Project Phase 3: OLAP Queries, and BI Dashboard

CSI4142 - Fundamentals of Data Science

Winter 2023

uOttawa

Faculté de génie
Faculty of Engineering

School of Electrical Engineering and Computer Science
University of Ottawa

Professor Yazan Otoum

Group 10:
Mazharul Maaz - 300128179
Bill Battushig - 300109257
Xiao Meng Li - 300109886

Due Date: Apr 11th, 2023

Table of Contents

Table of Contents **2**

Part A.1. Standard OLAP operations: **3**

a. Drill down and roll up: by using concept hierarchies in your data mart: 3

b. Slice, where only one dimension is selected: 4

c. Dice, where one creates a sub-cube: 4

d. Combining OLAP operations: 5

Part A.2. Explorative operation: **6**

a. Iceberg queries: 6

b. Windowing queries: 7

c. Using the Window clause: 8

Part B. BI dashboard and Information Visualization **9**

Screenshots of Business Intelligence (BI) Dashboard that show the functionality 9

Roll up and Drill Down: 10

Slicing: 11

Distribution of Tasks **13**

Part A.1. Standard OLAP operations:

a. Drill down and roll up: by using concept hierarchies in your data mart:

```
WITH
time_hierarchy AS (
    SELECT
        id,
        Month,
        CEIL(Month / 3.0)::INT AS Quarter,
        Year,
        (Year / 10) * 10 AS Decade
    FROM output_table
)
SELECT
    cs.Region,
    th.Year,
    th.Month,
    th.Quarter,
    th.Decade,
    SUM(cs.Total_cases) AS Total_cases,
    SUM(cs.New_cases) AS New_cases,
    SUM(cs.Total_deaths) AS Total_deaths,
    SUM(cs.New_deaths) AS New_deaths
FROM
    output_table cs
JOIN
    time_hierarchy th ON cs.id = th.id
GROUP BY
    ROLLUP (
        cs.Region,
        th.Year,
        th.Month,
        th.Quarter,
        th.Decade
    )
ORDER BY
    cs.Region,
    th.Year,
    th.Month,
    th.Quarter,
    th.Decade;
```

The above query calculates the Quarter and Decade based on the Month and Year columns. It then groups the data by Continent, Region, Year, Month, Quarter, and Decade. It returns the aggregates for each group.

b. Slice, where only one dimension is selected:

```
SELECT
    Year,
    Month,
    SUM(Total_cases) AS Total_cases,
    SUM(New_cases) AS New_cases,
    SUM(Total_deaths) AS Total_deaths,
    SUM(New_deaths) AS New_deaths
FROM
      output_table
WHERE
    Region = 'NORTHERN AFRICA                    '
GROUP BY
    Year,
    Month
ORDER BY
    Year,
    Month;
```

The above query creates a slice of the data where the Region dimension is fixed. The Region in this case is Northern Africa. It groups the data by Year and Month. It also calculates the aggregates for each group.

c. Dice, where one creates a sub-cube:

```
SELECT
    Region,
    Year,
    Month,
    SUM(Total_cases) AS Total_cases,
    SUM(New_cases) AS New_cases,
    SUM(Total_deaths) AS Total_deaths,
    SUM(New_deaths) AS New_deaths
FROM
    output_table
WHERE
```

```
    Region IN ('EASTERN EUROPE                    ', 'NORTHERN AFRICA
')
    AND Year IN (2020, 2021)
GROUP BY
    Region,
    Year,
    Month
ORDER BY
    Region,
    Year,
    Month;
```

The above query selects data for both Eastern Europe and Northern Africa from 2020 and 2021. This in turn creates a sub-cube of the data where both the Region and Year dimensions are fixed. It then groups the data and calculates the aggregates for each group.

d. Combining OLAP operations:

```
WITH
time_hierarchy AS (
    SELECT
        id,
        Year,
        Month,
        CEIL(Month / 3.0)::INT AS Quarter
    FROM output_table
),
unemployment_categories AS (
    SELECT
        id,
        CASE
            WHEN Unemployment_rate < 5 THEN 'Low'
            WHEN Unemployment_rate < 10 THEN 'Moderate'
            ELSE 'High'
        END AS Unemployment_category
    FROM output_table
)
SELECT
    cs.Region,
    th.Year,
    th.Quarter,
    uc.Unemployment_category,
```

```
    SUM(cs.Total_cases) AS Total_cases,
    SUM(cs.New_cases) AS New_cases,
    SUM(cs.Total_deaths) AS Total_deaths,
    SUM(cs.New_deaths) AS New_deaths
FROM
    output_table cs
JOIN
    time_hierarchy th ON cs.id = th.id
JOIN
    unemployment_categories uc ON cs.id = uc.id
GROUP BY
    ROLLUP (
        cs.Region,
        th.Year,
        th.Quarter,
        uc.Unemployment_category
    )
ORDER BY
    cs.Region,
    th.Year,
    th.Quarter,
    uc.Unemployment_category;
```

The above query calculates the Quarter based on the Month column and creates categories for the Unemployment_rate column: Low for less than 5%, Moderate for values between 5% and 10%, and High for more than 10%. It then groups the data by Region, Year, Quarter, and Unemployment_category. Using the ROLLUP function to create groups at different levels of the hierarchy. Finally, the query returns the aggregated values for each group.

Part A.2. Explorative operation:

a. Iceberg queries:

```
SELECT
    Year,
    Month,
    SUM(New_cases) AS Total_new_cases
FROM
    output_table
GROUP BY
    Year,
```

```
    Month
ORDER BY
    Total_new_cases DESC
LIMIT 5;
```

The above query calculates the total number of new cases for each group and sorts the results in descending order of the total number of cases. It displays the top 5 months with the highest number of new cases.

b. Windowing queries:

```
WITH
last_five_years AS (
    SELECT DISTINCT Year
    FROM output_table
    ORDER BY Year DESC
    LIMIT 5
)
SELECT
    cs.Year,
    cs.Country_name,
    cs.Literacy_rate,
    RANK() OVER (
        PARTITION BY cs.Year
        ORDER BY cs.Literacy_rate DESC
    ) AS Literacy_rank
FROM
    output_table cs
JOIN
    last_five_years lf ON cs.Year = lf.Year
WHERE
    cs.Literacy_rate IS NOT NULL
ORDER BY
    cs.Year DESC,
    Literacy_rank;
```

The above query selects the last five years from the fact table, and then for each year, it calculates the ranking of countries based on their literacy rates.

c. Using the Window clause:

```
WITH
denmark_data AS (
    SELECT
        Year,
        Month,
        SUM(Total_cases) AS Total_cases
    FROM
        output_table
    WHERE
        Country_name = 'Denmark'
    GROUP BY
        Year,
        Month
),
monthly_cases_with_lag_lead AS (
    SELECT
        Year,
        Month,
        Total_cases,
        LAG(Total_cases) OVER (ORDER BY Year, Month) AS Prev_month_cases,
        LEAD(Total_cases) OVER (ORDER BY Year, Month) AS Next_month_cases
    FROM
        denmark_data
)
SELECT
    Year,
    Month,
    Total_cases,
    Prev_month_cases,
    Next_month_cases,
    (Total_cases - Prev_month_cases) * 100.0 / Prev_month_cases AS
Prev_month_pct_change,
    (Next_month_cases - Total_cases) * 100.0 / Total_cases AS
Next_month_pct_change
FROM
    monthly_cases_with_lag_lead
ORDER BY
    Year,
    Month;
```

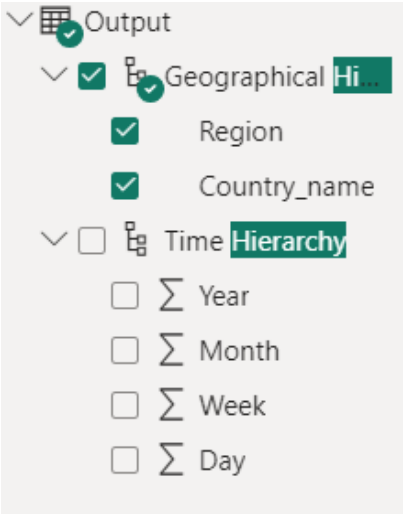
The above query selects the total number of COVID-19 cases in Denmark for each month. It then calculates the previous and next month's total cases using the window functions. Finally, the query calculates the percentage change in total cases compared to the previous and next months.

Part B. BI dashboard and Information Visualization

Screenshots of Business Intelligence (BI) Dashboard that show the functionality

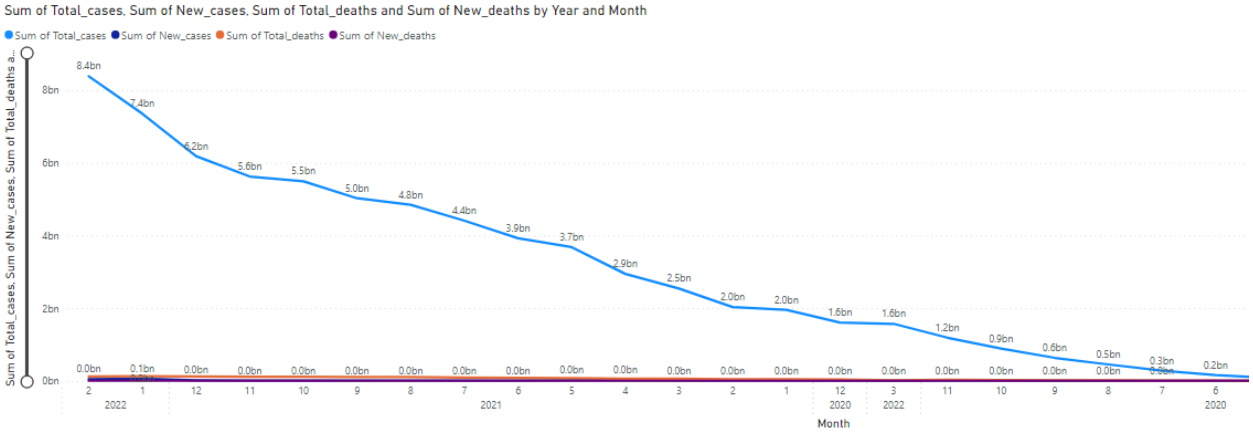
Dashboard Overall view:

Hierarchies:

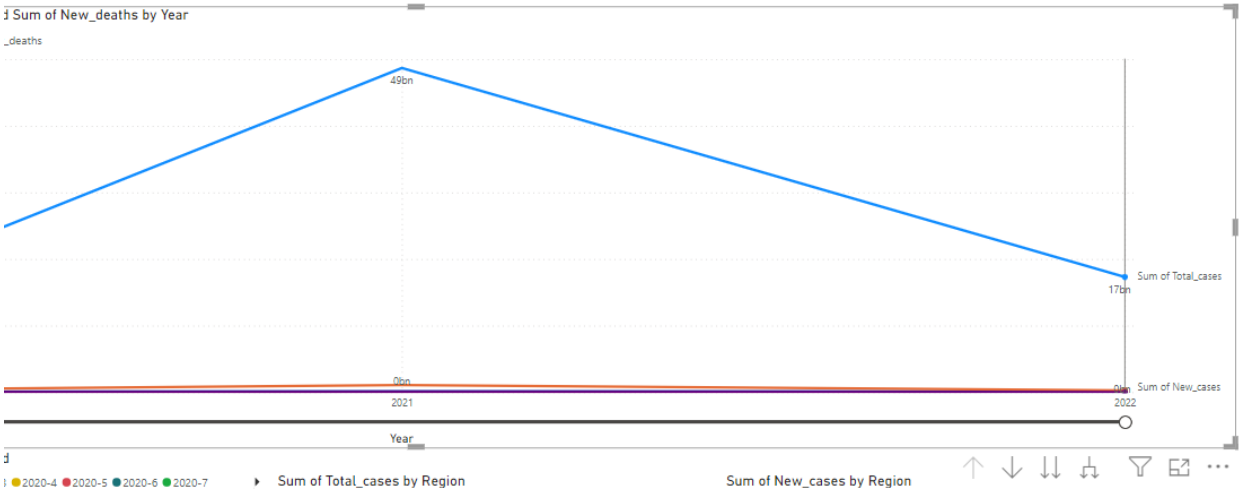


Roll up and Drill Down:

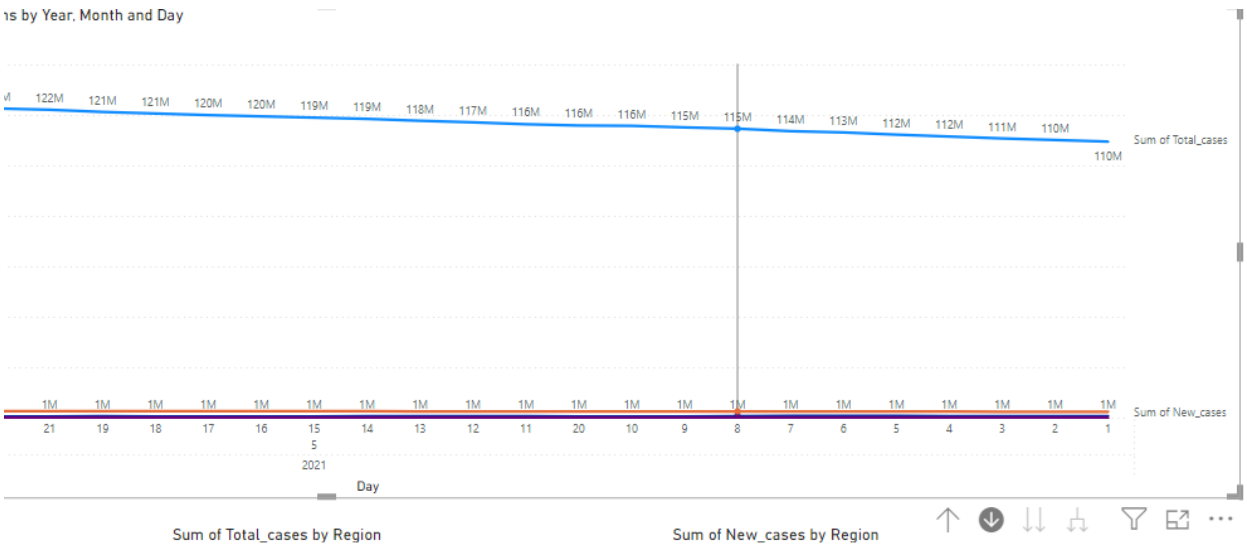
By Month:



By Year after clicking drill up:

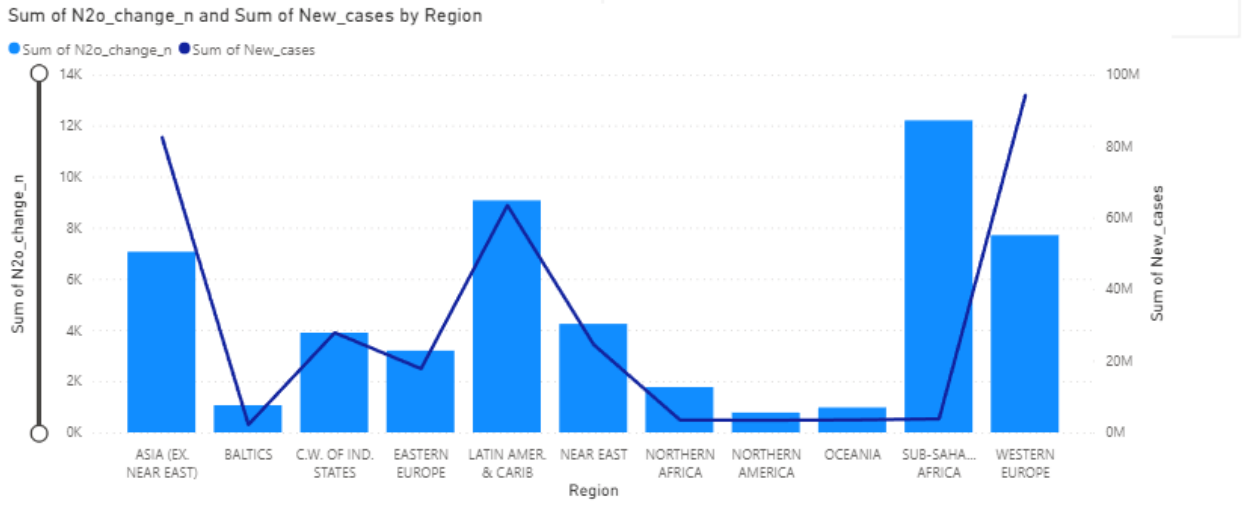


By Day after drill down:

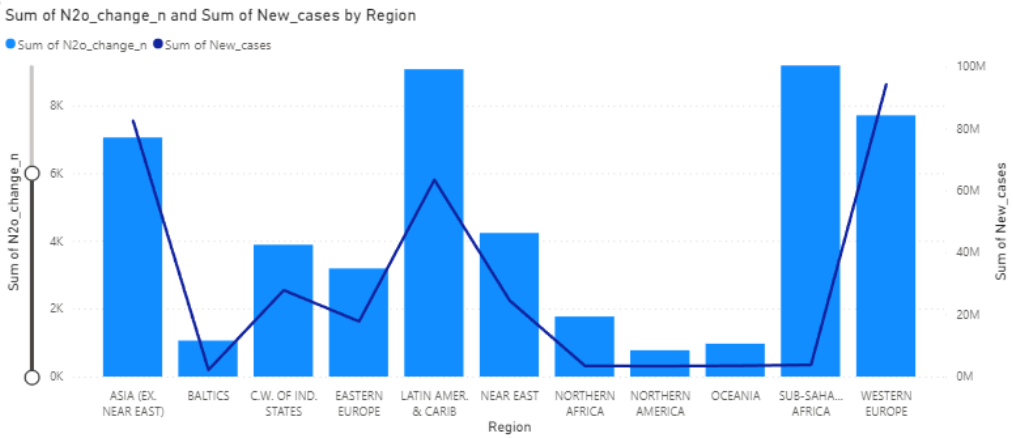


Slicing:

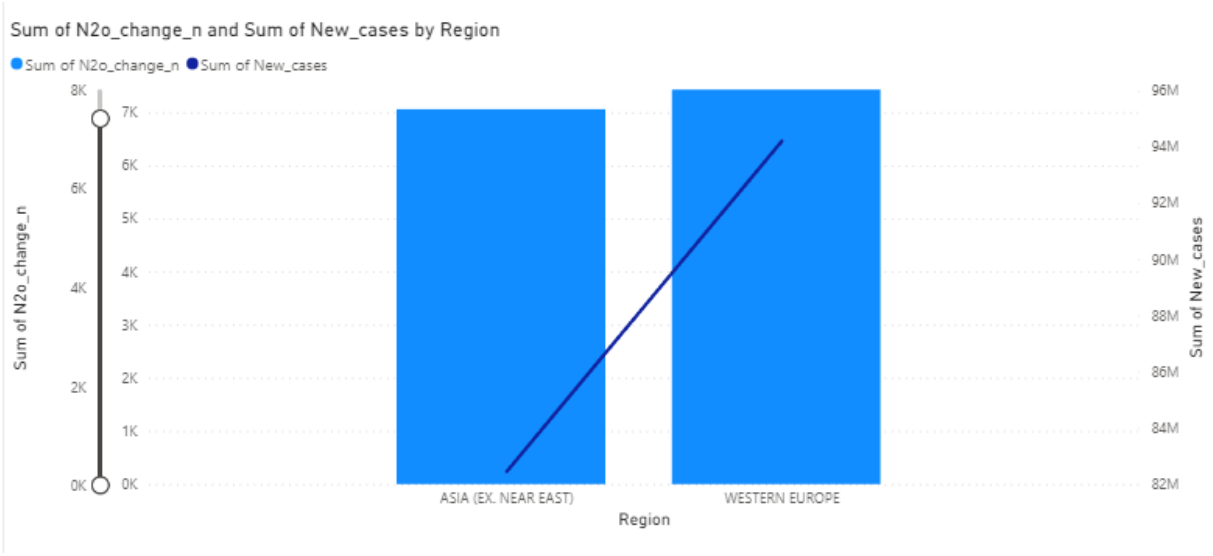
Using the slicing tool on the graph:

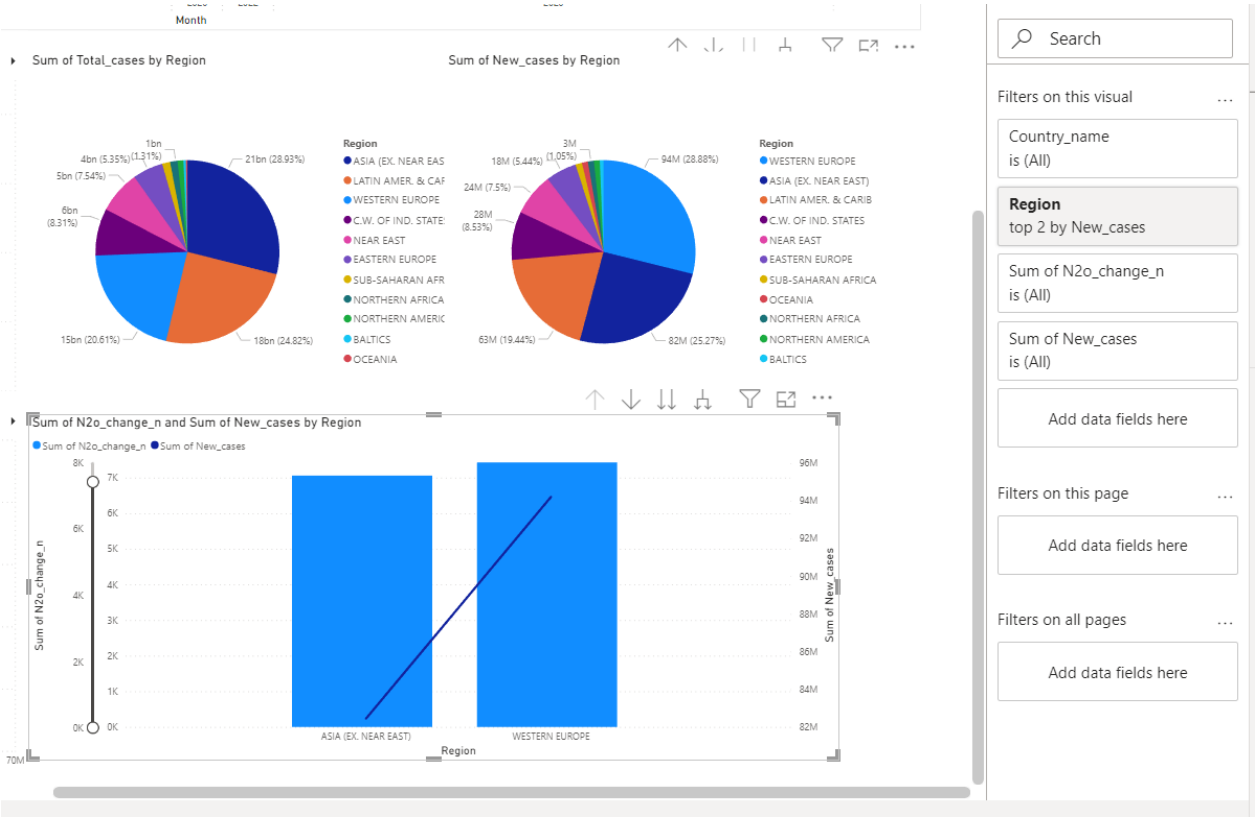


Sliced to around 6k



And we can further dice by adding filter to make a sub cube combined with slicing that was done before and also adding top N, for example get Top 2 of new cases by Region





Distribution of Tasks

CSI4142 - Project 2023
Phase 3- OLAP and BI Dashboard
Teamwork - breakdown of duties

Deliverable checklist	Responsible team member(s)	Expected completion date	Actual completion date	Estimated time (hours) to complete	Actual time (hours) to complete
OLAP queries					
Drill down, roll up	Bill Battushig	April 11th	April 07th	0.25	0.25
Drill down, roll up	Bill Battushig	April 11th	April 07th	0.25	0.25
Drill down, roll up	Bill Battushig	April 11th	April 07th	0.25	0.25
Drill down, roll up	Bill Battushig	April 11th	April 07th	0.25	0.25
Icebergs	Mazharul Maaz	April 11th	April 11th	0.25	0.25
Windowing - partition	Mazharul Maaz	April 11th	April 11th	0.25	0.25
Window	Mazharul Maaz	April 11th	April 11th	0.25	0.25
BI dashboard					
Design of data mart	Xiao Meng Li	April 11th	April 09th	0.25	0.33
Importing data	Xiao Meng Li	April 11th	April 09th	0.25	0.33
OLAP queries	Xiao Meng Li	April 11th	April 09th	0.25	0.33
Figures	Xiao Meng Li	April 11th	April 09th	0.25	0.33
Other tasks - please specify					