

Assignment 2 - MySQL

Marnie Blades
Data Skills Bootcamp
11/19/24

CONTENTS

Task 1:	3
List the different types of relationships in relational databases and provide examples.....	3
Task 2:	3
What is Normalization and why is it important to database development?.....	3
Task 3:	4
Using count, get the number of cities in the USA.....	4
Task 4:	4
Find out what the population and life expectancy for people in Argentina (ARG) is.....	4
Task 5:	5
Using ORDER BY, LIMIT, what country has the highest life expectancy?	5
Task 6:	5
Select 25 cities around the world that start with the letter 'F' in a single SQL query.	5
Task 7:	6
Create a SQL statement to display columns Id, Name, Population from the city table and limit results to first 10 rows only.	6
Task 8:	6
Create a SQL statement to display columns Id, Name, Population from the city table and limit results to first 10 rows only.	6
Task 9:	6
Create a SQL statement to find all city names from city table whose name begins with “Be” prefix.	6
Task 10:	7
Create a SQL statement to find only those cities from city table whose population is between 500000-1000000.....	7
Task 11:	7
Create a SQL statement to find a city with the lowest population in the city table.....	7
Task 12:	8
Create a SQL statement to show the population of Switzerland and all the languages spoken there.	8
Task 13:	8
Create a SQL statement to find the capital of Spain (ESP).....	8
Task 14:	9
Create a SQL statement to find the country with the highest life expectancy.....	9
Task 15:	9
Create a SQL statement to find all cities from the Europe continent.....	9

Task 16:	10
Create a SQL statement to find the most populated city in the city table	10
Task 17:	10
Create a SQL statement to find the total population of each continent	10
Task 18:	11
Create a SQL statement to find the average life expectancy by continent	11
Task 19:	12
Create a SQL statement to list the number of cities in each country	12
Task 20:	13
Create a SQL statement to find the total population of each country based on its cities	13
Task 21:	14
Create a SQL statement to find the most spoken language in each continent	14
Task 22:	14
Create a SQL statement to find countries where the official language is either 'English', 'Spanish', or 'French'	14
Task 23:	15
Write a query to display the total population for each continent.....	15
Task 24:	15
Write a query to list countries that have more than three official languages. (joins, group by, having.....	15
Task 25:	16
Find countries whose population is greater than the average population of all countries.	16
Task 26:	16
Creating an EER Diagram	16
Task 27:	17
Identify the primary key in country table.	17
Identify the primary key in city table.	17
Identify the primary key in countrylanguage table.	17
Identify the foreign key in city table.	17
Identify the foreign key in countrylanguage table.....	17

TASK 1:

LIST THE DIFFERENT TYPES OF RELATIONSHIPS IN RELATIONAL DATABASES AND PROVIDE EXAMPLES.

One to One Relationship:

A one-to-one relationship is where a record in one table is linked to one record in another table. An example of this would be in a driving school database, there would be a table called Students containing student's names and each student will have a corresponding record in the StudentDetails table that stores their driving licence number.

Other examples can include Passport Numbers to Person, Employee ID to Employee, and Country to Capital City.

One to Many Relationship:

A one-to-many relationship is the most common relationship in relational databases. It is where one record in a table is linked to multiple records in another table. An example of this would be having a countries table and a cities table. Each country would have multiple corresponding records in the cities table, but each city would only have one corresponding record in the countries table.

Other examples include Authors to Books, Customers to Orders, and Categories to Products.

Many to Many Relationship:

A many to many relationship is when multiple records from one table are linked to multiple records in another table. An example of this would be having a Students table and a Classes table. Each student takes multiple classes, and each class has multiple students who take the class.

Other examples include Movies to Actors, Projects to Employees, and Products to Orders.

TASK 2:

WHAT IS NORMALIZATION AND WHY IS IT IMPORTANT TO DATABASE DEVELOPMENT?

Normalization is a common practice when dealing with large databases. It is the process of breaking down, usually large and complex, tables into smaller and much simpler tables in a way that keeps the relationships between the data. Normalization is important as it removes anomalies from the data. This includes reducing redundancy, where the same piece of data is stored multiple times in the database, normalization ensures the data is only stored in one place which makes the database more efficient. It also prevents insertion, update, deletion anomalies which cause issues with adding new data, deleting data and updating the data. Additionally, it improves query performance, making queries faster to run, and data integrity and consistency.

TASK 3:

USING COUNT, GET THE NUMBER OF CITIES IN THE USA

SELECT COUNT(name) as Count, CountryCode

FROM city

WHERE CountryCode = 'USA';

The screenshot shows a SQL query editor with the following code:

```
10 #task 3:USING COUNT, GET THE NUMBER OF CITIES IN THE USA#
11 • select count(name) as Count, CountryCode
12 from city
13 where CountryCode = 'USA';
```

Below the code, there is a toolbar with options like "Result Grid", "Filter Rows", "Export", and "Wrap Cell Content". Below the toolbar, a table displays the results of the query:

Count	CountryCode
274	USA

At the bottom, there is a tab labeled "Result 13" with a close button.

TASK 4:

FIND OUT WHAT THE POPULATION AND LIFE EXPECTANCY FOR PEOPLE IN ARGENTINA (ARG) IS

SELECT Name, Population, LifeExpectancy

FROM country

Where Name = 'Argentina';

The screenshot shows a SQL query editor with the following code:

```
11 • SELECT COUNT(Name) as Count, CountryCode
12 FROM city
13 WHERE CountryCode = 'USA';
14
15 #TASK 4 Find out what the population and life expectancy for people in Argentina (ARG) is#
16 • SELECT Name, Population, LifeExpectancy
17 FROM country
18 where Name = 'Argentina';
```

Below the code, there is a toolbar with options like "Limit to 1000 rows", "Filter Rows", "Export", and "Wrap Cell Content". Below the toolbar, a table displays the results of the query:

Name	Population	LifeExpectancy
Argentina	37032000	75.1

At the bottom, there is a tab labeled "country 17" with a close button and a "Read O" link.

TASK 5:

USING ORDER BY, LIMIT, WHAT COUNTRY HAS THE HIGHEST LIFE EXPECTANCY?

SELECT Name as country, LifeExpectancy

FROM country

ORDER BY LifeExpectancy desc

LIMIT 1;

```
19
20 #TASK 5 USING ORDER BY, LIMIT, WHAT COUNTRY HAS THE HIGHEST LIFE EXPECTANCY?#
21 • SELECT Name as country, LifeExpectancy
22 FROM country
23 ORDER BY LifeExpectancy desc
24 LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

	country	LifeExpectancy
▶	Andorra	83.5

TASK 6:

SELECT 25 CITIES AROUND THE WORLD THAT START WITH THE LETTER 'F' IN A SINGLE SQL QUERY.

SELECT Name as City

from city

where name like 'F%'

limit 25;

```
26 #Select 25 cities around the world that start with the letter 'F' in a single SQL query.#
27 • SELECT Name as City
28 from city
29 where name like 'F%'
30 limit 25;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

	City
▶	Fagatogo
	Florencio Varela
	Formosa
	Fransistown
	Fortaleza

city 23 x Read Only

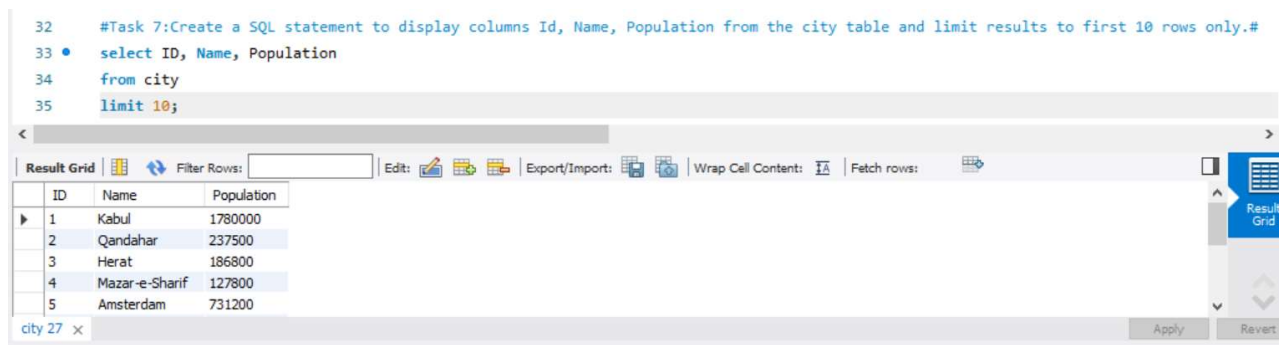
TASK 7:

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;



TASK 8:

CREATE A SQL STATEMENT TO DISPLAY COLUMNS ID, NAME, POPULATION FROM THE CITY TABLE
AND LIMIT RESULTS TO FIRST 10 ROWS ONLY.

Same as above (Task 7).

TASK 9:

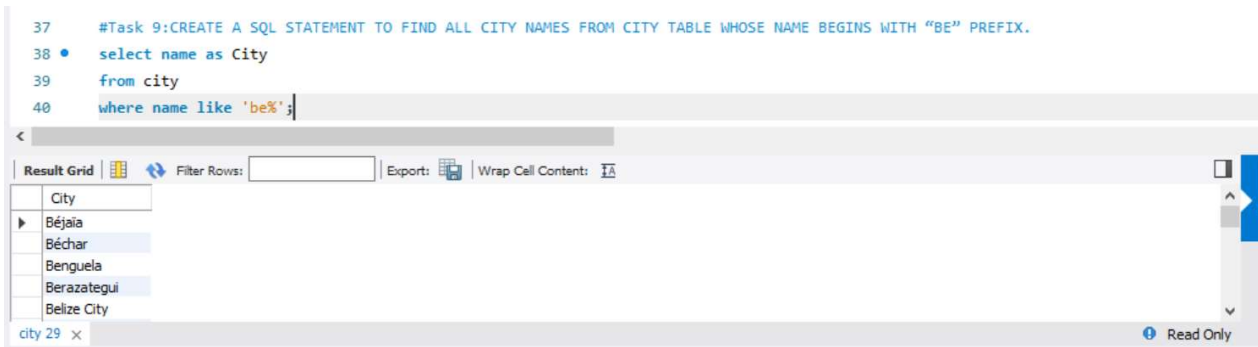
CREATE A SQL STATEMENT TO FIND ALL CITY NAMES FROM CITY TABLE WHOSE NAME BEGINS WITH
"BE" PREFIX.

select name as City

from city

where name like 'be%';

```
37 #Task 9:CREATE A SQL STATEMENT TO FIND ALL CITY NAMES FROM CITY TABLE WHOSE NAME BEGINS WITH "BE" PREFIX.
38 • select name as City
39 from city
40 where name like 'be%';
```



City
Béjaia
Béchar
Benguela
Berazategui
Belize City

TASK 10:

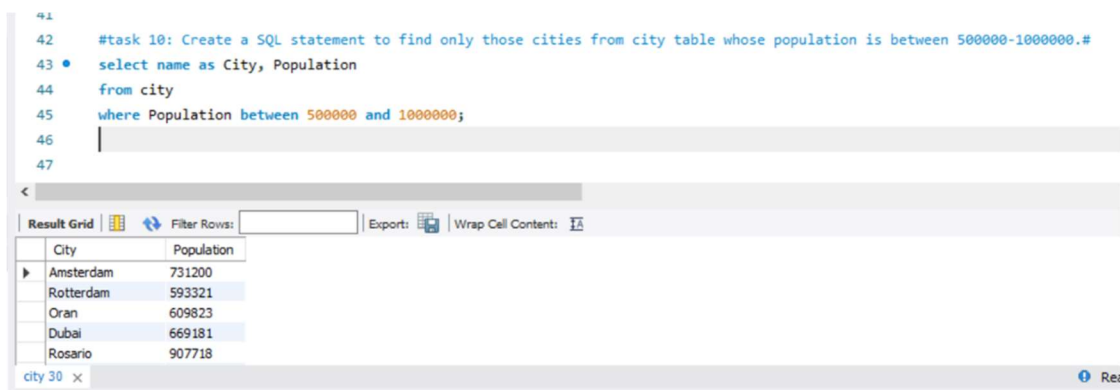
CREATE A SQL STATEMENT TO FIND ONLY THOSE CITIES FROM CITY TABLE WHOSE POPULATION IS BETWEEN 500000-1000000.

select name as City, Population

from city

where Population between 500000 and 1000000;

```
41
42 #task 10: Create a SQL statement to find only those cities from city table whose population is between 500000-1000000.#
43 • select name as City, Population
44 from city
45 where Population between 500000 and 1000000;
46
47
```



City	Population
Amsterdam	731200
Rotterdam	593321
Oran	609823
Dubai	669181
Rosario	907718

TASK 11:

CREATE A SQL STATEMENT TO FIND A CITY WITH THE LOWEST POPULATION IN THE CITY TABLE.

1. select *

from city

order by Population

limit 1;

2. select name, Population

from city

where Population = (select min(Population) from city);

```
48 • select *
49   from city
50   order by Population
51   limit 1;
52
53 • select name, Population
54   from city
55   where Population = (select min(Population) from city);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

name	Population
Adamstown	42

TASK 12:

CREATE A SQL STATEMENT TO SHOW THE POPULATION OF SWITZERLAND AND ALL THE LANGUAGES SPOKEN THERE.

select ct.name, cl.language, ct.population

from countrylanguage as cl

inner join country as ct

on cl.countrycode = ct.code

where CountryCode = 'CHE';

```
60
61 • select ct.name, cl.language, ct.population
62   from countrylanguage as cl
63   inner join country as ct
64   on cl.countrycode = ct.code
65   where CountryCode = 'CHE';
66
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

name	language	population
Switzerland	French	7160400
Switzerland	German	7160400
Switzerland	Italian	7160400
Switzerland	Romansh	7160400

TASK 13:

CREATE A SQL STATEMENT TO FIND THE CAPITAL OF SPAIN (ESP).

select c.id, c.name as City, ct.Name as Country, ct.Capital

from city as c

inner join country as ct

on c.CountryCode = ct.code

where ct.capital = c.ID and ct.name ='spain';

```
66
67 #task 13: CREATE A SQL STATEMENT TO FIND THE CAPITAL OF SPAIN (ESP).#
68 • select c.id, c.name as City, ct.Name as Country, ct.Capital
69 from city as c
70 inner join country as ct
71 on c.CountryCode = ct.code
72 where ct.capital = c.ID and ct.name ='spain';
73
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	City	Country	Capital
▶ 653	Madrid	Spain	653

TASK 14:

CREATE A SQL STATEMENT TO FIND THE COUNTRY WITH THE HIGHEST LIFE EXPECTANCY.

select name, LifeExpectancy

from country

where LifeExpectancy = (select max(LifeExpectancy) from country);

TASK 15:

CREATE A SQL STATEMENT TO FIND ALL CITIES FROM THE EUROPE CONTINENT.

select ct.name as Country, ct.Continent, c.name as City

from country as ct

inner join city as c

on ct.Code = c.CountryCode

where Continent = 'europe';

```

78
79 #task 15; Create a SQL statement to find all cities from the Europe continent.#
80 • select ct.name as Country, ct.Continent, c.name as City
81 from country as ct
82 inner join city as c
83 on ct.Code = c.CountryCode
84 where Continent = 'Europe';
85

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Country	Continent	City
Albania	Europe	Tirana
Andorra	Europe	Andorra la Vella
Austria	Europe	Wien
Austria	Europe	Graz
Austria	Europe	Linz

result 63 x Read O

TASK 16:

CREATE A SQL STATEMENT TO FIND THE MOST POPULATED CITY IN THE CITY TABLE.

1. select *
from city
order by population desc
limit 1;

2. select *
from city
where Population = (select max(Population) from city);

```

88 from city
89 order by population desc
90 limit 1;
91
92 • select *
93 from city
94 where Population = (select max(Population) from city);
95

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

ID	Name	CountryCode	District	Population
1024	Mumbai (Bombay)	IND	Maharashtra	10500000

TASK 17:

CREATE A SQL STATEMENT TO FIND THE TOTAL POPULATION OF EACH CONTINENT
select continent, sum(population) as population

from country

group by Continent;

```
100
101
102 #Task 17: CREATE A SQL STATEMENT TO FIND THE TOTAL POPULATION OF EACH CONTINENT#
103 • select continent, sum(population) as population
104 from country
105 group by Continent;
```

continent	population
North America	482993000
Asia	3705025700
Africa	784475000
Europe	730074600
South America	345780000

Task 18:

CREATE A SQL STATEMENT TO FIND THE AVERAGE LIFE EXPECTANCY BY CONTINENT

1. select continent, avg(lifeexpectancy)

from country

group by Continent;

2. select continent, round(avg(lifeexpectancy),2) as AvgLifeExpectancy

from country

group by Continent;

- 1.

```
100
101 #task 18: CREATE A SQL STATEMENT TO FIND THE AVERAGE LIFE EXPECTANCY BY CONTINENT#
102 • select continent, avg(lifeexpectancy) as AvgLifeExpectancy
103 from country
104 group by Continent;
```

continent	AvgLifeExpectancy
North America	72.99189
Asia	67.44118
Africa	52.57193
Europe	75.14773
South America	70.94615

- 2.

```

101 #task 18: CREATE A SQL STATEMENT TO FIND THE AVERAGE LIFE EXPECTANCY BY CONTINENT#
102 • select continent, avg(lifeexpectancy)
103 from country
104 group by Continent;
105
106 • select continent, round(avg(lifeexpectancy),2) as AvgLifeExpectancy
107 from country
108 group by Continent;

```

continent	AvgLifeExpectancy
North America	72.99
Asia	67.44
Africa	52.57
Europe	75.15
South America	70.95

TASK 19:

CREATE A SQL STATEMENT TO LIST THE NUMBER OF CITIES IN EACH COUNTRY

select ct.name as Country, count(c.name) as 'Number of Cities'

from country as ct

inner join city as c

on ct.code = c.countrycode

group by ct.name;

```

109
110 #Task 19: Create a SQL statement to list the number of cities in each country#
111 • select ct.name as Country, count(c.name) as 'Number of Cities'
112 from country as ct
113 inner join city as c
114 on ct.code = c.countrycode
115 group by ct.name;
116

```

Country	Number of Cities
Aruba	1
Afghanistan	4
Angola	5
Anguilla	2


TASK 20:

CREATE A SQL STATEMENT TO FIND THE TOTAL POPULATION OF EACH COUNTRY BASED ON ITS CITIES

```
1. select ct.name as country, sum(c.population) as 'Total City Population'
from country as ct
inner join city as c
on ct.code = c.CountryCode
group by ct.name;
```

```
2. select ct.name as Country,
ifnull(c.name, 'total') as City,
ifnull(sum(c.population), 0) as CityPopulation
from country as ct
inner join city as c
on ct.code = c.CountryCode
group by ct.name, c.name
with rollup;
```

1.



```
117 #Task 20: Create a SQL statement to find the total population of each country based on its cities#
118 • select ct.name as country, sum(c.population) as 'Total City Population'
119 from country as ct
120 inner join city as c
121 on ct.code = c.CountryCode
122 group by ct.name;
```

country	Total City Population
Aruba	29034
Afghanistan	2332100
Angola	2561600
Anguilla	1556
...	...

Result 97 x Read Only

2.

```

124 • select ct.name as Country,
125        ifnull(c.name, 'total') as City,
126        ifnull(sum(c.population), 0) as CityPopulation
127 from country as ct
128 inner join city as c
129 on ct.code = c.CountryCode
130 group by ct.name, c.name
131 with rollup;

```

Country	City	CityPopulation
Argentina	Tigre	296226
Argentina	Tres de Febrero	352311
Argentina	Vicente López	288341
Argentina	total	19996563
Armenia	Gjumri	211700

TASK 21:

CREATE A SQL STATEMENT TO FIND THE MOST SPOKEN LANGUAGE IN EACH CONTINENT

select ct.name as Country, cl.Language as 'Most Spoken Language', cl.percentage

from countrylanguage as cl

inner join country as ct

on cl.countrycode = ct.code

where cl.percentage = (select max(cl.Percentage) from countrylanguage cl where cl.CountryCode = ct.Code)

order by ct.name;

```

132
133 #Task 21:CREATE A SQL STATEMENT TO FIND THE MOST SPOKEN LANGUAGE IN EACH CONTINENT#
134 • select ct.name as Country, cl.Language as 'Most Spoken Language', cl.percentage
135 from countrylanguage as cl
136 inner join country as ct
137 on cl.countrycode = ct.code
138 where cl.percentage = (select max(cl.Percentage) from countrylanguage cl where cl.CountryCode = ct.Code)
139 order by ct.name;

```

Country	Most Spoken Language	percentage
Afghanistan	Pashto	52.4
Albania	Albaniana	97.9
Algeria	Arabic	86.0
American Samoa	Samoan	90.6

TASK 22:

CREATE A SQL STATEMENT TO FIND COUNTRIES WHERE THE OFFICIAL LANGUAGE IS EITHER

'ENGLISH', 'SPANISH', OR 'FRENCH'

select ct.name as country, cl.Language, IsOfficial

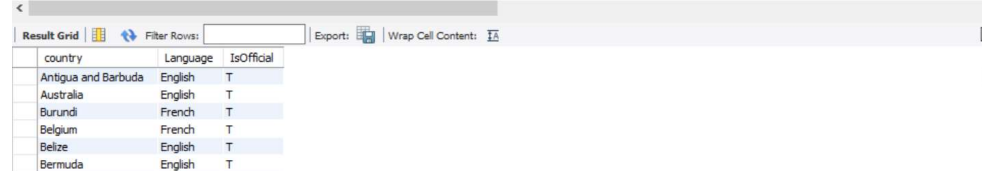
from countrylanguage as cl

inner join country as ct

on cl.countrycode = ct.code

where language in ('English', 'Spanish', 'French') and IsOfficial = 'T';

```
140
141 #CREATE A SQL STATEMENT TO FIND COUNTRIES WHERE THE OFFICIAL LANGUAGE IS EITHER 'ENGLISH', 'SPANISH', OR 'FRENCH'#
142 • select ct.name as country, cl.Language, IsOfficial
143 from countrylanguage as cl
144 inner join country as ct
145 on cl.countrycode = ct.code
146 where language in ('English', 'Spanish', 'french') and IsOfficial = 'T';
```



country	Language	IsOfficial
Antigua and Barbuda	English	T
Australia	English	T
Burundi	French	T
Belgium	French	T
Belize	English	T
Bermuda	English	T

TASK 23:

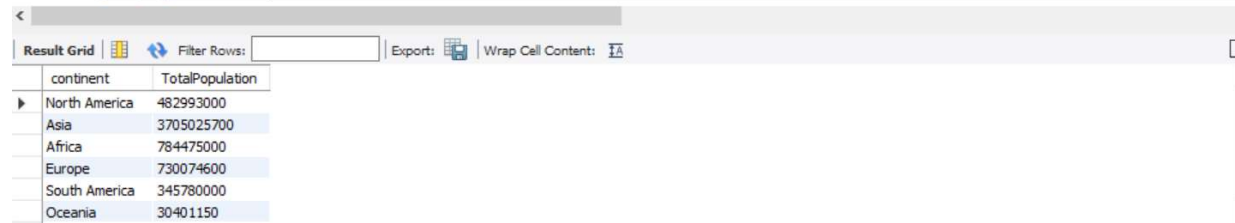
WRITE A QUERY TO DISPLAY THE TOTAL POPULATION FOR EACH CONTINENT.

select continent, sum(population) as TotalPopulation

from country

group by continent;

```
147
148 #Task 23: Write a query to display the total population for each continent.#
149 • select continent, sum(population) as TotalPopulation
150 from country
151 group by continent;
```



continent	TotalPopulation
North America	482993000
Asia	3705025700
Africa	784475000
Europe	730074600
South America	345780000
Oceania	30401150

TASK 24:

WRITE A QUERY TO LIST COUNTRIES THAT HAVE MORE THAN THREE OFFICIAL LANGUAGES. (JOINS, GROUP BY, HAVING)

select ct.name as Country, count(cl.Language) as 'Number of Official languages'

from countrylanguage as cl

inner join country as ct

on cl.countrycode = ct.code

where IsOfficial = 'T'

group by ct.name

having count(cl.Language) > 3;


```

154 • select ct.name as Country, count(c1.Language) as 'Number of Official languages'
155 from countrylanguage as c1
156 inner join country as ct
157 on c1.countrycode = ct.code
158 where IsOfficial = 'T'
159 group by ct.name
160 having count(c1.Language) > 3;
161

```

Country	Number of Official languages
Switzerland	4
South Africa	4

Task 25:

FIND COUNTRIES WHOSE POPULATION IS GREATER THAN THE AVERAGE POPULATION OF ALL COUNTRIES.

select name as Country, Population

from country

where population > (select avg(population) from country);

```

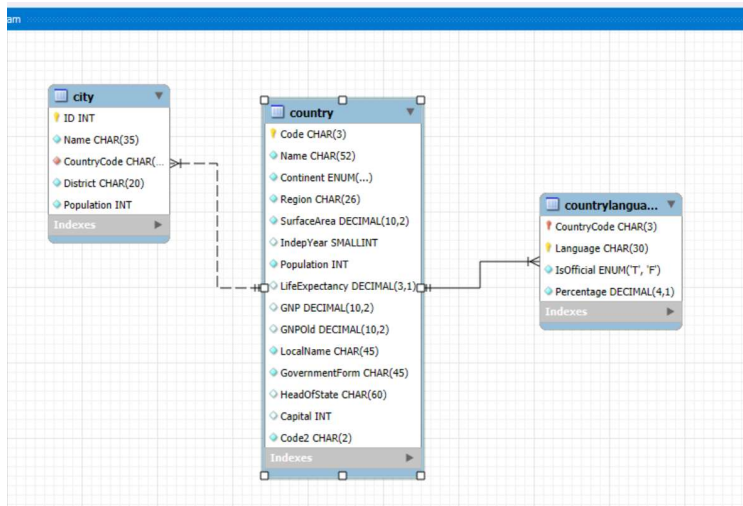
162 #Task 25: Find countries whose population is greater than the average population of all countries.#
163 • select name as Country, Population
164 from country
165 where population > (select avg(population) from country);
166
167 • select avg(population)
168 from country;

```

Country	Population
Argentina	37032000
Bangladesh	129155000
Brazil	170115000
Canada	31147000
China	1277558000
Congo, The Democratic Republic of the	51654000

Task 26:

CREATING AN EER DIAGRAM



TASK 27:

IDENTIFY THE PRIMARY KEY IN COUNTRY TABLE.

- Code

IDENTIFY THE PRIMARY KEY IN CITY TABLE.

- ID

IDENTIFY THE PRIMARY KEY IN COUNTRYLANGUAGE TABLE.

- CountryCode and Language

IDENTIFY THE FOREIGN KEY IN CITY TABLE.

- CountryCode

IDENTIFY THE FOREIGN KEY IN COUNTRYLANGUAGE TABLE.

- CountryCode