

Fall 23 CPSC 5175 – Midterm 2

Youtube Video Link: <https://youtu.be/naO43Crytjw>

SQL Questions And Answers

1. Select the **country code** of the country, followed by the **province** that the city is in, along with the **city name** and find the city or cities with the highest population.

Beijing
Population: 18960744

Screenshot:

The screenshot shows a Microsoft SQL Server Management Studio window. The query window contains the following T-SQL code:

```
--Q1. Select the country code of the country, followed by the province that the city is in, along with the city name and find the city or cities with the highest population.
Select *
From Country country
Inner Join City city on country.Capital = city.Name
Select Country.Code AS country_code,Country.Capital,City.Name AS city_name,
MAX(city.population) AS population
FROM Country country
Inner Join City city on country.Capital = city.Name
Group by country.Code,country.Capital,City.Name
Order by Population DESC
/**ANSWER: Bejing pop: 18960744**/
```

The results window displays a table with the following data:

	country_code	Capital	city_name	population
1	CN	Beijing	Beijing	18960744
2	R	Moskva	Moskva	13010112
3	CGO	Kinshasa	Kinshasa	11575000
4	IND	Delhi	Delhi	11034555
5	MW	Lilongwe	Lilongwe	9893188
6	RI	Jakarta	Jakarta	9607787
7	ROK	Seoul	Seoul	9586195
8	ET	Al Qahirah	Al Qahirah	9293612
9	J	Tokyo	Tokyo	9272740
10	GB	London	London	8888608
11	MEX	Ciudad de México	Ciudad de México	8843706
12	IR	Tehran	Tehran	8693706
13	PE	Lima	Lima	8574974
14	THA	Bangkok	Bangkok	8305218
15	BD	Dhaka	Dhaka	7423137
16	HK	Hong Kong	Hong Kong	7413070
17	CO	Bogotá	Bogotá	7387400

The status bar at the bottom indicates "Query executed successfully." and "246 rows".

2.

- Pull each country together with the count of neighboring countries. For simplicity exclude the countries that have no neighbors. Your resulting Select clause should have the **country code**, followed by the **count** of neighboring countries.

Screenshot:

The screenshot shows a Microsoft SQL Server Management Studio window. The query window contains the following T-SQL code:

```
16 --Q3 part A. Pull each country together with the count of neighboring countries. For simplicity exclude the countries that have no neighbors. Your resulting Select clause should have the country code, followed by the count of neighboring countries.
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
```

The results pane displays a table with columns: CountryName, CountryCode, and num_bordering_countries. The data is as follows:

CountryName	CountryCode	num_bordering_countries
China	CN	14
Brazil	BR	9
Austria	A	8
Congo Dem Rep.	DGO	7
Algeria	DZ	7
Cameroun	CAM	6
Yemen	YEM	6
Hungary	HU	6
Iraq	IRQ	6
Slovenia	SLV	5
Burkina Faso	BF	5
Yemen	YEM	5
Baluchistan	BY	5
Germany	DE	5
Yugoslavia	YAK	5
Tanzania	EAT	5
Mosambique	MOC	5

- Now also include the countries with no neighbors. Hint: Use an outer join. Select the **country code** of the country followed by the **count** of neighboring countries.

Screenshot:

The screenshot shows a Microsoft SQL Server Management Studio window. The query window contains the following T-SQL code:

```
32 --Q3 part b. Now also include the countries with no neighbors. Hint: Use an outer join. Select the country code of the country followed by the count of neighboring countries...
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
```

The results pane displays a table with columns: CountryName, CountryCode, and num_bordering_countries. The data is as follows:

CountryName	CountryCode	num_bordering_countries
China	CN	14
Brazil	BR	9
Austria	A	8
Congo Dem Rep.	DGO	7
Algeria	DZ	7
Cameroun	CAM	6
Yemen	YEM	6
Hungary	HU	6
Iraq	IRQ	6
Slovenia	SLV	5
Burkina Faso	BF	5
Yemen	YEM	5
Baluchistan	BY	5
Germany	DE	5
Yugoslavia	YAK	5
Tanzania	EAT	5
Mosambique	MOC	5
Burkina Faso	BF	5
Bulgaria	BG	5
Malta	MLT	5
Bolivia	BOL	5
Germany	D	5
Yugoslavia	YAK	5
Tanzania	EAT	5
Colombia	CO	4
Egypt	ET	4
Benin	BEN	4
Switzerland	CH	4
Cote d'Ivoire	CI	4
Albania	AL	4

3. List the **name** and **percentage** of all the ethnic groups in the United States such that the percentage of the population of each of the ethnic group is lower than the average percentage across all ethnic groups in the United States?

Screenshot:

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'Questions.sql - DE_7JTRF4P\Tina (53)' is open, displaying the following T-SQL code:

```

54 -- Q3. List the name and percentage of all the ethnic groups in the United States such that the percentage of the population of each of the ethnic group is lower than the average percentage across all ethnic groups in the United States?-
55
56 Select *
57 From EthnicGroup
58
59 --To view all data in the USA---
60 Select *
61 From EthnicGroup
62 Where Country = 'USA'
63 Order By Percentage desc
64
65 --Calculate the avg ethnic percentage in the USA, so our results should be less than 3.746%
66 Select AVG(percentage) as avg_percentage
67 From EthnicGroup
68 Where Country = 'USA'
69
70 --Final answer--
71 Select Name as EthnicGroup, Percentage
72 From EthnicGroup
73 Where Country = 'USA'
74 AND Percentage < (select AVG(percentage) as AvgUSAEthnicPercentage From EthnicGroup where Country = 'USA')
75
76 Group by name, Percentage
77 Order by Percentage desc;
78
79

```

The results pane shows a table with two columns: 'EthnicGroup' and 'Percentage'. The data is as follows:

	EthnicGroup	Percentage
1	French	3
2	Polish	2.8
3	Scot	2.7
4	Jewish	2.1
5	Norwegian	1.3
6	Dutch	1.2
7	Chinese	1.2
8	Filipino	1.1
9	Indian	1
10	Arab	0.6
11	Korean	0.5
12	Vietnamese	0.5
13	Japanese	0.4
14	Cherokee	0.26
15	Hawaiian	0.17
16	Navajo	0.1
17	Yupik	0.01
18	Dakota	0.01

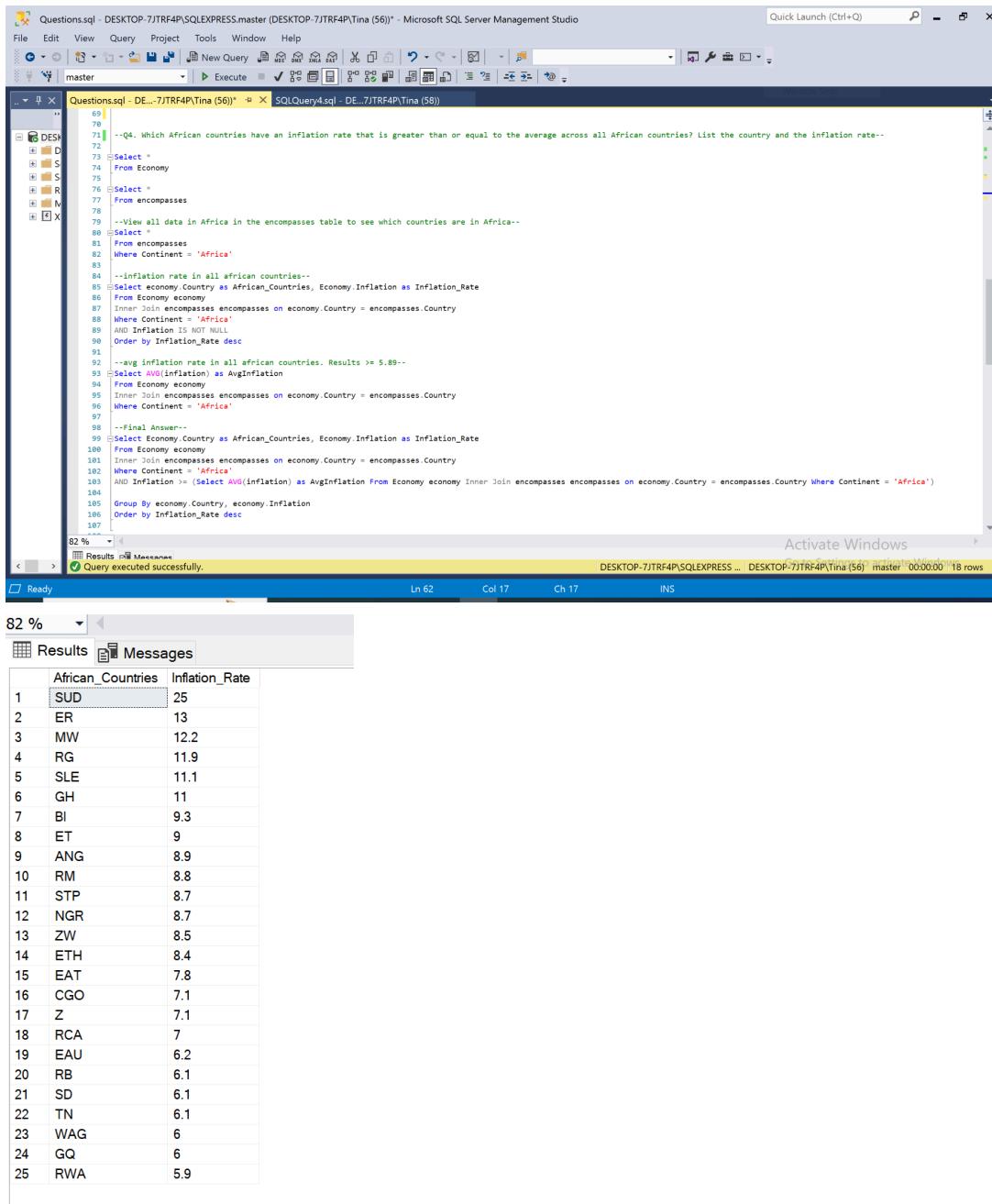
A message at the bottom of the results pane says 'Query executed successfully.' and '18 rows'.

This is a zoomed-in view of the 'Results' tab from the previous screenshot. It displays the same table of ethnic groups and their percentages, ordered by percentage in descending order. The table has two columns: 'EthnicGroup' and 'Percentage'.

	EthnicGroup	Percentage
1	French	3
2	Polish	2.8
3	Scot	2.7
4	Jewish	2.1
5	Norwegian	1.3
6	Dutch	1.2
7	Chinese	1.2
8	Filipino	1.1
9	Indian	1
10	Arab	0.6
11	Korean	0.5
12	Vietnamese	0.5
13	Japanese	0.4
14	Cherokee	0.26
15	Hawaiian	0.17
16	Navajo	0.1
17	Yupik	0.01
18	Dakota	0.01

4. Which African countries have an inflation rate that is greater than or equal to the average across all African countries? List the **country** and the **inflation rate**

Screenshot:



The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays the following SQL code:

```

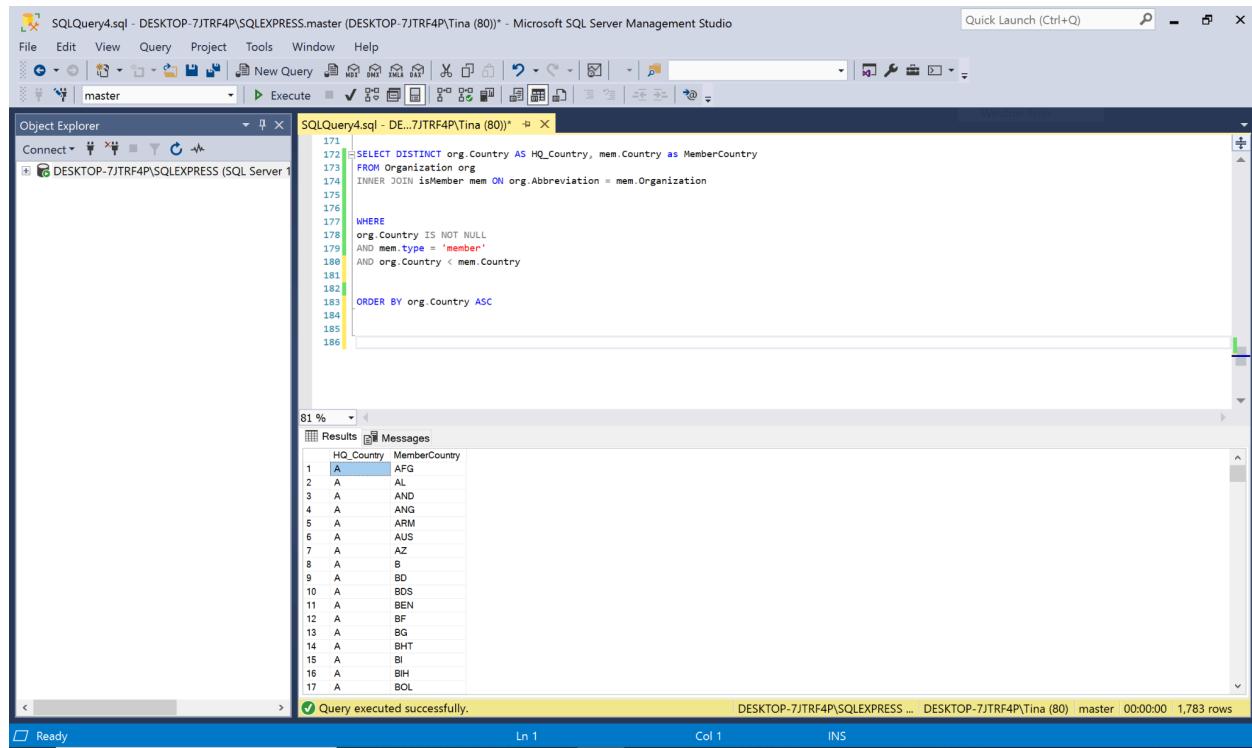
69 --Q4. Which African countries have an inflation rate that is greater than or equal to the average across all African countries? List the country and the inflation rate-
70
71 --Select *
72 --From Economy
73 --Select *
74 --From encompasses
75 --View all data in Africa in the encompasses table to see which countries are in Africa--
76 --Select *
77 --From encompasses
78 --Where Continent = 'Africa'
79
80 --inflation rate in all african countries--
81 Select economy.Country as African_Countries, Economy.Inflation as Inflation_Rate
82 From Economy
83 Inner Join encompasses encompass ON economy.Country = encompasses.Country
84 Where Continent = 'Africa'
85 AND Inflation IS NOT NULL
86 Order by Inflation_Rate desc
87
88 --avg inflation rate in all african countries. Results >= 5.89--
89 Select AVG(inflation) as AvgInflation
90 From Economy
91 Inner Join encompasses encompass ON economy.Country = encompasses.Country
92 Where Continent = 'Africa'
93
94 --Final Answer--
95 Select Economy.Country as African_Countries, Economy.Inflation as Inflation_Rate
96 From Economy
97 Inner Join encompasses encompass ON economy.Country = encompasses.Country
98 Where Continent = 'Africa'
99 AND Inflation >= (Select AVG(inflation) as AvgInflation From Economy)
100 Inner Join encompasses encompass ON economy.Country = encompasses.Country Where Continent = 'Africa')
101
102 Group By economy.Country, economy.Inflation
103 Order by Inflation_Rate desc
104
105
106
107

```

The status bar at the bottom indicates "Query executed successfully." The results grid shows the following data:

	African_Countries	Inflation_Rate
1	SUD	25
2	ER	13
3	MW	12.2
4	RG	11.9
5	SLE	11.1
6	GH	11
7	BI	9.3
8	ET	9
9	ANG	8.9
10	RM	8.8
11	STP	8.7
12	NGR	8.7
13	ZW	8.5
14	ETH	8.4
15	EAT	7.8
16	CGO	7.1
17	Z	7.1
18	RCA	7
19	EAU	6.2
20	RB	6.1
21	SD	6.1
22	TN	6.1
23	WAG	6
24	GQ	6
25	RWA	5.9

5. Provide a list of all pairs of countries that are members of the same organizations. List each pair (A, B) only once by including only (A, B) but not the symmetric pair (B, A) in the result (hint: use "<" to compare the strings). Exclude countries that are members of no organizations. Select the pair of **country codes**.



The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays the following T-SQL code:

```
171
172 --SELECT DISTINCT org.Country AS HQ_Country, mem.Country AS MemberCountry
173 FROM Organization org
174 INNER JOIN isMember mem ON org.Abbreviation = mem.Organization
175
176
177 WHERE
178 org.Country IS NOT NULL
179 AND mem.type = 'member'
180 AND org.Country < mem.Country
181
182 ORDER BY org.Country ASC
183
184
185
186
```

The results pane shows a table with two columns: HQ_Country and MemberCountry. The data is as follows:

HQ_Country	MemberCountry
A	AFG
A	AL
A	AND
A	ANG
A	ARM
A	AUS
A	AZ
A	B
A	BD
A	BDS
A	BEN
A	BF
A	BG
A	BHT
A	BI
A	BIH
A	BOL

Below the results, a message indicates "Query executed successfully." and provides system information: DESKTOP-7JTRF4P\SQLEXPRESS ... DESKTOP-7JTRF4P\Tina (80) master 00:00:00 1,783 rows.

STEP-BY-STEP PROCESS

QUESTION 1:

- Connect Data Base:
 - Download the zip file
 - Open 'schema' and 'mondial-inputs-mssql' file in SMMS
 - Create a database:
 - on the 'mondial-inputs-mssql' file:
 - Line 1: CREATE DATABASE mondial

Line 2: GO

NOTE: To double check if that worked, go to the side bar and refresh. Click on databases, you should see 'mondial' as part of it.

- Execute both files
- Copy and paste each file to their own query and execute
- Open a new query for question 1
 - To check if everything has loaded correctly:

```
SELECT *  
FROM City  
execute
```

- look at each table on its own first, decide if you need to join tables

```
Select *  
From Country
```

```

1  /***Select the country code of the country, followed by the province that the city is in,
2   | along with the city name and find the city or cities with the highest population.***/
3
4  /***look at each table on its own first, decide if you need to join tables***/
5  Select *
6    From City
7
8  Select *
9    From Country
10
11

```

- It was decided that we needed to do left join. To see the entire table itself to see what you're pulling from:

country_code	Province	city_name	population
CN	Beijing	Beijing	18960744
R	Moscow	Moskva	13010112
CGO	Kinshasa	Kinshasa	11575000
IND	Delhi	Delhi	11034655
MW	Malawi	Lilongwe	9893188
RI	DKI Jakarta	Jakarta	9607787
ROK	South Korea	Seoul	9586195
ET	Egypt	Al Qahirah	923612
J	Tokyo	Tokyo	9272740
GB	London	London	8888608
BR	District Federal	Ciudad de México	8843706
MEX	District Federal	Ciudad de México	8843706
RA	District Federal	Ciudad de México	8843706
YV	District Federal	Ciudad de México	8843706
IR	Tehran	Tehran	8693706
PE	Lima City	Lima	8574974
THA	Thailand	Bangkok	8305218

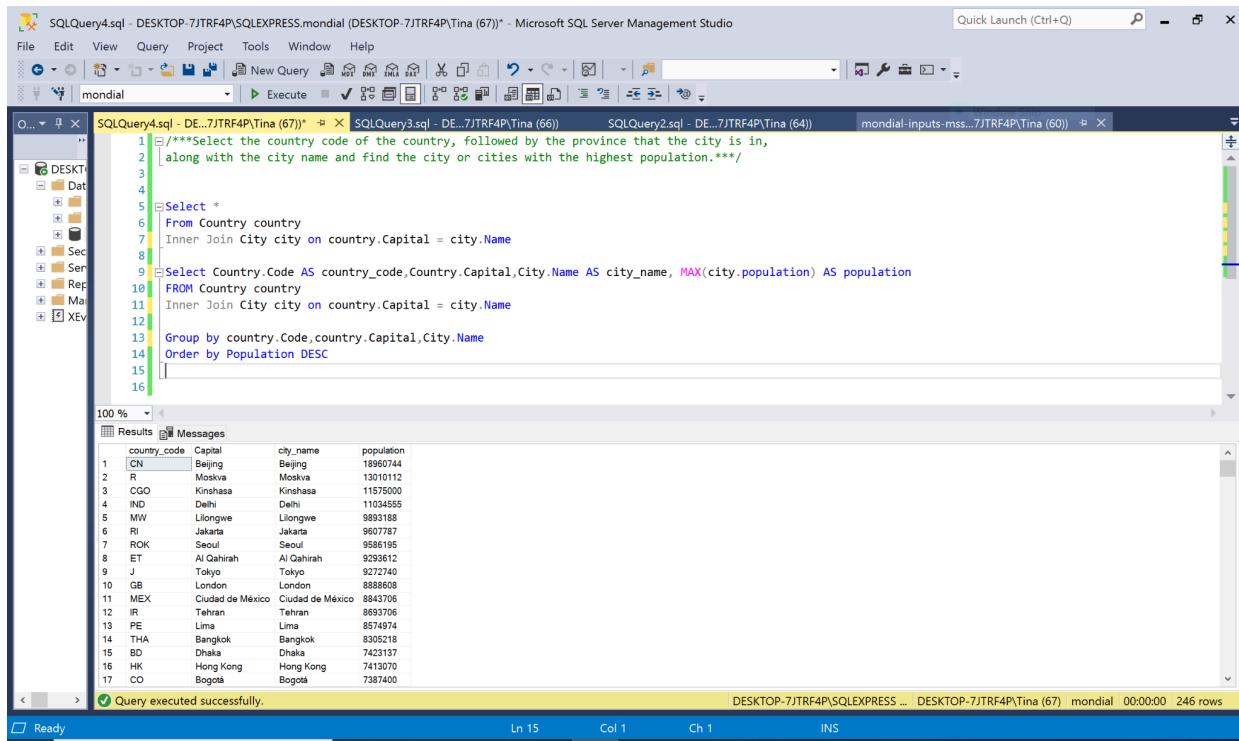
Query executed successfully.

Answer Beijing

As you can see there seems to be an error in the database...the highlighted above keeps repeating. The CODE needs to be the primary key not PROVINCE. So to fix that:

Look at ER Diagram to find unique value that connects city and country...seems to be capital. It is on both tables, but named differently (name and capital)

Make changes to your code:



The screenshot shows the Microsoft SQL Server Management Studio interface. In the center, there is a results grid titled 'Results' showing a list of cities from the mondial database. The columns are labeled 'country_code', 'Capital', 'city_name', and 'population'. The data includes major cities like Beijing, Moskva, Kinshasa, Delhi, Lilongwe, Jakarta, Seoul, Al Qahirah, Tokyo, London, Ciudad de México, Tehran, Lima, Bangkok, Dhaka, Hong Kong, and Bogotá, along with their respective populations.

country_code	Capital	city_name	population
CN	Beijing	Beijing	18960744
RU	Moskva	Moskva	13010112
CGO	Kinshasa	Kinshasa	11575000
IND	Delhi	Delhi	11034555
MW	Lilongwe	Lilongwe	9893188
RI	Jakarta	Jakarta	9607787
ROK	Seoul	Seoul	9586195
ET	Al Qahirah	Al Qahirah	9293612
J	Tokyo	Tokyo	9272740
GB	London	London	8886608
MEX	Ciudad de México	Ciudad de México	8843706
IR	Tehran	Tehran	8693706
PE	Lima	Lima	8574974
THA	Bangkok	Bangkok	8305218
BD	Dhaka	Dhaka	7423137
HK	Hong Kong	Hong Kong	7413070
CO	Bogotá	Bogotá	7387400

QUESTION 2:

1. Take a look at the RS pdf file. Border table seems to be the best fit for this question.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'master' database is selected. In the center pane, a query is being run against the 'master' database:

```
SQLQuery4.sql - DESKTOP-7JTRF4P\Tina (65)* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
File New Query Execute
master
SQLQuery4.sql - DESKTOP-7JTRF4P\Tina (65)* SQLQuery2.sql - not connected schema.sql - not connected SQLQuery3.sql - not connected
SQLQuery4.sql - DE...7JTRF4P\Tina (65)*
22 /**
23 * For simplicity exclude the countries that have no neighbors.
24 Your resulting Select clause should have the country code, followed by the count of neighboring countries.
25 */
26
27 Select *
28 From borders
29
30
31 Select
32 Country1,
33 COUNT(1) As num_neighbors
34 From borders
35 Group By Country1
```

The results pane displays the output of the query:

Country1	num_neighbors
A	8
AD	1
AFG	6
AL	4
AND	2
ANG	4
ARM	4
AZ	4
B	4
BD	2
BEN	4
BF	5
BG	5
BHT	2
BI	3
BIH	3
BOL	5

At the bottom of the screen, a message indicates the query was executed successfully.

this is wrong!

This is why:

once you get to the second sentence of 'excluding the countries that have no neighbors' you run into an issue. This is where you realize, if you look at the borders table compared to the countries table, that the borders table does not include the countries with no neighbors already. So what you have to do is do a join.

When you write it this way, it only counted the repeating code in country1 column not actually counting the number of countries that surround that country (the one in country1 column).

Remember: an inner join only pulls matching records from both tables. an outer join will pull matching and unique values (so everything).

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "SQLQuery4.sql - DESKTOP-7JTRF4P\SQLEXPRESS.master (DESKTOP-7JTRF4P\Tina (65)) - Microsoft SQL Server Management Studio". The query editor window contains the following SQL code:

```

21 /**
22 **Q2 part A. Pull each country together with the count of neighboring countries.
23 For simplicity exclude the countries that have no neighbors.
24 Your resulting Select clause should have the country code, followed by the count of neighboring countries.
25 */
26
27 Select *
28 From borders
29
30 Select *
31 From Country
32
33 Select Country.Name as CountryName, Country.Code as CountryCode, COUNT(country1) as num_bordering_countries
34 From Country country
35 Inner Join borders borders on country.Code = borders.Country1
36
37 Group by country.Name, country.Code
38 Order by num_bordering_countries desc
39
40 /**
41 **Q2 part B. Now also include the countries with no neighbors. Hint: Use an outer join.
42 Select the country code of the country followed by the count of neighboring countries.**/
43
44 Select distinct country.name as CountryName, Country.Code as CountryCode, COUNT(country2) as num_bordering_countries
45 From Country country
46 Full Outer Join borders borders on country.Code = borders.Country1

```

The results grid shows the following data:

	CountryName	CountryCode	num_bordering_countries
1	China	CN	14
2	Brazil	BR	9
3	Austria	A	8
4	Congo, Dem.Rep.	CGO	7
5	Algeria	DZ	7
6	Cameroon	CAM	6
7	Afghanistan	AFG	6
8	Himana	H	6

Below the results grid, a message bar says "Query executed successfully."

To fix:

Taking a look at the borders and country table, all the countries that do not have neighbors are not included in the borders table. We want to combine all the MATCHING countries in the countries table to the borders table.

Syntax:

SELECT

table_A.columnX,

table_A.columnY,

table_B.columnZ

FROM table_A

INNER JOIN table_b ON table_A.foreign_key = table_B.primary_key;

Code: #2 part A

```
Select Country.Name as CountryName, Country.Code as CountryCode,  
COUNT(country1) as num_neighboring_countries  
From Country country  
Inner Join borders borders on country.Code = borders.Country1  
Group by country.Name, country.Code  
Order by num_neighboring_countries desc
```

Getting the count of neighboring countries

from the country table named country

table A (country) joins table B (borders) based on the foreign key "Code" in the countries table, this should match the primary key "country1" in the borders table.

Country's Code column will join/combine into Borders Country1 column. It will only pull matching results.

The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

so you're wanting to find the num of neighboring countries for each country so you want to group by the country code. We also added country name to make things look cohesive and easier to read and understand what is going on.

We could stop at Group by, but if we wanted to see the country with the most neighboring countries, we could use order by function in descending order

Code: #2 part B

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center, the SQL Query Editor window displays the following T-SQL code:

```
39
40 /*Q2 part B. Now also include the countries with no neighbors. Hint: Use an outer join.
41 Select the country code of the country followed by the count of neighboring countries.*/
42
43 Select distinct country.name as CountryName, Country.Code as CountryCode, COUNT(country2) as num_bordering_countries
44 From Country country
45 Full Outer Join borders borders on country.Code = borders.Country1
46
47 Group by country.Name, country.Code
48 Order by num_bordering_countries desc
49
```

Below the editor, the Results pane shows the output of the query:

	CountryName	CountryCode	num_bordering_countries
1	China	CN	14
2	Brazil	BR	9
3	Austria	A	8
4	Congo, Dem.Rep.	CGO	7
5	Algeria	DZ	7
6	Cameroon	CAM	6
7	Afghanistan	AFG	6
8	Hungary	H	6

At the bottom of the Results pane, a message indicates: "Query executed successfully."

The **FULL OUTER JOIN** keyword returns all records when there is a match in left (table1) or right (table2) table records.

Tip: **FULL OUTER JOIN** and **FULL JOIN** are the same.

Now we want to include the countries with no neighbors (ie cuba) if you look at the current results from the above step, cuba is not there. That is a way to check later to see if this is done correctly.

We use outer join because we want to combine all unique and matching values(everything) into the resulting table.

Repeat the same thing except Full Outer Join. And use COUNT(country2) instead. Because country 2 names the actual bordering countries. Country 1 just repeats the primary country that we joined on.

Question 3:

1. View EthnicGroup table
2. View all data in the USA

Select *

From EthnicGroup

Where Country = 'USA'

Order By Percentage desc

3. Calculate the avg ethnic percentage in the USA, so our results should be less than 3.746%. I double checked by selecting the entire percentage column in usa and copy and pasting into an excel sheet to find the avg. I confirmed that the avg is 3.746%.

Select AVG(percentage) as avg_percentage

From EthnicGroup

Where Country = 'USA'

4. I have to define USA in the main and subquery. The first WHERE statement tells us to look for only in the US ethnic groups and the second one makes sure that we are looking at the US ethnic groups with the percentage lower than the average, which makes it more specific.

REMEMBER: Group By is required for most queries and Order By is good practice.

SQLQuery4.sql - DESKTOP-7JTRF4P\SQLEXPRESS.master (DESKTOP-7JTRF4P\Tina (56)) - Microsoft SQL Server Management Studio

```

54 -- Q3. List the name and percentage of all the ethnic groups in the United States such that the percentage of the population of each of the ethnic group is lower than the
55 -- average percentage across all ethnic groups in the United States?-
56 Select *
57 From EthnicGroup
58
59 --To view all data in the USA---
60 Select *
61 From EthnicGroup
62 Where Country = 'USA'
63 Order By Percentage desc
64
65 --Calculate the avg ethnic percentage in the USA, so our results should be less than 3.746%
66 Select AVG(percentage) as avg_percentage
67 From EthnicGroup
68 Where Country = 'USA'
69
70 --Final answer--
71 Select Name as EthnicGroup, Percentage
72 From EthnicGroup
73 Where Country = 'USA'
74 AND Percentage < (select AVG(percentage) as AvgUSAEthnicPercentage From EthnicGroup where Country = 'USA')
75
76 Group by name, Percentage
77 Order by Percentage desc
78
79

```

Results

EthnicGroup	Percentage
French	3
Polish	28
Scot	27
Afghan	21
Norwegian	13
Dutch	12
Chinese	12

Activate Windows

Query executed successfully.

DESKTOP-7JTRF4P\SQLEXPRESS ... DESKTOP-7JTRF4P\Tina (56) master 00:00:00 18 rows

Question 4

SQLQuery4.sql - DESKTOP-7JTRF4P\SQLEXPRESS.master (DESKTOP-7JTRF4P\Tina (56)) - Microsoft SQL Server Management Studio

```

77 Order by Percentage desc
78
79
80 -- Q4. Which African countries have an inflation rate that is greater than or equal to the average across all African countries? List the country and the inflation
81 -- rate--
82 Select *
83 From Economy
84
85 Select *
86 From encompasses
87
88 --View all data in Africa in the encompasses table to see which countries are in africa--
89 Select *
90 From encompasses
91 Where Continent = 'Africa'
92
93
94 --inflation rate in all african countries--
95 Select Economy.Country as African_Countries, Economy.Inflation as inflation_rate
96 From Economy economy
97 Inner Join encompasses encompasses on economy.Country = encompasses.Country
98 Where Continent = 'Africa'
99 AND Inflation IS NOT NULL
100 Order by inflation_rate desc
101
102 --avg inflation rate in all african countries--
103 Select AVG(inflation) as AvgInflation
104 From Economy economy
105 Inner Join encompasses encompasses on economy.Country = encompasses.Country
106 Where Continent = 'Africa'
107
108 --Wrong Answer--
109 Select Economy.Country as African_Countries, Economy.Inflation as Inflation_Rate
110 From Economy economy
111 Inner Join encompasses encompasses on economy.Country = encompasses.Country

```

Results

Activate Windows

Query executed successfully.

DESKTOP-7JTRF4P\SQLEXPRESS ... DESKTOP-7JTRF4P\Tina (56) master 00:00:00 18 rows

1. View both economy and encompasses tables. If you look at the RS pdf file, encompasses shows information to which continents a country belongs to.
2. view all data in Africa in the encompasses table to see which countries are in africa. There should be 60 countries. Notice that there are some null values.
3. Find the inflation rate in all african countries. Be sure to remove the null values.
4. find the average inflation rate in all african countries

initial answer: This is WRONG!

The screenshot shows a Microsoft SQL Server Management Studio interface. The query window contains the following SQL code:

```

--avg inflation rate in all african countries--
=Select AVG(Inflation) as AvgInflation
From Economy economy
Inner Join encompasses encompassess on economy.Country = encompasses.Country
Where Continent = 'Africa'
--Wrong Answer-
--Avg Inflation per Country as African_Countries, Economy.Inflation as Inflation_Rate
From Economy economy
Inner Join encompasses encompassess on economy.Country = encompasses.Country
Where Continent = 'Africa'
AND Inflation >= (select AVG(Inflation) as AvgAfricanInflation from economy where Continent = 'Africa')
Group by economy.Country, economy.Inflation
Order by inflation_rate desc

```

The results grid displays the following data:

African_Countries	Inflation_Rate
ET	9
ANG	8.9
RM	8.8
STP	8.7
NGR	8.7
ZW	8.5
ETH	8.4
EAT	7.8
CGO	7.1
Z	7.1
RCA	7
EAU	6.2
RB	6.1
TN	6.1
SD	6.1
WAG	6
GQ	6
RWA	5.9
RSA	5.8
EAK	5.8
NAM	5.5
LB	5.2
LS	5

At the bottom of the results grid, it says "Activate Windows". Below the grid, a status bar shows "Query executed successfully." and "30 rows".

Why:

The results come up with inflation rates lower than the inflation avg of african countries, which is: 5.896. The results should have ended on 25.

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "SQLQuery4.sql - DESKTOP-7JTRF4P\SQLEXPRESS.master (DESKTOP-7JTRF4P\Tina (S6)) - Microsoft SQL Server Management Studio". The main window contains a query editor with the following code:

```

103 Order by inflation_rate desc
104
105 --avg inflation rate in all african countries--
106 Select AVG(inflation) as AvgInflation
107 From Economy.economy
108 Inner Join encompasses encompasses.on economy.Country = encompasses.Country
109 Where Continent = 'Africa'
110
111 --Wrong Answer--
112 Select Economy.Country as African_Countries, Economy.Inflation as Inflation_Rate
113 From Economy.economy
114 Inner Join encompasses encompasses.on economy.Country = encompasses.Country
115 Where Continent = 'Africa'
116 AND Inflation >= (select AVG(inflation) as AvgAfricanInflation from economy where Continent = 'Africa')
117
118
119 Group by economy.Country, economy.Inflation
120 Order by inflation_rate desc

```

The code is highlighted in blue, indicating syntax errors. A message box in the center says "Msg 207, Level 16, State 1, Line 116 Invalid column name 'Continent'." Below the message, it says "Completion time: 2024-02-28T16:28:29.6031500-05:00". At the bottom of the editor, a yellow bar says "Query completed with errors". The status bar at the bottom right shows "DESKTOP-7JTRF4P\SQLEXPRESS ... DESKTOP-7JTRF4P\Tina (S6) master 00:00:00 0 rows".

If you highlight the subquery and execute it, it will tell you the error. 'Continent is not connected'. We need to define Continent since it isn't in the economy table. We have already done the work. The –avg inflation rate in all african countries– query is what is missing in the final query. Just copy and paste into the subquery.

The first inner join was to join is to find the countries we found in africa in the economy table. The second inner join is to find the average inflation rate in all african countries where we join the encompasses table with the economy table to get the continent column

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "SQLQuery4.sql - DESKTOP-7JTRF4P\SQLEXPRESS.master (DESKTOP-7JTRF4P\Tina (56)) - Microsoft SQL Server Management Studio". The Object Explorer sidebar shows the database structure for "DESKTOP-7JTRF4P\SQLEXPRESS". The main pane displays a T-SQL query and its results.

```
121
122
123
124 --Final Answer--
125 Select Economy_Country as African_Countries, Economy_Inflation as Inflation_Rate
126 From Economy economy
127 Inner Join encompasses encompasses on economy.Country = encompasses.Country
128 Where Continent = 'Africa'
129 AND Inflation >= ( Select AVG(inflation) as AvgInflation From Economy economy Inner Join encompasses encompasses on economy.Country =
    encompasses.Country Where Continent = 'Africa')
130
131
132 Group by economy.Country, economy.Inflation
133 Order by inflation_rate desc
```

The Results grid shows the following data:

African_Countries	Inflation_Rate
SLE	11.1
GH	11
BI	9.3
ET	9
ANG	8.9
RM	8.8
STP	8.7
NGR	8.7
ZW	8.5
ETH	8.4
CAT	7.8
CGO	7.1
Z	7.1
RCA	7
EAU	6.2
RB	6.1
SD	6.1
TN	6.1
WAG	6
GQ	6
RWA	5.9

At the bottom, a message bar says "Query executed successfully." and shows the execution details: "DESKTOP-7JTRF4P\SQLEXPRESS ... DESKTOP-7JTRF4P\Tina (56) master 00:00:00 25 rows".

This is the final answer.

Question 5:

1. look at each table
2. Notice that there are some null values
3. Inner join is needed

SQLQuery4.sql - DESKTOP-7JTRF4P\SQLEXPRESS.master (DESKTOP-7JTRF4P\Tina (80)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Execute

master

Object Explorer

SQLQuery4.sql - DE...7JTRF4P\Tina (80)*

```
171 SELECT DISTINCT org.Country AS HQ_Country, mem.Country AS MemberCountry
172 FROM Organization org
173 INNER JOIN isMember mem ON org.Abbreviation = mem.Organization
174
175
176
177 WHERE
178 org.Country IS NOT NULL
179 AND mem.type = 'member'
180 AND org.Country < mem.Country
181
182
183 ORDER BY org.Country ASC
184
185
186
```

81 %

Results Messages

	HQ_Country	MemberCountry
1	A	AFG
2	A	AL
3	A	AND
4	A	ANG
5	A	ARM
6	A	AUS
7	A	AZ
8	A	B
9	A	BD
10	A	BDS
11	A	BEN
12	A	BF
13	A	BG
14	A	BHT
15	A	BI
16	A	BIH
17	A	BOL

Query executed successfully.

DESKTOP-7JTRF4P\SQLEXPRESS ... DESKTOP-7JTRF4P\Tina (80) master 00:00:00 1,783 rows

Ready Ln 1 Col 1 INS