**Data Engineering Challenge – Ernesto Rountri**

## Exercise 1: Loading Data

*Programmatically (Using Python/R or other programming language) retrieve JSON data from europe.eu (link below) and load the data to a PostgreSQL Database (install it locally).*

*Data source 1: Covid Data:* [*https://www.ecdc.europa.eu/en/publications-data/data-national-14-day-notification-rate-covid-19*](https://www.ecdc.europa.eu/en/publications-data/data-national-14-day-notification-rate-covid-19) *(use the Json file)*

*Load the CSV table Countries of the word.*

*Data source 2: Countries data:* [*https://www.kaggle.com/fernandol/countries-of-the-world/data?select=countries+of+the+world.csv*](https://www.kaggle.com/fernandol/countries-of-the-world/data?select=countries+of+the+world.csv)

*Alternative: Load the data manually (using some importing tool).*

**A: Please refer to the script “Exercise 1” attached on the email.**

## Exercise 2: Create a Pipeline

*Create a Data Pipeline that extracts the last version of the data (Covid-19 data, Data Source 1) and adds to the PostgreSQL database only the new records. (Note the Data source 1, Covid-19 dataset, changes one time per day)*

**A: Please refer to the script “Exercise 2” attached on the email.**

## Exercise 3: Create a View

*Create a view with the data of the table “Countries of the word” with the latest number of cases, “Cumulative\_number\_for\_14\_days\_of\_COVID-19\_cases\_per\_100000” and date when the Information was extracted.*

* **A: The below answers are also included in the “sql script.sql” file.**
* **A: Frist I changed the column names on the countries table:**

ALTER TABLE countries

RENAME COLUMN "Birthrate" TO birthrate;

ALTER TABLE countries

RENAME COLUMN "Deathrate" TO deathrate;

ALTER TABLE countries

RENAME COLUMN "Agriculture" TO agriculture;

ALTER TABLE countries

RENAME COLUMN "Industry" TO industry;

ALTER TABLE countries

RENAME COLUMN "Service" TO service;

ALTER TABLE countries

RENAME COLUMN "Country" TO country;

ALTER TABLE countries

RENAME COLUMN "Region" TO region;

ALTER TABLE countries

RENAME COLUMN "Population" TO population;

ALTER TABLE countries

RENAME COLUMN "Climate" TO climate;

ALTER TABLE countries

RENAME COLUMN pop\_density TO pop\_density\_sq\_mi;

ALTER TABLE countries

RENAME COLUMN "Area (sq. mi.)" TO area\_sq\_mi;

ALTER TABLE countries

RENAME COLUMN "Coastline (coast/area ratio)" TO coastline\_ratio;

ALTER TABLE countries

RENAME COLUMN "Net migration" TO net\_migration;

ALTER TABLE countries

RENAME COLUMN "Infant mortality (per 1000 births)" TO infant\_mortality;

ALTER TABLE countries

RENAME COLUMN "GDP ($ per capita)" TO gdp\_percapita;

ALTER TABLE countries

RENAME COLUMN "Literacy (%)" TO literacy\_perc;

ALTER TABLE countries

RENAME COLUMN "Phones (per 1000)" TO phones\_perthousands;

ALTER TABLE countries

RENAME COLUMN "Arable (%)" TO arable\_perc;

ALTER TABLE countries

RENAME COLUMN "Crops (%)" TO crops\_perc;

ALTER TABLE countries

RENAME COLUMN "Other (%)" TO other\_perc;

* **A: Removed the space at the end of the country name in the contry column in the countries database.**

UPDATE countries

SET country = RTRIM(country);

* **A: Created a new column to store the first day of the year-week in date format, as these were stored as text.**

ALTER TABLE covid\_data

ADD COLUMN year\_week\_date date;

UPDATE covid\_data

SET year\_week\_date = TO\_DATE(year\_week, 'IYYY-IW');

* **A: Create view query:**

CREATE VIEW covid\_cases\_per\_country AS

SELECT

c.country, c.region, c.population, c.area\_sq\_mi,

c.pop\_density\_sq\_mi, c.coastline\_ratio,

c.net\_migration, c.infant\_mortality,

c.gdp\_percapita, c.literacy\_perc, c.phones\_perthousands, c.arable\_perc,

c.crops\_perc, c.other\_perc, c.climate, c.birthrate, c.deathrate,

c.agriculture, c.industry, c.service, o.weekly\_count,

SUM (o.cumulative\_count) OVER (

ORDER BY year\_week\_date

ROWS BETWEEN 1 PRECEDING AND CURRENT ROW

) AS cumulative\_number\_for\_14\_days\_of\_COVID\_19\_cases\_per\_100000,

o.year\_week, o.year\_week\_date

FROM

countries c

JOIN

covid\_data o

ON

c.country = o.country

WHERE

o.indicator = 'cases'

ORDER BY year\_week\_date;

## Exercise 4: Queries

1. *What is the country with the highest number of Covid-19 cases per 100 000 Habitants at 31/07/2020?*

|  |  |
| --- | --- |
| country | total\_cases\_per\_100k\_hab |
| Spain | 5196026 |

* **A: Here I had to further alter the column types to accomplish the exercise. Converted the weekly\_count, cumulative\_count and pop\_density\_sq\_mi to numeric formats and then I had to recreate the view.**

ALTER TABLE covid\_data

ALTER COLUMN weekly\_count TYPE integer

ALTER TABLE covid\_data

ALTER COLUMN cumulative\_count TYPE integer

ALTER TABLE covid\_data

ALTER COLUMN pop\_density\_sq\_mi TYPE double precision;

* **A: SELECT query:**

SELECT

country,

SUM(cumulative\_number\_for\_14\_days\_of\_COVID\_19\_cases\_per\_100000) AS total\_cases\_per\_100k\_hab

FROM covid\_cases\_per\_country

WHERE year\_week\_date <= '2020-07-27'

GROUP BY country

ORDER BY total\_cases\_per\_100k\_hab DESC

LIMIT 1;

**Performance: 46 msec**

1. *What is the top 10 countries with the lowest number of Covid-19 cases per 100 000 Habitants at 31/07/2020?*

|  |  |
| --- | --- |
| country | total\_cases\_peer\_100k\_hab |
| Lithuania | 215841 |
| Latvia | 225464 |
| Liechtenstein | 286270 |
| Norway | 380127 |
| Malta | 429302 |
| Iceland | 488494 |
| Finland | 582915 |
| Ireland | 674286 |
| Slovakia | 676019 |
| Estonia | 743503 |

* **A: SELECT query**

SELECT

country,

SUM(cumulative\_number\_for\_14\_days\_of\_COVID\_19\_cases\_per\_100000) AS total\_cases\_per\_100k\_hab

FROM covid\_cases\_per\_country

WHERE year\_week\_date <= '2020-07-27'

GROUP BY country

ORDER BY total\_cases\_per\_100k\_hab ASC

LIMIT 10;

**Performance: 56 msec**

1. *What is the top 10 countries with the highest number of cases among the top 20 richest countries (by GDP per capita)?*

|  |  |  |
| --- | --- | --- |
| **country** | **total\_cases** | **gdp\_percapita** |
| France | 40086999 | 27600 |
| Germany | 38437756 | 27600 |
| Italy | 26345573 | 26700 |
| Spain | 13980340 | 22000 |
| Netherlands | 8618815 | 28600 |
| Austria | 6084155 | 30000 |
| Portugal | 5621015 | 18000 |
| Greece | 5405742 | 20000 |
| Belgium | 4803737 | 29100 |
| Denmark | 3105726 | 31100 |

* **A: Query**

WITH top20\_gdpcountries AS (

SELECT

country,

SUM(weekly\_count) AS total\_cases,

MAX(gdp\_percapita) as gdp\_percapita

FROM covid\_cases\_per\_country

GROUP BY country

ORDER BY gdp\_percapita DESC

LIMIT 20

)

SELECT

country,

total\_cases,

gdp\_percapita

FROM top20\_gdpcountries

ORDER BY total\_cases DESC

LIMIT 10;

**Performance 70 msec**

1. *List all the regions with the number of cases per million of inhabitants and display information on population density, for 31/07/2020.*

|  |  |  |
| --- | --- | --- |
| **region** | **cases\_per\_million** | **population\_density** |
| WESTERN EUROPE | 19180090 | 3545.1 |
| EASTERN EUROPE | 6149040 | 681.2 |
| BALTICS | 347180 | 119.5 |
| NEAR EAST | 61560 | 84.8 |

* **A: For this query I had to transform the values of the pop\_density\_sq\_mi column as they were comma separated values. I used the following queries to create a new column with the transformed values with “.” as their decimal separator.**

ALTER TABLE countries

ADD COLUMN pop\_density\_numeric numeric;

UPDATE countries

SET pop\_density\_numeric = REPLACE(pop\_density\_sq\_mi, ',', '.')::numeric;

ALTER TABLE countries

DROP COLUMN pop\_density\_sq\_mi;

ALTER TABLE countries

RENAME COLUMN pop\_density\_numeric TO pop\_density\_sq\_mi;

* **A: Now here is the query for the main request:**

SELECT

region,

SUM(cumulative\_number\_for\_14\_days\_of\_covid\_19\_cases\_per\_100000) \* 10 AS cases\_per\_million,

SUM (pop\_density\_sq\_mi) AS population\_density

FROM

covid\_cases\_per\_country

WHERE

year\_week = '2020-31'

GROUP BY

region

ORDER BY

cases\_per\_million DESC;

**Performance: 36 msec**

1. *Query the data to find duplicated records.*

* **For the covid data table I was able to run two queries using the following considerations:**

1. **A: 1st query includes the columns country, weekly\_count, cumulative\_count, rate\_14\_day and year\_week. It brings back 532 duplicate rows.**

WITH covid\_data\_duplicates AS (

SELECT country, weekly\_count, cumulative\_count, rate\_14\_day, year\_week, COUNT(\*) AS duplicate\_count

FROM covid\_data

GROUP BY country, weekly\_count, cumulative\_count, rate\_14\_day, year\_week

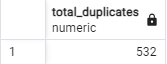
HAVING COUNT(\*) > 1

)

SELECT

SUM(duplicate\_count) AS total\_duplicates

FROM covid\_data\_duplicates;



1. **A: 2nd query only considers columns country, weekly\_count and year week. It brings back 592 duplicate rows. To which I think is mostly influenced by weekly\_counts with null values and some of them with actual duplicate counts for the same week (submitting the same weekly count record more than once). I consider this query to be the most accurate as the other columns: cumulative\_count and rate\_14\_day are derived from the column weekly\_count.**

WITH covid\_data\_duplicates AS (

SELECT country, weekly\_count, year\_week, COUNT(\*) AS duplicate\_count

FROM covid\_data

GROUP BY country, weekly\_count, year\_week

HAVING COUNT(\*) > 1

)

SELECT

SUM(duplicate\_count) AS total\_duplicates

FROM covid\_data\_duplicates;



**Performance: 71 msec**

1. *Analyze the performance of all the queries and describes what you see. Get improvements suggestions.*

**A**: By running all the queries, I found that all of them run under 71, but there is always room for improvement. Here are a couple of thinks I can do to improve query performance:

1. Create indexes on the tables referenced by the view covid\_cases\_per\_country. I would create the index on the country column of both the covid\_data table and the countries table.
2. I can remove the rows with NULL values in the weekly\_count column.

## Exercise 5:

## *Enrich the information with any other piece of data (available on the web) and justify the choice.*

A: I was not able to complete this step, but my inclination is to look for an API that contains the data relevant to my research interests. Once I find such an API, I will proceed to install it and extract the necessary information from the API's response. To do this, I'll map the fields in the API response to match the schema of my dataset, and I'll ensure that I handle the data types correctly during this process. I would use an API for the following reasons:

1. Because an alternative would be to use web scrapping, which takes more effort and time to extract the data. With web scrapping I may need to perform additional data cleaning and transformation.
2. APIs are designed for programmatic data access, making data retrieval more reliable and structured. APIs return data in a structured format as JSON or XML.
3. APIs usually come with detailed documentation.
4. APIs are maintained by the service provider.

## Exercise 6:

## *Produce a report showing the most useful discoveries you have made. Graphical representation is welcome.*

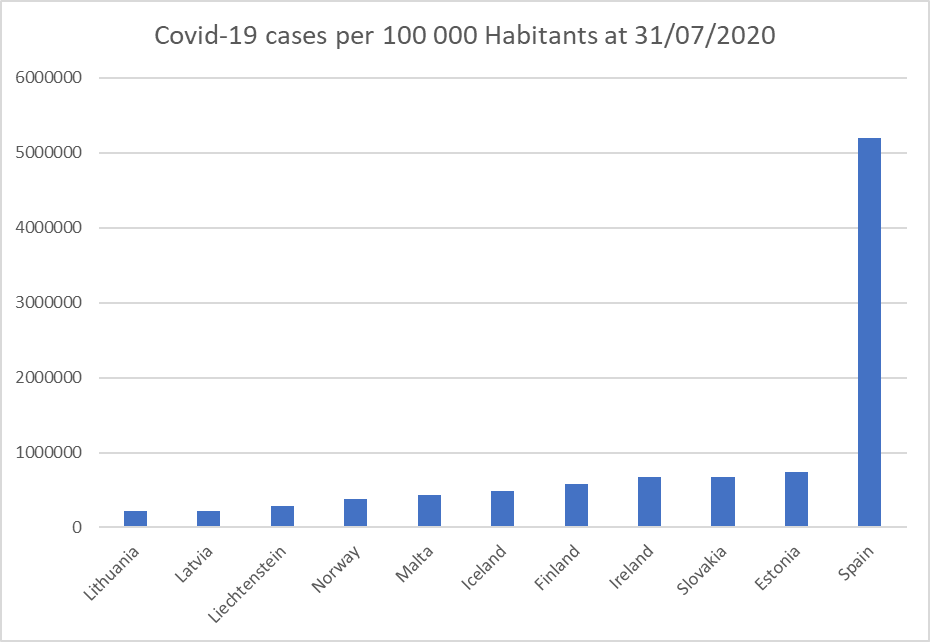
**A**: Countries with lower population density experienced significantly fewer cases per 100,000 inhabitants. This trend can be attributed to the fact that the coronavirus primarily spreads through the air, making transmission more likely among individuals in close proximity.

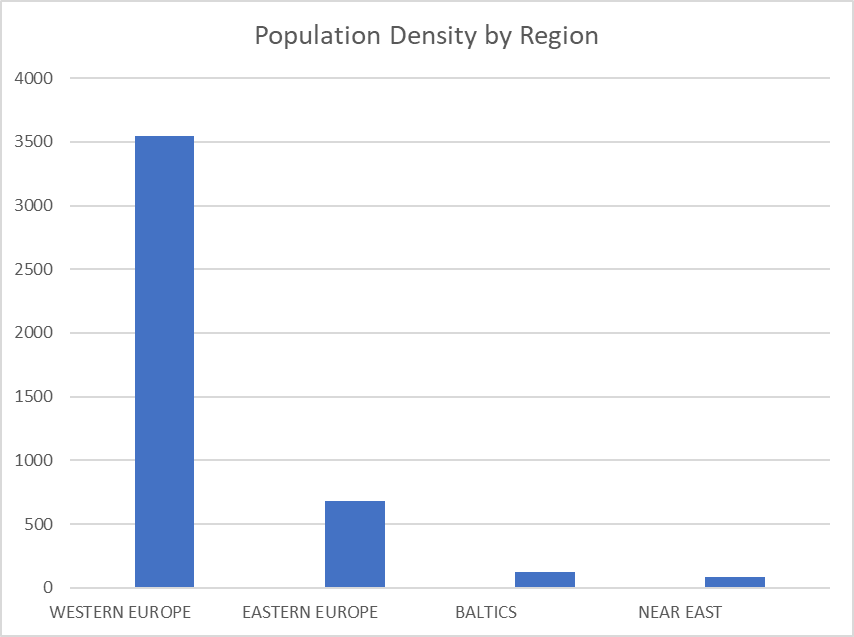
Many of the countries with low case numbers took proactive measures to contain the virus's spread. These measures included implementing early lockdowns, enforcing social distancing, mandating mask usage, and imposing travel restrictions. However, Spain stood out with a high number of COVID-19 cases per 100,000 inhabitants by the 31st week of 2020.

Spain's population density, with approximately 80 people per square mile, is relatively high compared to countries like Lithuania (55) and Estonia (29.3). Additionally, Spain was dealing with the challenges of a second wave of cases during that period:

*Spain lifted a nationwide state of emergency on June 21 as it emerged from a strict three-month lockdown imposed to gain control over one of Europe’s worst outbreaks. The rapid spread of the virus in Spain left some 28,400 people dead by the official count — almost surely understated — with most of them over the age of 70.*

*https://www.nytimes.com/2020/07/23/world/europe/spain-coronavirus-reopening.html*





|  |  |  |  |
| --- | --- | --- | --- |
| **Western Europe** | **Eastern Europe** | **Baltics** | **Near East** |
| Liechtenstein  Norway  Iceland  Ireland  Spain  France  Germany  Italy  Netherlands  Austria  Portugal  Belgium  Denmark | Slovakia | Lithuania  Latvia  Estonia | Malta  Finland  Greece |