DATA WAREHOUSING AND OLAP

MODULE CODE:7BUIS010W COURSEWORK1 (2022-23)

NAME: KAMALJEET KAUR SIDHU

STUDENT ID: 19489382

Table of Contents

[1 Design a Conceptual Data warehouse Model 2](#_Toc130892973)

[i List of Functional Dependencies for the Integrated Relational Schema 2](#_Toc130892974)

[ 8 Primary keys corresponding to 8 FD’s 2](#_Toc130892975)

[ii Build the Attribute Tree from Integrated Relational Schema 2](#_Toc130892976)

[ Construction of Attribute tree 2](#_Toc130892977)

[ Pruning of the Attribute Tree 4](#_Toc130892978)

[iii Build the Fact Schema from Attribute Tree 5](#_Toc130892979)

[ Identification of Facts: 7](#_Toc130892980)

[ Identification of Measures: 7](#_Toc130892981)

[ Identification of Dimensions: 7](#_Toc130892982)

[2 Map the DFM model to logical model. Clearly display the main facts table(s) and dimensions. 8](#_Toc130892983)

[3 Provide the DDL statements to create the proposed data-warehouse schema. 8](#_Toc130892984)

[4 Decide whether and which materialized views are convenient to improve response time of the frequent queries. 10](#_Toc130892985)

[ Calculation of Fact table(s) size 10](#_Toc130892986)

[ Matrix Specification, including involved queries against Group BY and where clauses 10](#_Toc130892987)

[ Justified choice of materialized view 10](#_Toc130892991)

[5 Provide and implement a materialised view(s) to answer the directors frequent queries 11](#_Toc130892992)

[1. **For each region and month report the** COVID-19 cases and deaths, Intensive care unit (ICU) cases as reported by different sources and vaccination rates for each vaccine. 12](#_Toc130892993)

[2. **For each country and quarter report the** COVID-19 cases and deaths, Intensive care unit cases and vaccination rates for each vaccine. 13](#_Toc130892994)

[i Visual representation based on resulted data from first query 14](#_Toc130892995)

[ii Visual representation based on resulted data from second query 16](#_Toc130892996)

[iii References: 17](#_Toc130892997)

# Design a Conceptual Data warehouse Model

## List of Functional Dependencies for the Integrated Relational Schema

## 8 Primary keys corresponding to 8 FD’s

**V-ID (VIRUS)** Name, Family, Type, Infection-Rate

**L-ID (LOCATION)** Country, Region, Population

**T-ID (TIME)** Year, Month, Date, Start-Date

**Vac-ID (VACCINE)** Name

**Trg-ID (TARGET GROUP)** Age-Range

**L-ID, T-ID, V-ID (PANDEMIC)** Current-Cases, New-Cases, Recovered, Deaths, Country, Region, Population, Year, Month, Date, Start-Date, Name, Family, Type, Infection-Rate

**Vac-ID, T-ID, L-ID, Trg-ID** Doses-Received, First-Dose, Second-Dose, Third-Dose, Year, Month, Date, Start-Date, Country, Region, Population, Age-Range

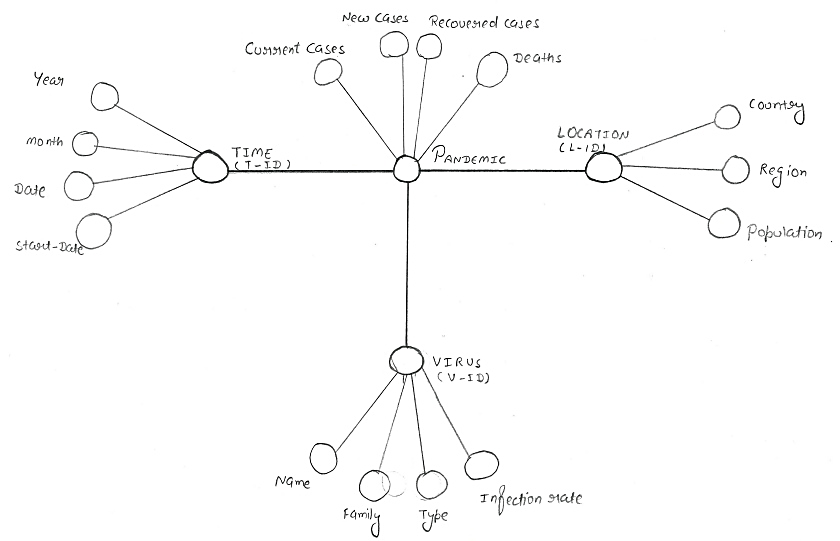
**L-ID, T-ID** Indicator, Source, ICU-Cases, Country, Region, Population, Year, Month, Date, Start-Date

## Build the Attribute Tree from Integrated Relational Schema

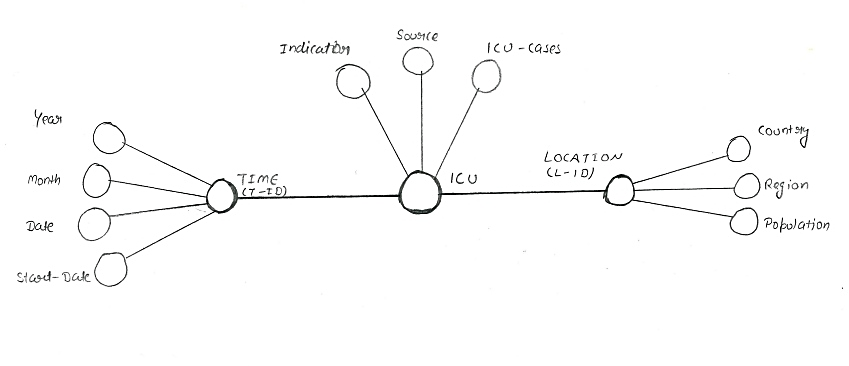
## Construction of Attribute tree

**Note: zoom in the pictures for clear visibility.**

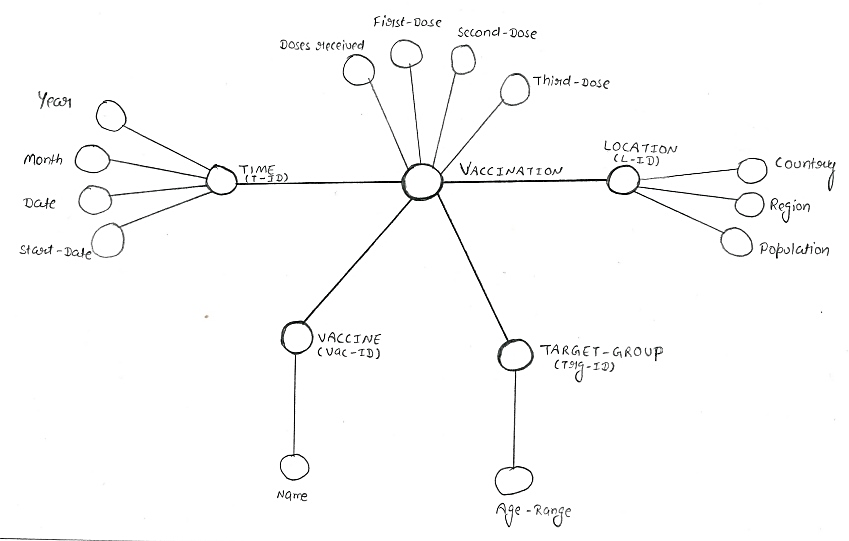
(PANDEMIC ATTRIBUTE TREE)

****

(ICU Attribute Tree)

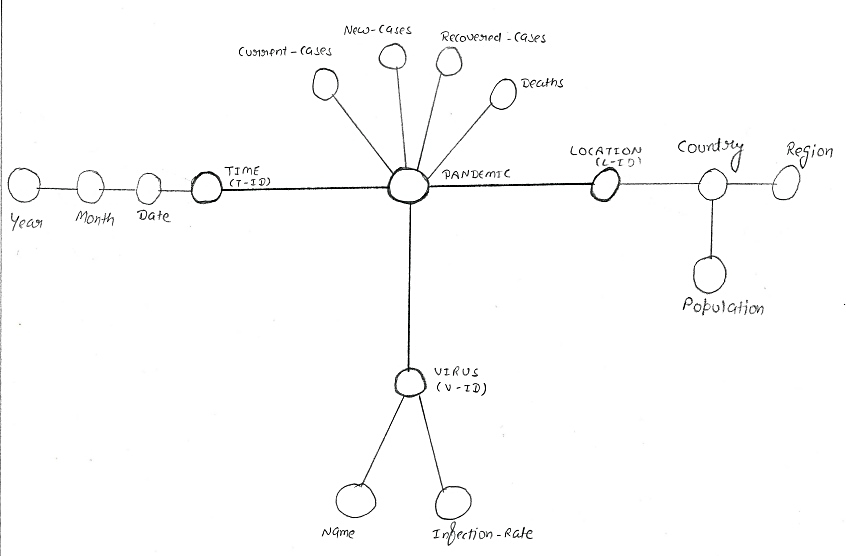


(Vaccination Attribute Tree)

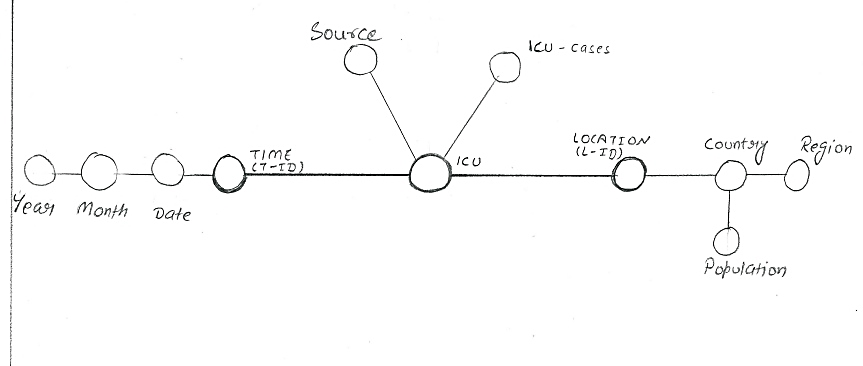


* Pruning of the Attribute Tree

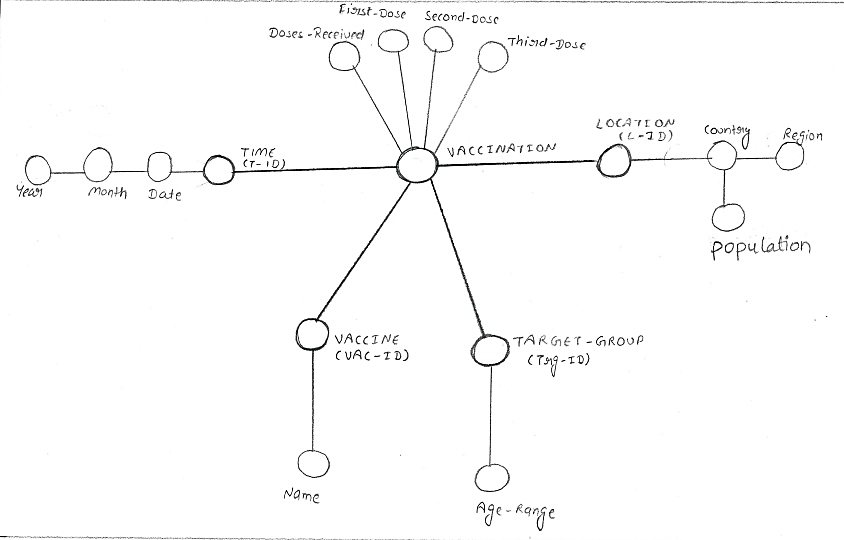
(Pandemic, ICU, Vaccination)

****

Pruned Attributes (Pandemic): Time (Start-date), Virus (Family, Type)



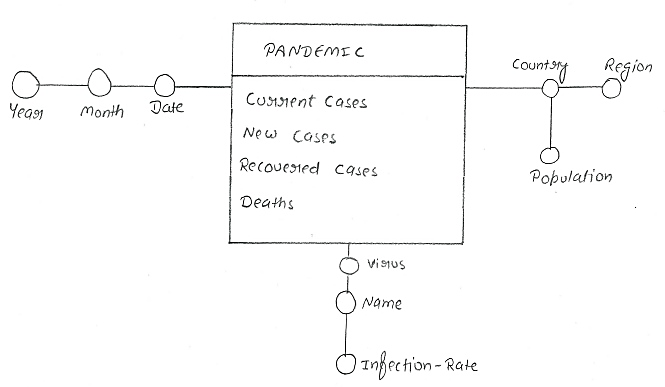
Pruned Attributes (ICU): Time (Start-date), ICU (Indicators)

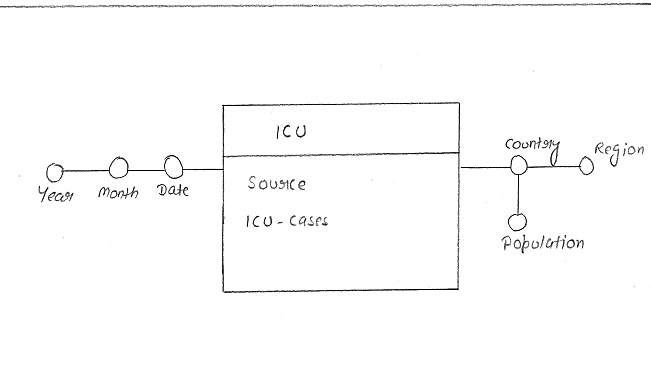
****

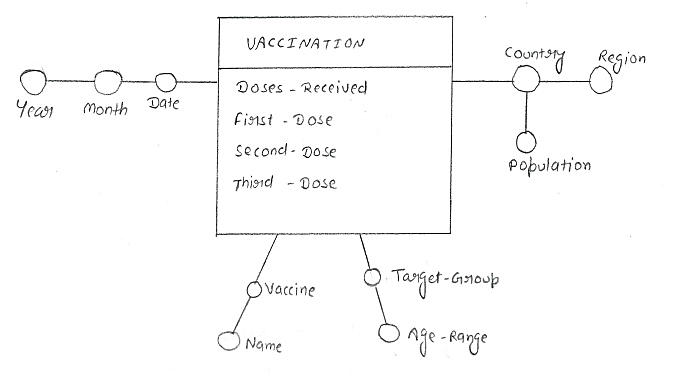
Pruned Attributes (Vaccination): Time (Start-date)

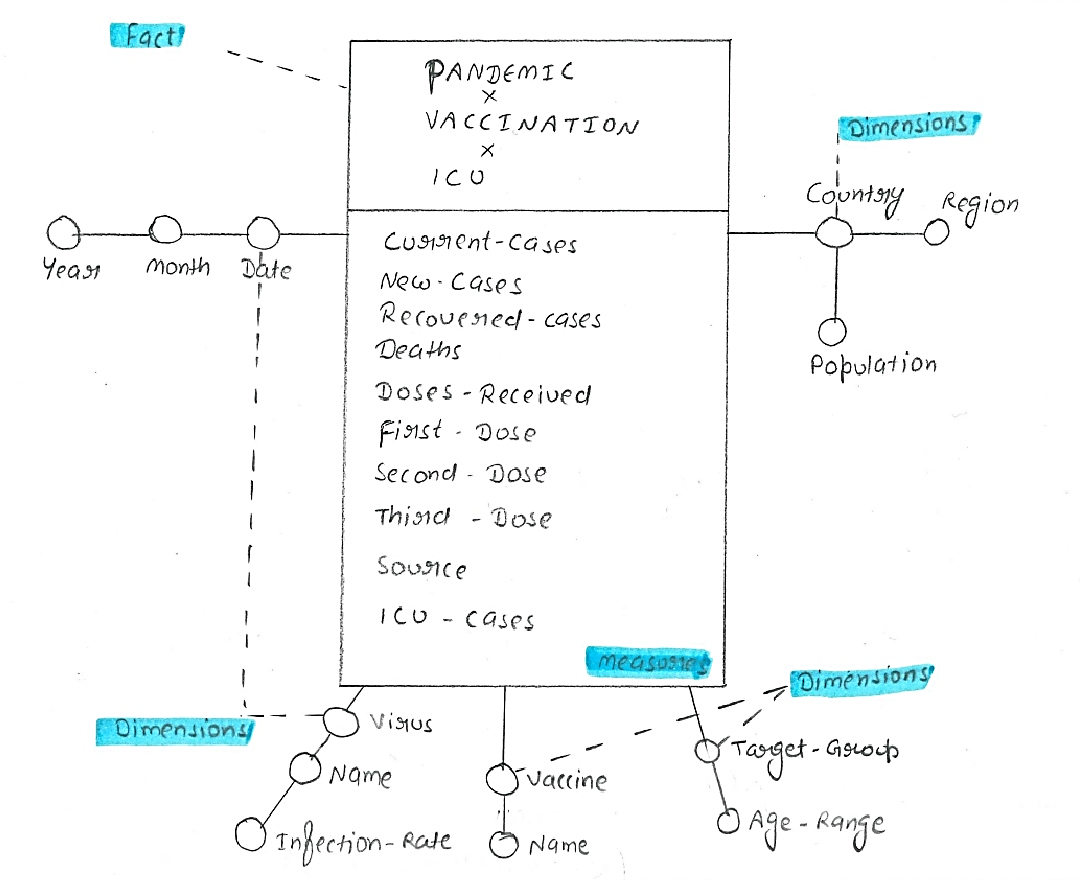
## Build the Fact Schema from Attribute Tree

(Identification of Measures, Facts and Dimensions)



****

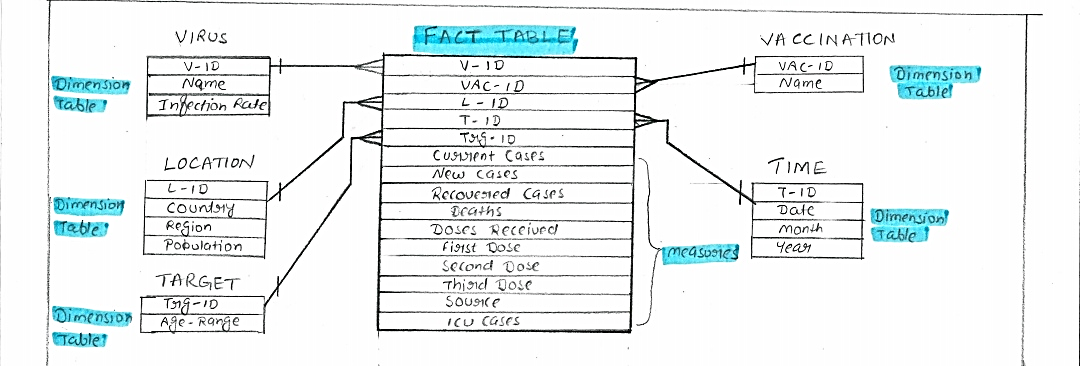
****

****

Merged Fact Schema (Pandemic, ICU, Vaccination ):

* Identification of Facts: PANDEMIC, ICU, VACCINATION
* Identification of Measures: Current-Cases, New-cases, Recovered-Cases, Deaths, Doses-Received, First-Dose, Second-Dose, Third-Dose, Source, ICU-Cases
* Identification of Dimensions: Date, Country, Vaccine, Target-Group, Virus

# Map the DFM model to logical model. Clearly display the main facts table(s) and dimensions.



Fact Tables: COVID19\_FACT\_TABLE

Dimension Tables: VIRUS, LOCATION, TARGET GROUP, VACCINATION, TIME

# Provide the DDL statements to create the proposed data-warehouse schema.

import sqlite3

import pandas as pd

conn = sqlite3.connect('.\sql\_db\Covid\_19\_Data.db')

cur = conn.cursor()

# create Dimension table Time in database

cur.execute('''CREATE TABLE TIME\_TABLE(

                T\_ID INTEGER PRIMARY KEY NOT NULL,

                Year INTEGER NULL,

                Month CHAR NULL,

                Date DATE NULL);

                ''')

# create Dimension table Virus in database

cur.execute('''CREATE TABLE VIRUS\_TABLE(

                V\_ID INTEGER PRIMARY KEY NOT NULL,

                Name CHAR NULL,

                Type CHAR NULL);

                ''')

# create Dimension table Location in database

cur.execute('''CREATE TABLE LOCATION\_TABLE(

              L\_ID INTEGER PRIMARY KEY NOT NULL,

                Country CHAR NULL,

                Region CHAR NULL,

                Population INTEGER NULL);

                ''')

# create Dimension table target Group in database

cur.execute('''CREATE TABLE TARGET\_GROUP\_TABLE(

              TRG\_ID INTEGER PRIMARY KEY NOT NULL,

                Age\_range CHAR NULL);

                ''')

# create Dimension table Vaccine in database

cur.execute('''CREATE TABLE VACCINE\_TABLE(

              VAC\_ID INTEGER PRIMARY KEY NOT NULL,

                Name CHAR NULL);

                ''')

# create Fact table COVID19\_FACT\_TABLE in database

cur.execute('''CREATE TABLE COVID19\_FACT\_TABLE(

                T\_ID INTEGER,

                L\_ID INTEGER,

                V\_ID INTEGER,

                VAC\_ID INTEGER,

                TRG\_ID INTEGER,

               Current\_Cases INTEGER NULL,

               New\_Cases INTEGER NULL,

               Recovered\_Cases INTEGER NULL,

               Deaths INTEGER NULL,

               Doses\_received INTEGER NULL,

               First\_Dose INTEGER NULL,

               Second\_Dose INTEGER NULL,

               Third\_Dose INTEGER NULL,

               ICU\_cases INTEGER NULL,

               Source CHAR NULL,

               FOREIGN KEY (T\_ID) REFERENCES TIME\_TABLE(T\_ID ),

               FOREIGN KEY (L\_ID) REFERENCES LOCATION\_TABLE(L\_ID ),

               FOREIGN KEY (V\_ID) REFERENCES VIRUS\_TABLE(V\_ID ),

               FOREIGN KEY (VAC\_ID) REFERENCES VACCINE\_TABLE(VAC\_ID ),

               FOREIGN KEY (TRG\_ID) REFERENCES TARGET\_GROUP\_TABLE(TRG\_ID)

               PRIMARY KEY(L\_ID,T\_ID,V\_ID,VAC\_ID,TRG\_ID)

               );

                ''')

# Decide whether and which materialized views are convenient to improve response time of the frequent queries.

## Calculation of Fact table(s) size

|  |  |
| --- | --- |
| **COVID19\_FACT\_TABLE Column Names(Data type)** | **Number of Bytes** |
| T\_ID (INTEGER) | 4 |
| L\_ID (INTEGER) | 4 |
| V\_ID (INTEGER) | 4 |
| VAC\_ID (INTEGER) | 4 |
| TRG\_ID (INTEGER) | 4 |
| Current\_Cases (INTEGER) | 4 |
| New\_Cases (INTEGER) | 4 |
| Recovered\_cases (INTEGER) | 4 |
| Deaths (INTEGER) | 4 |
| Doses\_received (INTEGER) | 4 |
| First\_Dose (INTEGER) | 4 |
| Second\_Dose (INTEGER) | 4 |
| Third\_Dose (INTEGER) | 4 |
| ICU\_Cases (INTEGER) | 4 |
| Source (CHAR) | 1 |
| **TOTAL** | 57 |

Size (COVID19\_FACT\_TABLE) = Number of bytes per column \* Total number of rows in the table

Size (COVID19\_FACT\_TABLE) = 57(Bytes) \* 48

Size (COVID19\_FACT\_TABLE) = 2736 Bytes

## Matrix Specification, including involved queries against Group BY and where clauses

|  |  |  |  |
| --- | --- | --- | --- |
| Individual query implementation | Group by Clause | Where Clause | Order by Clause |
| Materialized View(matz\_view) | Region, Country, Month, Vaccination Name, Source | Region, Country, T\_ID | Region, Country, T\_ID, Source |
| Query One | Region, Month, Vaccination Name | Region, Month | Region, Month |
| Query Two | Country, Quarter, Vaccine Name | Country, Quarter | Country, Quarter |

## Justified choice of materialized view

To answer both the queries frequently the following materialized view has been created to improve the response. It includes the attributes from different dimension tables. Like, LOCATION (LOC\_ID, Region, Country, and Population), TIME (T\_ID, Year, Month, and Quarter), VIRUS (V\_ID, Name), VACCINE (VAC\_ID, Name), and TARGET GROUP (VAC\_ID). From the COVID19\_FACT\_TABLE Covid Cases are calculated by adding the current and new cases, Deaths, ICU Cases and Source are added into the materialized view.

Additionally, Vaccination rate is calculated by taking the percentage of Doses Received divided by the population of each country. However, Infection rate is calculated by taking the percentage of infected people (Current Cases, New Cases) divided by population of each country.

Apart from answering the current queries this materialized view can help to analyse the total proportion of vaccinated population in each country. And in which country population is less infected by the virus. This view can provide the insights about the difference in the number of Covid Cases, Deaths and ICU Cases each month. Moreover, data can be filtered using the Primary Keys (LOC\_ID, T\_ID, V\_ID, VAC\_ID and VAC\_ID) of dimension tables.

# Provide and implement a materialised view(s) to answer the directors frequent queries

Note: Kindly refer this link to access code file for complete and appropriately formed results.

<https://github.com/kamalsidhu224/Datawarehouse-Coursework-files/blob/main/Covid_19_Coursework(2022_23)_DW%20(1).ipynb>

 # Create Materialized view

cur.execute('''CREATE VIEW matz\_view AS

                          Select

                          lt.Region as Region,

                          lt.Country as Country,

                          lt.Population as Population,

                          tt.Year as Year,

                          tt.Month as Month,

                          case when tt.Month in ('January') then 1

                          when tt.Month in ('February') then 1

                          when tt.Month in ('March') then 1

                          when tt.Month in ('April') then 2

                          when tt.Month in ('May') then 2

                          when tt.Month in ('June') then 2

                          when tt.Month in ('July') then 3

                          when tt.Month in ('August') then 3

                          when tt.Month in ('September') then 3

                          when tt.Month in ('October') then 4

                          when tt.Month in ('November') then 4

                          when tt.Month in ('December')then 4 end as Quarter,

                          covid.Current\_Cases,

                          covid.New\_Cases,

                          (covid.Current\_cases+covid.New\_cases) as Covid\_Cases,

                          covid.Deaths as Deaths,

                          covid.ICU\_cases as ICU\_Cases,

                          covid.Source as Source,

                          vt.Name as Vaccine\_Name,

                          ROUND((covid.Doses\_received\*100.00/lt.Population),2) as Vacc\_Rate\_Percentage ,

                          vit.Name as Virus\_name,

                          ROUND(((covid.Current\_cases+covid.New\_cases)\*100.00/lt.Population),2) as Infection\_Rate\_Percentage,

                          covid.Doses\_received as Doses\_Received,

                          covid.First\_Dose as First\_Dose,

                          covid.Second\_Dose as Second\_Dose,

                          covid.Third\_Dose as Third\_Dose,

                          covid.L\_ID as L\_ID,

                          covid.T\_ID as T\_ID,

                          covid.V\_ID as V\_ID,

                          covid.VAC\_ID as VAC\_ID,

                          covid.Trg\_ID as Trg\_ID,

                          tt.T\_ID as Month\_ID

 from COVID19\_FACT\_TABLE covid

 left join LOCATION\_TABLE lt on covid.L\_ID=lt.L\_ID

 left join TIME\_TABLE tt on covid.T\_ID=tt.T\_ID

 left join VACCINE\_TABLE vt on covid.VAC\_ID=vt.VAC\_ID

 left join VIRUS\_TABLE vit on covid.V\_ID=vit.V\_ID

 left join TARGET\_GROUP\_TABLE tgt on covid.TRG\_ID=tgt.Trg\_ID

 group by lt.Region,lt.Country,tt.Month,vt.Name,Source

 order by lt.Region,lt.Country,covid.T\_ID,Source;''')

## **For each region and month report the** COVID-19 cases and deaths, Intensive care unit (ICU) cases as reported by different sources and vaccination rates for each vaccine.

# Write query1 from matz\_view Materialized view

dataframe\_query\_one=pd.read\_sql\_query('''

                          Select

                          Region,

                          Month,

                          sum(Covid\_Cases) as Covid\_Cases,

                          sum(Deaths) as Deaths,

                          sum(ICU\_cases) as ICU\_Cases,

                          Source,

                          Vaccine\_Name,

                          sum(Vacc\_Rate\_Percentage)

from matz\_view

 group by Region,Month,Vaccine\_Name

 order by Region,Month\_ID;''',conn)

# commit and save changes to database

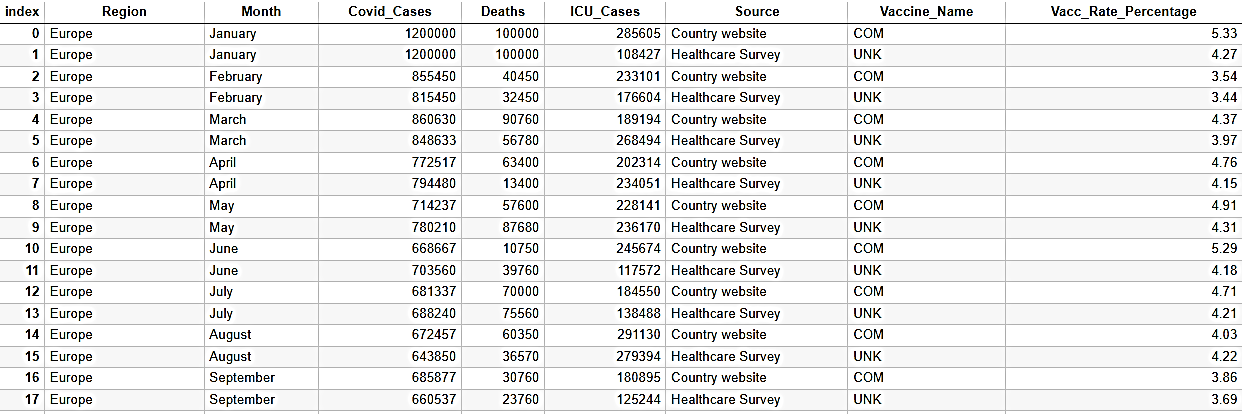
#where covid.T\_ID=tt.T\_ID and covid.L\_ID=lt.L\_ID

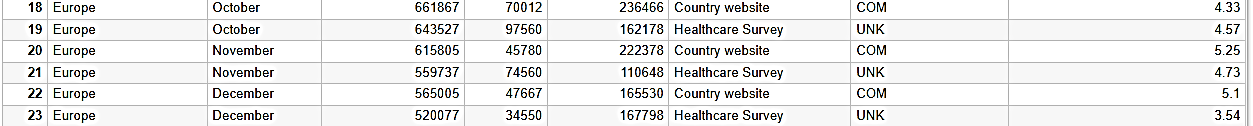
conn.commit()

dataframe\_query\_one

**Results based on Query one for one particular region.**

Note: Corrections filter from the format tab is applied on the picture to make it more visible.





## **For each country and quarter report the** COVID-19 cases and deaths, Intensive care unit cases and vaccination rates for each vaccine.

# Rewrite query2 from matz\_view Materialized view

dataframe\_query\_one=pd.read\_sql\_query('''

                          Select

                          Country,

                          case when Month in ('January') then 1

                          when Month in ('February') then 1

                          when Month in ('March') then 1

                          when Month in ('April') then 2

                          when Month in ('May') then 2

                          when Month in ('June') then 2

                          when Month in ('July') then 3

                          when Month in ('August') then 3

                          when Month in ('September') then 3

                          when Month in ('October') then 4

                          when Month in ('November') then 4

                          when Month in ('December')then 4 end as Quarter,

                          sum(Covid\_Cases) as Covid\_Cases,

                          sum(Deaths) as Deaths,

                          sum(ICU\_cases) as ICU\_Cases,

                          Vaccine\_Name,

                          Vacc\_Rate\_Percentage

 from matz\_view

 group by Country,Quarter,Vaccine\_Name

 order by Country,Quarter;''',conn)

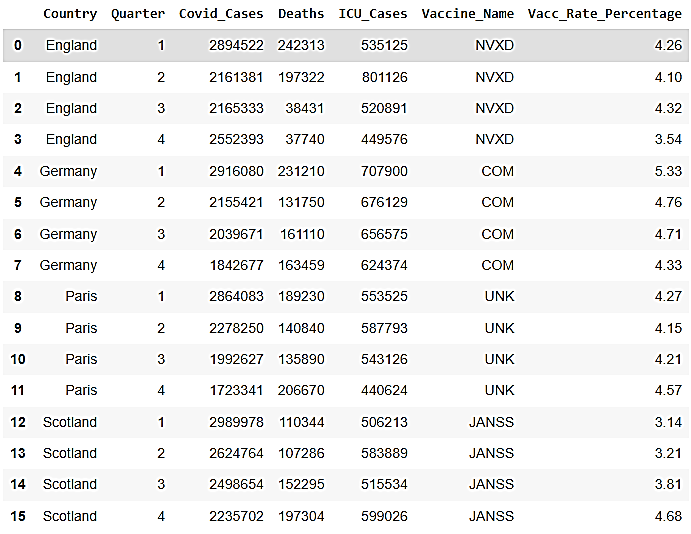
# commit and save changes to database

#where covid.T\_ID=tt.T\_ID and covid.L\_ID=lt.L\_ID

conn.commit()

dataframe\_query\_one

**Results based on Query two**



## Visual representation based on resulted data from first query

Note: Kindly move the cursor over the lines to know about the specific month and value as per four categories mentioned in the line graph.

## Visual representation based on resulted data from second query

## References:

Course Material

[Case When statement used to find the Quarter column](https://www.w3schools.com/sql/sql_ref_case.asp)

[Foreign key references used in the fact table](https://www.w3schools.com/sql/sql_foreignkey.asp)