Introduction

In this project we will work with the data from CIA WORLD FACTBOOK that contains demographic information like names, population, population growth, birth rate, death rate, area land and area water. In this project we will analyze the data by using SQL programming language.

Load SQL

In [6]:

```
!conda install -yc conda-forge ipython-sql
%load_ext sql
%sql sqlite:///factbook.db
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.
```

Out[6]:

```
'Connected: @factbook.db'
```

Data Exploration

We will output first five rows to check how the data look like.

In [7]:

```
%%sql

SELECT *
    FROM facts
LIMIT 5;
```

```
 * sqlite:///factbook.db
Done.
```

Out[7]:

id	code	name	area	area_land	area_water	population	population_growth	birth_rate	death_rate	migrati
1	af	Afghanistan	652230	652230	0	32564342	2.32	38.57	13.89	
2	al	Albania	28748	27398	1350	3029278	0.3	12.92	6.58	
3	ag	Algeria	2381741	2381741	0	39542166	1.84	23.67	4.31	
4	an	Andorra	468	468	0	85580	0.12	8.13	6.96	
5	ao	Angola	1246700	1246700	0	19625353	2.78	38.78	11.49	

This shows that our data has:

name — the name of the country.
area— the country's total area (both land and water).
area_land — the country's land area in square kilometers.
area_water — the country's waterarea in square kilometers.
population — the country's population.
population_growth— the country's population growth as a percentage.
birth_rate — the country's birth rate, or the number of births per year per 1,000 people.
death_rate — the country's death rate, or the number of death per year per 1,000 people

Further Exploration

Now we will explore the data deeper by finding the minimum and maximum for population and population growth.

In [8]:

```
%%sql
SELECT
    MIN(population) AS Min_pop,
    MAX(population) AS Max_pop,
    MIN(population_growth) AS Min_pop_growth,
    MAX(population_growth) AS Max_pop_growth
    FROM facts;
```

```
 * sqlite:///factbook.db
Done.
```

Out[8]:

Min_pop	Max_pop	Min_pop_growth	Max_pop_growth
0	7256490011	0.0	4.02

We found there is a country with zero population and a country with more than 7.2 billion people. We will zoom in just these countries.

First we will explore the country with zero population.

In [9]:

```
%%sql
SELECT name AS Country, Population
    FROM facts
    WHERE population == (SELECT Min(population)
                            FROM facts);
```

```
 * sqlite:///factbook.db
```

```
Done.
```

Out[9]:

Country	population
Antarctica	0

We found that the country with zero population is Antartica which seems a match.

Now we will explore the country with maximum population

In [13]:

```
%%sql
SELECT name AS Country, population AS Population
    FROM facts
    WHERE population == (SELECT Max(population)
                            FROM facts);
```

```
 * sqlite:///factbook.db
Done.
```

Out[13]:

Country	Population
World	7256490011

We discovered that there is a row with the name "World" that contains the total population of the world "7.2 billion". This seems an error in our data so we have to remove this particular row.

Removing Data

The following code will help us to remove the world's population row.

In [14]:

```
%%sql
DELETE FROM facts WHERE name == 'World';
SELECT
    Min(population) AS Min_pop,
    Min(population_growth) AS Min_pop_growth,
    Max(population) AS Max_pop,
    Max(population_growth) AS Max_pop_growth
    FROM facts;
```

```
 * sqlite:///factbook.db
1 rows affected.
Done.
```

Out[14]:

Min_pop	Min_pop_growth	Max_pop	Max_pop_growth

0	0.0	1367485388	4.02

Output "1 row affected" indicates that one row was affected in our data. After removing the "world" row we output the maximum population again but this time the Maximum population is 1.4 billion which means we were able to sucessfully remove the row containing world's population.

Exploring Average Population and Area

Our further analysis includes finding average population and area.

In [15]:

```
%%sql
SELECT AVG(population) AS Average_pop, AVG(area) AS Average_area
    FROM facts;
```

```
 * sqlite:///factbook.db
Done.
```

Out[15]:

Average_pop	Average_area
32242666.56846473	555093.546184739

By using the above result we will find the densely populated country that have:

Above- average values for population
Below- average values for area

In [44]:

```
%%sql
SELECT code AS Code, name AS Country, population AS Population, area AS Area
    FROM facts
    WHERE population > (SELECT AVG(population)
                          FROM facts)
    AND area < (SELECT AVG(area)
                          FROM facts)
    ORDER BY population DESC;
```

```
 * sqlite:///factbook.db
Done.
```

Out[44]:

Code	Country	Population	Area
bg	Bangladesh	168957745	148460
ja	Japan	126919659	377915

rp	Philippines	100998376	300000
vm	Vietnam	94348835	331210
gm	Germany	80854408	357022
th	Thailand	67976405	513120
uk	United Kingdom	64088222	243610
it	Italy	61855120	301340
ks	Korea, South	49115196	99720
sp	Spain	48146134	505370
pl	Poland	38562189	312685
ug	Uganda	37101745	241038
iz	Iraq	37056169	438317
mo	Morocco	33322699	446550

We see some countries that are known as densely populated countries so we have confidence in our results.

Further Exploration of the Data

Now we will find top 5 countries that have highest population.

First we will find the average population and use it to find the countries with population greater than the average.

In [27]:

```
%%sql
SELECT code AS Code, name AS Country, population AS Population
    FROM facts
    WHERE population > (SELECT AVG(population)
    FROM facts)
    ORDER BY population DESC
    LIMIT 5;
```

```
 * sqlite:///factbook.db
Done.
```

Out[27]:

Code	Country	Population

ch	China	1367485388
in	India	1251695584
ee	European Union	513949445
us	United States	321368864
id	Indonesia	255993674

Now lets find out TOP 5 countries with highest growth rate

In [32]:

```
%%sql
SELECT code AS Code, name AS Country, population_growth AS Population_Growth
    FROM facts
    WHERE population_growth > (SELECT AVG(population_growth)
                    FROM facts)
    ORDER BY population_growth DESC
    LIMIT 5;
```

```
 * sqlite:///factbook.db
Done.
```

Out[32]:

Code	Country	Population_Growth
od	South Sudan	4.02
mi	Malawi	3.32
by	Burundi	3.28
ng	Niger	3.25
ug	Uganda	3.24

Finding the countries which have more water than land

In [34]:

```
%%sql
SELECT name AS Country, area_water AS Water, area_land AS Land
    FROM facts
    WHERE area_water > area_land
    ORDER BY area_water
```

```
 * sqlite:///factbook.db
Done.
```

Out[34]:

Country	Water	Land
Virgin Islands	1564	346
British Indian Ocean Territory	54340	60

We will find TOP 10 countries which will be adding most people to their population next year.

In [35]:

```
%%sql
SELECT name AS Country, migration_rate AS Migration_rate
    FROM facts
    WHERE migration_rate > (SELECT AVG(migration_rate)
    FROM facts)
    ORDER BY migration_rate DESC
    LIMIT 10;
```

```
 * sqlite:///factbook.db
Done.
```

Out[35]:

Country	Migration_rate
Qatar	22.39
American Samoa	21.13
Micronesia, Federated States of	20.93
Syria	19.79
Tonga	17.84
British Virgin Islands	17.28
Luxembourg	17.16
Cayman Islands	14.4
Singapore	14.05
Nauru	13.63

Finding the countries which have more death rate than birth rate

In [37]:

```
%%sql
SELECT name AS Country, death_rate AS Death_Rate, birth_rate AS Birth_Rate
    FROM facts
    WHERE death_rate > birth_rate
    ORDER BY death_rate DESC;
```

```
 * sqlite:///factbook.db
Done.
```

Out[37]:

Country	Death_Rate	Birth_Rate
Ukraine	14.46	10.72
Bulgaria	14.44	8.92
Latvia	14.31	10.0
Lithuania	14.27	10.1
Russia	13.69	11.6
Serbia	13.66	9.08
Belarus	13.36	10.7
Hungary	12.73	9.16
Moldova	12.59	12.0
Estonia	12.4	10.51
Croatia	12.18	9.45
Romania	11.9	9.14
Germany	11.42	8.47
Slovenia	11.37	8.42
Greece	11.09	8.66
Portugal	11.02	9.27
Czech Republic	10.34	9.63
Italy	10.19	8.74
Poland	10.19	9.74
Bosnia and Herzegovina	9.75	8.87
Saint Pierre and Miquelon	9.72	7.42
Japan	9.51	7.93
Austria	9.42	9.41
Monaco	9.24	6.65

Finally we will find TOP 10 countries with highest Population per Area Ratio by dividing population and area to get population per area ratio.

In [45]:

```
%%sql
SELECT name AS Country, CAST(population AS float)/CAST(area AS float) AS
"Population Per Area Ratio"
    FROM facts
    ORDER BY CAST(population AS float)/CAST(area AS float) DESC
    LIMIT 10;
```

```
 * sqlite:///factbook.db
Done.
```

Out[45]:

Country	Population Per Area Ratio
Macau	21168.964285714286
Monaco	15267.5
Singapore	8141.279770444763
Hong Kong	6445.041516245487
Gaza Strip	5191.819444444444
Gibraltar	4876.333333333333
Bahrain	1771.8592105263158
Maldives	1319.6409395973155
Malta	1310.01582278481
Bermuda	1299.9259259259259

Conclusion

SQL makes it easy to analyse and interact with data. We were able to draw useful insights from the factbook by including some calculations and grouping of data.