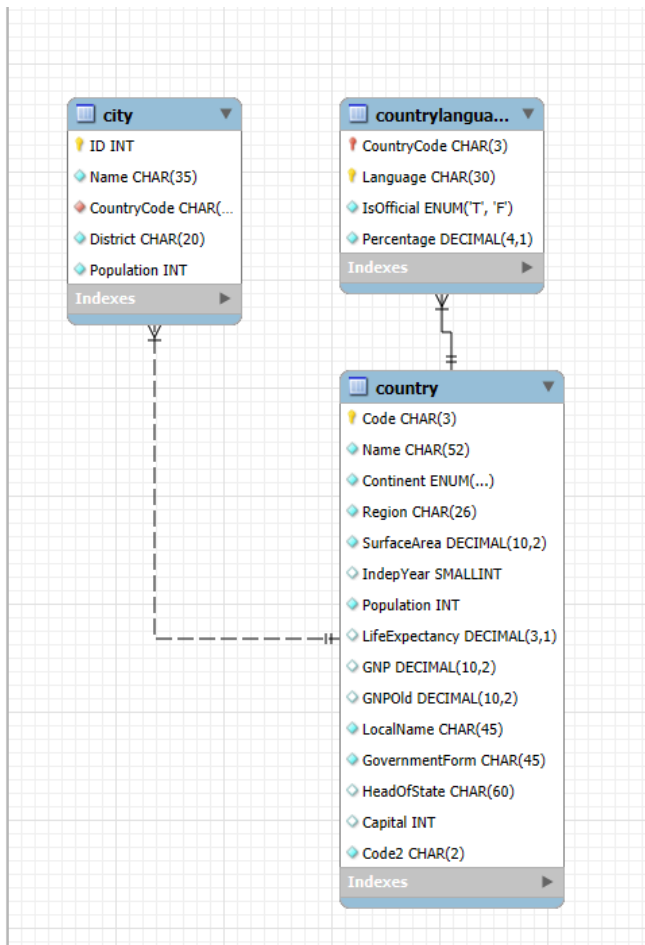


Part 1: Install the World Database

```
5433 • /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
5434 • /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
5435 • SET autocommit=@old_autocommit;
5436
5437 -- Dump completed on 2020-01-22 9:56:18
5438
5439 • SHOW TABLES;
```

Result Grid	Filter Rows: <input type="text"/>	Export:	Wrap Cell Content:
Tables_in_world			
city			
country			
countrylanguage			

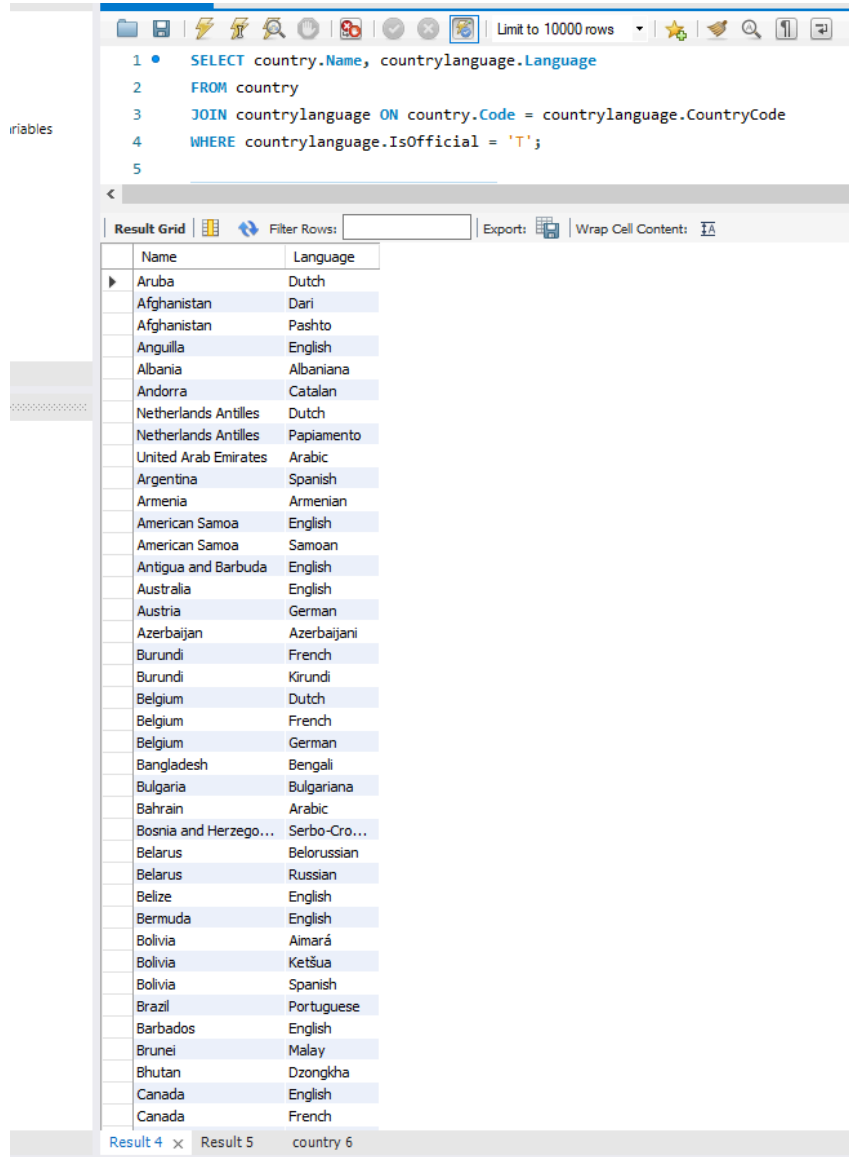
Part 2: Reverse Engineer the Database



Part 3: SQL Queries

Basic Queries

1. Retrieve all country names and their official languages.



The screenshot shows a database query tool interface. The SQL query is as follows:

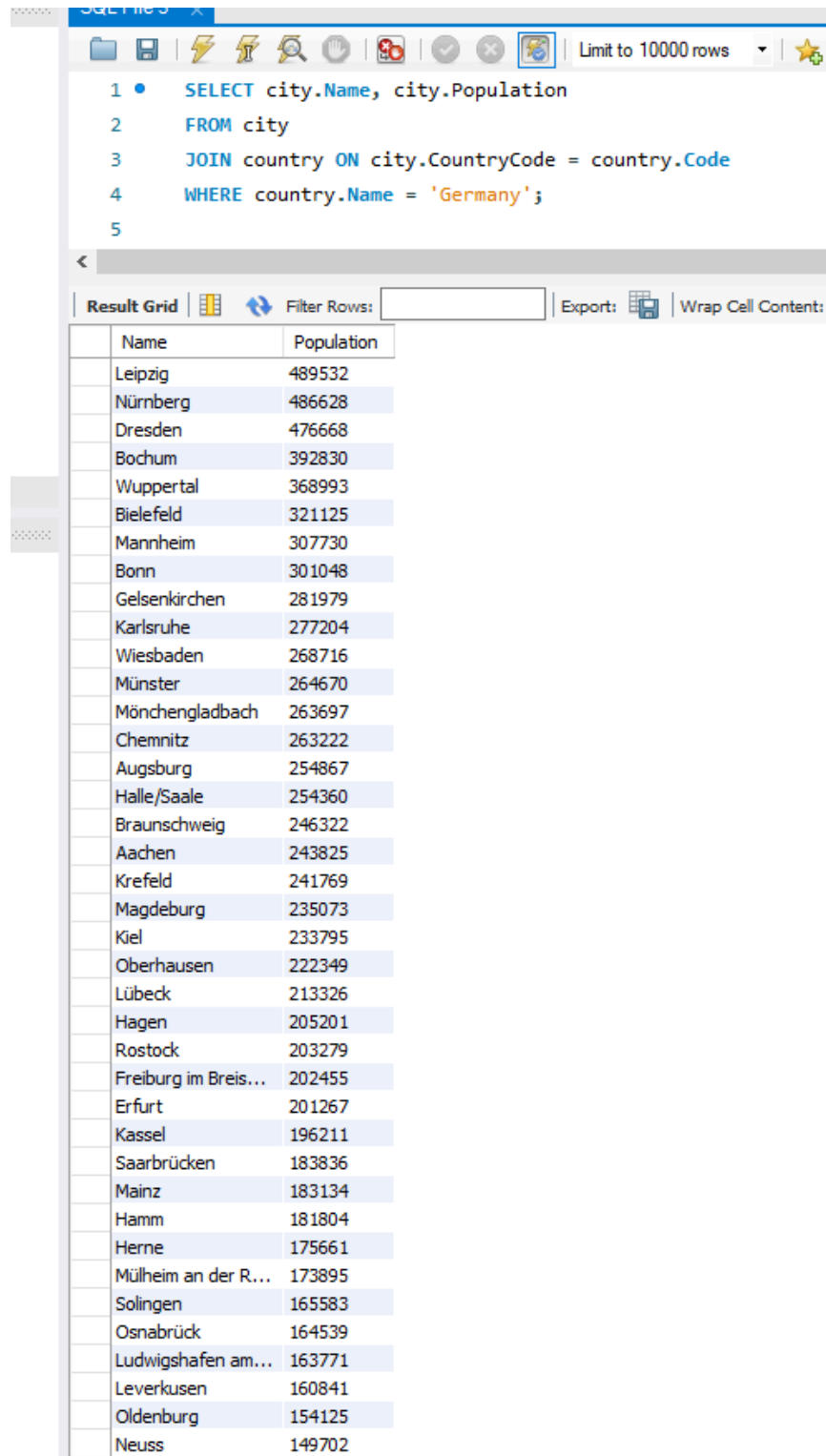
```
1 • SELECT country.Name, countrylanguage.Language
2 FROM country
3 JOIN countrylanguage ON country.Code = countrylanguage.CountryCode
4 WHERE countrylanguage.IsOfficial = 'T';
5
```

The results are displayed in a table with the following columns: Name and Language. The table contains 40 rows of data, listing various countries and their official languages.

Name	Language
Aruba	Dutch
Afghanistan	Dari
Afghanistan	Pashto
Anguilla	English
Albania	Albaniana
Andorra	Catalan
Netherlands Antilles	Dutch
Netherlands Antilles	Papiamentu
United Arab Emirates	Arabic
Argentina	Spanish
Armenia	Armenian
American Samoa	English
American Samoa	Samoan
Antigua and Barbuda	English
Australia	English
Austria	German
Azerbaijan	Azerbaijani
Burundi	French
Burundi	Kirundi
Belgium	Dutch
Belgium	French
Belgium	German
Bangladesh	Bengali
Bulgaria	Bulgariana
Bahrain	Arabic
Bosnia and Herzego...	Serbo-Cro...
Belarus	Belorussian
Belarus	Russian
Belize	English
Bermuda	English
Bolivia	Aimará
Bolivia	Ketšua
Bolivia	Spanish
Brazil	Portuguese
Barbados	English
Brunei	Malay
Bhutan	Dzongkha
Canada	English
Canada	French

The interface also shows a toolbar with various icons and a status bar at the bottom indicating 'Result 4' and 'country 6'.

2. List all cities in Germany along with their population.



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 10000 rows' dropdown. The SQL editor contains the following query:

```
1 • SELECT city.Name, city.Population
2 FROM city
3 JOIN country ON city.CountryCode = country.Code
4 WHERE country.Name = 'Germany';
5
```

Below the editor, the 'Result Grid' tab is active, displaying a table with two columns: 'Name' and 'Population'. The table lists 30 German cities and their populations, sorted in descending order. The interface also includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox.

Name	Population
Leipzig	489532
Nürnberg	486628
Dresden	476668
Bochum	392830
Wuppertal	368993
Bielefeld	321125
Mannheim	307730
Bonn	301048
Gelsenkirchen	281979
Karlsruhe	277204
Wiesbaden	268716
Münster	264670
Mönchengladbach	263697
Chemnitz	263222
Augsburg	254867
Halle/Saale	254360
Braunschweig	246322
Aachen	243825
Krefeld	241769
Magdeburg	235073
Kiel	233795
Oberhausen	222349
Lübeck	213326
Hagen	205201
Rostock	203279
Freiburg im Breis...	202455
Erfurt	201267
Kassel	196211
Saarbrücken	183836
Mainz	183134
Hamm	181804
Herne	175661
Mülheim an der R...	173895
Solingen	165583
Osnabrück	164539
Ludwigshafen am...	163771
Leverkusen	160841
Oldenburg	154125
Neuss	149702

3. Find the five smallest countries by surface area.

SQL File 3*

```
1 • SELECT Name, SurfaceArea
2 FROM country
3 ORDER BY SurfaceArea ASC
4 LIMIT 5;
```

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

Name	SurfaceArea
Holy See (Vatican City State)	0.40
Monaco	1.50
Gibraltar	6.00
Tokelau	12.00
Cocos (Keeling) Islands	14.00

Filtering & Aggregation

4. Find all countries with a population greater than 50 million and sort them in descending order of population.

SQL File 3*

```
1 • SELECT Name, Population
2 FROM country
3 WHERE Population > 50000000
4 ORDER BY Population DESC;
```

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

Name	Population
China	1277558000
India	1013662000
United States	278357000
Indonesia	212107000
Brazil	170115000
Pakistan	156483000
Russian Federation	146934000
Bangladesh	129155000
Japan	126714000
Nigeria	111506000
Mexico	98881000
Germany	82164700
Vietnam	79832000
Philippines	75967000
Egypt	68470000
Iran	67702000
Turkey	66591000
Ethiopia	62565000
Thailand	61399000
United Kingdom	59623400
France	59225700
Italy	57680000
Congo, The Demo...	51654000
Ukraine	50456000

5. Retrieve the average life expectancy per continent.

```

22 • SELECT Continent, AVG(LifeExpectancy) AS AvgLifeExpectancy
23 FROM country
24 GROUP BY Continent;
25

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [A](#)

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

6. Calculate the total population per region.

```

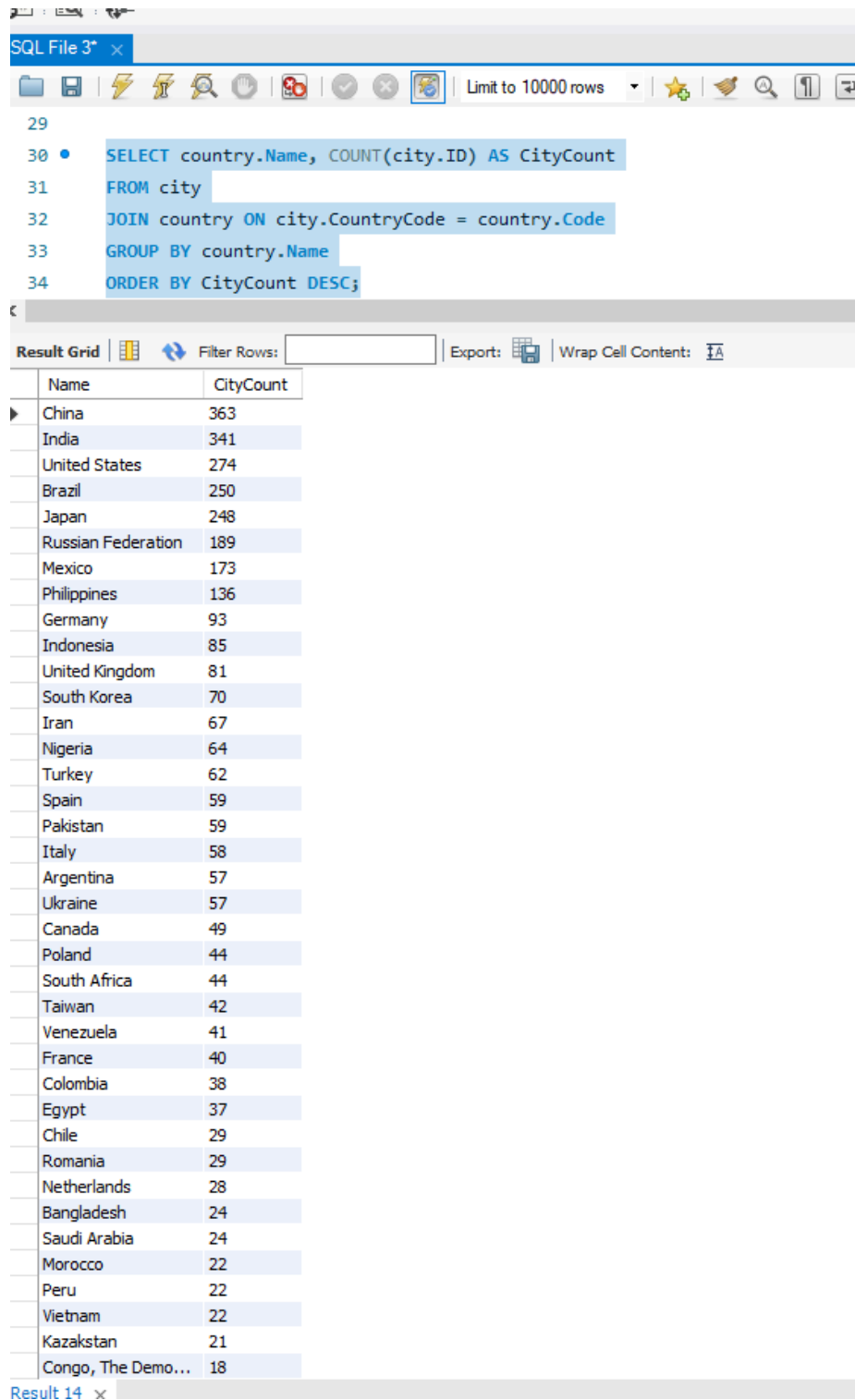
25
26 • SELECT Region, SUM(Population) AS TotalPopulation
27 FROM country
28 GROUP BY Region;
29

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [A](#)

Region	TotalPopulation
Caribbean	38140000
Southern and Central Asia	1490776000
Central Africa	95652000
Southern Europe	144674200
Middle East	188380700
South America	345780000
Polynesia	633050
Antarctica	0
Australia and New Zealand	22753100
Western Europe	183247600
Eastern Africa	246999000
Western Africa	221672000
Eastern Europe	307026000
Central America	135221000
North America	309632000
Southeast Asia	518541000
Southern Africa	46886000
Eastern Asia	1507328000
Nordic Countries	24166400
Northern Africa	173266000
Baltic Countries	7561900
Melanesia	6472000
Micronesia	543000
British Islands	63398500
Micronesia/Caribbean	0

7. Count the number of cities in each country and sort by the highest count.



The screenshot shows a SQL query editor window titled "SQL File 3*" with a toolbar and a "Limit to 10000 rows" dropdown. The query is as follows:

```
29
30 • SELECT country.Name, COUNT(city.ID) AS CityCount
31 FROM city
32 JOIN country ON city.CountryCode = country.Code
33 GROUP BY country.Name
34 ORDER BY CityCount DESC;
```

Below the query editor is the "Result Grid" tab, which displays the results of the query. The grid has two columns: "Name" and "CityCount". The results are sorted in descending order of city count.

Name	CityCount
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
South Korea	70
Iran	67
Nigeria	64
Turkey	62
Spain	59
Pakistan	59
Italy	58
Argentina	57
Ukraine	57
Canada	49
Poland	44
South Africa	44
Taiwan	42
Venezuela	41
France	40
Colombia	38
Egypt	37
Chile	29
Romania	29
Netherlands	28
Bangladesh	24
Saudi Arabia	24
Morocco	22
Peru	22
Vietnam	22
Kazakhstan	21
Congo, The Demo...	18

At the bottom of the window, the text "Result 14" is visible.

Joins & Subqueries

8. Display the top 10 largest cities along with their country name.

SQL File 3*

```

39  country.Name AS CountryName,
40  city.Population
41  FROM
42  city
43  JOIN
44  country ON city.CountryCode = country.Code
45  ORDER BY city.Population DESC
46  LIMIT 10;

```

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

	CityName	CountryName	Population
▶	Mumbai (Bombay)	India	10500000
	Seoul	South Korea	9981619
	São Paulo	Brazil	9968485
	Shanghai	China	9696300
	Jakarta	Indonesia	9604900
	Karachi	Pakistan	9269265
	Istanbul	Turkey	8787958
	Ciudad de México	Mexico	8591309
	Moscow	Russian Federation	8389200
	New York	United States	8008278

9. Retrieve the names of all countries that have an official language of French.

```

42
43 • SELECT DISTINCT country.Name
44 FROM country
45 JOIN countrylanguage ON country.Code = countrylanguage.CountryCode
46 WHERE countrylanguage.Language = 'French' AND countrylanguage.IsOfficial = 'T';
47

```

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

	Name
▶	Burundi
	Belgium
	Canada
	Switzerland
	France
	Guadeloupe
	Haiti
	Luxembourg
	Monaco
	Madagascar
	Martinique
	Mayotte
	New Caledonia
	French Poly...
	Rwanda
	Saint Pierre ...
	Seychelles
	Vanuatu

10. Find all countries where English is spoken, but it is not the official language.

```

47
48 • SELECT DISTINCT country.Name
49 FROM country
50 JOIN countrylanguage ON country.Code = countrylanguage.CountryCode
51 WHERE countrylanguage.Language = 'English' AND countrylanguage.IsOfficial = 'F';
52

```

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

Name
Aruba
Netherlands Antilles
Bahrain
Brunei
Cook Islands
Denmark
Iceland
Japan
Kuwait
Macao
Monaco
Maldives
Malaysia
Norway
Puerto Rico
Trinidad and Toba...

Advanced Queries

11. Find countries where the population tripled in the past 50 years (if historical data is available).

SQL File 3*

```

47
48 • SELECT Name
49 FROM country
50 WHERE Population >= 3 * (SELECT Population FROM country_history WHERE Year = YEAR(CURDATE()) - 50 AND country.Code = c
51

```

Limit to 10000 rows

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

Continent	RichestCountry	HighestGNP	PoorestCountry	LowestGNP
North America	Virgin Islands, U.S.	8510700.00	Anguilla	0.00
Asia	Yemen	3787042.00	Afghanistan	0.00
Africa	Zimbabwe	116729.00	Algeria	0.00
Europe	Yugoslavia	2133367.00	Albania	0.00
South America	Venezuela	776739.00	Argentina	0.00
Oceania	Wallis and Futuna	351182.00	American Samoa	0.00
Antarctica	South Georgia and the South Sandwich Islands	0.00	Antarctica	0.00

12. List the richest and poorest countries in each continent based on GNP (Gross National Product).

SQL File 3* x

Limit to 10000 rows

```

51
52 • SELECT Continent,
53     MAX(Name) AS RichestCountry, MAX(GNP) AS HighestGNP,
54     MIN(Name) AS PoorestCountry, MIN(GNP) AS LowestGNP
55 FROM country
56 GROUP BY Continent;
57

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [fA](#)

	Continent	RichestCountry	HighestGNP	PoorestCountry	LowestGNP
▶	North America	Virgin Islands, U.S.	8510700.00	Anguilla	0.00
	Asia	Yemen	3787042.00	Afghanistan	0.00
	Africa	Zimbabwe	116729.00	Algeria	0.00
	Europe	Yugoslavia	2133367.00	Albania	0.00
	South America	Venezuela	776739.00	Argentina	0.00
	Oceania	Wallis and Futuna	351182.00	American Samoa	0.00
	Antarctica	South Georgia and the South Sandwich Islands	0.00	Antarctica	0.00

13. Identify countries with a life expectancy below the global average.

SQL File 3* x

Limit to 10000 rows

```

55 FROM country
56 GROUP BY Continent;
57
58 • SELECT Name, LifeExpectancy
59 FROM country
60 WHERE LifeExpectancy < (SELECT AVG(LifeExpectancy) FROM country);
61

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	Name	LifeExpectancy
▶	Afghanistan	45.9
	Angola	38.3
	Armenia	66.4
	Azerbaijan	62.9
	Burundi	46.2
	Benin	50.2
	Burkina Faso	46.7
	Bangladesh	60.2
	Bolivia	63.7
	Brazil	62.9
	Bhutan	52.4
	Botswana	39.3
	Central Afri...	44.0
	Côte d'Ivoire	45.2
	Cameroon	54.8
	Congo, The...	48.8
	Congo	47.4
	Comoros	60.0
	Djibouti	50.8
	Egypt	63.3
	Eritrea	55.8
	Western Sa...	49.8
	Ethiopia	45.2
	Gabon	50.1
	Georgia	64.5
	Ghana	57.4
	Guinea	45.6
	Gambia	53.2
	Guinea-Bissau	49.0
	Equatorial G...	53.6
	Grenada	64.5
	Guatemala	66.2
	Guyana	64.0

country 18 x

14. Retrieve the capital cities of countries with a population above 100 million

SQL File 3* x

Limit to 10000 rows

```
61
62 • SELECT city.Name AS CapitalCity, country.Name AS CountryName
63 FROM country
64 JOIN city ON country.Capital = city.ID
65 WHERE country.Population > 100000000;
66
```

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

	CapitalCity	CountryName
▶	Dhaka	Bangladesh
	Brasilia	Brazil
	Peking	China
	Jakarta	Indonesia
	New Delhi	India
	Tokyo	Japan
	Abuja	Nigeria
	Islamabad	Pakistan
	Moscow	Russian Federation
	Washington	United States

15. Find the continent with the highest number of countries.

SQL File 3* x

Limit to 10000 rows

```
67 • SELECT Continent, COUNT(Name) AS CountryCount
68 FROM country
69 GROUP BY Continent
70 ORDER BY CountryCount DESC
71 LIMIT 1;
72
```

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

	Continent	CountryCount
▶	Africa	58