Data Science Homework 4

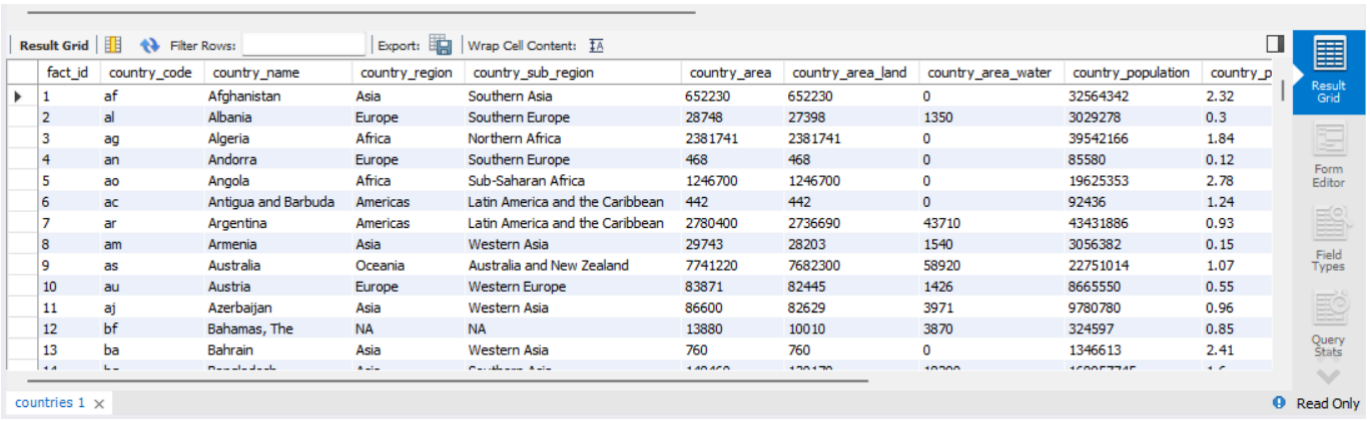
By: Rania Siddiqui 07494

Note: I have used MySql for this homework. In Mysql, when creating a table, you need to specify which database you need to create your table in. Since 'sakila' was the default database that was there when i installed mysql, so i have created all my tables in 'sakila' database.

Queries for creating tables for both of the csv files
Below is the query:

```
USE sakila;
CREATE TABLE countries (
    fact_id INT,
    country_code VARCHAR(5),
    country_name VARCHAR(100),
    country_region VARCHAR(100),
    country_sub_region VARCHAR(100),
    country_area INT,
    country_area_land INT,
    country_area_water INT,
    country_population BIGINT,
    country_population_growth FLOAT,
    country_birth_rate FLOAT,
    country_death_rate FLOAT,
    country_migration_rate FLOAT
);
```

For checking and loading/importing the country dataset into a table
Below is my screen of my sql server showing data has been properly imported.
SELECT * FROM sakila.countries;



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

fact_id	country_code	country_name	country_region	country_sub_region	country_area	country_area_land	country_area_water	country_population	country_p
1	af	Afghanistan	Asia	Southern Asia	652230	652230	0	32564342	2.32
2	al	Albania	Europe	Southern Europe	28748	27398	1350	3029278	0.3
3	ag	Algeria	Africa	Northern Africa	2381741	2381741	0	39542166	1.84
4	an	Andorra	Europe	Southern Europe	468	468	0	85580	0.12
5	ao	Angola	Africa	Sub-Saharan Africa	1246700	1246700	0	19625353	2.78
6	ac	Antigua and Barbuda	Americas	Latin America and the Caribbean	442	442	0	92436	1.24
7	ar	Argentina	Americas	Latin America and the Caribbean	2780400	2736690	43710	43431886	0.93
8	am	Armenia	Asia	Western Asia	29743	28203	1540	3056382	0.15
9	as	Australia	Oceania	Australia and New Zealand	7741220	7682300	58920	22751014	1.07
10	au	Austria	Europe	Western Europe	83871	82445	1426	8665550	0.55
11	aj	Azerbaijan	Asia	Western Asia	86600	82629	3971	9780780	0.96
12	bf	Bahamas, The	NA	NA	13880	10010	3870	324597	0.85
13	ba	Bahrain	Asia	Western Asia	760	760	0	1346613	2.41

countries 1 | Read Only

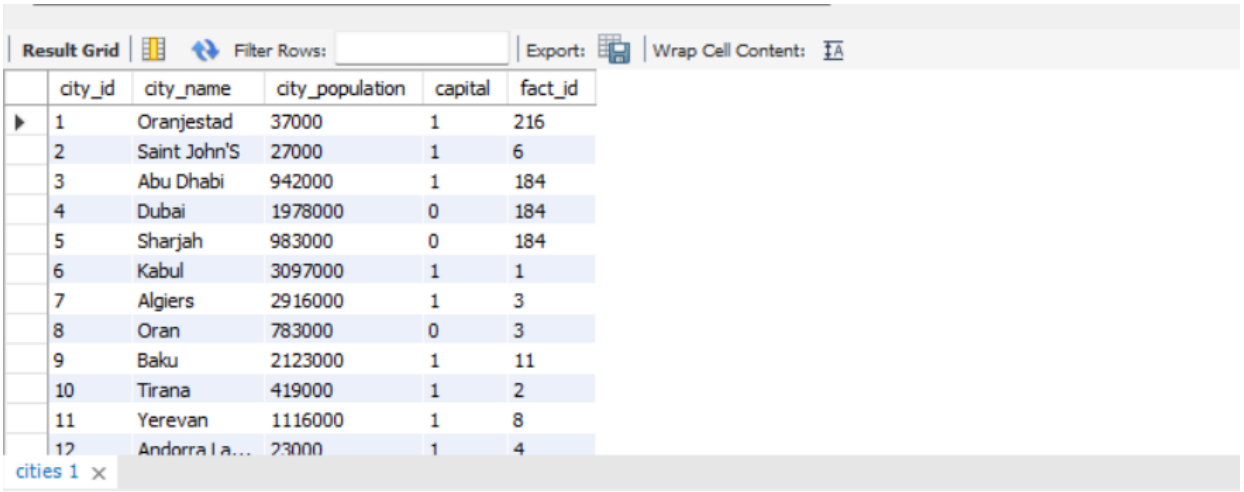
For cities.csv

```
USE sakila;
CREATE TABLE cities (
    city_id INT,
    city_name VARCHAR(255),
```

```
    city_population INT,
    capital BOOLEAN,
    fact_id INT
);
```

For checking and loading/importing the cities dataset into a table
Below is my screen of my sql server showing data has been properly imported.

SELECT * FROM sakila.cities;



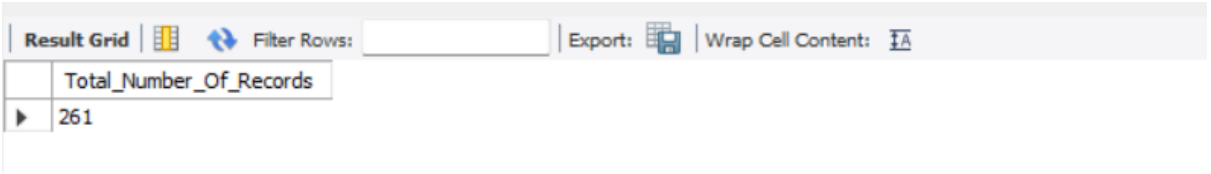
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

city_id	city_name	city_population	capital	fact_id
1	Oranjestad	37000	1	216
2	Saint John'S	27000	1	6
3	Abu Dhabi	942000	1	184
4	Dubai	1978000	0	184
5	Sharjah	983000	0	184
6	Kabul	3097000	1	1
7	Algiers	2916000	1	3
8	Oran	783000	0	3
9	Baku	2123000	1	11
10	Tirana	419000	1	2
11	Yerevan	1116000	1	8
12	Andorra La...	23000	1	4

cities 1

The data in both these tables were imported manually through the CSV files through mysql's direct access.

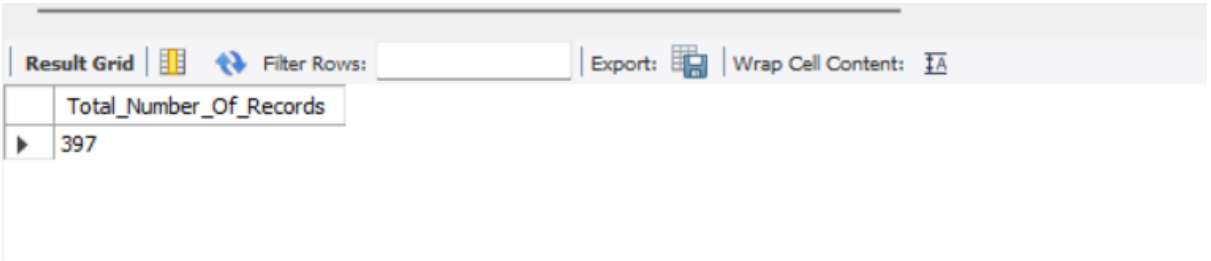
1. **The count of total number of records in each table.**
   select count(*) as Total_Number_Of_Records from sakila.countries



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

Total_Number_Of_Records
261

   select count(*) as Total_Number_Of_Records from sakila.cities



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

Total_Number_Of_Records
397

2. **the count of number of cities for each country in descending order of count (use group by)**

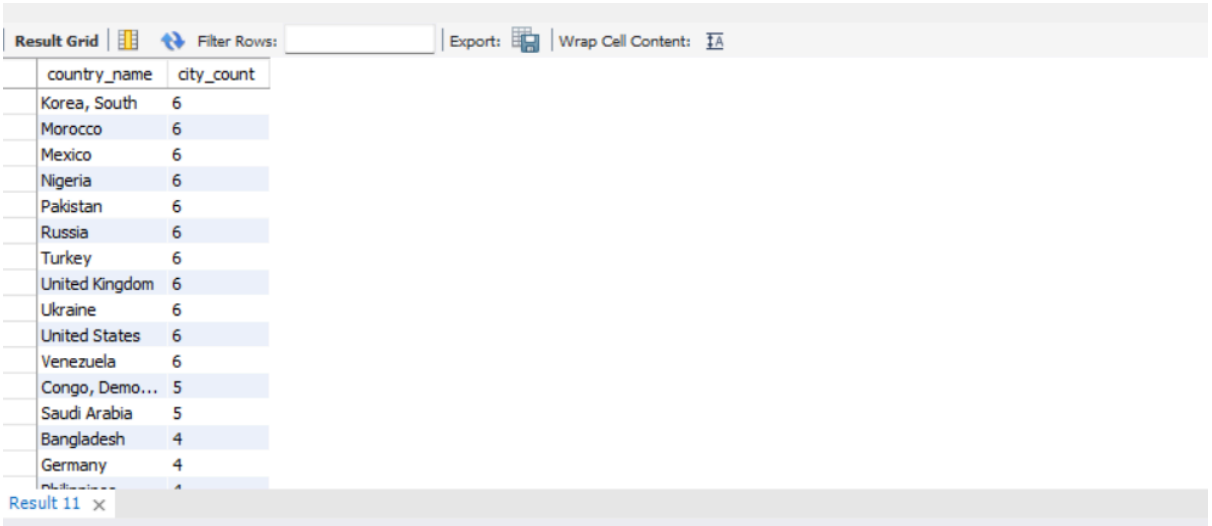
```
SELECT
    sakila.countries.country_name,
    COUNT(sakila.cities.city_id) AS city_count
FROM
    sakila.cities
```

```
JOIN
   sakila.countries
ON
   sakila.cities.fact_id =sakila. countries.fact_id
GROUP BY
   sakila.countries.country_name
ORDER BY
   city_count DESC;
```

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

country_name	city_count
South Africa	7
Argentina	6
Australia	6
Brazil	6
Canada	6
China	6
Colombia	6
France	6
Indonesia	6
India	6
Iran	6
Italy	6

Result 3



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

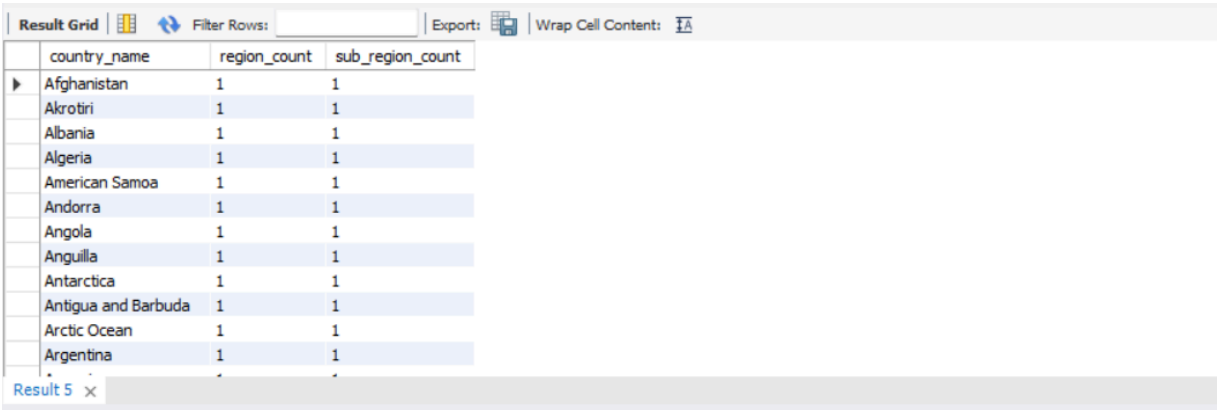
country_name	city_count
Korea, South	6
Morocco	6
Mexico	6
Nigeria	6
Pakistan	6
Russia	6
Turkey	6
United Kingdom	6
Ukraine	6
United States	6
Venezuela	6
Congo, Demo...	5
Saudi Arabia	5
Bangladesh	4
Germany	4

Result 11

Here also the output was really big, so i have added only a few snippets.

3. **the count of regions and sub-regions in each country. Sort them by ascending order of country name. (use group by)**

```
SELECT
   sakila.countries.country_name,
   COUNT(DISTINCT sakila.countries.country_region) AS region_count,
   COUNT(DISTINCT sakila.countries.country_sub_region) AS sub_region_count
FROM
   sakila.countries
GROUP BY
   sakila.countries.country_name
ORDER BY
   sakila.countries.country_name ASC;
```

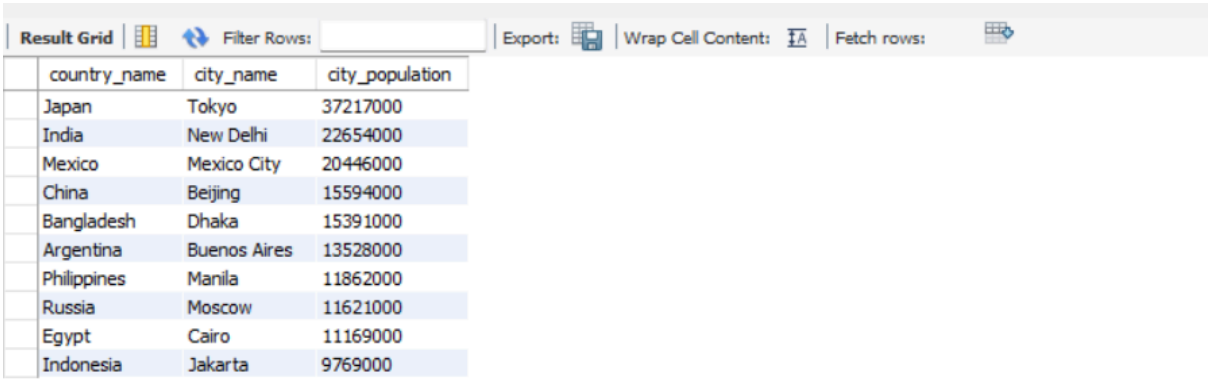


country_name	region_count	sub_region_count
Afghanistan	1	1
Akrotiri	1	1
Albania	1	1
Algeria	1	1
American Samoa	1	1
Andorra	1	1
Angola	1	1
Anguilla	1	1
Antarctica	1	1
Antigua and Barbuda	1	1
Arctic Ocean	1	1
Argentina	1	1

Basically region_count and subregion_count is 1 for all the countries. Because the output was really big, I pasted only a snippet of it.

4. **Top 10 most populous capital cities. Display country, city and population in descending order.**

```
SELECT
    sakila.countries.country_name,
    sakila.cities.city_name,
    sakila.cities.city_population
FROM
    sakila.cities
JOIN
    sakila.countries
ON
    sakila.cities.fact_id = sakila.countries.fact_id
WHERE
    sakila.cities.capital = 1
ORDER BY
    sakila.cities.city_population DESC
LIMIT 10;
```

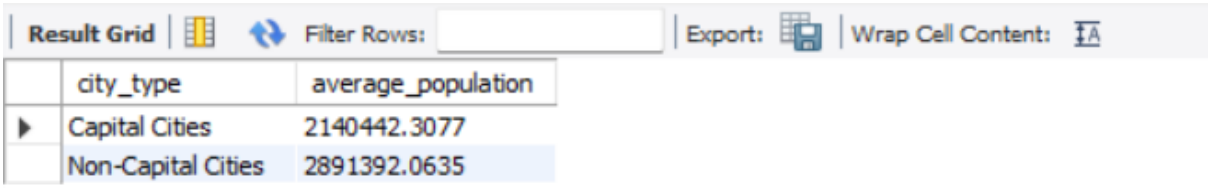


country_name	city_name	city_population
Japan	Tokyo	37217000
India	New Delhi	22654000
Mexico	Mexico City	20446000
China	Beijing	15594000
Bangladesh	Dhaka	15391000
Argentina	Buenos Aires	13528000
Philippines	Manila	11862000
Russia	Moscow	11621000
Egypt	Cairo	11169000
Indonesia	Jakarta	9769000

5. **Average city population of capital and non-capital cities. (use group by)**

```
SELECT
    CASE
        WHEN sakila.cities.capital = 1 THEN 'Capital Cities'
```

```
        ELSE 'Non-Capital Cities'
    END AS city_type,
    AVG(sakila.cities.city_population) AS average_population
FROM
    sakila.cities
GROUP BY
    sakila.cities.capital;
```

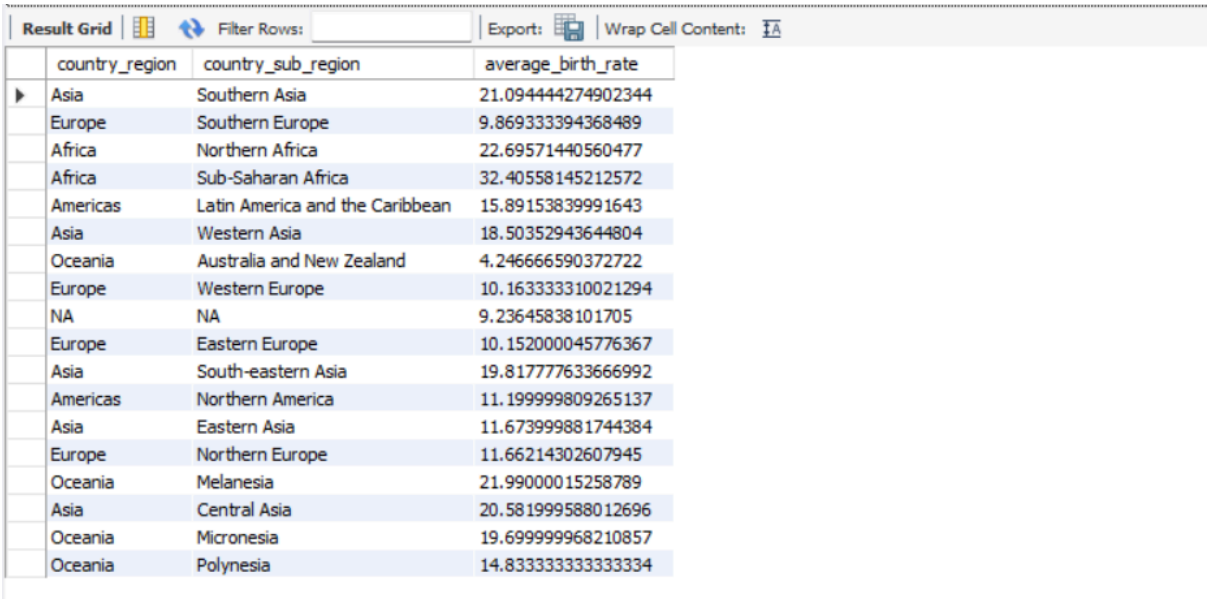


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

city_type	average_population
Capital Cities	2140442.3077
Non-Capital Cities	2891392.0635

6. **Average country birth rate for each region and sub-region (use group by)**

```
SELECT
    sakila.countries.country_region,
    sakila.countries.country_sub_region,
    AVG(sakila.countries.country_birth_rate) AS average_birth_rate
FROM
    sakila.countries
GROUP BY
    sakila.countries.country_region,
    sakila.countries.country_sub_region;
```



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

country_region	country_sub_region	average_birth_rate
Asia	Southern Asia	21.094444274902344
Europe	Southern Europe	9.869333394368489
Africa	Northern Africa	22.69571440560477
Africa	Sub-Saharan Africa	32.40558145212572
Americas	Latin America and the Caribbean	15.89153839991643
Asia	Western Asia	18.50352943644804
Oceania	Australia and New Zealand	4.246666590372722
Europe	Western Europe	10.163333310021294
NA	NA	9.23645838101705
Europe	Eastern Europe	10.152000045776367
Asia	South-eastern Asia	19.817777633666992
Americas	Northern America	11.199999809265137
Asia	Eastern Asia	11.673999881744384
Europe	Northern Europe	11.66214302607945
Oceania	Melanesia	21.99000015258789
Asia	Central Asia	20.581999588012696
Oceania	Micronesia	19.699999968210857
Oceania	Polynesia	14.833333333333334