PANDEMIC NEXUS

A Covid-19 Database Project

Nakul Krishnan (nxk230031@utdallas.edu)
Sai Anirudh Gandham (sxg230101@utdallas.edu)
Shobha Ojha (sxo210014@utdallas.edu)
Patrick Gervadis Ninan (pgn230000@utdallas.edu)
Shashank Marri (sxm220417@utdallas.edu)
Mittakanti Satyajeet Reddy (sxm230111@utdallas.edu)

TABLE OF CONTENT

INTRODUCTION

The COVID-19 pandemic has showcased the need for data management systems to track, analyze, and respond to the evolving public health situation. The implementation of an effective Database Management System (DBMS) for COVID-19 data is essential for organizing and managing information related to cases, testing, treatment, and vaccination. This project aims to design and implement a comprehensive DBMS focused on handling COVID-19 data.

Project Goals:

1. Data Organization: Establish a well-organized database structure to efficiently store and manage COVID-19- related data, ensuring easy retrieval and analysis.

2. Real-time Reporting: Implement mechanisms to support real-time reporting of COVID-19 statistics using Power Bi

3. Interconnected Information: Create relationships between different data entities to enable comprehensive analysis.

4. User Accessibility: Develop a user-friendly interface for authorized users to input, update, and retrieve data, catering to the needs of healthcare professionals, researchers, and administrators.

Assumptions:

1. Data Accuracy: It is assumed that the data entered into the system by healthcare providers and administrators is accurate and reflects the true state of COVID-19 cases, tests, and treatments.

2. Data Integration: It is assumed that data from various sources, such as testing centers, hospitals, and vaccination sites, can be integrated seamlessly into the DBMS to provide a unified and comprehensive view.

Limitations:

1. Schema Evolution Challenges: The project recognizes potential challenges in adapting the database schema to accommodate future data requirements. Rapid changes in data structures may pose difficulties in maintaining consistency and coherence.

2. Resource Intensiveness: The implementation of a scalable architecture might be resource-intensive, leading to potential constraints on hardware and infrastructure. Balancing scalability with resource availability may impact the system's overall performance.

CONCEPTUAL DESIGN

Here is the ER diagram generated based on our project description and real-life experiences.

2.1 ER diagram

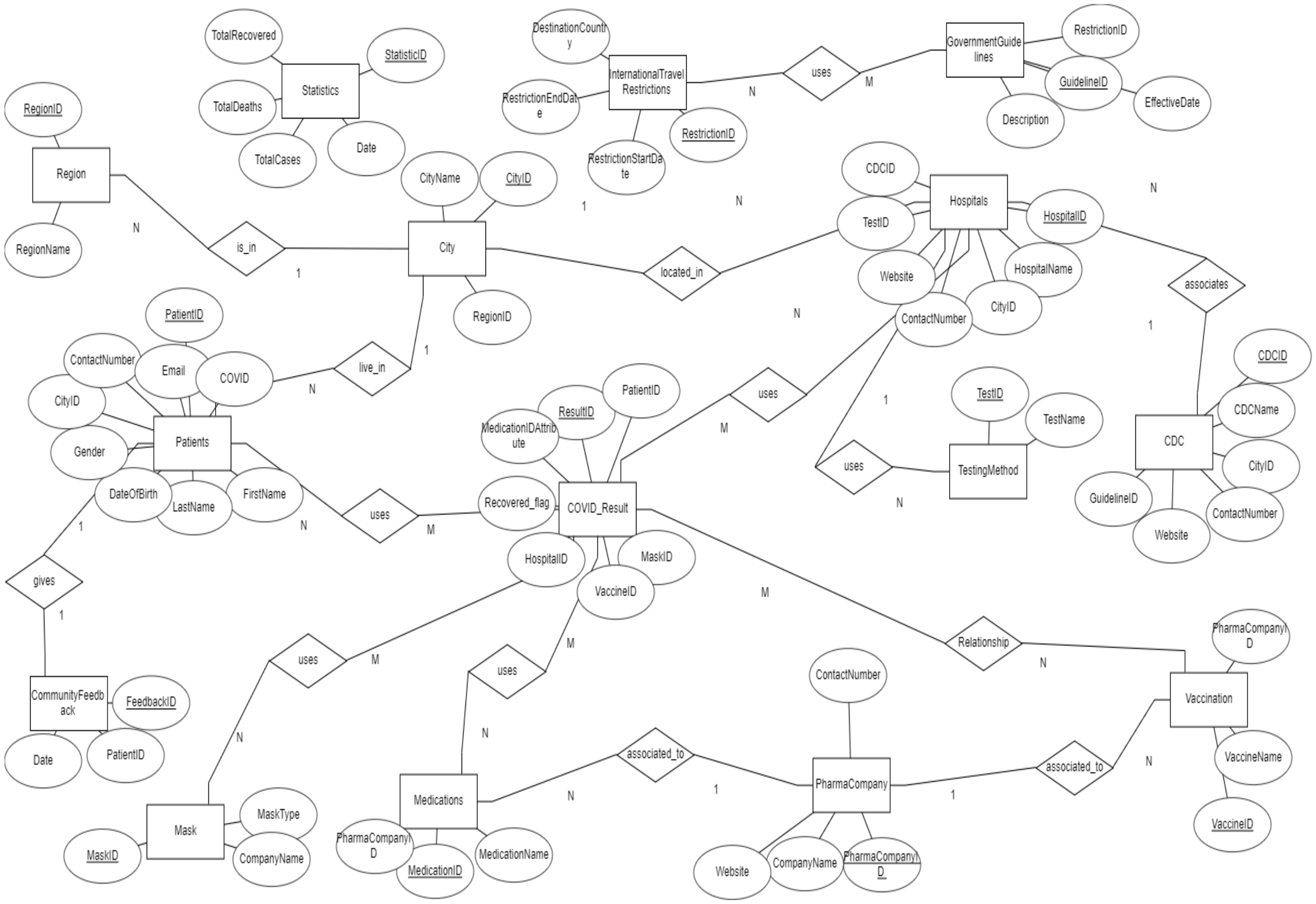
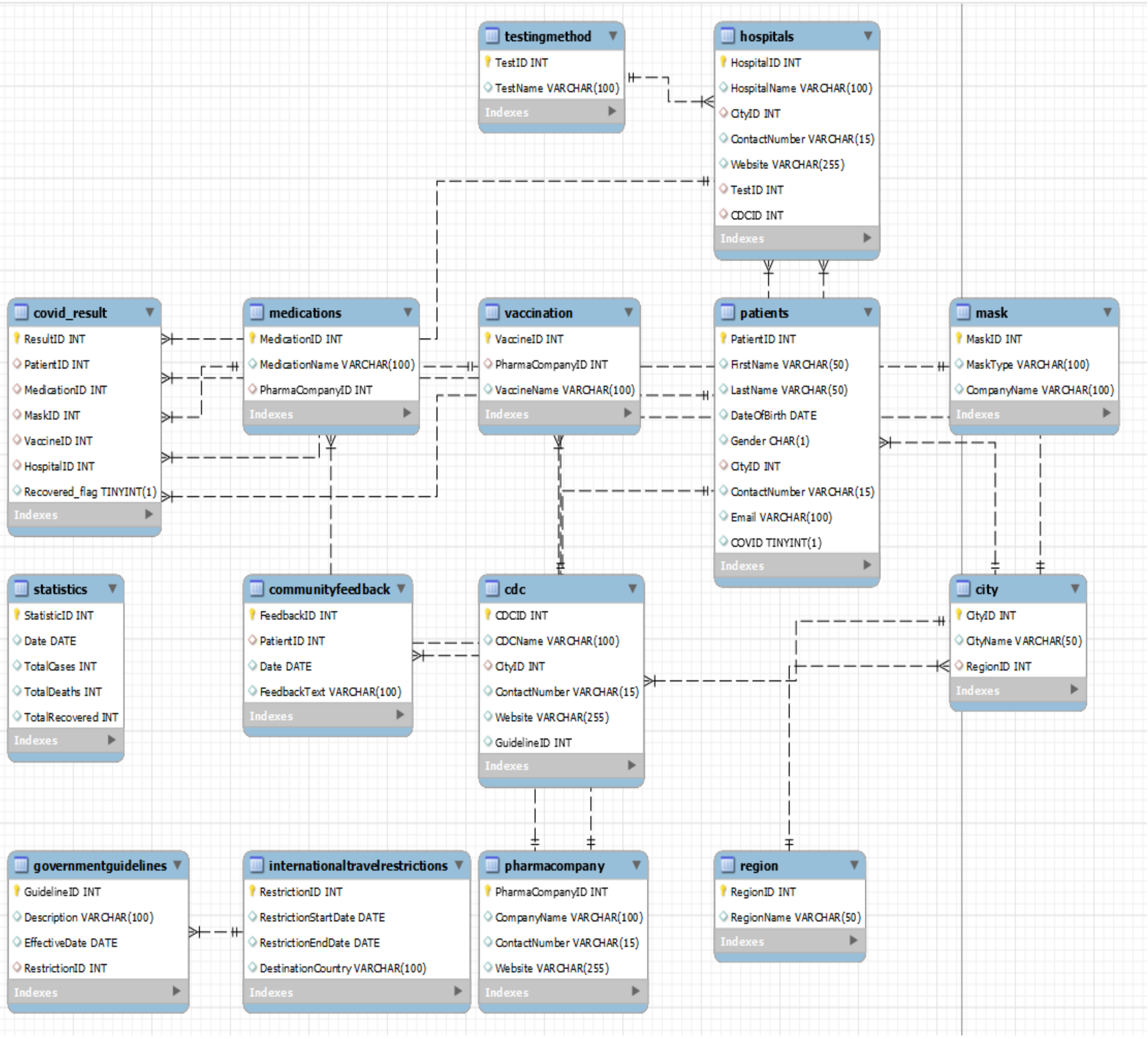


Figure 1 Final ER DIAGRAM

RELATIONAL SCHEMA

3.1. Relational Schema

The diagram highlights the relationships between each table by showing how each foreign key is connected to the primary key of the parent table. It is essentially a blueprint to show how information is correlated and retrieved in the database.



PHASE III. IMPLEMENTATION

1. Pre-Illumination

This report outlines the implementation phase of the database project, focusing on the creation of the database, table setup, data population and SQL queries. Our project utilizes the MySQL database management system. Part 2 is the modified relational schema. Part 3 is the creation of the database, including tables, all other structures as well as constraints, data type and format, Part 4 is the query scenario design and implementation.

2. Modified Relational Schema

Firstly, according to the requirement of the last phase and with the purpose of simplifying the relation model for this database, we created 4 new tables - InternationalTravelRestrictions, GovernmentGuidelines, Statistics and CommunityFeedback. These tables were added in order to enrich the database with more valuable and relevant information. It also makes the database robust and facilitates efficient data retrieval and analysis. All the tables are normalized to 3NF in order to maintain data integrity by reducing redundancy and removing duplicates, so as to enable flexible updates without affecting other tables and to simplify queries.

3. Creation Of Database With SQL Statements

- **Database Creation**

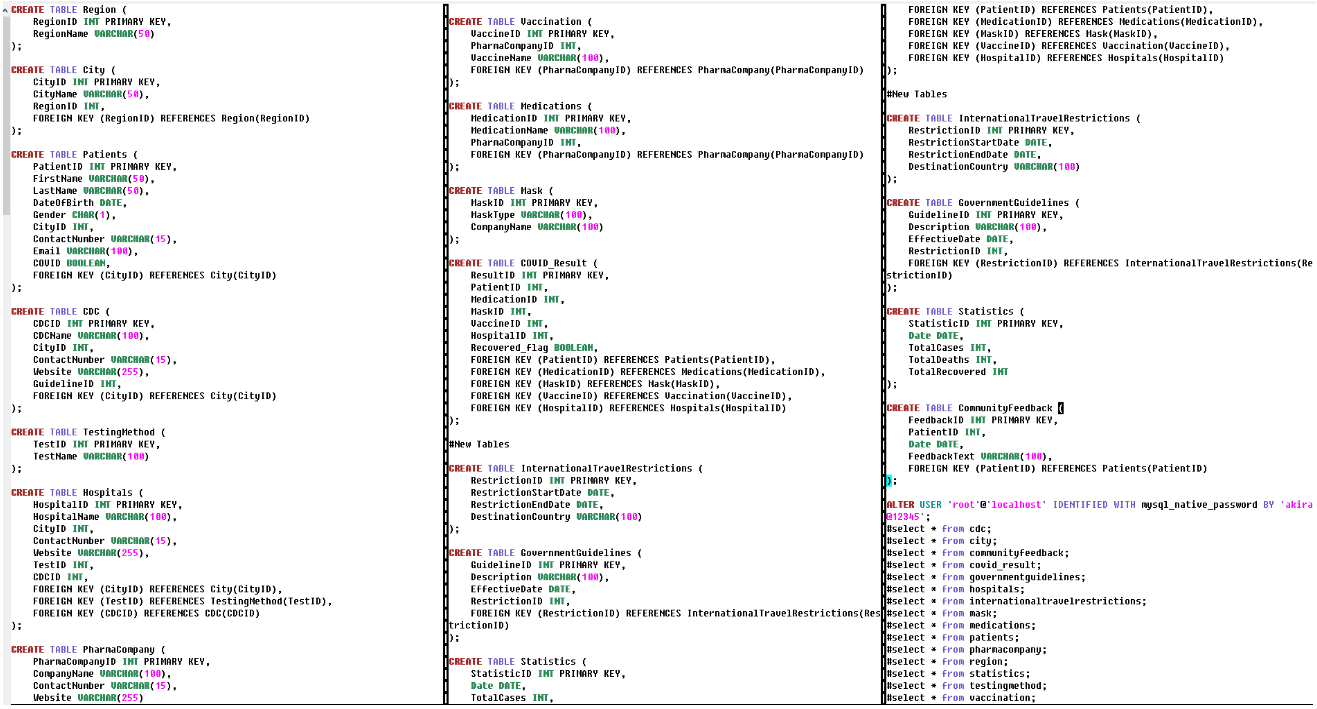
Firstly, we created Covid_Database using the following SQL statement:

```
CREATE DATABASE IF NOT EXISTS Covid_Database;

USE Covid_Database;
```

- **Table Creation**

Secondly, we created 15 Tables using the following SQL statement:

(Detailed SQL script file is uploaded and attached in submission)



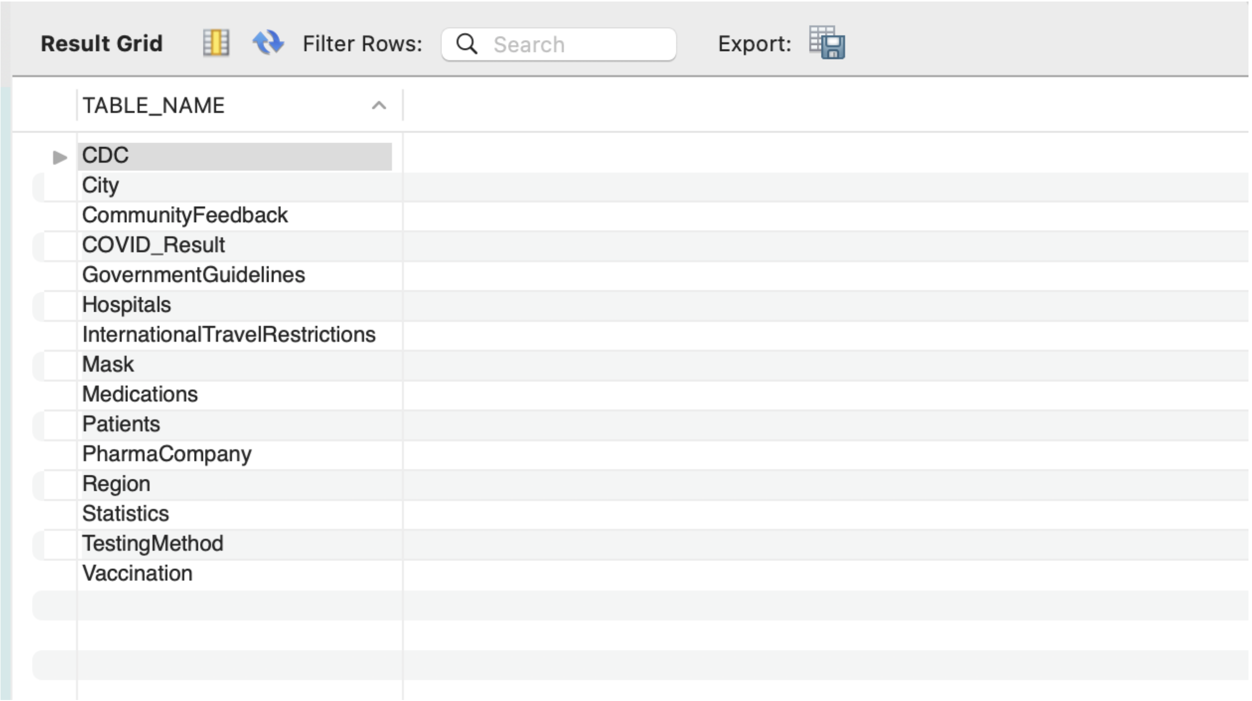
```
CREATE TABLE Region (
    RegionID INT PRIMARY KEY,
    RegionName VARCHAR(50)
);

CREATE TABLE City (
    CityID INT PRIMARY KEY,
    CityName VARCHAR(50),
    RegionID INT,
    FOREIGN KEY (RegionID) REFERENCES Region(RegionID)
);

CREATE TABLE Patients (
    PatientID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    DateOfBirth DATE,
    Gender CHAR(1),
    CityID INT,
    ContactNumber VARCHAR(15),
    Email VARCHAR(100),
    COVID BOOLEAN,
    FOREIGN KEY (CityID) REFERENCES City(CityID)
);

CREATE TABLE CDC (
    CDCID INT PRIMARY KEY,
    CDCName VARCHAR(100),
    CityID INT,
    ContactNumber VARCHAR(15),
    Website VARCHAR(255),
    GuidelineID INT,
    FOREIGN KEY (CityID) REFERENCES City(CityID)
);

CREATE TABLE TestingMethod (
    TestID INT PRIMARY KEY,
    TestName VARCHAR(100)
);

CREATE TABLE Hospitals (
    HospitalID INT PRIMARY KEY,
    HospitalName VARCHAR(100),
    CityID INT,
    ContactNumber VARCHAR(15),
    Website VARCHAR(255),
    TestID INT,
    CDCID INT,
    FOREIGN KEY (CityID) REFERENCES City(CityID),
    FOREIGN KEY (TestID) REFERENCES TestingMethod(TestID),
    FOREIGN KEY (CDCID) REFERENCES CDC(CDCID)
);

CREATE TABLE PharmaCompany (
    PharmaCompanyID INT PRIMARY KEY,
    CompanyName VARCHAR(100),
    ContactNumber VARCHAR(15),
    Website VARCHAR(255)
```

```

CREATE TABLE Vaccination (
    VaccineID INT PRIMARY KEY,
    PharmaCompanyID INT,
    VaccineName VARCHAR(100),
    FOREIGN KEY (PharmaCompanyID) REFERENCES PharmaCompany(PharmaCompanyID)
);

CREATE TABLE Medications (
    MedicationID INT PRIMARY KEY,
    MedicationName VARCHAR(100),
    PharmaCompanyID INT,
    FOREIGN KEY (PharmaCompanyID) REFERENCES PharmaCompany(PharmaCompanyID)
);

CREATE TABLE Mask (
    MaskID INT PRIMARY KEY,
    MaskType VARCHAR(100),
    CompanyName VARCHAR(100)
);

CREATE TABLE COVID_Result (
    ResultID INT PRIMARY KEY,
    PatientID INT,
    MedicationID INT,
    MaskID INT,
    VaccineID INT,
    HospitalID INT,
    Recovered_flag BOOLEAN,
    FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
    FOREIGN KEY (MedicationID) REFERENCES Medications(MedicationID),
    FOREIGN KEY (MaskID) REFERENCES Mask(MaskID),
    FOREIGN KEY (VaccineID) REFERENCES Vaccination(VaccineID),
    FOREIGN KEY (HospitalID) REFERENCES Hospitals(HospitalID)
);

#New Tables

CREATE TABLE InternationalTravelRestrictions (
    RestrictionID INT PRIMARY KEY,
    RestrictionStartDate DATE,
    RestrictionEndDate DATE,
    DestinationCountry VARCHAR(100)
);

CREATE TABLE GovernmentGuidelines (
    GuidelineID INT PRIMARY KEY,
    Description VARCHAR(100),
    EffectiveDate DATE,
    RestrictionID INT,
    FOREIGN KEY (RestrictionID) REFERENCES InternationalTravelRestrictions(Res
trictionID)
);

CREATE TABLE Statistics (
    StatisticID INT PRIMARY KEY,
    Date DATE,
    TotalCases INT,
```

```
    FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
    FOREIGN KEY (MedicationID) REFERENCES Medications(MedicationID),
    FOREIGN KEY (MaskID) REFERENCES Mask(MaskID),
    FOREIGN KEY (VaccineID) REFERENCES Vaccination(VaccineID),
    FOREIGN KEY (HospitalID) REFERENCES Hospitals(HospitalID)
);

#New Tables

CREATE TABLE InternationalTravelRestrictions (
    RestrictionID INT PRIMARY KEY,
    RestrictionStartDate DATE,
    RestrictionEndDate DATE,
    DestinationCountry VARCHAR(100)
);

CREATE TABLE GovernmentGuidelines (
    GuidelineID INT PRIMARY KEY,
    Description VARCHAR(100),
    EffectiveDate DATE,
    RestrictionID INT,
    FOREIGN KEY (RestrictionID) REFERENCES InternationalTravelRestrictions(Re
strictionID)
);

CREATE TABLE Statistics (
    StatisticID INT PRIMARY KEY,
    Date DATE,
    TotalCases INT,
    TotalDeaths INT,
    TotalRecovered INT
);

CREATE TABLE CommunityFeedback (
    FeedbackID INT PRIMARY KEY,
    PatientID INT,
    Date DATE,
    FeedbackText VARCHAR(100),
    FOREIGN KEY (PatientID) REFERENCES Patients(PatientID)
);

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'akira
@12345';
#select * from cdc;
#select * from city;
#select * from communityfeedback;
#select * from covid_result;
#select * from governmentguidelines;
#select * from hospitals;
#select * from internationaltravelrestrictions;
#select * from mask;
#select * from medications;
#select * from patients;
#select * from pharmacompany;
#select * from region;
#select * from statistics;
#select * from testingmethod;
#select * from vaccination;
```

- **Result**

15 empty tables were created with specified data types and keys.

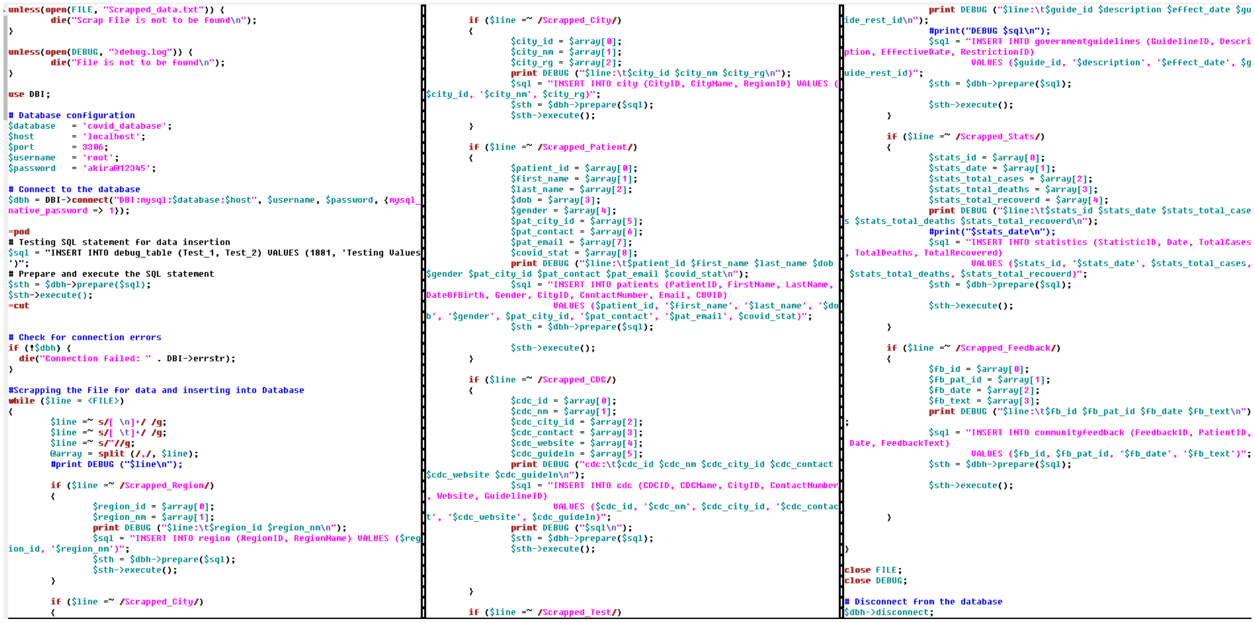


TABLE_NAME
CDC
City
CommunityFeedback
COVID_Result
GovernmentGuidelines
Hospitals
InternationalTravelRestrictions
Mask
Medications
Patients
PharmaCompany
Region
Statistics
TestingMethod
Vaccination

- **A Database State**

To ensure the database is populated for testing and development purposes, sample data was inserted into each table. In order to scrape the data covid_data_scrapping_tool.pl was created and used (attached). The following records were added to each table, maintaining data consistency and validity.

(Detailed Perl script file is uploaded and attached in submission)



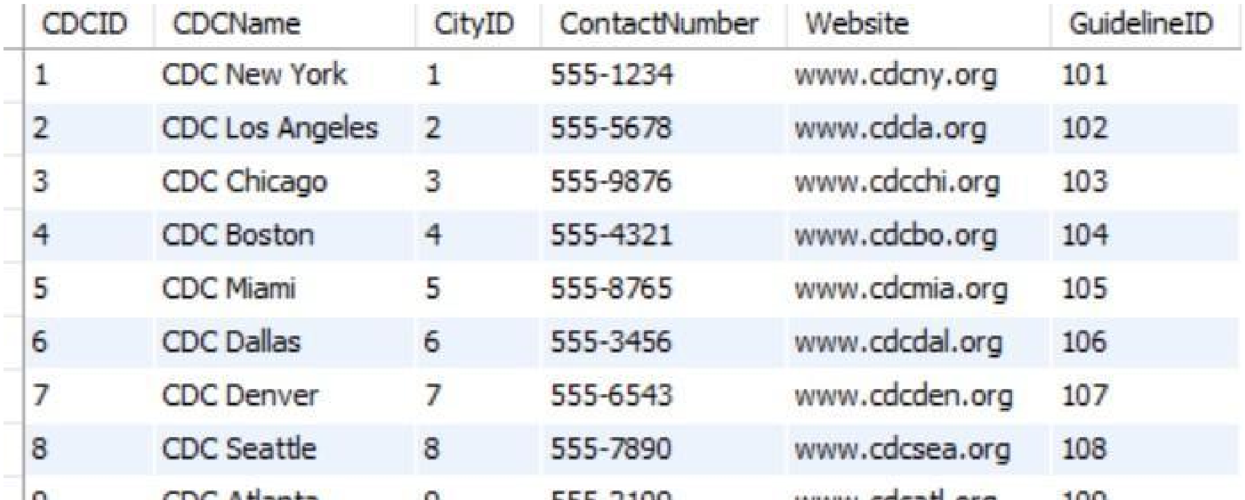
```perl
unless(open(FILE, "Scrapped_data.txt")) {
        die("Scrap File is not to be found\n");
}

unless(open(DEBUG, ">debug.log")) {
        die("File is not to be found\n");
}

use DBI;

# Database configuration
$database   = 'covid_database';
$host       = 'localhost';
$port       = 3306;
$username   = 'root';
$password   = 'akira@12345';

# Connect to the database
$dbh = DBI->connect("DBI:mysql:$database:$host", $username, $password, {mysql_
native_password => 1});

=pod
# Testing SQL statement for data insertion
$sql = "INSERT INTO debug_table (Test_1, Test_2) VALUES (1881, 'Testing Values
')";
# Prepare and execute the SQL statement
$sth = $dbh->prepare($sql);
$sth->execute();
=cut


# Check for connection errors
if (!$dbh) {
  die("Connection failed: " . DBI->errstr);
}

#Scrapping the File for data and inserting into Database
while ($line = <FILE>)
{
        $line =~ s/[ \n]+/ /g;
        $line =~ s/[ \t]+/ /g;
        $line =~ s/"//g;
        @array = split (/,/, $line);
        #print DEBUG ("$line\n");

        if ($line =~ /Scrapped_Region/)
        {
                $region_id = $array[0];
                $region_nm = $array[1];
                print DEBUG ("$line:\t$region_id $region_nm\n");
                $sql = "INSERT INTO region (RegionID, RegionName) VALUES ($reg
ion_id, '$region_nm')";
                $sth = $dbh->prepare($sql);
                $sth->execute();
        }

        if ($line =~ /Scrapped_City/)
        {
```

```perl
        if ($line =~ /Scrapped_City/)
        {
                $city_id = $array[0];
                $city_nm = $array[1];
                $city_rg = $array[2];
                print DEBUG ("$line:\t$city_id $city_nm $city_rg\n");
                $sql = "INSERT INTO city (CityID, CityName, RegionID) VALUES (
$city_id, '$city_nm', $city_rg)";
                $sth = $dbh->prepare($sql);
                $sth->execute();
        }

        if ($line =~ /Scrapped_Patient/)
        {
                $patient_id = $array[0];
                $first_name = $array[1];
                $last_name = $array[2];
                $dob = $array[3];
                $gender = $array[4];
                $pat_city_id = $array[5];
                $pat_contact = $array[6];
                $pat_email = $array[7];
                $covid_stat = $array[8];
                print DEBUG ("$line:\t$patient_id $first_name $last_name $dob
$gender $pat_city_id $pat_contact $pat_email $covid_stat\n");
                $sql = "INSERT INTO patients (PatientID, FirstName, LastName,
DateOfBirth, Gender, CityID, ContactNumber, Email, COVID)
                        VALUES ($patient_id, '$first_name', '$last_name', '$do
b', '$gender', $pat_city_id, '$pat_contact', '$pat_email', $covid_stat)";
                $sth = $dbh->prepare($sql);

                $sth->execute();
        }

        if ($line =~ /Scrapped_CDC/)
        {
                $cdc_id = $array[0];
                $cdc_nm = $array[1];
                $cdc_city_id = $array[2];
                $cdc_contact = $array[3];
                $cdc_website = $array[4];
                $cdc_guideln = $array[5];
                print DEBUG ("cdc:\t$cdc_id $cdc_nm $cdc_city_id $cdc_contact
$cdc_website $cdc_guideln\n");
                $sql = "INSERT INTO cdc (CDCID, CDCName, CityID, ContactNumber
, Website, GuidelineID)
                        VALUES ($cdc_id, '$cdc_nm', $cdc_city_id, '$cdc_contac
t', '$cdc_website', $cdc_guideln)";
                print DEBUG ("$sql\n");
                $sth = $dbh->prepare($sql);
                $sth->execute();



        }

        if ($line =~ /Scrapped_Test/)
```

```perl
                print DEBUG ("$line:\t$guide_id $description $effect_date $gu
ide_rest_id\n");
                #print("DEBUG $sql\n");
                $sql = "INSERT INTO governmentguidelines (GuidelineID, Descri
ption, EffectiveDate, RestrictionID)
                        VALUES ($guide_id, '$description', '$effect_date', $g
uide_rest_id)";
                $sth = $dbh->prepare($sql);

                $sth->execute();
        }

        if ($line =~ /Scrapped_Stats/)
        {
                $stats_id = $array[0];
                $stats_date = $array[1];
                $stats_total_cases = $array[2];
                $stats_total_deaths = $array[3];
                $stats_total_recoverd = $array[4];
                print DEBUG ("$line:\t$stats_id $stats_date $stats_total_case
s $stats_total_deaths $stats_total_recoverd\n");
                #print("$stats_date\n");
                $sql = "INSERT INTO statistics (StatisticID, Date, TotalCases
, TotalDeaths, TotalRecovered)
                        VALUES ($stats_id, '$stats_date', $stats_total_cases,
 $stats_total_deaths, $stats_total_recoverd)";
                $sth = $dbh->prepare($sql);

                $sth->execute();

        }

        if ($line =~ /Scrapped_Feedback/)
        {
                $fb_id = $array[0];
                $fb_pat_id = $array[1];
                $fb_date = $array[2];
                $fb_text = $array[3];
                print DEBUG ("$line:\t$fb_id $fb_pat_id $fb_date $fb_text\n")
;
                $sql = "INSERT INTO communityfeedback (FeedbackID, PatientID,
 Date, FeedbackText)
                        VALUES ($fb_id, $fb_pat_id, '$fb_date', '$fb_text')";
                $sth = $dbh->prepare($sql);

                $sth->execute();

        }

}

close FILE;
close DEBUG;

# Disconnect from the database
$dbh->disconnect;
```

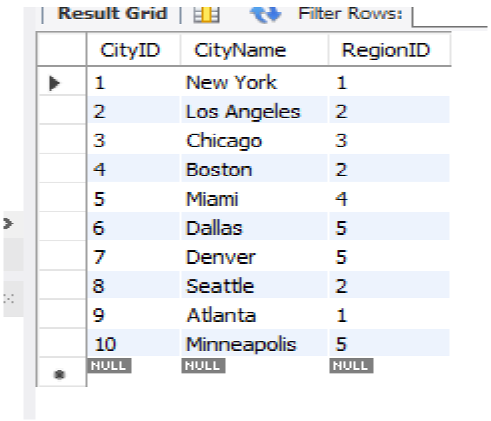
Tables After Data Population

☐ Table 1 - CDC



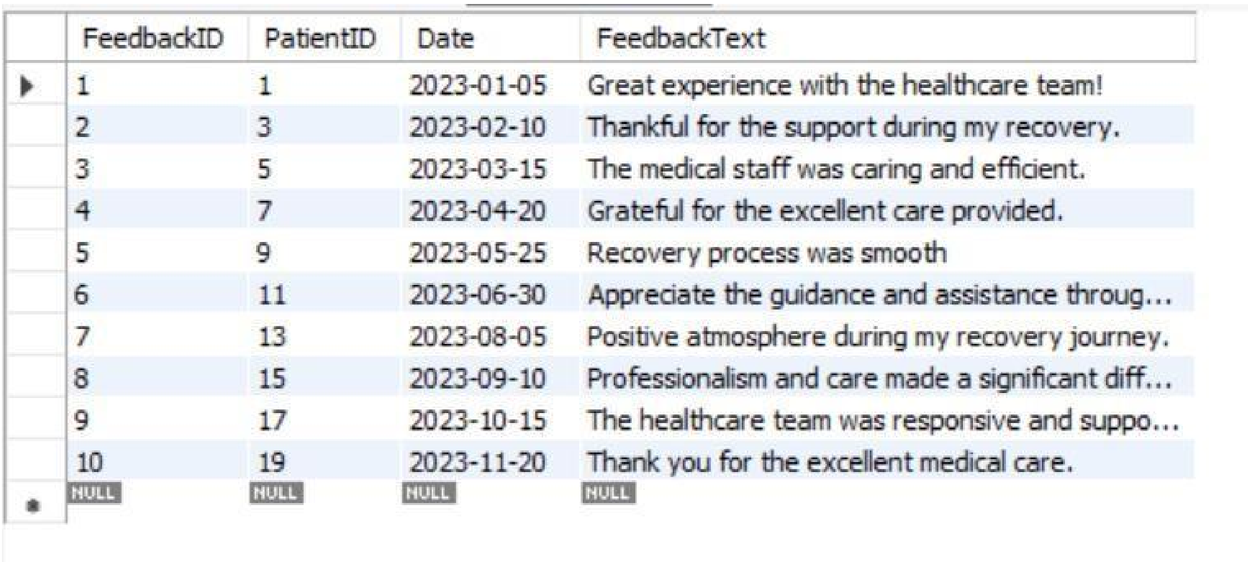
CDCID	CDCName	CityID	ContactNumber	Website	GuidelineID
1	CDC New York	1	555-1234	www.cdcny.org	101
2	CDC Los Angeles	2	555-5678	www.cdcla.org	102
3	CDC Chicago	3	555-9876	www.cdcchi.org	103
4	CDC Boston	4	555-4321	www.cdcbo.org	104
5	CDC Miami	5	555-8765	www.cdcmia.org	105
6	CDC Dallas	6	555-3456	www.cdcdal.org	106
7	CDC Denver	7	555-6543	www.cdcden.org	107
8	CDC Seattle	8	555-7890	www.cdcsea.org	108
9	CDC Atlanta	9	555-2109	www.cdcatl.org	109

- ☐ Table 2 - City



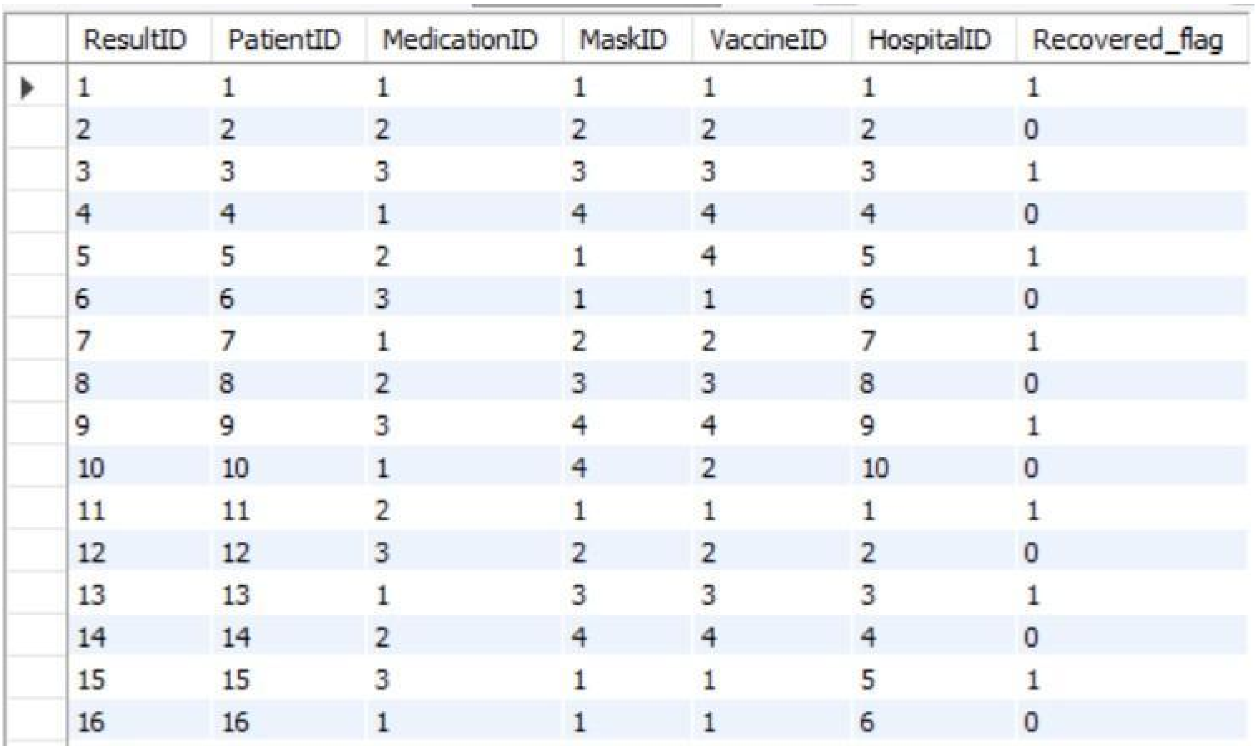
CityID	CityName	RegionID
1	New York	1
2	Los Angeles	2
3	Chicago	3
4	Boston	2
5	Miami	4
6	Dallas	5
7	Denver	5
8	Seattle	2
9	Atlanta	1
10	Minneapolis	5
NULL	NULL	NULL

- ☐ Table 3 - Community Feedback



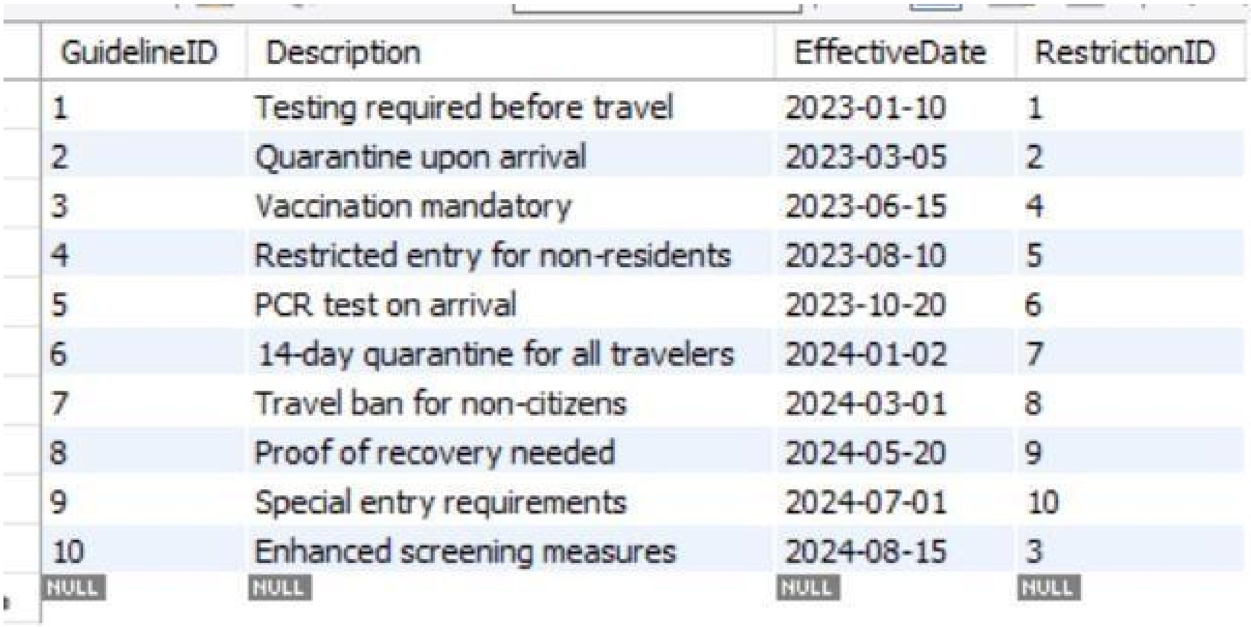
FeedbackID	PatientID	Date	FeedbackText
1	1	2023-01-05	Great experience with the healthcare team!
2	3	2023-02-10	Thankful for the support during my recovery.
3	5	2023-03-15	The medical staff was caring and efficient.
4	7	2023-04-20	Grateful for the excellent care provided.
5	9	2023-05-25	Recovery process was smooth
6	11	2023-06-30	Appreciate the guidance and assistance throug...
7	13	2023-08-05	Positive atmosphere during my recovery journey.
8	15	2023-09-10	Professionalism and care made a significant diff...
9	17	2023-10-15	The healthcare team was responsive and suppo...
10	19	2023-11-20	Thank you for the excellent medical care.
NULL	NULL	NULL	NULL

☐ Table 4 - COVID_Result



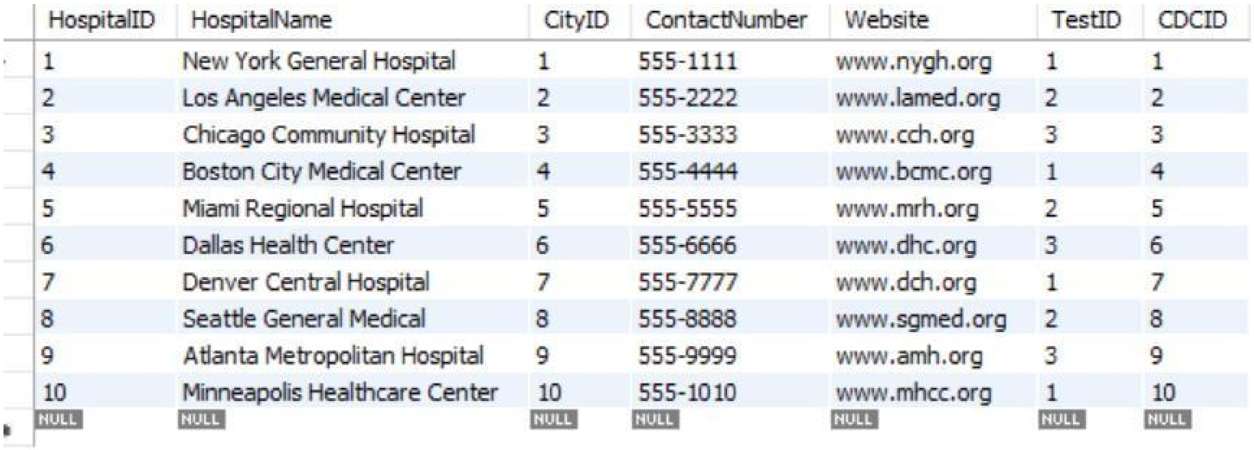
ResultID	PatientID	MedicationID	MaskID	VaccineID	HospitalID	Recovered_flag
1	1	1	1	1	1	1
2	2	2	2	2	2	0
3	3	3	3	3	3	1
4	4	1	4	4	4	0
5	5	2	1	4	5	1
6	6	3	1	1	6	0
7	7	1	2	2	7	1
8	8	2	3	3	8	0
9	9	3	4	4	9	1
10	10	1	4	2	10	0
11	11	2	1	1	1	1
12	12	3	2	2	2	0
13	13	1	3	3	3	1
14	14	2	4	4	4	0
15	15	3	1	1	5	1
16	16	1	1	1	6	0

☐ Table 5 : GovernmentGuidelines



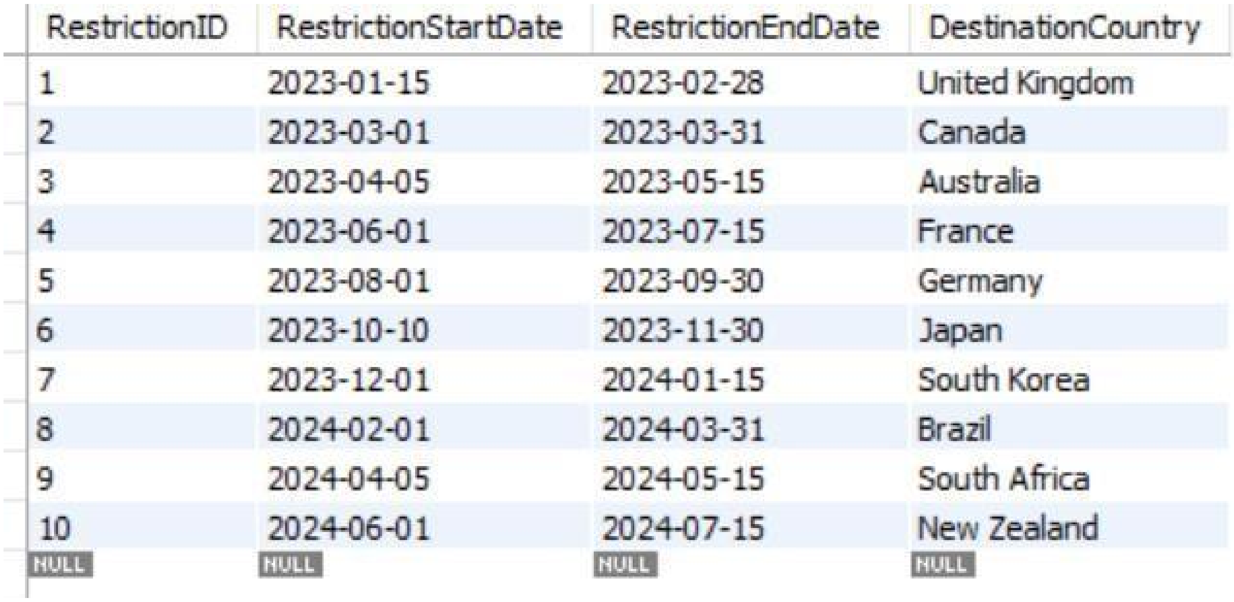
GuidelineID	Description	EffectiveDate	RestrictionID
1	Testing required before travel	2023-01-10	1
2	Quarantine upon arrival	2023-03-05	2
3	Vaccination mandatory	2023-06-15	4
4	Restricted entry for non-residents	2023-08-10	5
5	PCR test on arrival	2023-10-20	6
6	14-day quarantine for all travelers	2024-01-02	7
7	Travel ban for non-citizens	2024-03-01	8
8	Proof of recovery needed	2024-05-20	9
9	Special entry requirements	2024-07-01	10
10	Enhanced screening measures	2024-08-15	3
NULL	NULL	NULL	NULL

□ Table 6 : Hospitals



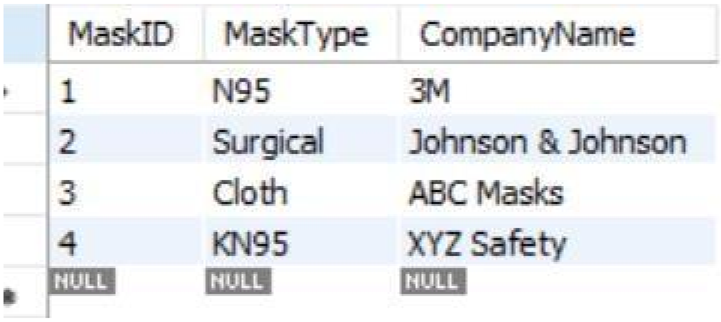
HospitalID	HospitalName	CityID	ContactNumber	Website	TestID	CDCID
1	New York General Hospital	1	555-1111	www.nygh.org	1	1
2	Los Angeles Medical Center	2	555-2222	www.lamed.org	2	2
3	Chicago Community Hospital	3	555-3333	www.cch.org	3	3
4	Boston City Medical Center	4	555-4444	www.bcmc.org	1	4
5	Miami Regional Hospital	5	555-5555	www.mrh.org	2	5
6	Dallas Health Center	6	555-6666	www.dhc.org	3	6
7	Denver Central Hospital	7	555-7777	www.dch.org	1	7
8	Seattle General Medical	8	555-8888	www.sgmed.org	2	8
9	Atlanta Metropolitan Hospital	9	555-9999	www.amh.org	3	9
10	Minneapolis Healthcare Center	10	555-1010	www.mhcc.org	1	10
NULL	NULL	NULL	NULL	NULL	NULL	NULL

□ Table 7 : InternationalTravelRestrictions



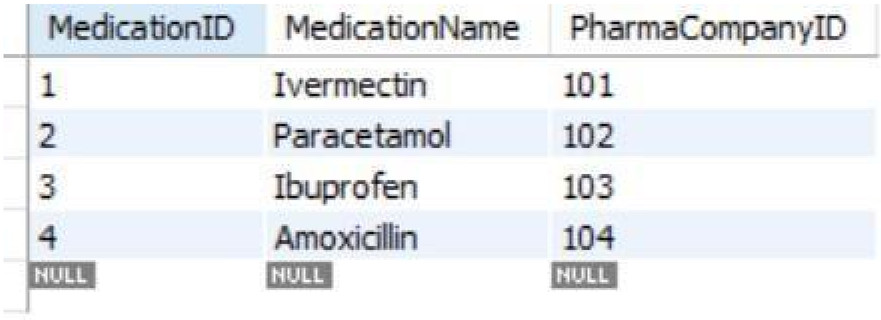
RestrictionID	RestrictionStartDate	RestrictionEndDate	DestinationCountry
1	2023-01-15	2023-02-28	United Kingdom
2	2023-03-01	2023-03-31	Canada
3	2023-04-05	2023-05-15	Australia
4	2023-06-01	2023-07-15	France
5	2023-08-01	2023-09-30	Germany
6	2023-10-10	2023-11-30	Japan
7	2023-12-01	2024-01-15	South Korea
8	2024-02-01	2024-03-31	Brazil
9	2024-04-05	2024-05-15	South Africa
10	2024-06-01	2024-07-15	New Zealand
NULL	NULL	NULL	NULL

□ Table 8 : Mask



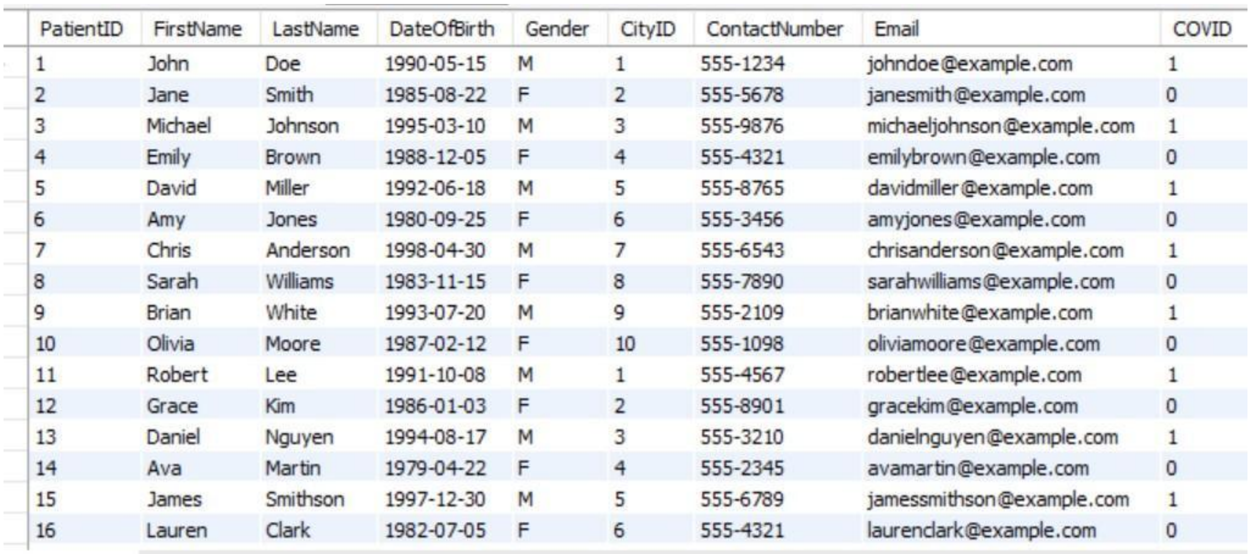
MaskID	MaskType	CompanyName
1	N95	3M
2	Surgical	Johnson & Johnson
3	Cloth	ABC Masks
4	KN95	XYZ Safety
NULL	NULL	NULL

□ Table 9 : Medications



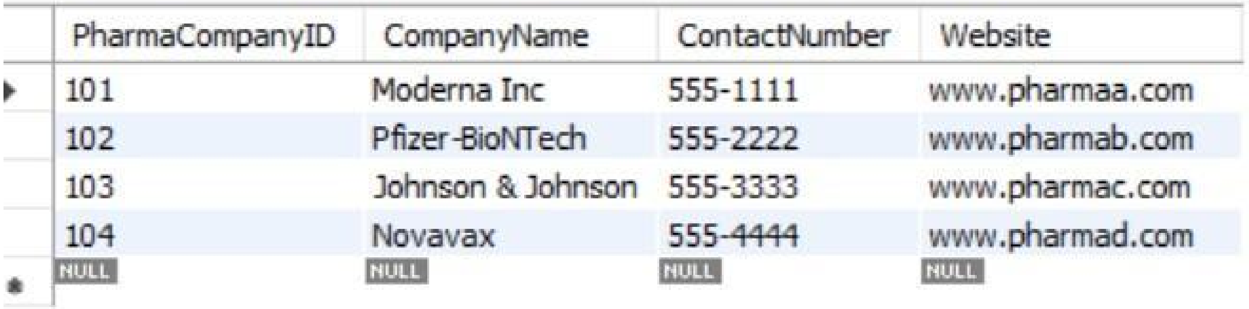
MedicationID	MedicationName	PharmaCompanyID
1	Ivermectin	101
2	Paracetamol	102
3	Ibuprofen	103
4	Amoxicillin	104
NULL	NULL	NULL

□ Table 10 : Patients



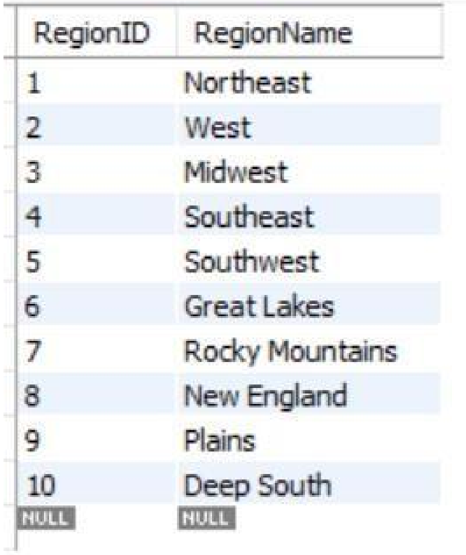
PatientID	FirstName	LastName	DateOfBirth	Gender	CityID	ContactNumber	Email	COVID
1	John	Doe	1990-05-15	M	1	555-1234	johndoe@example.com	1
2	Jane	Smith	1985-08-22	F	2	555-5678	janesmith@example.com	0
3	Michael	Johnson	1995-03-10	M	3	555-9876	michaeljohnson@example.com	1
4	Emily	Brown	1988-12-05	F	4	555-4321	emilybrown@example.com	0
5	David	Miller	1992-06-18	M	5	555-8765	davidmiller@example.com	1
6	Amy	Jones	1980-09-25	F	6	555-3456	amyjones@example.com	0
7	Chris	Anderson	1998-04-30	M	7	555-6543	chrisanderson@example.com	1
8	Sarah	Williams	1983-11-15	F	8	555-7890	sarahwilliams@example.com	0
9	Brian	White	1993-07-20	M	9	555-2109	brianwhite@example.com	1
10	Olivia	Moore	1987-02-12	F	10	555-1098	oliviamoore@example.com	0
11	Robert	Lee	1991-10-08	M	1	555-4567	robertlee@example.com	1
12	Grace	Kim	1986-01-03	F	2	555-8901	gracekim@example.com	0
13	Daniel	Nguyen	1994-08-17	M	3	555-3210	danielnguyen@example.com	1
14	Ava	Martin	1979-04-22	F	4	555-2345	avamartin@example.com	0
15	James	Smithson	1997-12-30	M	5	555-6789	jamessmithson@example.com	1
16	Lauren	Clark	1982-07-05	F	6	555-4321	laurenclark@example.com	0

□ Table 11 : PharmaCompany



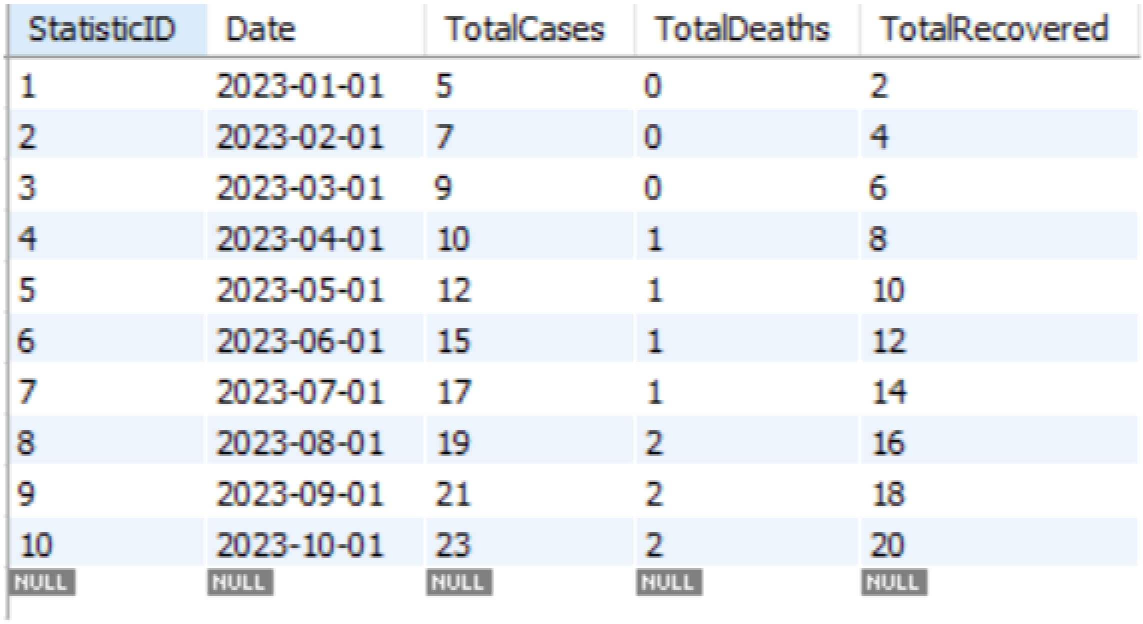
PharmaCompanyID	CompanyName	ContactNumber	Website
101	Moderna Inc	555-1111	www.pharmaa.com
102	Pfizer-BioNTech	555-2222	www.pharmab.com
103	Johnson & Johnson	555-3333	www.pharmac.com
104	Novavax	555-4444	www.pharmad.com
NULL	NULL	NULL	NULL

□ Table 12 : Region



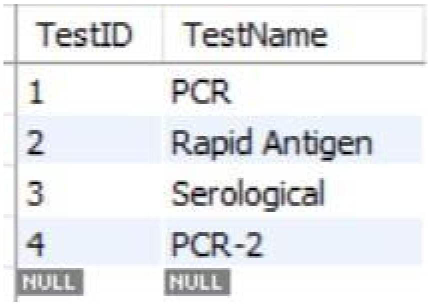
RegionID	RegionName
1	Northeast
2	West
3	Midwest
4	Southeast
5	Southwest
6	Great Lakes
7	Rocky Mountains
8	New England
9	Plains
10	Deep South
NULL	NULL

- □ Table 13 : Statistics



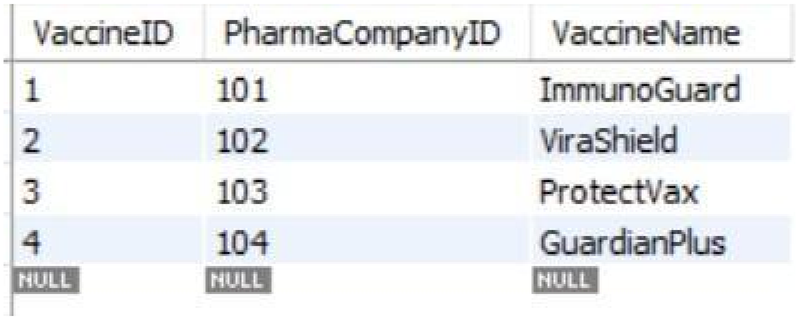
StatisticID	Date	TotalCases	TotalDeaths	TotalRecovered
1	2023-01-01	5	0	2
2	2023-02-01	7	0	4
3	2023-03-01	9	0	6
4	2023-04-01	10	1	8
5	2023-05-01	12	1	10
6	2023-06-01	15	1	12
7	2023-07-01	17	1	14
8	2023-08-01	19	2	16
9	2023-09-01	21	2	18
10	2023-10-01	23	2	20
NULL	NULL	NULL	NULL	NULL

- □ Table 14 : TestingMethod



TestID	TestName
1	PCR
2	Rapid Antigen
3	Serological
4	PCR-2
NULL	NULL

- □ Table 15 :Vaccination



VaccineID	PharmaCompanyID	VaccineName
1	101	ImmunoGuard
2	102	ViraShield
3	103	ProtectVax
4	104	GuardianPlus
NULL	NULL	NULL

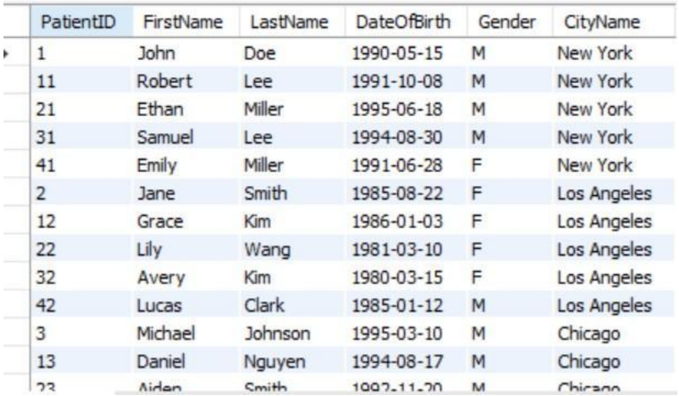
QUERY DESIGN SCENARIOS

Query 01: Retrieve Patient Information with City Names.

Description: Retrieves patient information along with their city names. It combines details such as PatientID, FirstName, LastName, DateOfBirth, Gender, and CityName from the Patients and City tables.

```
SELECT P.PatientID, P.FirstName, P.LastName,

P.DateOfBirth, P.Gender, C.CityName FROM Patients

P

JOIN City C ON P.CityID = C.CityID;
```

Result Of Query 01:



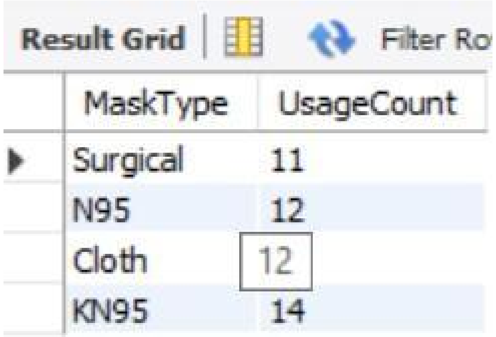
PatientID	FirstName	LastName	DateOfBirth	Gender	CityName
1	John	Doe	1990-05-15	M	New York
11	Robert	Lee	1991-10-08	M	New York
21	Ethan	Miller	1995-06-18	M	New York
31	Samuel	Lee	1994-08-30	M	New York
41	Emily	Miller	1991-06-28	F	New York
2	Jane	Smith	1985-08-22	F	Los Angeles
12	Grace	Kim	1986-01-03	F	Los Angeles
22	Lily	Wang	1981-03-10	F	Los Angeles
32	Avery	Kim	1980-03-15	F	Los Angeles
42	Lucas	Clark	1985-01-12	M	Los Angeles
3	Michael	Johnson	1995-03-10	M	Chicago
13	Daniel	Nguyen	1994-08-17	M	Chicago
23	Aiden	Smith	1992-11-20	M	Chicago

Query 02: Find the mask types count among COVID patients.

Description: Counts the usage of different mask types among COVID patients. It joins the Mask and COVID_Result tables on MaskID and calculates the count of each mask type.

```
 SELECT MaskType, COUNT(*) AS UsageCount FROM

Mask JOIN COVID_Result ON Mask.MaskID =

COVID_Result.MaskID GROUP BY MaskType ORDER BY

UsageCount ;
```

Result Of Query 02:



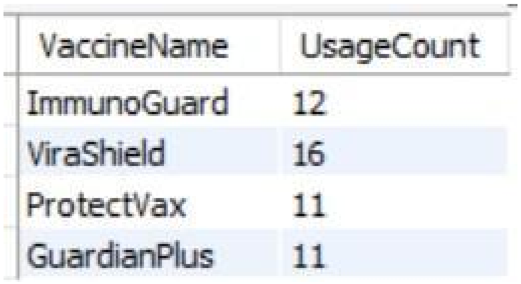
MaskType	UsageCount
Surgical	11
N95	12
Cloth	12
KN95	14

Query 03: Show count of different vaccine types used.

Description: Counts the usage of different vaccine types among patients. It joins the Vaccination and COVID_Result tables on VaccineID and calculates the count of each vaccine type.

```
SELECT VaccineName, COUNT(*) AS UsageCount

FROM Vaccination JOIN COVID_Result ON

Vaccination.VaccineID =

COVID_Result.VaccineID GROUP BY

VaccineName;
```

Result Of Query 03:



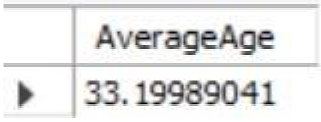
VaccineName	UsageCount
ImmunoGuard	12
ViraShield	16
ProtectVax	11
GuardianPlus	11

Query 04: Get average age of COVID-positive patients.

Description: Calculates the average age of patients who tested positive for COVID-19. It uses the DATEDIFF function to calculate age based on the DateOfBirth column in the Patients table where the COVID column is TRUE.

```
SELECT AVG(DATEDIFF(CURDATE(), DateOfBirth)/365) AS

AverageAge FROM Patients WHERE COVID = TRUE;
```

Result Of Query 04:



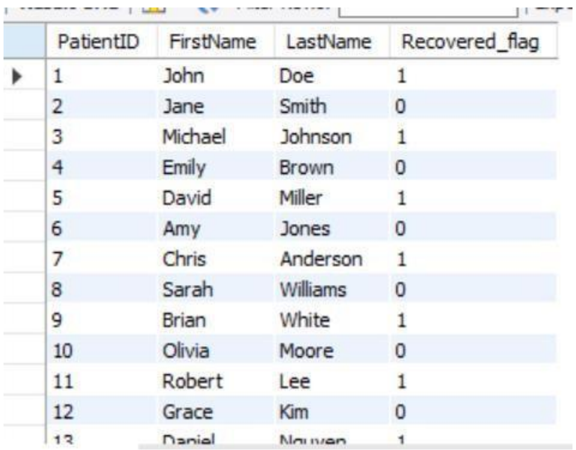
	AverageAge
▶	33.19989041

Query 05: List Patients and Their COVID Results.

Description: Lists patients along with their COVID results. It includes patient details such as PatientID, FirstName, LastName, and the recovery status (Recovered_flag) from the COVID_Result table.

```
SELECT P.PatientID, P.FirstName, P.LastName, CR.Recovered_flag
 FROM Patients P

 JOIN COVID_Result CR ON P.PatientID = CR.PatientID;
```

Result Of Query 05:



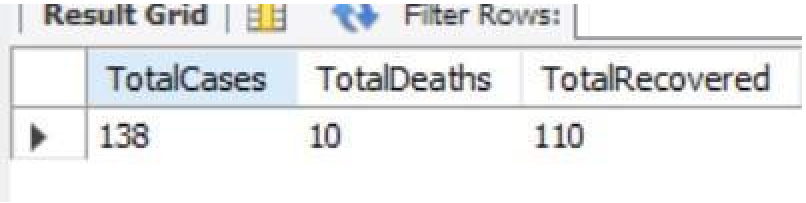
	PatientID	FirstName	LastName	Recovered_flag
▶	1	John	Doe	1
	2	Jane	Smith	0
	3	Michael	Johnson	1
	4	Emily	Brown	0
	5	David	Miller	1
	6	Amy	Jones	0
	7	Chris	Anderson	1
	8	Sarah	Williams	0
	9	Brian	White	1
	10	Olivia	Moore	0
	11	Robert	Lee	1
	12	Grace	Kim	0
	13	Daniel	Nguyen	1

Query 06: Calculate Total Cases, Deaths, and Recovered.

Description: Calculates the total number of COVID-19 cases, deaths, and recoveries. It uses the SUM function on the TotalCases, TotalDeaths, and TotalRecovered columns from the Statistics table.

```
SELECT SUM(TotalCases) AS TotalCases, SUM(TotalDeaths) AS
TotalDeaths, SUM(TotalRecovered) AS TotalRecovered

FROM Statistics;
```

Result Of Query 06:



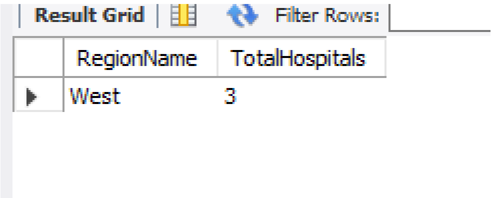
TotalCases	TotalDeaths	TotalRecovered
138	10	110

Query 07: Identify Regions with the Most Hospitals.

Description: Identifies regions with the highest number of hospitals. It counts the total hospitals in each region by joining the Region, City, and Hospitals tables.

```
SELECT                    R.RegionName,

COUNT(H.HospitalID)  AS  TotalHospitals

FROM Region R

LEFT   JOIN   City   C   ON

R.RegionID   =   C.RegionID

LEFT  JOIN  Hospitals  H  ON

C.CityID  =  H.CityID  GROUP

BY R.RegionName

ORDER BY

TotalHospitals

DESC limit 1;
```

Result Of Query 07:



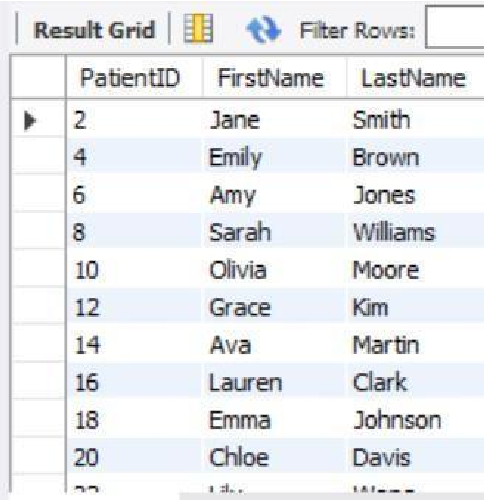
RegionName	TotalHospitals
West	3

Query 08: List Patients who have not Recovered.

Description: Lists patients who have not recovered. It includes patient details such as PatientID, FirstName, and LastName from the Patients table, filtered by the Recovered_flag in the COVID_Result table.

```
SELECT P.PatientID,

P.FirstName, P.LastName FROM

Patients P

JOIN COVID_Result CR ON

P.PatientID = CR.PatientID WHERE

CR.Recovered_flag = false;
```

Result Of Query 08:



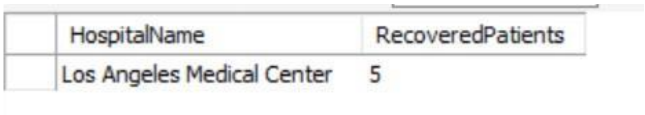
PatientID	FirstName	LastName
2	Jane	Smith
4	Emily	Brown
6	Amy	Jones
8	Sarah	Williams
10	Olivia	Moore
12	Grace	Kim
14	Ava	Martin
16	Lauren	Clark
18	Emma	Johnson
20	Chloe	Davis

Query 09: Retrieve Hospitals with the Highest Number of Recovered Patients.

Description: Retrieves hospitals with the highest number of recovered patients. It counts the recovered patients for each hospital and orders the results in descending order by the count.

```
SELECT HospitalName, COUNT(cr.PatientID)

AS RecoveredPatients FROM Hospitals

LEFT JOIN COVID_Result cr ON Hospitals.HospitalID =
cr.HospitalID
WHERE

cr.Recovered_flag

= 'Y' GROUP BY

HospitalName

ORDER BY

RecoveredPatients

DESC LIMIT 1;
```

Result Of Query 09:



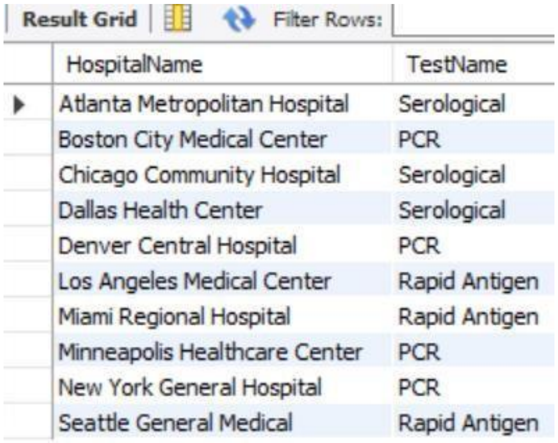
	HospitalName	RecoveredPatients
	Los Angeles Medical Center	5

Query 10: Retrieve Hospitals with Testing Methods Ordered by Hospital Name.

Description: Retrieves hospitals offering specific testing methods and orders the results by hospital name. It includes hospital names and corresponding testing methods from the Hospitals and TestingMethod tables.

```
SELECT h.HospitalName,

tm.TestName FROM

Hospitals h

JOIN TestingMethod tm ON

h.TestID = tm.TestID ORDER BY

h.HospitalName;
```

Result Of Query 10:



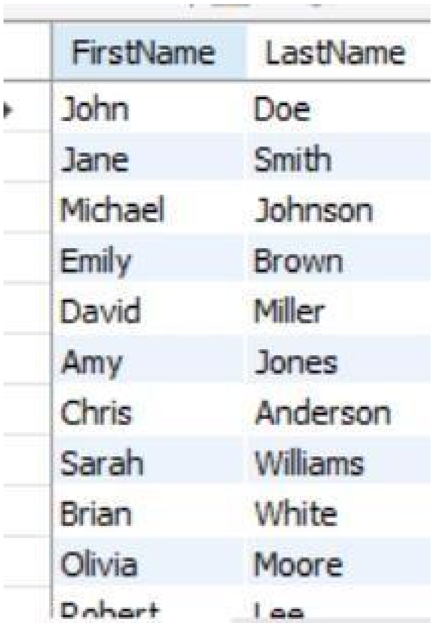
HospitalName	TestName
Atlanta Metropolitan Hospital	Serological
Boston City Medical Center	PCR
Chicago Community Hospital	Serological
Dallas Health Center	Serological
Denver Central Hospital	PCR
Los Angeles Medical Center	Rapid Antigen
Miami Regional Hospital	Rapid Antigen
Minneapolis Healthcare Center	PCR
New York General Hospital	PCR
Seattle General Medical	Rapid Antigen

Query 11: Retrieve Patients Who Have Received a Vaccine.

Description: Retrieves a list of patients who have received any vaccine. It includes patient details such as FirstName and LastName from the Patients table, filtered by the existence of a vaccine record in the COVID_Result table.

```
SELECT FirstName,

LastName FROM

Patients

WHERE

  PatientID IN
  ( SELECT
  PatientID
  FROM
  COVID_Result
  WHERE VaccineID IS NOT NULL
);
```

Result Of Query 11:



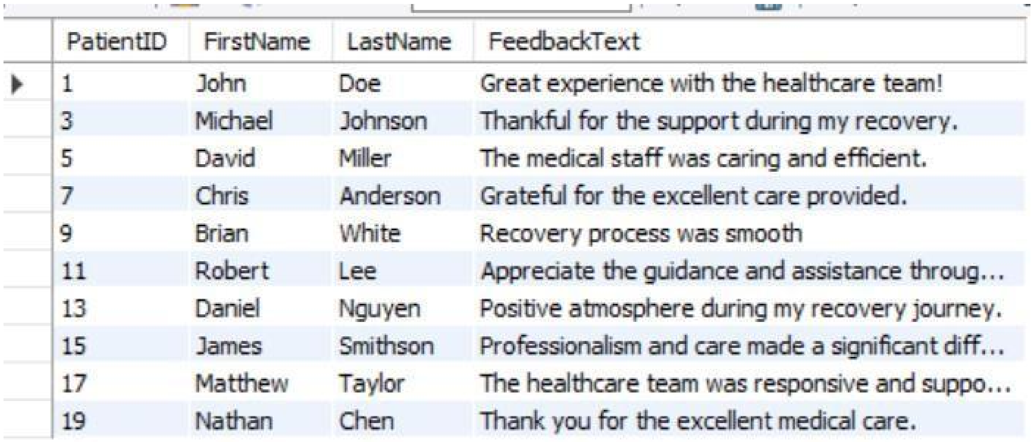
FirstName	LastName
John	Doe
Jane	Smith
Michael	Johnson
Emily	Brown
David	Miller
Amy	Jones
Chris	Anderson
Sarah	Williams
Brian	White
Olivia	Moore
Robert	Lee

Query 12: Find Patients who have Provided Feedback.

Description: This query retrieves information about patients who have provided feedback. It includes patient details such as PatientID, FirstName, LastName, and the feedback text from the CommunityFeedback table.

```
SELECT P.PatientID, P.FirstName,
P.LastName, CF.FeedbackText FROM
Patients P
JOIN CommunityFeedback CF ON P.PatientID = CF.PatientID;
```

Result Of Query 12:



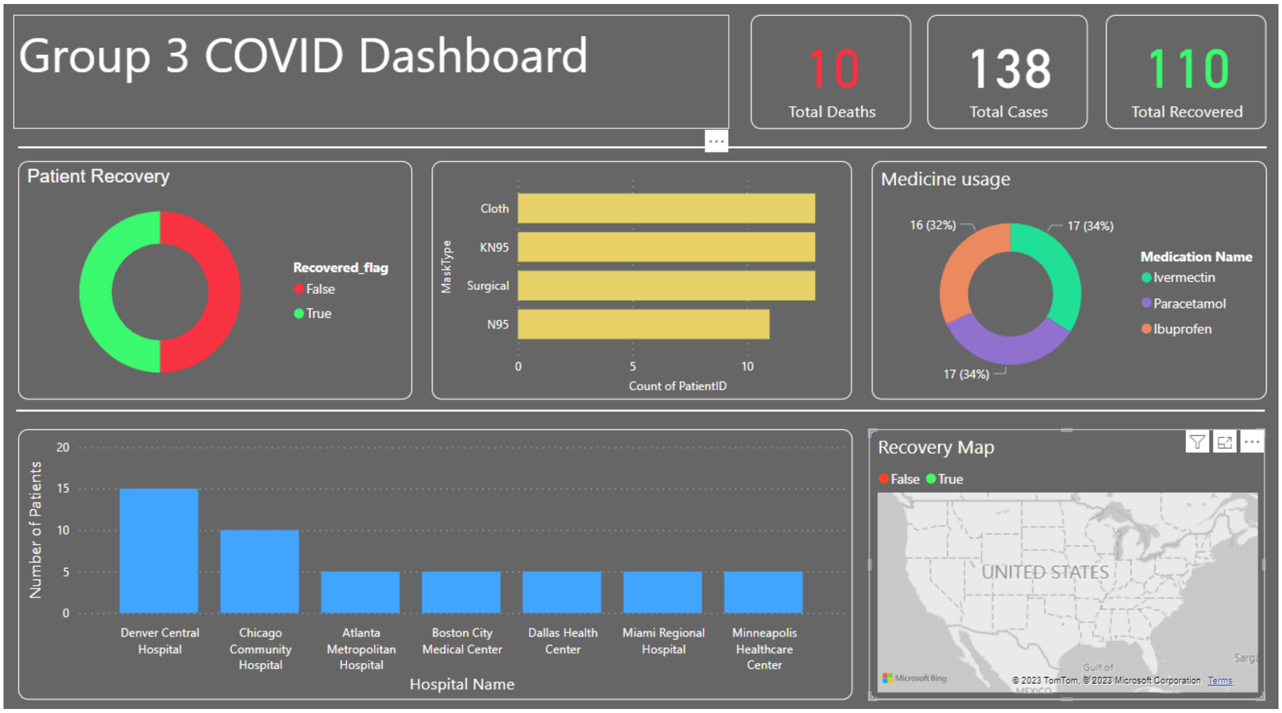
PatientID	FirstName	LastName	FeedbackText
1	John	Doe	Great experience with the healthcare team!
3	Michael	Johnson	Thankful for the support during my recovery.
5	David	Miller	The medical staff was caring and efficient.
7	Chris	Anderson	Grateful for the excellent care provided.
9	Brian	White	Recovery process was smooth
11	Robert	Lee	Appreciate the guidance and assistance throug...
13	Daniel	Nguyen	Positive atmosphere during my recovery journey.
15	James	Smithson	Professionalism and care made a significant diff...
17	Matthew	Taylor	The healthcare team was responsive and suppo...
19	Nathan	Chen	Thank you for the excellent medical care.

CHALLENGES FACED

The development of a COVID database faced numerous challenges, including:

1. **Data collection and integration:** Gathering COVID-related data from diverse sources, such as healthcare facilities, government agencies, and international organizations, presented a significant challenge due to varying data formats, inconsistencies in reporting practices, and potential data quality issues to the database. Following data scraping and cleaning, we developed a script to seamlessly transfer the data from the file to the database. This proved invaluable for data testing, enabling us to make necessary modifications and push the data repeatedly without the tedium of manual table insertions.
2. **Data accuracy and completeness:** Ensuring the accuracy and completeness of COVID data was crucial for the reliability of the database. Missing data, inconsistencies in data entry, and potential errors in coding or reporting posed significant challenges. Data validation procedures, including data quality checks, anomaly detection, and missing data imputation techniques, were essential for maintaining data integrity.
3. **Data visualization and analysis:** Effectively communicating and analysing the vast amount of COVID data required sophisticated data visualization tools and analytical techniques. We Developed an interactive dashboard using Power BI and employed statistical methods which were essential for extracting meaningful insights from the data and supporting evidence-based decision-making.

POWER BI DASHBOARD



Overall Observations:

- The dashboard displays a prominent "Total Recovered" figure. It's crucial to compare this number with reliable sources like the World Health Organization (WHO) or national health agencies. Analyzing discrepancies can reveal potential underreporting, reporting delays, or differences in counting methodologies.
- The visualization presents a breakdown of mask usage, highlighting cloth (34%), KN95 (34%), and N95 (17%) as the most popular choices.
- Ivermectin and Paracetamol appear to be the leading medications used (34% and 17% respectively), warranting further investigation into their prevalence.
- A map depicting recovery progress across regions offers valuable insights into potential geographical disparities.

CONCLUSION

This database system will empower public health specialists to:

- Analyze: the efficacy of various interventions like vaccines, medications, and government actions against COVID-19.
- Identify: trends and patterns in patient data to predict future outbreaks and inform effective resource allocation.
- Draw conclusions: about the effectiveness of different strategies and refine their approaches to mitigate the impact of COVID-19.
- Visualize: data in a clear and comprehensible manner, facilitating collaboration and communication across different stakeholders.
- Make informed decisions: based on comprehensive data analysis, ultimately leading to improved public health outcomes.

By providing a centralized and integrated data platform, this system will equip healthcare professionals with the knowledge and tools necessary to combat COVID-19 and safeguard public health.

Note : Please refer to the ReadMe in the .tar file provided for usage instructions.