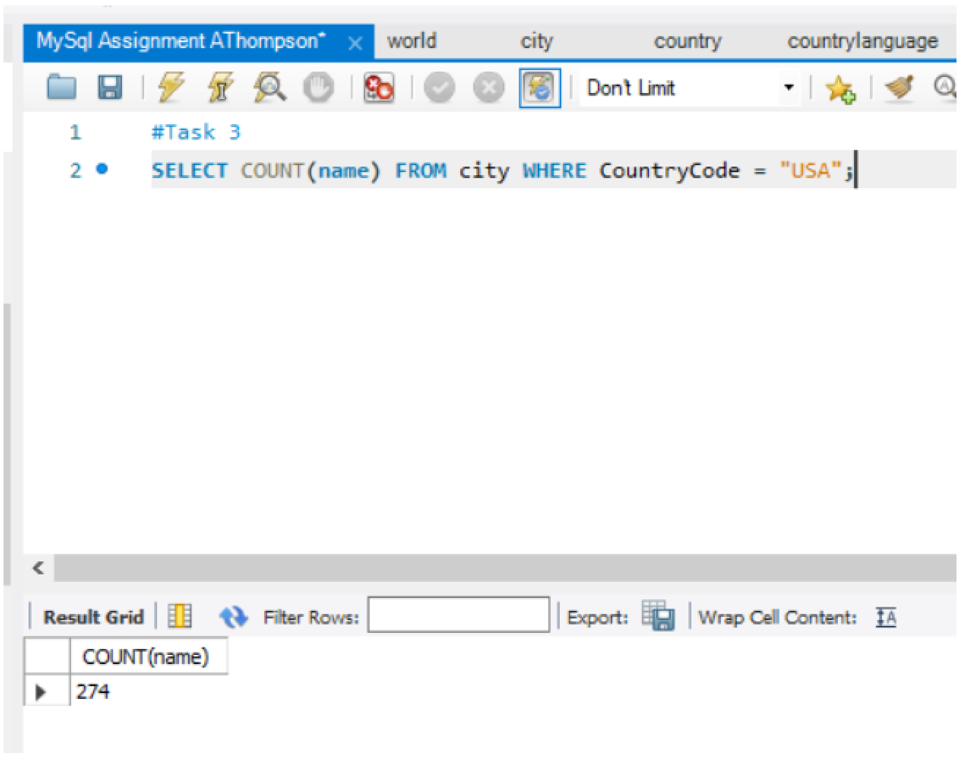
Data normalization is the process of reducing data redundancy, duplication and improving integrity within a table or database. Data normalization ensures that your data remains clean, consistent, and error-free by breaking it into smaller tables and linking them through relationships. This also works to improve database performance and protect against errors appearing within the database as data is changed or updated.

Set Up:



SQL Tasks

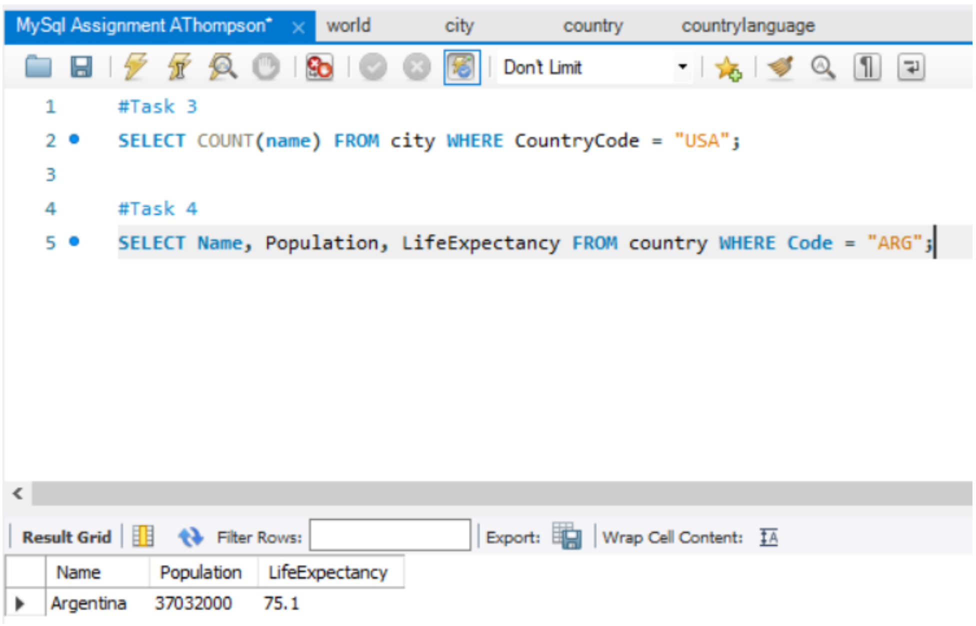
Task 3: Using count, get the number of cities in the USA.



```
#Task 3
SELECT COUNT(name) FROM city WHERE CountryCode = "USA";
```

```
COUNT(name)
274
```

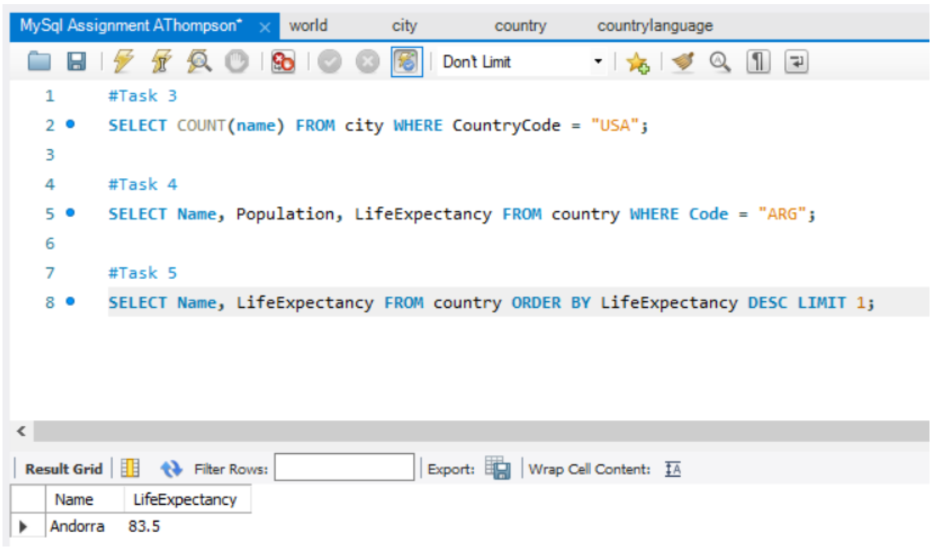
Task 4: Find out what the population and life expectancy for people in Argentina is.



```
#Task 3
SELECT COUNT(name) FROM city WHERE CountryCode = "USA";

#Task 4
SELECT Name, Population, LifeExpectancy FROM country WHERE Code = "ARG";
```

```
Name        Population  LifeExpectancy
Argentina   37032000    75.1
```

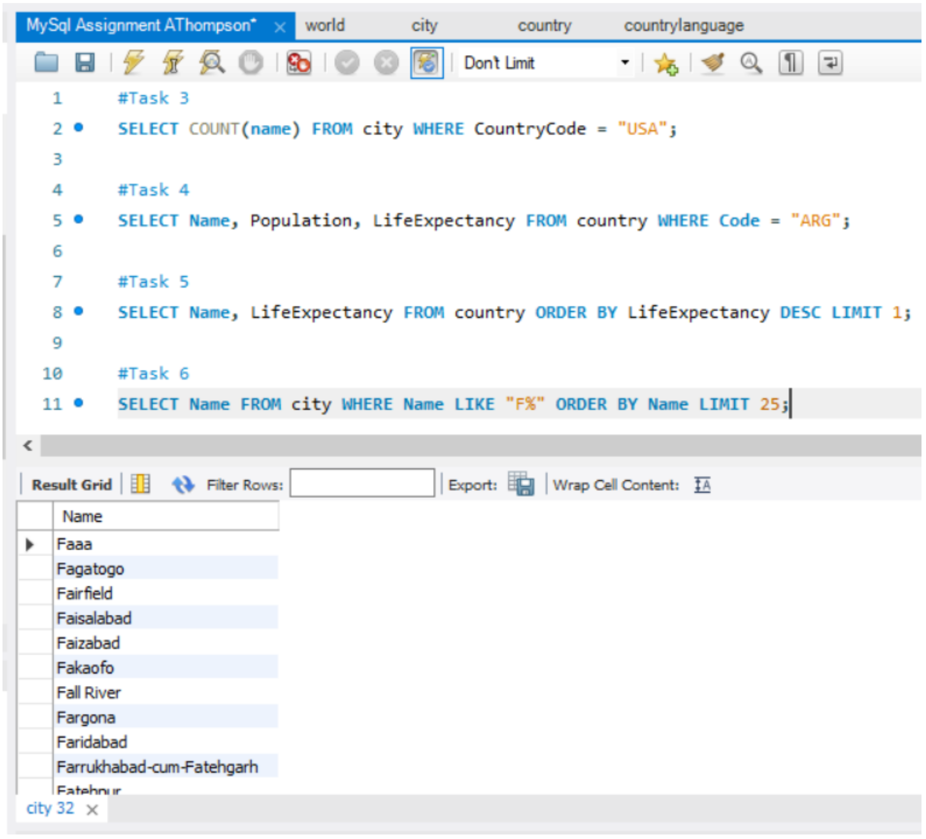
Task 5: Using ORDER BY, LIMIT what country has the highest life expectancy?



```
#Task 3
SELECT COUNT(name) FROM city WHERE CountryCode = "USA";

#Task 4
SELECT Name, Population, LifeExpectancy FROM country WHERE Code = "ARG";

#Task 5
SELECT Name, LifeExpectancy FROM country ORDER BY LifeExpectancy DESC LIMIT 1;
```

Name	LifeExpectancy
Andorra	83.5

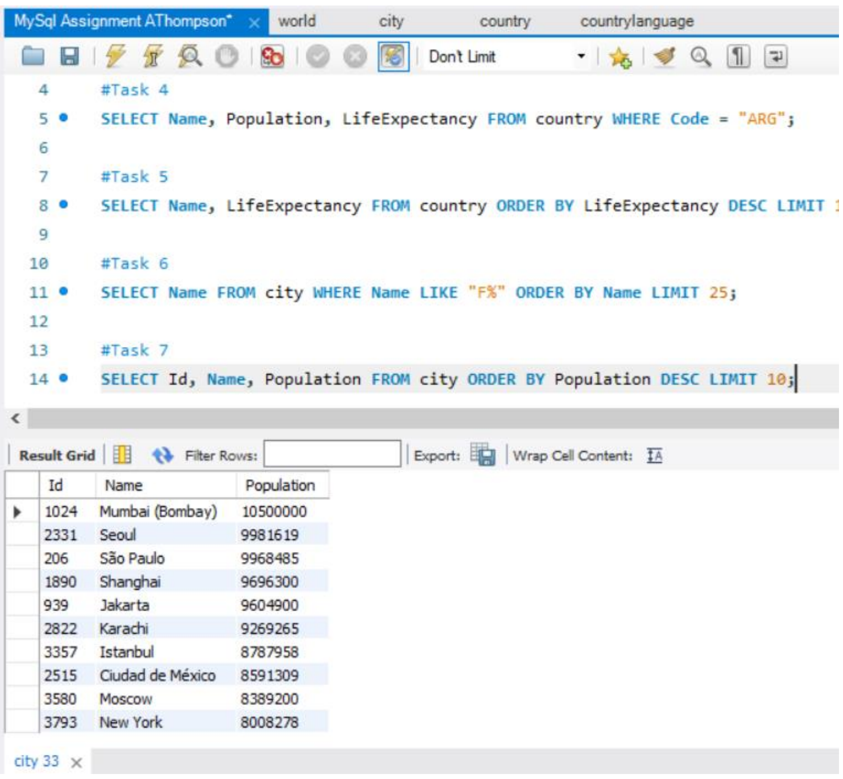
Task 6: Select 25 cities around the world that start with the letter 'F' in a single SQL query.



```
#Task 3
SELECT COUNT(name) FROM city WHERE CountryCode = "USA";

#Task 4
SELECT Name, Population, LifeExpectancy FROM country WHERE Code = "ARG";

#Task 5
SELECT Name, LifeExpectancy FROM country ORDER BY LifeExpectancy DESC LIMIT 1;

#Task 6
SELECT Name FROM city WHERE Name LIKE "F%" ORDER BY Name LIMIT 25;
```

Name
Faaa
Fagatogo
Fairfield
Faisalabad
Faizabad
Fakaofo
Fall River
Fargona
Faridabad
Farrukhabad-cum-Fatehgarh
Fatehpur

Task 7: Create a SQL statement to display columns Id, Name, Population from the city table and limit to first 10 rows only.

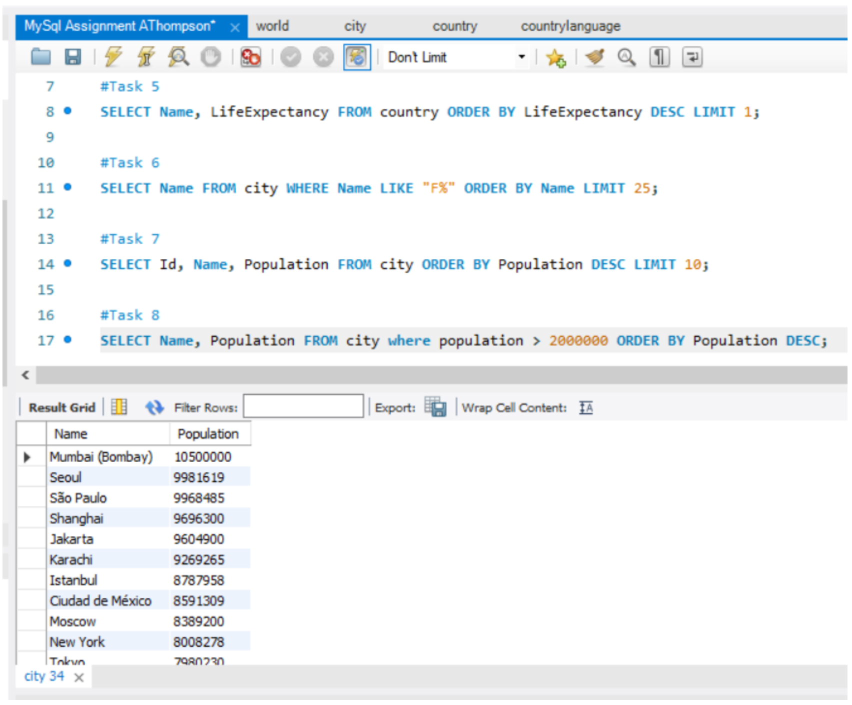


```sql
#Task 4
SELECT Name, Population, LifeExpectancy FROM country WHERE Code = "ARG";

#Task 5
SELECT Name, LifeExpectancy FROM country ORDER BY LifeExpectancy DESC LIMIT

#Task 6
SELECT Name FROM city WHERE Name LIKE "F%" ORDER BY Name LIMIT 25;

#Task 7
SELECT Id, Name, Population FROM city ORDER BY Population DESC LIMIT 10;
```

Id	Name	Population
1024	Mumbai (Bombay)	10500000
2331	Seoul	9981619
206	São Paulo	9968485
1890	Shanghai	9696300
939	Jakarta	9604900
2822	Karachi	9269265
3357	Istanbul	8787958
2515	Ciudad de México	8591309
3580	Moscow	8389200
3793	New York	8008278

Note: As this task doesn't specify how the data is ordered I've chosen to show the 10 most populous cities.

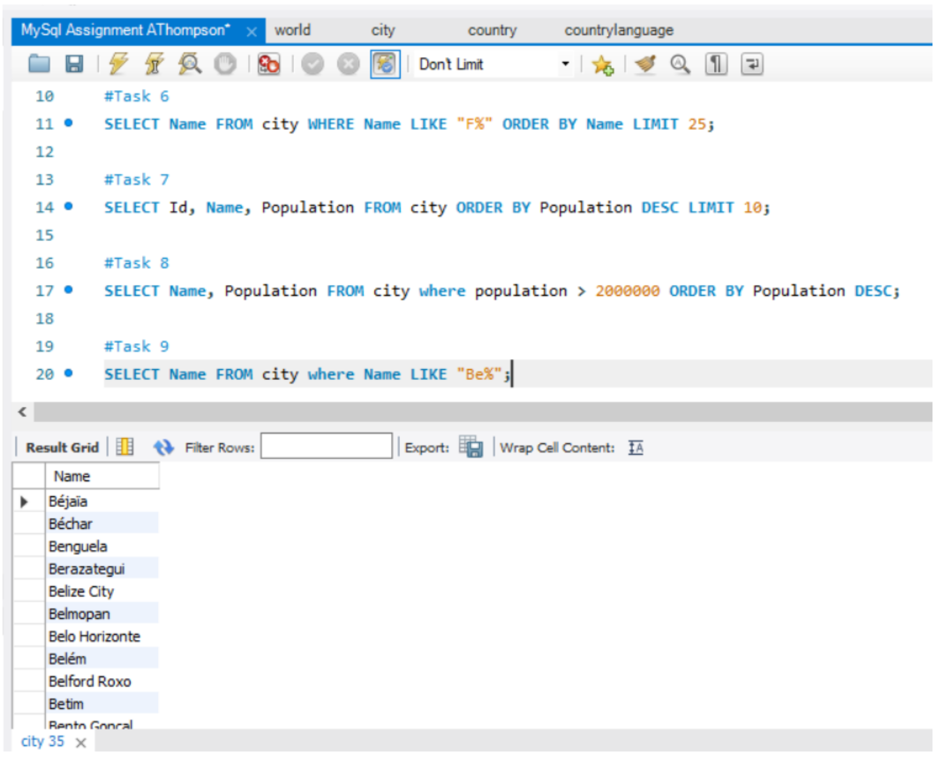
Task 8: Create a SQL statement to find only those cities from city table whose population is larger than 2000000.



```sql
#Task 5
SELECT Name, LifeExpectancy FROM country ORDER BY LifeExpectancy DESC LIMIT 1;

#Task 6
SELECT Name FROM city WHERE Name LIKE "F%" ORDER BY Name LIMIT 25;

#Task 7
SELECT Id, Name, Population FROM city ORDER BY Population DESC LIMIT 10;

#Task 8
SELECT Name, Population FROM city where population > 2000000 ORDER BY Population DESC;
```

Name	Population
Mumbai (Bombay)	10500000
Seoul	9981619
São Paulo	9968485
Shanghai	9696300
Jakarta	9604900
Karachi	9269265
Istanbul	8787958
Ciudad de México	8591309
Moscow	8389200
New York	8008278
Tokyo	7980230

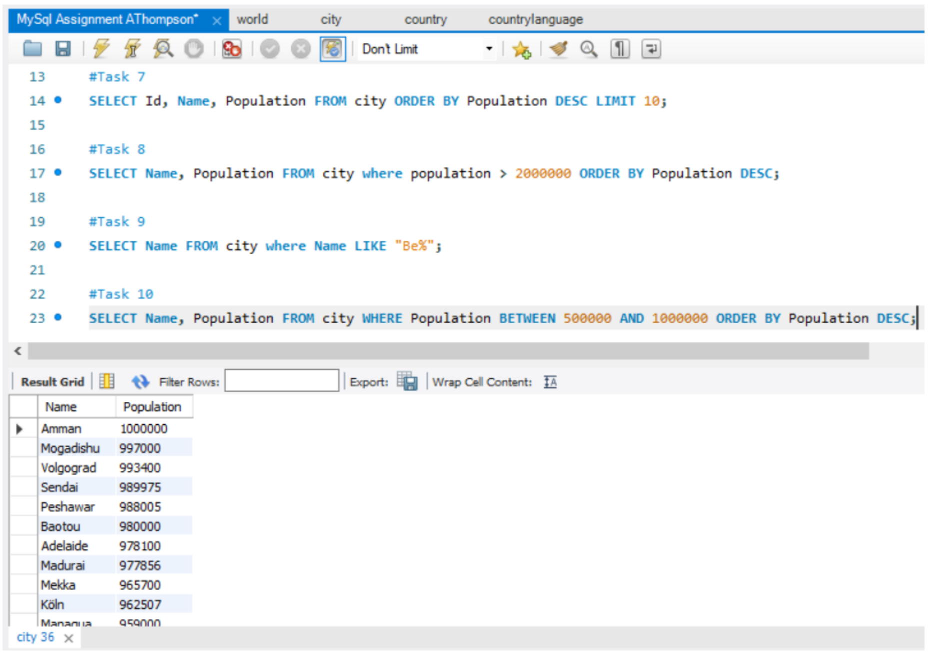
Task 9: Create a SQL statement to find all city names from city table whose name begins with 'Be' prefix.



```
#Task 6
SELECT Name FROM city WHERE Name LIKE "F%" ORDER BY Name LIMIT 25;

#Task 7
SELECT Id, Name, Population FROM city ORDER BY Population DESC LIMIT 10;

#Task 8
SELECT Name, Population FROM city where population > 2000000 ORDER BY Population DESC;

#Task 9
SELECT Name FROM city where Name LIKE "Be%";
```

Name
Béjaïa
Béchar
Benguela
Berazategui
Belize City
Belmopan
Belo Horizonte
Belém
Belford Roxo
Betim
Bento Gonçal

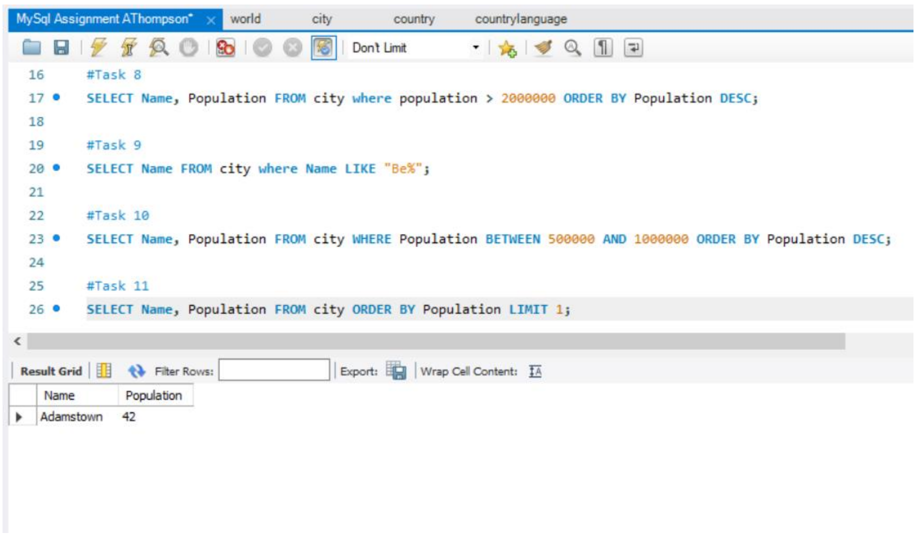
Task 10: Create a SQL statement to only those cities from city table whose population is between 500000 – 10000000.



```
#Task 7
SELECT Id, Name, Population FROM city ORDER BY Population DESC LIMIT 10;

#Task 8
SELECT Name, Population FROM city where population > 2000000 ORDER BY Population DESC;

#Task 9
SELECT Name FROM city where Name LIKE "Be%";

#Task 10
SELECT Name, Population FROM city WHERE Population BETWEEN 500000 AND 1000000 ORDER BY Population DESC;
```

Name	Population
Amman	1000000
Mogadishu	997000
Volgograd	993400
Sendai	989975
Peshawar	988005
Baotou	980000
Adelaide	978100
Madurai	977856
Mekka	965700
Köln	962507
Managua	959000

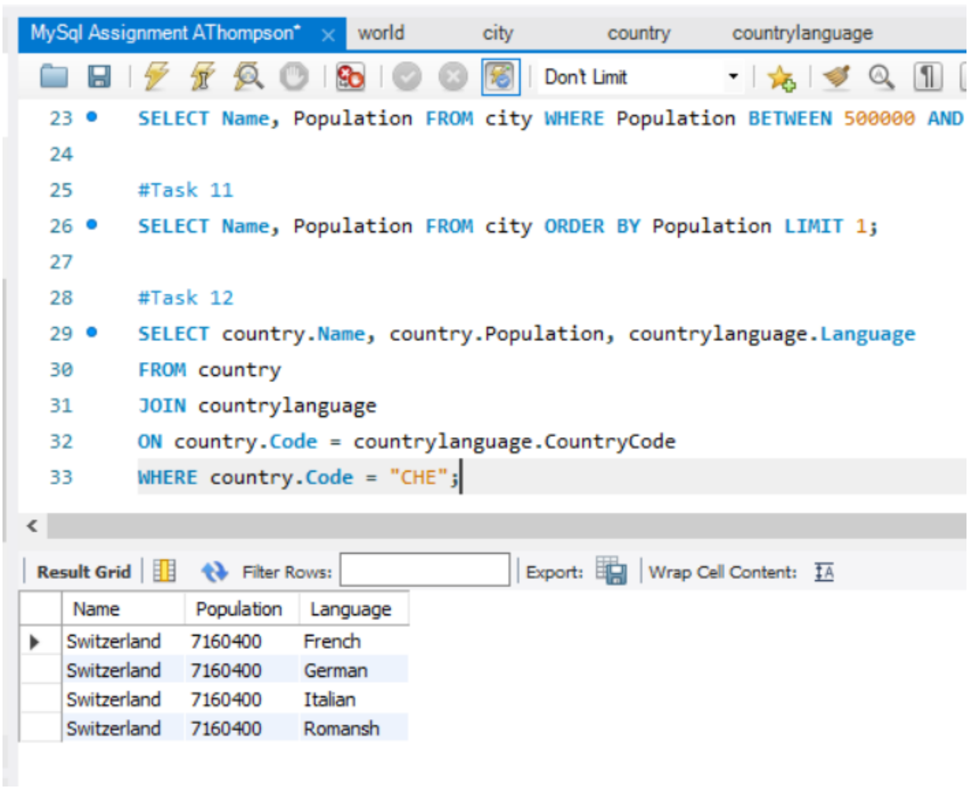
Task 11: Create a SQL statement to find a city with the lowest population in the city table.



```sql
#Task 8
SELECT Name, Population FROM city where population > 2000000 ORDER BY Population DESC;

#Task 9
SELECT Name FROM city where Name LIKE "Be%";

#Task 10
SELECT Name, Population FROM city WHERE Population BETWEEN 500000 AND 1000000 ORDER BY Population DESC;

#Task 11
SELECT Name, Population FROM city ORDER BY Population LIMIT 1;
```

```
Name        Population
Adamstown   42
```

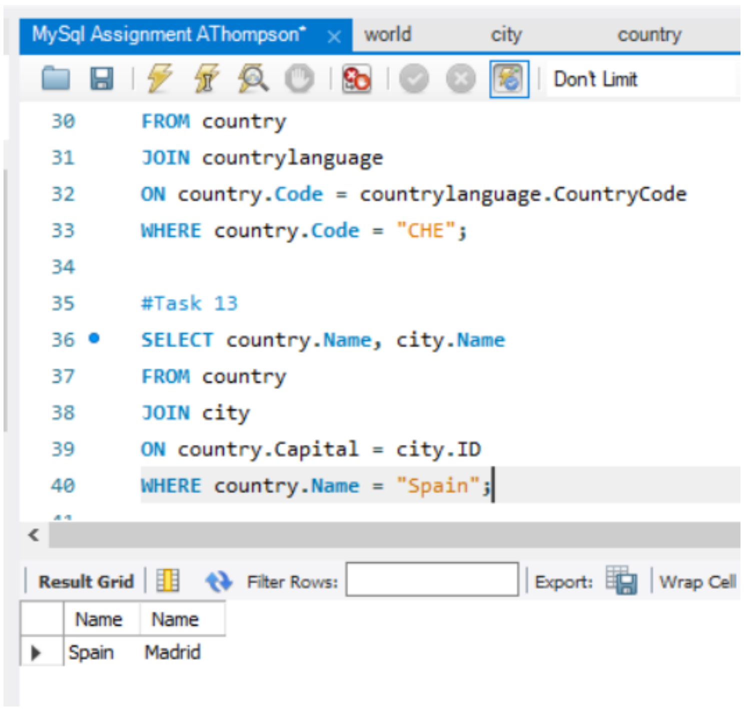
Task 12: Create a SQL statement to show the population of Switzerland and all the languages spoken there.



```sql
SELECT Name, Population FROM city WHERE Population BETWEEN 500000 AND

#Task 11
SELECT Name, Population FROM city ORDER BY Population LIMIT 1;

#Task 12
SELECT country.Name, country.Population, countrylanguage.Language
FROM country
JOIN countrylanguage
ON country.Code = countrylanguage.CountryCode
WHERE country.Code = "CHE";
```

```
Name          Population   Language
Switzerland   7160400      French
Switzerland   7160400      German
Switzerland   7160400      Italian
Switzerland   7160400      Romansh
```

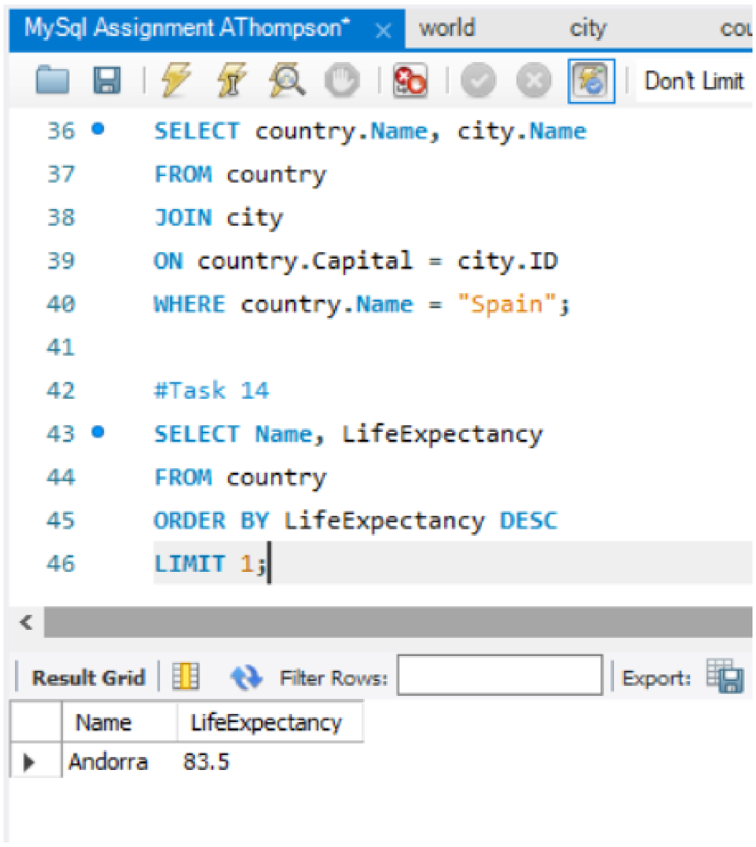
Task 13: Create a SQL statement to find the capital of Spain.



```
FROM country
JOIN countrylanguage
ON country.Code = countrylanguage.CountryCode
WHERE country.Code = "CHE";

#Task 13
SELECT country.Name, city.Name
FROM country
JOIN city
ON country.Capital = city.ID
WHERE country.Name = "Spain";
```

Name	Name
Spain	Madrid

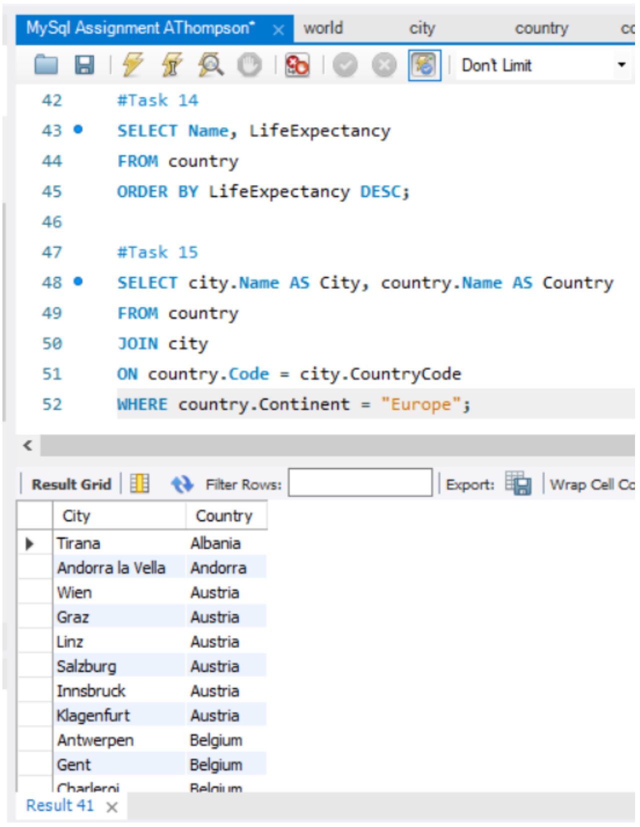
Task 14: Create a SQL statement to find the country with the highest life expectancy.



```
SELECT country.Name, city.Name
FROM country
JOIN city
ON country.Capital = city.ID
WHERE country.Name = "Spain";

#Task 14
SELECT Name, LifeExpectancy
FROM country
ORDER BY LifeExpectancy DESC
LIMIT 1;
```

Name	LifeExpectancy
Andorra	83.5

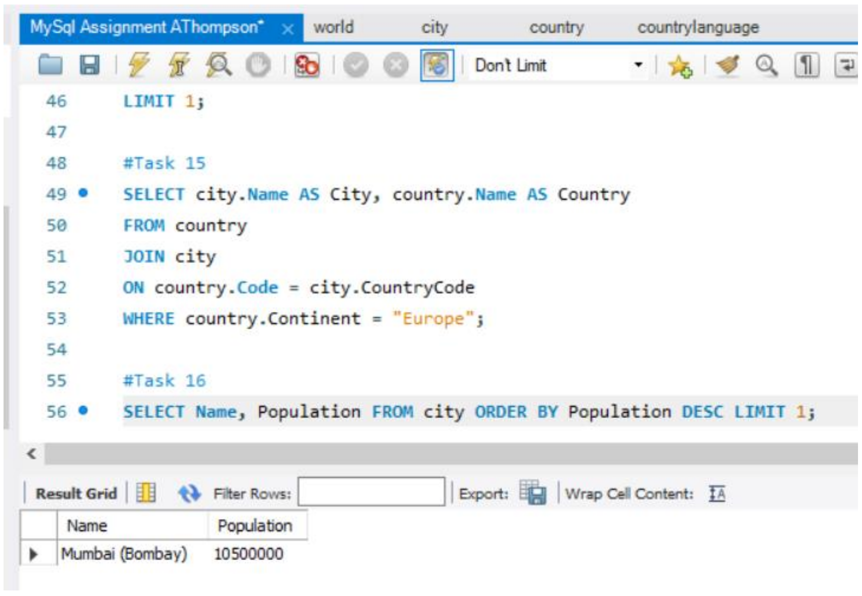
Task 15: Create a SQL statement to find all cities from the Europe continent.



```
#Task 14
SELECT Name, LifeExpectancy
FROM country
ORDER BY LifeExpectancy DESC;

#Task 15
SELECT city.Name AS City, country.Name AS Country
FROM country
JOIN city
ON country.Code = city.CountryCode
WHERE country.Continent = "Europe";
```

City	Country
Tirana	Albania
Andorra la Vella	Andorra
Wien	Austria
Graz	Austria
Linz	Austria
Salzburg	Austria
Innsbruck	Austria
Klagenfurt	Austria
Antwerpen	Belgium
Gent	Belgium
Charleroi	Belgium

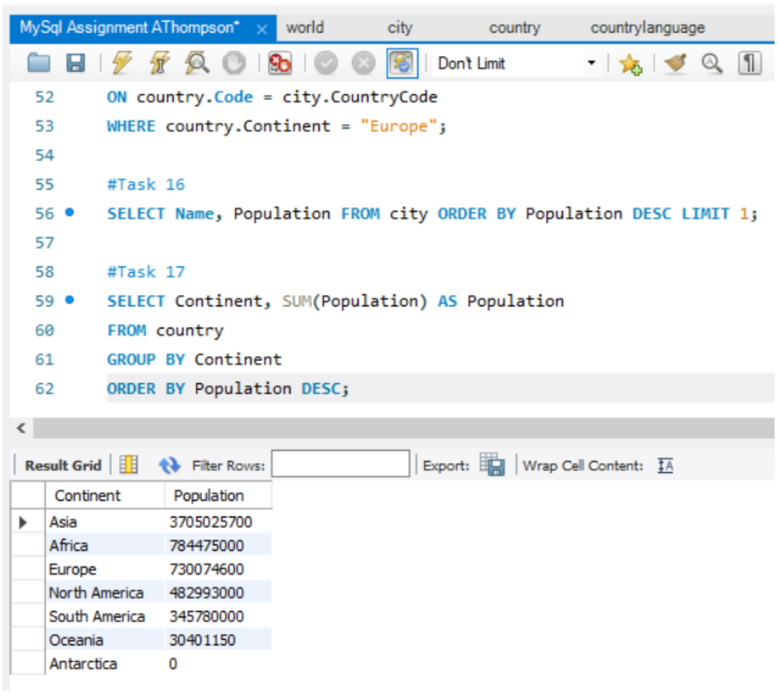
Task 16: Create a SQL statement to find the most populated city in the city table.



```
LIMIT 1;

#Task 15
SELECT city.Name AS City, country.Name AS Country
FROM country
JOIN city
ON country.Code = city.CountryCode
WHERE country.Continent = "Europe";

#Task 16
SELECT Name, Population FROM city ORDER BY Population DESC LIMIT 1;
```

Name	Population
Mumbai (Bombay)	10500000

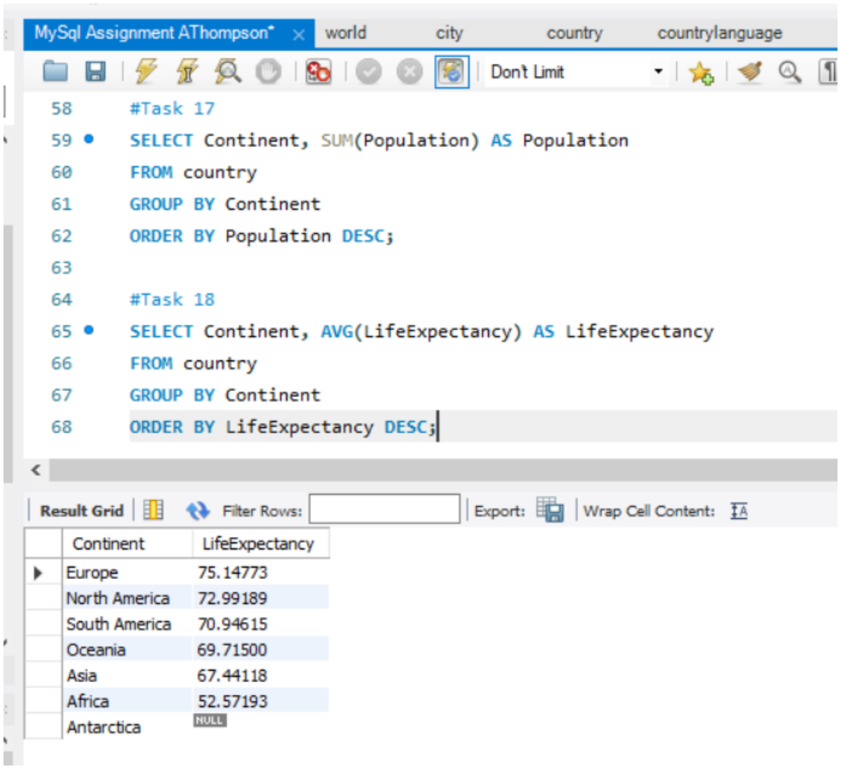
Task 17: Create a SQL statement to find population of each continent.



```
ON country.Code = city.CountryCode
WHERE country.Continent = "Europe";

#Task 16
SELECT Name, Population FROM city ORDER BY Population DESC LIMIT 1;

#Task 17
SELECT Continent, SUM(Population) AS Population
FROM country
GROUP BY Continent
ORDER BY Population DESC;
```

Continent	Population
Asia	3705025700
Africa	784475000
Europe	730074600
North America	482993000
South America	345780000
Oceania	30401150
Antarctica	0

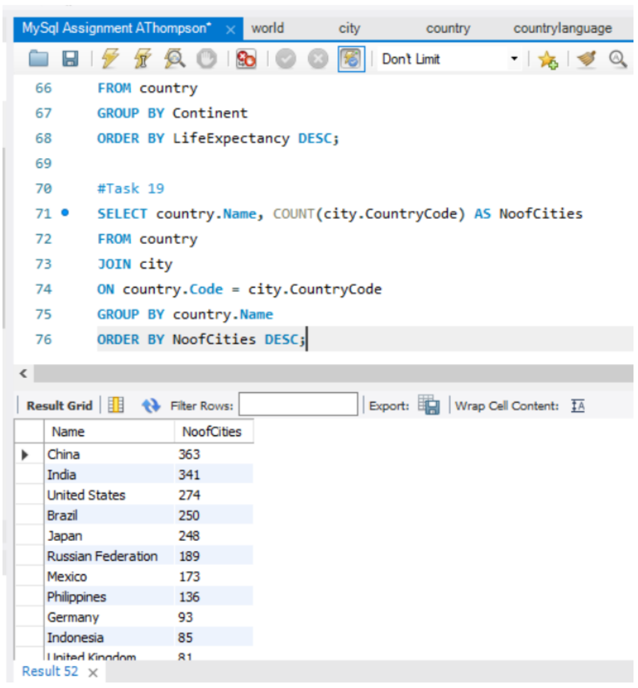
Task 18: Create a SQL statement to find the average life expectancy by continent.



```
#Task 17
SELECT Continent, SUM(Population) AS Population
FROM country
GROUP BY Continent
ORDER BY Population DESC;

#Task 18
SELECT Continent, AVG(LifeExpectancy) AS LifeExpectancy
FROM country
GROUP BY Continent
ORDER BY LifeExpectancy DESC;
```

Continent	LifeExpectancy
Europe	75.14773
North America	72.99189
South America	70.94615
Oceania	69.71500
Asia	67.44118
Africa	52.57193
Antarctica	NULL

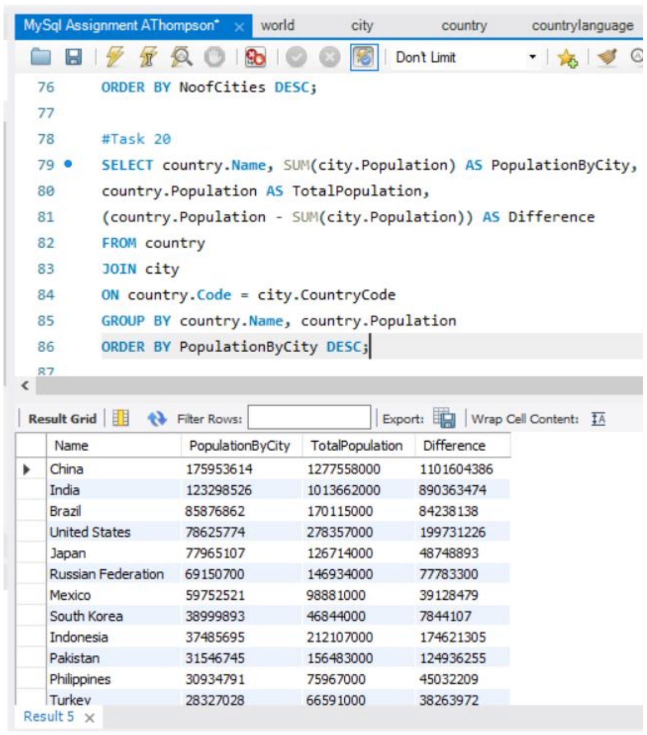
Task 19: Create a SQL statement to list the number of cities in each country.



```
FROM country
GROUP BY Continent
ORDER BY LifeExpectancy DESC;

#Task 19
SELECT country.Name, COUNT(city.CountryCode) AS NoofCities
FROM country
JOIN city
ON country.Code = city.CountryCode
GROUP BY country.Name
ORDER BY NoofCities DESC;
```

Name	NoofCities
China	363
India	341
United States	274
Brazil	250
Japan	248
Russian Federation	189
Mexico	173
Philippines	136
Germany	93
Indonesia	85
United Kingdom	81

Task 20: Create a SQL statement to find the total population of each country based on its cities

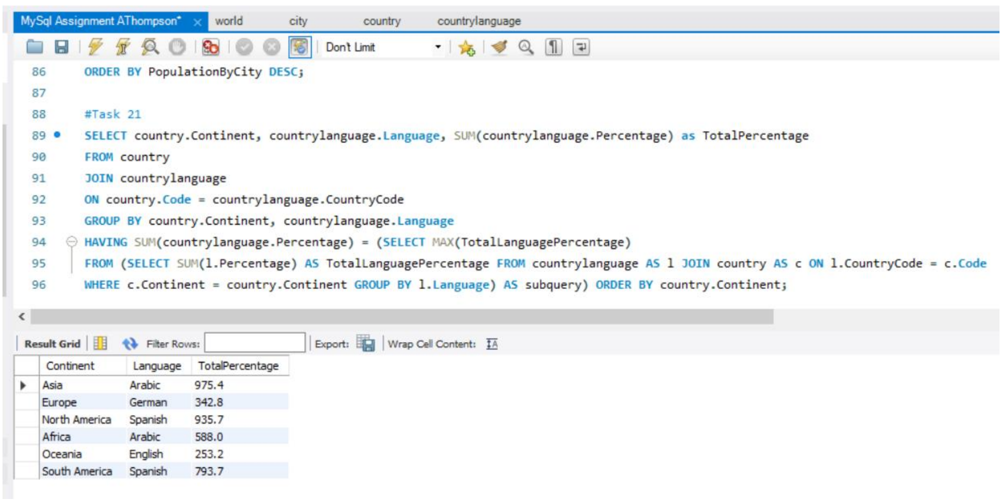


```
ORDER BY NoofCities DESC;

#Task 20
SELECT country.Name, SUM(city.Population) AS PopulationByCity,
country.Population AS TotalPopulation,
(country.Population - SUM(city.Population)) AS Difference
FROM country
JOIN city
ON country.Code = city.CountryCode
GROUP BY country.Name, country.Population
ORDER BY PopulationByCity DESC;
```

Name	PopulationByCity	TotalPopulation	Difference
China	175953614	1277558000	1101604386
India	123298526	1013662000	890363474
Brazil	85876862	170115000	84238138
United States	78625774	278357000	199731226
Japan	77965107	126714000	48748893
Russian Federation	69150700	146934000	77783300
Mexico	59752521	98881000	39128479
South Korea	38999893	46844000	7844107
Indonesia	37485695	212107000	174621305
Pakistan	31546745	156483000	124936255
Philippines	30934791	75967000	45032209
Turkey	28327028	66591000	38263972

Note: I got cocky and included the difference between population and the population totals by city.

Task 21: Create a SQL statement to find the most spoken language on each continent.

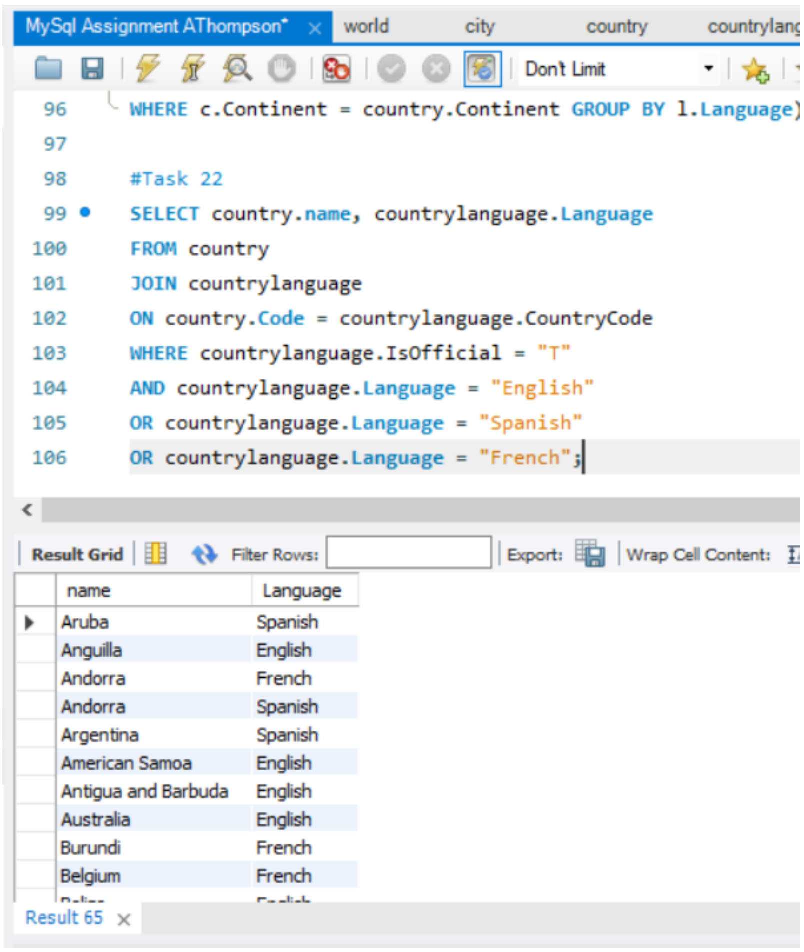


```sql
ORDER BY PopulationByCity DESC;

#Task 21
SELECT country.Continent, countrylanguage.Language, SUM(countrylanguage.Percentage) as TotalPercentage
FROM country
JOIN countrylanguage
ON country.Code = countrylanguage.CountryCode
GROUP BY country.Continent, countrylanguage.Language
HAVING SUM(countrylanguage.Percentage) = (SELECT MAX(TotalLanguagePercentage)
FROM (SELECT SUM(l.Percentage) AS TotalLanguagePercentage FROM countrylanguage AS l JOIN country AS c ON l.CountryCode = c.Code
WHERE c.Continent = country.Continent GROUP BY l.Language) AS subquery) ORDER BY country.Continent;
```

Continent	Language	TotalPercentage
Asia	Arabic	975.4
Europe	German	342.8
North America	Spanish	935.7
Africa	Arabic	588.0
Oceania	English	253.2
South America	Spanish	793.7

Note: Even with Chat GPT's help this still looks wrong, but it was the best I could do.

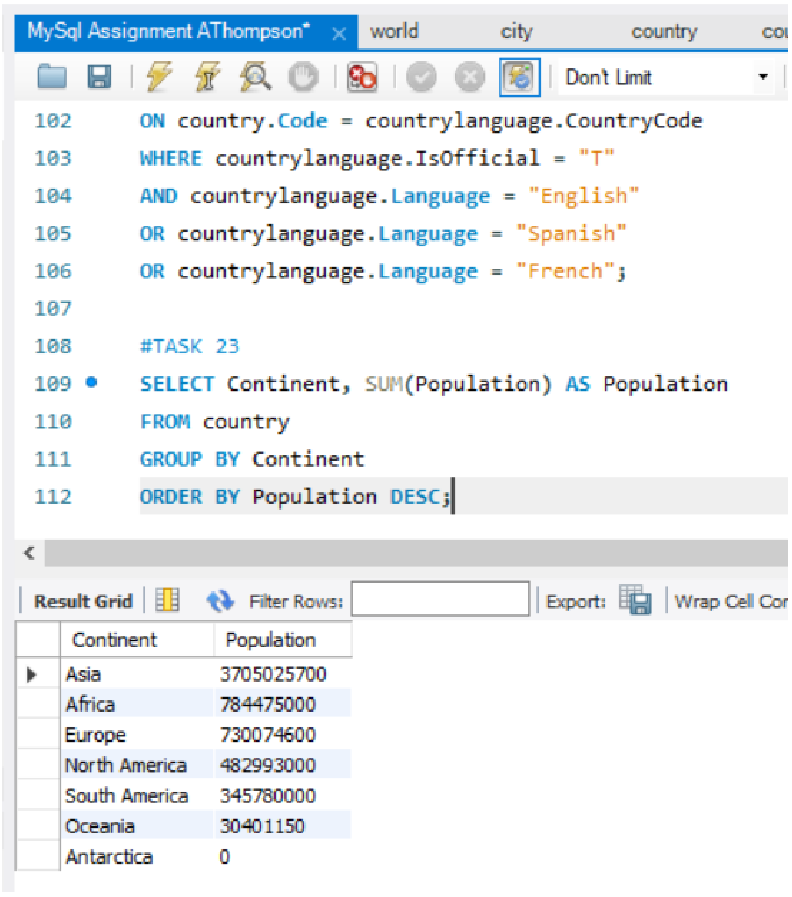
Task 22: Create a SQL statement to find countries where the official language is either 'English', 'Spanish' or 'French'.



```
WHERE c.Continent = country.Continent GROUP BY l.Language)

#Task 22
SELECT country.name, countrylanguage.Language
FROM country
JOIN countrylanguage
ON country.Code = countrylanguage.CountryCode
WHERE countrylanguage.IsOfficial = "T"
AND countrylanguage.Language = "English"
OR countrylanguage.Language = "Spanish"
OR countrylanguage.Language = "French";
```

name	Language
Aruba	Spanish
Anguilla	English
Andorra	French
Andorra	Spanish
Argentina	Spanish
American Samoa	English
Antigua and Barbuda	English
Australia	English
Burundi	French
Belgium	French

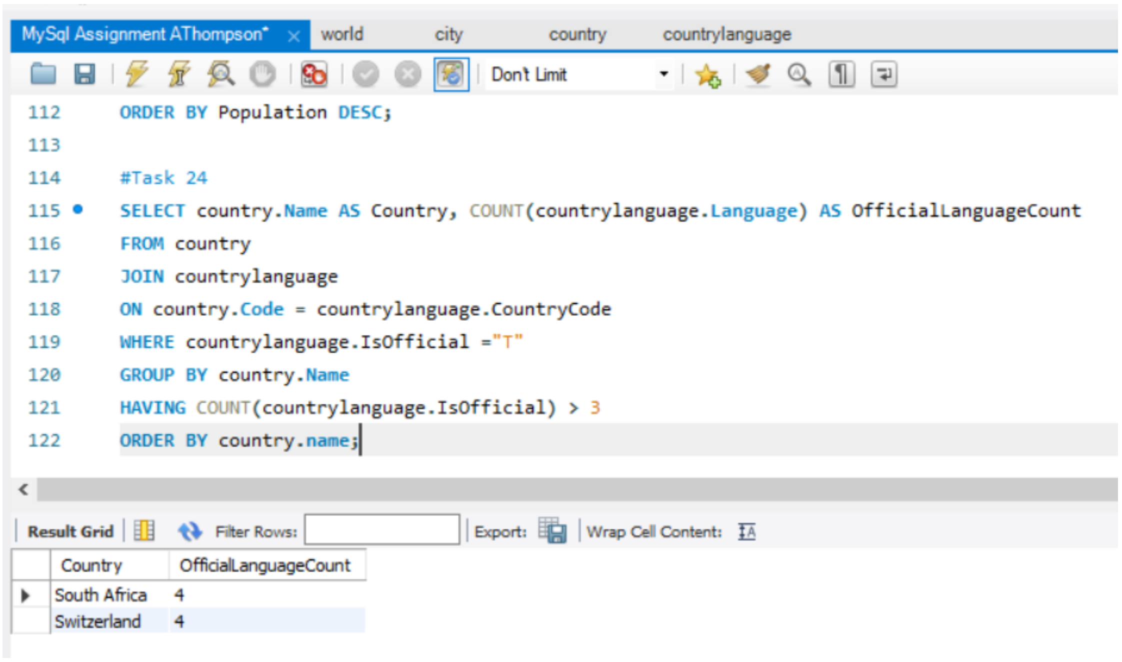
Task 23: Write a query to display the total population of each continent.



```
ON country.Code = countrylanguage.CountryCode
WHERE countrylanguage.IsOfficial = "T"
AND countrylanguage.Language = "English"
OR countrylanguage.Language = "Spanish"
OR countrylanguage.Language = "French";

#TASK 23
SELECT Continent, SUM(Population) AS Population
FROM country
GROUP BY Continent
ORDER BY Population DESC;
```

Continent	Population
Asia	3705025700
Africa	784475000
Europe	730074600
North America	482993000
South America	345780000
Oceania	30401150
Antarctica	0

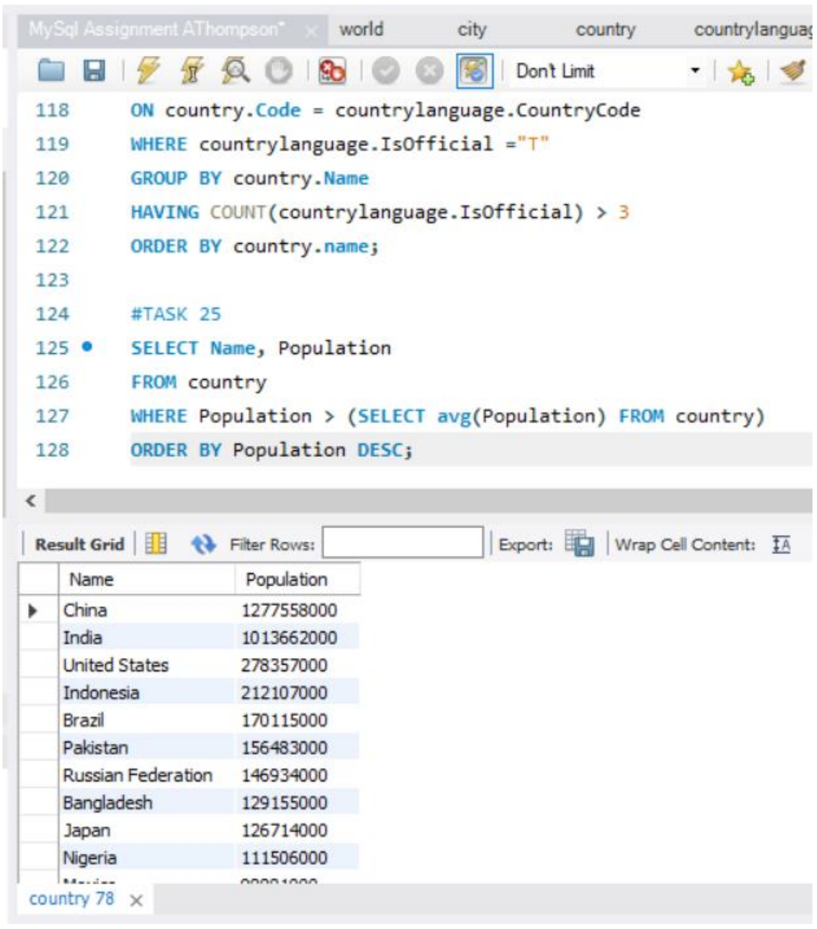
Task 24: Write a query to list countries that have more than three official languages.



```
ORDER BY Population DESC;

#Task 24
SELECT country.Name AS Country, COUNT(countrylanguage.Language) AS OfficialLanguageCount
FROM country
JOIN countrylanguage
ON country.Code = countrylanguage.CountryCode
WHERE countrylanguage.IsOfficial ="T"
GROUP BY country.Name
HAVING COUNT(countrylanguage.IsOfficial) > 3
ORDER BY country.name;
```

Country	OfficialLanguageCount
South Africa	4
Switzerland	4

Task 25: Find countries whose population is greater than the average population of all countries.



```
ON country.Code = countrylanguage.CountryCode
WHERE countrylanguage.IsOfficial ="T"
GROUP BY country.Name
HAVING COUNT(countrylanguage.IsOfficial) > 3
ORDER BY country.name;

#TASK 25
SELECT Name, Population
FROM country
WHERE Population > (SELECT avg(Population) FROM country)
ORDER BY Population DESC;
```

Name	Population
China	1277558000
India	1013662000
United States	278357000
Indonesia	212107000
Brazil	170115000
Pakistan	156483000
Russian Federation	146934000
Bangladesh	129155000
Japan	126714000
Nigeria	111506000

Task 26:
Primary Key in country Table – ‘Code’
Primary Key in city Table – ‘ID’
Primary Key in countrylanguage Table – ‘CountryCode’
Foreign Key in city Table – ‘CountryCode’
Foreign Key in countrylanguage Table – ‘CountryCode’