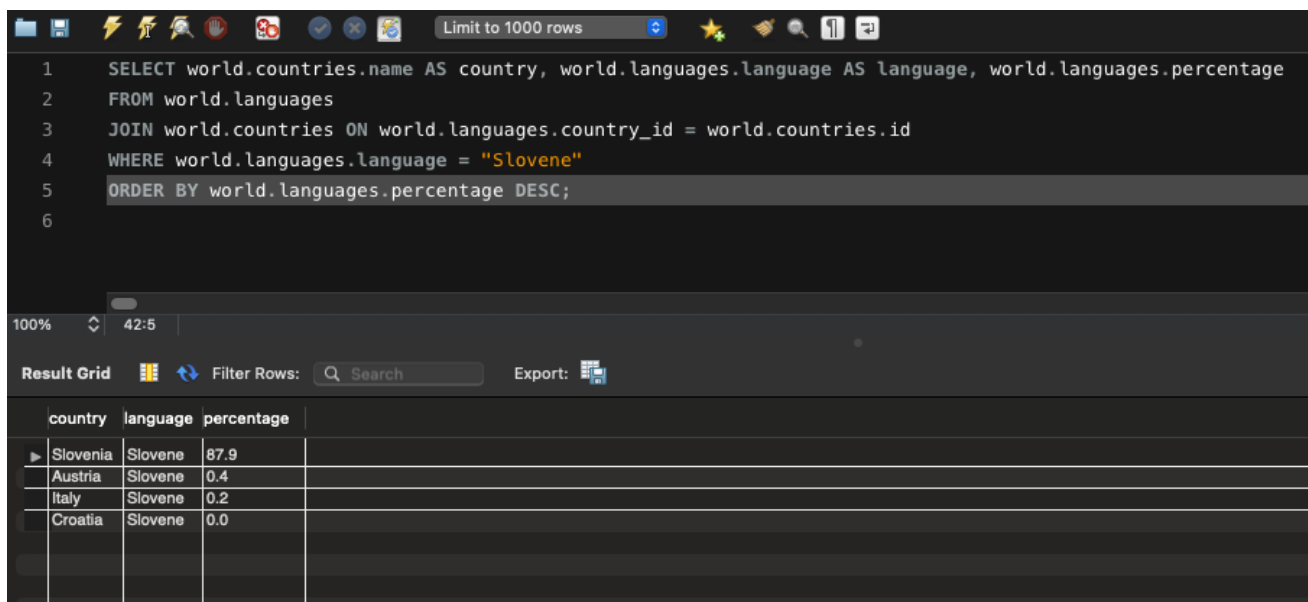


## MySQL Countries Assignment – Francis Frago, 18Dec21

1. What query would you run to get all the countries that speak Slovene? Your query should return the name of the country, language and language percentage. Your query should arrange the result by language percentage in descending order. (1)

```
SELECT world.countries.name AS country, world.languages.language AS language,  
world.languages.percentage  
FROM world.languages  
JOIN world.countries ON world.languages.country_id = world.countries.id  
WHERE world.languages.language = "Slovene"  
ORDER BY world.languages.percentage DESC;
```



The screenshot shows a MySQL query editor with the following SQL query entered:

```
1 SELECT world.countries.name AS country, world.languages.language AS language, world.languages.percentage  
2 FROM world.languages  
3 JOIN world.countries ON world.languages.country_id = world.countries.id  
4 WHERE world.languages.language = "Slovene"  
5 ORDER BY world.languages.percentage DESC;  
6
```

Below the query editor, the results are displayed in a table with the following columns: country, language, and percentage. The results are sorted by percentage in descending order.

country	language	percentage
Slovenia	Slovene	87.9
Austria	Slovene	0.4
Italy	Slovene	0.2
Croatia	Slovene	0.0

2. What query would you run to display the total number of cities for each country? Your query should return the name of the country and the total number of cities. Your query should arrange the result by the number of cities in descending order. (3)

```
SELECT countries.name AS country, COUNT(cities.name) AS number_of_cities  
FROM cities  
JOIN countries ON cities.country_id = countries.id  
GROUP BY countries.name  
ORDER BY number_of_cities DESC;
```

The screenshot shows a SQL IDE interface. At the top, there's a toolbar with various icons and a 'Limit to 1000 rows' dropdown. Below the toolbar, a SQL query is entered in a text area:

```

6 SELECT countries.name AS country, COUNT(cities.name) AS number_of_cities
7 FROM cities
8 JOIN countries ON cities.country_id = countries.id
9 GROUP BY countries.name
10 ORDER BY number_of_cities DESC;

```

Below the query editor, there's a 'Result Grid' section. It includes a 'Filter Rows' search bar and an 'Export' button. The result grid displays the following data:

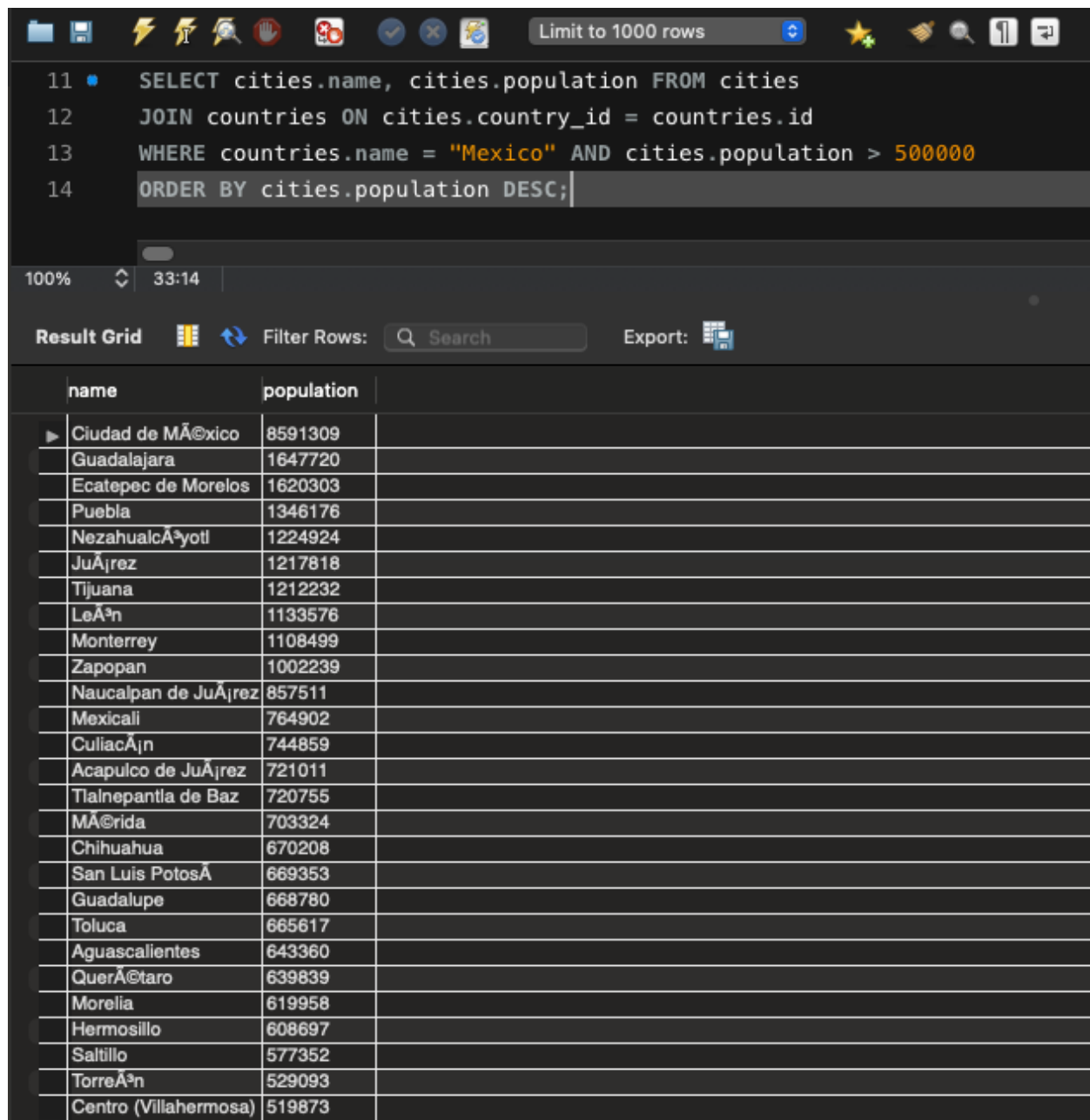
	country	number_of_cities
▶	China	363
	India	341
	United States	274
	Brazil	250
	Japan	248
	Russian Federation	189
	Mexico	173
	Philippines	136
	Germany	93
	Indonesia	85
	United Kingdom	81
	South Korea	70
	Iran	67
	Nigeria	64
	Turkey	62
	Spain	59
	Pakistan	59
	Italy	58

3. What query would you run to get all the cities in Mexico with a population of greater than 500,000? Your query should arrange the result by population in descending order. (1)

```

SELECT cities.name, cities.population FROM cities
JOIN countries ON cities.country_id = countries.id
WHERE countries.name = "Mexico" AND cities.population > 500000
ORDER BY cities.population DESC;

```



The screenshot shows a SQL IDE interface. At the top, there's a toolbar with various icons and a 'Limit to 1000 rows' button. Below the toolbar, a SQL query is entered in a text area:

```

11 SELECT cities.name, cities.population FROM cities
12 JOIN countries ON cities.country_id = countries.id
13 WHERE countries.name = "Mexico" AND cities.population > 500000
14 ORDER BY cities.population DESC;

```

Below the query editor, there's a 'Result Grid' section. It includes a 'Filter Rows' search bar and an 'Export' button. The results are displayed in a table with two columns: 'name' and 'population'.

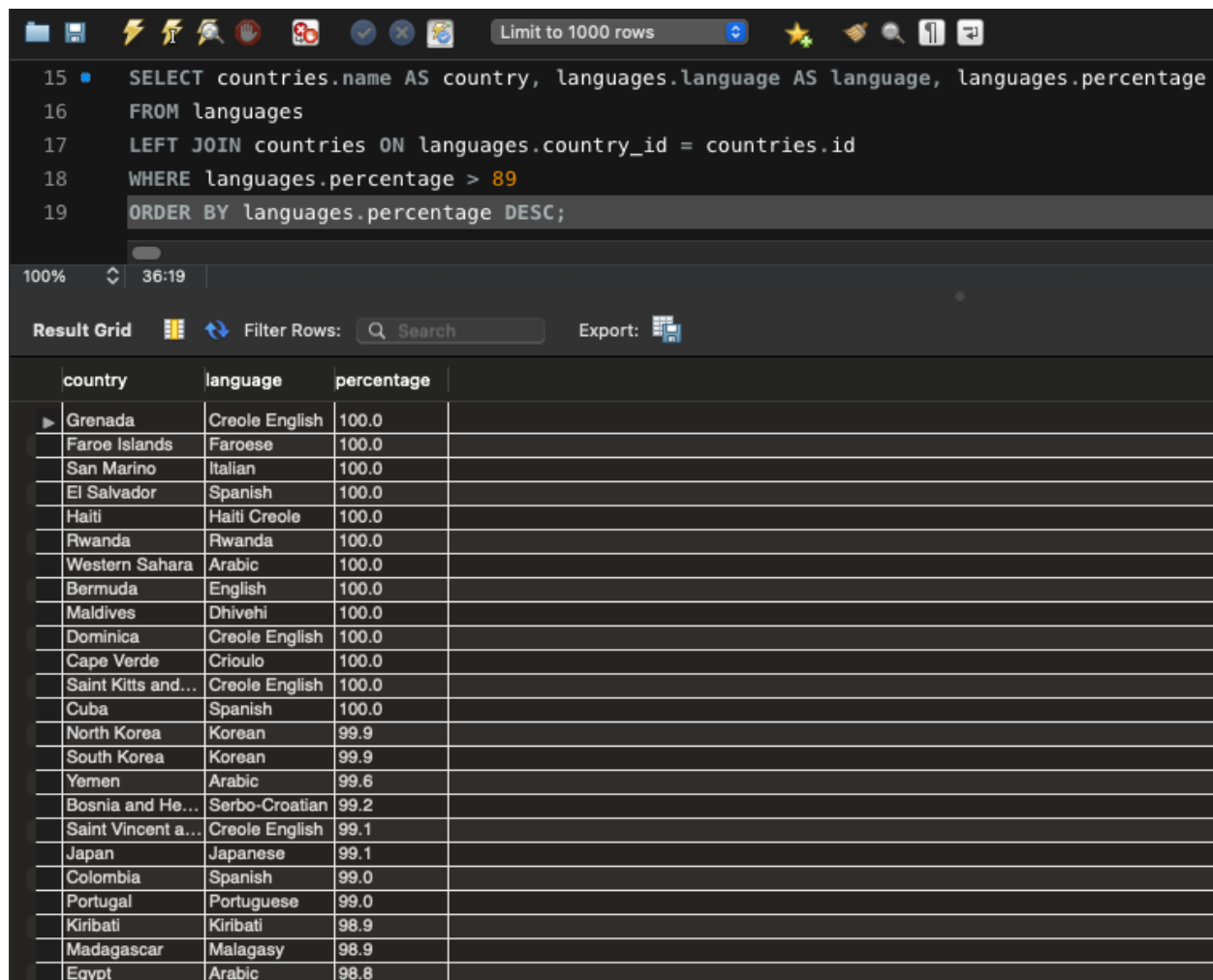
	name	population
▶	Ciudad de México	8591309
▶	Guadalajara	1647720
▶	Ecatepec de Morelos	1620303
▶	Puebla	1346176
▶	Nezahualcóyotl	1224924
▶	Juárez	1217818
▶	Tijuana	1212232
▶	León	1133576
▶	Monterrey	1108499
▶	Zapopan	1002239
▶	Naucalpan de Juárez	857511
▶	Mexicali	764902
▶	Culiacán	744859
▶	Acapulco de Juárez	721011
▶	Tlalnepantla de Baz	720755
▶	Mérida	703324
▶	Chihuahua	670208
▶	San Luis Potosí	669353
▶	Guadalupe	668780
▶	Toluca	665617
▶	Aguascalientes	643360
▶	Querétaro	639839
▶	Morelia	619958
▶	Hermosillo	608697
▶	Saltillo	577352
▶	Torreón	529093
▶	Centro (Villahermosa)	519873

4. What query would you run to get all languages in each country with a percentage greater than 89%? Your query should arrange the result by percentage in descending order. (1)

```

SELECT countries.name AS country, languages.language AS language,
languages.percentage
FROM languages
LEFT JOIN countries ON languages.country_id = countries.id
WHERE languages.percentage > 89
ORDER BY languages.percentage DESC;

```



```

15 SELECT countries.name AS country, languages.language AS language, languages.percentage
16 FROM languages
17 LEFT JOIN countries ON languages.country_id = countries.id
18 WHERE languages.percentage > 89
19 ORDER BY languages.percentage DESC;

```

100% 36:19

Result Grid Filter Rows: Search Export:

country	language	percentage
Grenada	Creole English	100.0
Faroe Islands	Faroese	100.0
San Marino	Italian	100.0
El Salvador	Spanish	100.0
Haiti	Haiti Creole	100.0
Rwanda	Rwanda	100.0
Western Sahara	Arabic	100.0
Bermuda	English	100.0
Maldives	Dhivehi	100.0
Dominica	Creole English	100.0
Cape Verde	Crioulo	100.0
Saint Kitts and...	Creole English	100.0
Cuba	Spanish	100.0
North Korea	Korean	99.9
South Korea	Korean	99.9
Yemen	Arabic	99.6
Bosnia and He...	Serbo-Croatian	99.2
Saint Vincent a...	Creole English	99.1
Japan	Japanese	99.1
Colombia	Spanish	99.0
Portugal	Portuguese	99.0
Kiribati	Kiribati	98.9
Madagascar	Malagasy	98.9
Egypt	Arabic	98.8

5. What query would you run to get all the countries with Surface Area below 501 and Population greater than 100,000? (2)

```

SELECT countries.name, countries.surface_area, countries.population FROM countries
WHERE countries.surface_area < 501 AND countries.population > 100000;

```

The screenshot shows a SQL IDE interface. The query editor contains the following SQL code:

```

17  -- LEFT JOIN countries ON languages.country_id = countries.id
18  -- WHERE languages.percentage > 89
19  -- ORDER BY languages.percentage DESC;
20  SELECT countries.name, countries.surface_area, countries.population FROM countries
21  WHERE countries.surface_area < 501 AND countries.population > 100000;

```

Below the query editor, the 'Result Grid' is displayed, showing a table with the following data:

name	surface_area	population
Aruba	193.00	103000
Barbados	430.00	270000
Macao	18.00	473000
Maldives	298.00	286000
Malta	316.00	380200
Mayotte	373.00	149000
Saint Vincent and the Grenadines	388.00	114000

6. What query would you run to get countries with only Constitutional Monarchy with a capital greater than 200 and a life expectancy greater than 75 years? (1)

```

SELECT countries.name, countries.government_form, countries.capital,
countries.life_expectancy
FROM countries
WHERE countries.government_form = 'Constitutional Monarchy'
AND countries.capital > 200
AND countries.life_expectancy > 75;

```

The screenshot shows a SQL IDE interface. The query editor contains the following SQL code:

```

22  SELECT countries.name, countries.government_form, countries.capital, countries.life_expectancy
23  FROM countries
24  WHERE countries.government_form = 'Constitutional Monarchy'
25  AND countries.capital > 200
26  AND countries.life_expectancy > 75;

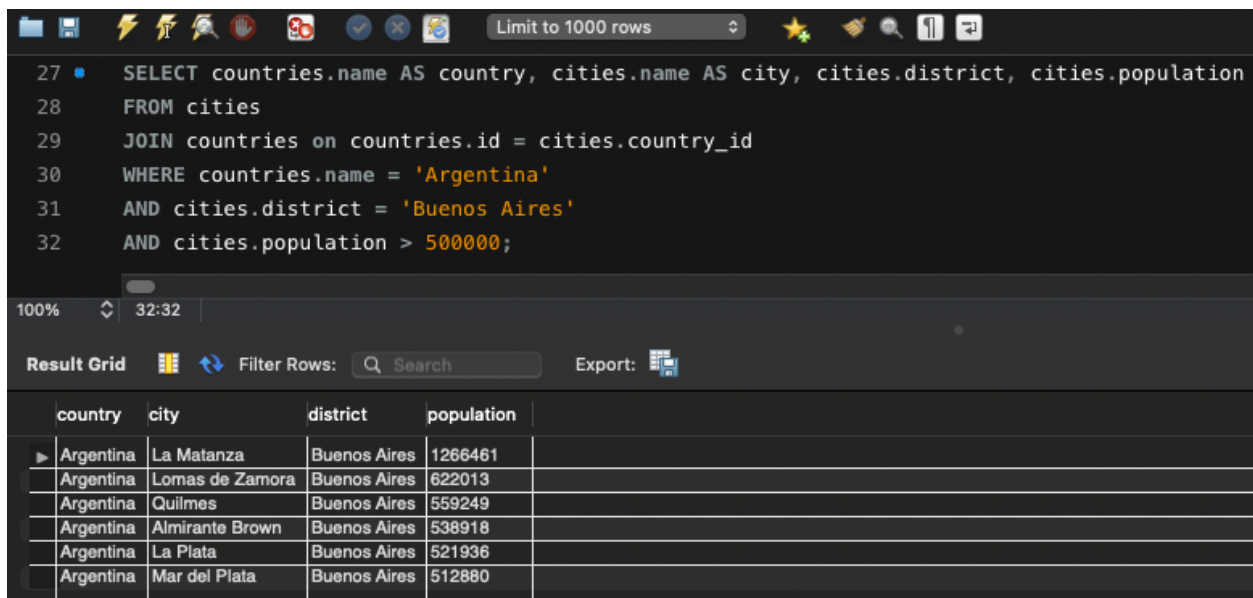
```

Below the query editor, the 'Result Grid' is displayed, showing a table with the following data:

name	government_form	capital	life_expectancy
Denmark	Constitutional Monarchy	3315	76.5
Spain	Constitutional Monarchy	653	78.8
United Kingdom	Constitutional Monarchy	456	77.7
Jamaica	Constitutional Monarchy	1530	75.2
Jordan	Constitutional Monarchy	1786	77.4
Japan	Constitutional Monarchy	1532	80.7
Liechtenstein	Constitutional Monarchy	2446	78.8
Luxembourg	Constitutional Monarchy	2452	77.1
Monaco	Constitutional Monarchy	2695	78.8
Norway	Constitutional Monarchy	2807	78.7
New Zealand	Constitutional Monarchy	3499	77.8
Sweden	Constitutional Monarchy	3048	79.6

7. What query would you run to get all the cities of Argentina inside the Buenos Aires district and have the population greater than 500, 000? The query should return the Country Name, City Name, District and Population. (2)

```
SELECT countries.name AS country, cities.name AS city, cities.district, cities.population
FROM cities
JOIN countries on countries.id = cities.country_id
WHERE countries.name = 'Argentina'
AND cities.district = 'Buenos Aires'
AND cities.population > 500000;
```



The screenshot shows a SQL query editor with a dark theme. The query is entered in a text area, and below it, a 'Result Grid' displays the results of the query. The query is:   
SELECT countries.name AS country, cities.name AS city, cities.district, cities.population  
FROM cities  
JOIN countries on countries.id = cities.country\_id  
WHERE countries.name = 'Argentina'  
AND cities.district = 'Buenos Aires'  
AND cities.population > 500000;  
The result grid shows 6 rows of data. The first row is highlighted. The columns are: country, city, district, and population.

	country	city	district	population
▶	Argentina	La Matanza	Buenos Aires	1266461
▶	Argentina	Lomas de Zamora	Buenos Aires	622013
▶	Argentina	Quilmes	Buenos Aires	559249
▶	Argentina	Almirante Brown	Buenos Aires	538918
▶	Argentina	La Plata	Buenos Aires	521936
▶	Argentina	Mar del Plata	Buenos Aires	512880

8. What query would you run to summarize the number of countries in each region? The query should display the name of the region and the number of countries. Also, the query should arrange the result by the number of countries in descending order. (2)

```
SELECT countries.region, COUNT(countries.name) AS countries
FROM countries
GROUP BY countries.region
ORDER BY countries DESC;
```

33	SELECT countries.region, COUNT(countries.name) AS countries
34	FROM countries
35	GROUP BY countries.region
36	ORDER BY countries DESC;
37	
38	

100%
1:37

Result Grid
Filter Rows:
Search
Export:

	region	countries	
▶	Caribbean	24	
	Eastern Africa	20	
	Middle East	18	
	Western Africa	17	
	Southern Europe	15	
	Southern and Central Asia	14	
	South America	14	
	Southeast Asia	11	
	Polynesia	10	
	Eastern Europe	10	
	Central Africa	9	
	Western Europe	9	
	Central America	8	
	Eastern Asia	8	
	Nordic Countries	7	
	Northern Africa	7	
	Micronesia	7	
	Antarctica	5	
	Australia and New Zealand	5	
	North America	5	
	Southern Africa	5	
	Melanesia	5	
	Baltic Countries	3	
	British Islands	2	
	Micronesia/Caribbean	1	