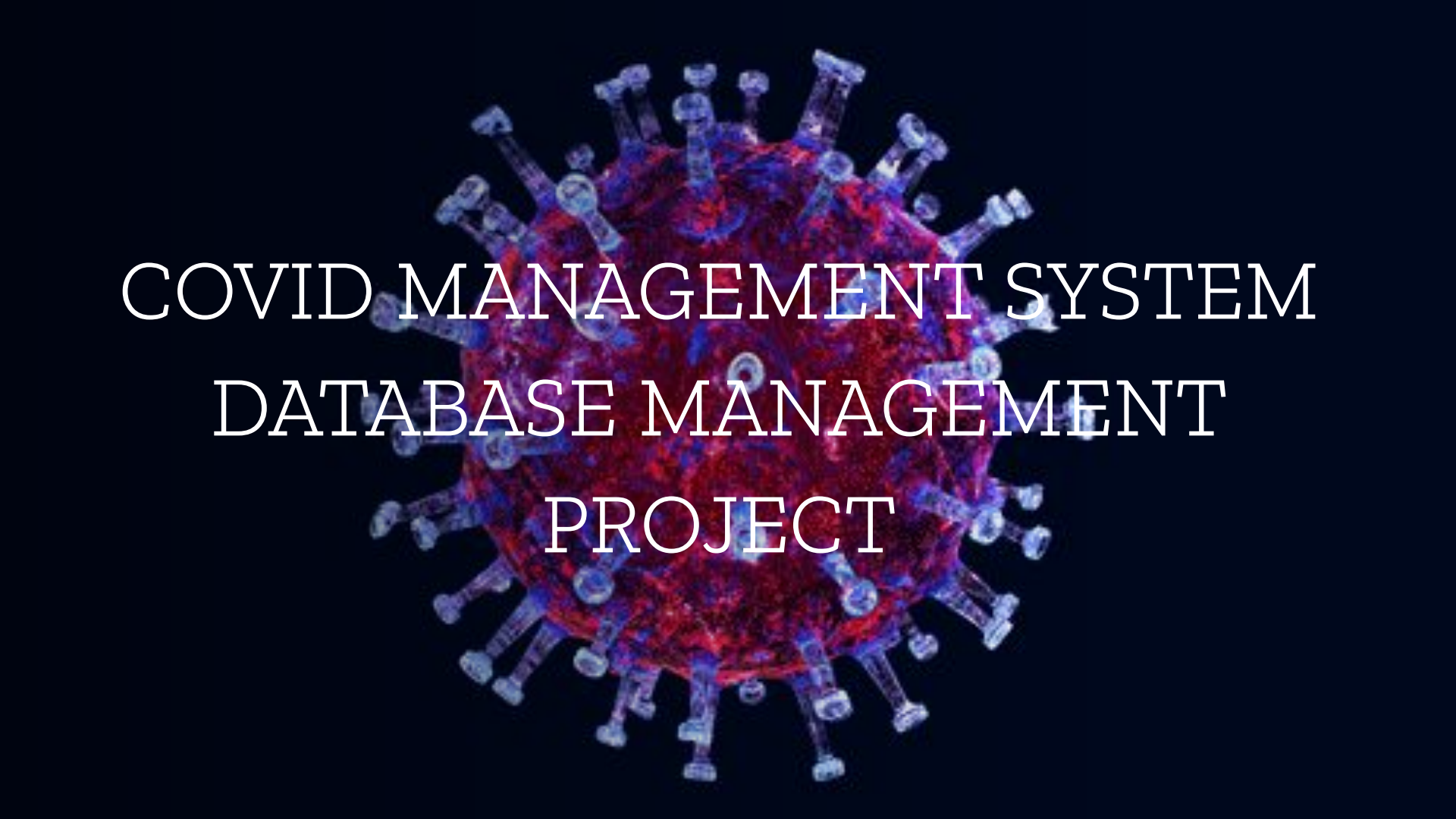

COVID MANAGEMENT SYSTEM
DATABASE MANAGEMENT
PROJECT

TEAM PRESENTATION

Ronit Chougule

Shreya Soni

Smruti Jethwani

Vini Wategaonkar

TOPICS

- DATA DEFINITION POPULATION
- ENTITY RELATIONSHIP DIAGRAM
- OBJECTIVE
- PROCEDURES, FUNCTIONS & VIEWS
- PROBLEM STATEMENT
- TRIGGERS, ENCRYPTION & VISUALIZATION

PROBLEM STATEMENT

- Over the past two years, there has been an outbreak of Coronavirus (Covid) that is spreading rapidly worldwide.

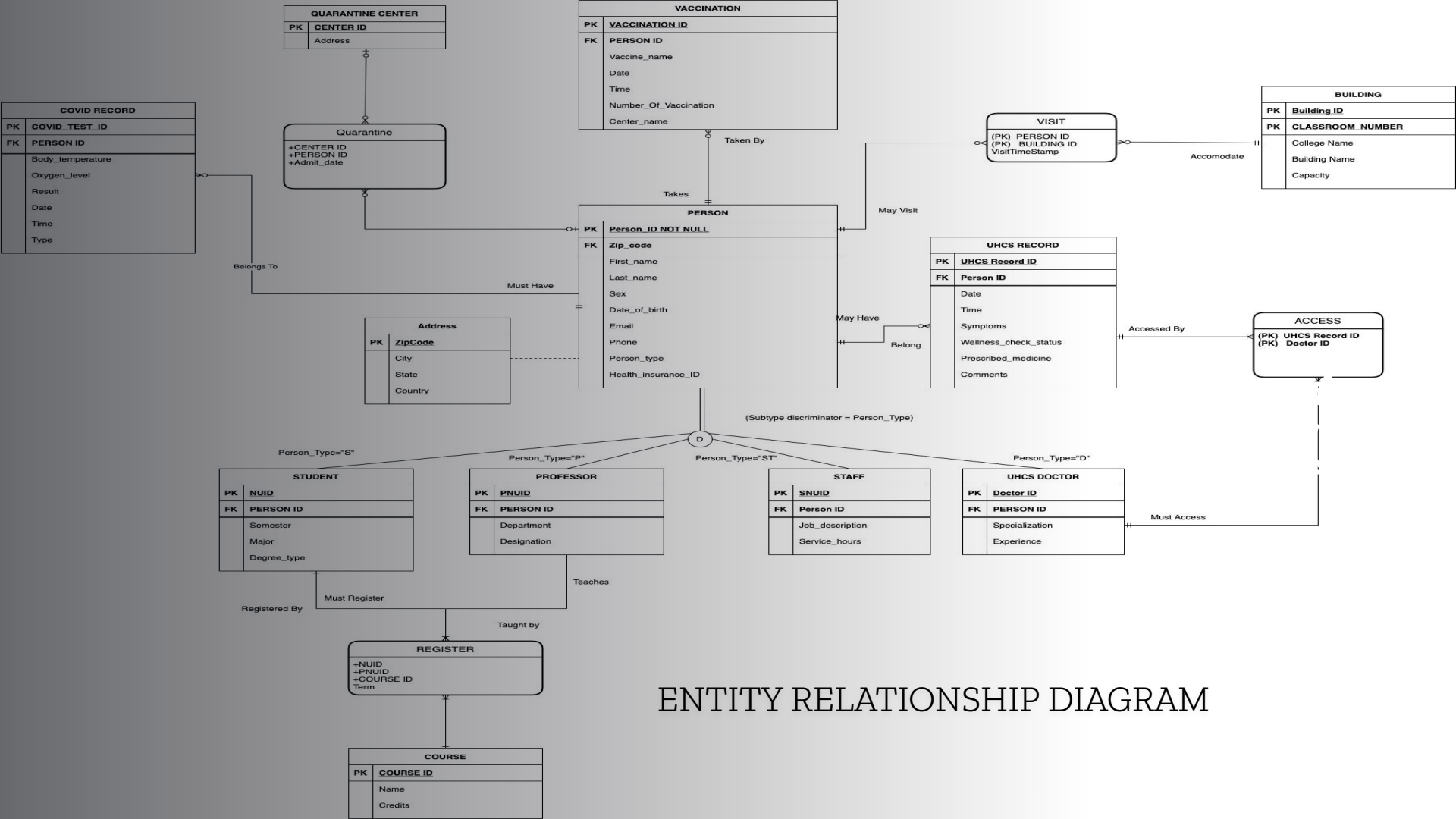
- This has resulted in the shutdown of many universities, restaurants, and offices.

OBJECTIVE

- This project aims to control the spread of Coronavirus by closely monitoring student, faculty and staff health & wellbeing.

- This project will ensure smooth functioning and safe working conditions at the university.

ENTITY RELATIONSHIP DIAGRAM



ENTITY RELATIONSHIP DIAGRAM

DATA DEFINITION LANGUAGE

Person Table.

```
CREATE TABLE PERSON (
 personID INT NOT NULL PRIMARY KEY IDENTITY(1,1),
 zipCode INT FOREIGN KEY REFERENCES ADDRESS_DIVISION(zipCode),
 firstName VARCHAR(20),
 lastName VARCHAR(20),
 sex VARCHAR(10),
 dateOfBirth DATE,
 email VARCHAR(50),
 phone VARCHAR(10),
 personType VARCHAR(10),
 healthInsuranceID VARCHAR(20)
)
```

Covid Record.

```
CREATE TABLE COVID_RECORD(
 covidTestID INT NOT NULL PRIMARY KEY IDENTITY(100,1),
 personID INT FOREIGN KEY REFERENCES PERSON(personID),
 bodyTemperature FLOAT,
 oxygenLevel FLOAT,
 result VARCHAR(10),
 CovidTestTimestamp DATETIME,
 typeOfTest VARCHAR(20)
);
```

DATA MANIPULATION LANGUAGE

Person Table

```
INSERT INTO PERSON VALUES
(02215, 'Simon', 'Frachie', 'female', '11/30/1996', 'simonfrachie@gmail.com',
'8976549012', 'st', 'NUQ873412980'),
(02215, 'Shimona', 'Gosling', 'female', '09/10/1996', 'shimonagosling@gmail.com',
'7609126589', 'st','NUQ873412891'),
(02215, 'Shinchan', 'Huyang', 'male', '07/24/1996', 'shinchanhuyang@gmail.com',
'8976546789', 'st', 'NUQ873489034'),
(02215, 'Christine', 'Rosling', 'female', '11/21/1997', 'christinerosling@gmail.com',
'6174709975', 'st', 'NUQ873409908')
```

Vaccination Table.

```
INSERT INTO VACCINATION VALUES (2, 'Covaxine','07-06-2021',1,'412 Park avenue center 1');
INSERT INTO VACCINATION VALUES (3, 'Pfizer','11-07-2021',2,'Dubai vaccine covid center 9');
INSERT INTO VACCINATION VALUES (4, 'covishield','12-05-2021',2,'6 Street park covid center 2');
INSERT INTO VACCINATION VALUES (5, 'Covishield','10-06-2021',2,'12 Beaker street covid center');
```

STORED PROCEDURES

Display student, professor, course relation.

```
CREATE PROCEDURE getRegisterationDetails AS
BEGIN
SELECT CONCAT(P1.firstName, ' ',P1.lastName) AS STUDENT,
CONCAT(P2.firstName,' ',P2.lastName) AS PROFESSOR, C.courseName
FROM REGISTER R JOIN PROFESSOR PF ON R.PNUID=PF.PNUID JOIN STUDENT S ON S.NUID=R.NUID
JOIN COURSE C ON R.courseID=C.courseID JOIN PERSON P1 ON P1.personID=S.personID JOIN
PERSON P2
ON P2.personID=PF.personID
ORDER BY C.courseName
END;
```

A person visited given building for a given date.

```
CREATE PROCEDURE getBuildingNoAndTime @buildingID int , @visitTimeStamp date AS
BEGIN
SELECT distinct p.personID, v.classRoomNumber, CONCAT(p.firstName,' ', p.lastName) AS
FullName,
case when personType = 'pf' then 'Professor'
when personType = 'st' then 'Student' else 'Staff'
END AS PersonType
FROM visit v join person p on p.personID=v.personID join building b
on b.buildingID=v.buildingID where b.buildingID=@buildingID and CONVERT(VARCHAR(10),
visitTimeStamp, 111)=@visitTimeStamp
END;
```

Find person last week visiting history if find out to be covid positive.

```
CREATE PROCEDURE dbo.getCovidCasesStatus @numberOfDays INT
AS
SELECT distinct p.personID, C.result, CONCAT(p.firstName,' ', p.lastName) AS FullName,
CONCAT(b.collegeName,' ',b.buildingName) AS BUILDING,
case when personType = 'pf' then 'Professor'
when personType = 'st' then 'Student' else 'Staff'
END AS PersonType
FROM PERSON p join COVID_RECORD C on p.personID= C.personID join VISIT V
on V.personID=p.personID JOIN BUILDING b ON b.buildingID = V.buildingID
where c.result = 'Positive'
AND CONVERT(VARCHAR(10), visitTimeStamp, 111) BETWEEN GETDATE() - @numberOfDays and
GETDATE();
```

Person details vaccinated by given vaccine name

```
CREATE PROCEDURE dbo.getVaccinationStatus @vaccine VARCHAR(30)
AS
SELECT distinct p.personID, CONCAT(p.firstName,' ', p.lastName) AS Full_Name,
case when personType = 'pf' then 'Professor'
when personType = 'st' then 'Student' else 'Staff'
END AS PersonType , v.vaccineName AS Vaccine_Name, v.numberOfVaccination AS
Total_Vaccinations
FROM PERSON p join VACCINATION v on p.personID = v.personID
where v.vaccineName = @vaccine
```

VIEWS

List of people belong to a particular course.

```
CREATE VIEW getRegistrationForDMDD AS
SELECT CONCAT(P1.firstName, ' ',P1.lastName) AS STUDENT,
CONCAT(P2.firstName,' ',P2.lastName) AS PROFESSOR, C.courseName
FROM REGISTER R JOIN PROFESSOR PF ON R.PNUID=PF.PNUID JOIN STUDENT S ON S.NUID=R.NUID
JOIN COURSE C ON R.courseID=C.courseID JOIN PERSON P1 ON P1.personID=S.personID JOIN
PERSON P2
ON P2.personID=PF.personID
WHERE C.courseName = 'DMDD';
```

List of people vaccinated by each quarter of year 2021

```
CREATE VIEW GetVaccinatedPeopleFirstQ AS
select P.personID ,CONCAT(p.firstName,' ', p.lastName) AS FullName,
case when P.personType = 'pf' then 'Professor'
when P.personType = 'st' then 'Student' else 'Staff'
END AS PersonType, V.vaccineName,V.dateOfVaccination, V.numberOfVaccination
from PERSON P JOIN VACCINATION V ON P.personID=V.personID WHERE CONVERT(VARCHAR(10),
dateOfVaccination, 111)
BETWEEN '2021/01/01' and  '2021/04/30';
```

USER DEFINED FUNCTION

No. of the students opted for a particular course.

```
CREATE FUNCTION getStudentsCountOptedForCourse(
@course_name VARCHAR(25))
RETURNS INT
AS
BEGIN
DECLARE @COUNT INT
SELECT @COUNT = COUNT(R.NUID) FROM REGISTER R JOIN COURSE C ON R.courseID=C.courseID
WHERE C.courseName=@course_name
RETURN @COUNT
END;
```

Average body temperature or oxygen level of the people who are positive.

```
CREATE FUNCTION getAverageBodyTemp()
RETURNS FLOAT
AS
BEGIN
DECLARE @avgTemp FLOAT
select @avgTemp = AVG(bodyTemperature) from COVID_RECORD WHERE result = 'positive'
RETURN @avgTemp
END;
```

TRIGGER

What are the available quarantine centers if the subject is positive

```
create TRIGGER covidCenterDetails
ON COVID_RECORD
FOR insert
as
BEGIN
select * from QUARANTINE_CENTER WHERE (select [result] from INSERTED) = 'Positive'
END;
```

ENCRYPTION

Add encryption on health insurance id

```
ALTER TABLE PERSON ADD [Encrypted_Health_Insurance] varbinary(400) NULL
```

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'PERSON@1234';
```

```
SELECT name KeyName,
symmetric_key_id KeyID,
key_length KeyLength,
algorithm_desc KeyAlgorithm
FROM sys.symmetric_keys;
```

Continue…

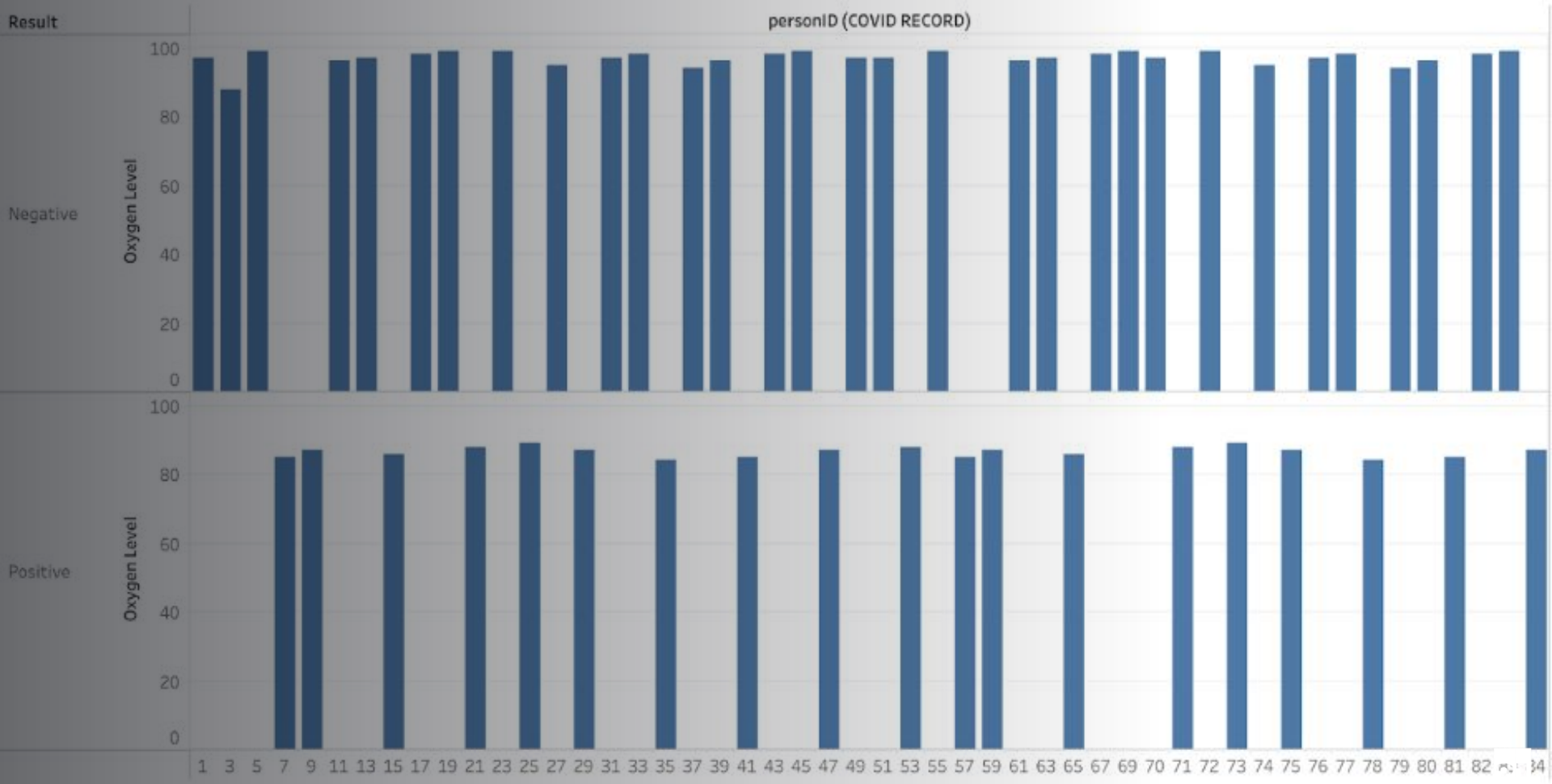
```
CREATE CERTIFICATE healthInsuranceNumber WITH SUBJECT = 'Person Health Insurance Number';

CREATE SYMMETRIC KEY HealthPass_SM WITH ALGORITHM = AES_256 ENCRYPTION BY CERTIFICATE healthInsuranceNumber;

UPDATE PERSON set [Encrypted_Health_Insurance] = EncryptByKey(Key_GUID('HealthPass_SM'), CONVERT(varbinary,[healthInsuranceID]) )

OPEN SYMMETRIC KEY HealthPass_SM DECRYPTION BY CERTIFICATE healthInsuranceNumber;
```

VISUALIZATION

POWER BI & TABLEAU

Sheet 1

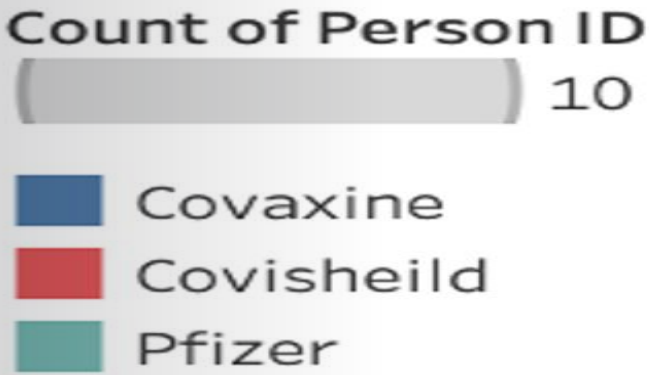
It shows oxygen level of people with covid results positive & negative



Sum of Oxygen Level for each personID (COVID RECORD) broken down by Result. The view is filtered on Exclusions (personID (COVID RECORD), Result), which keeps 50 members.

Sheet 1

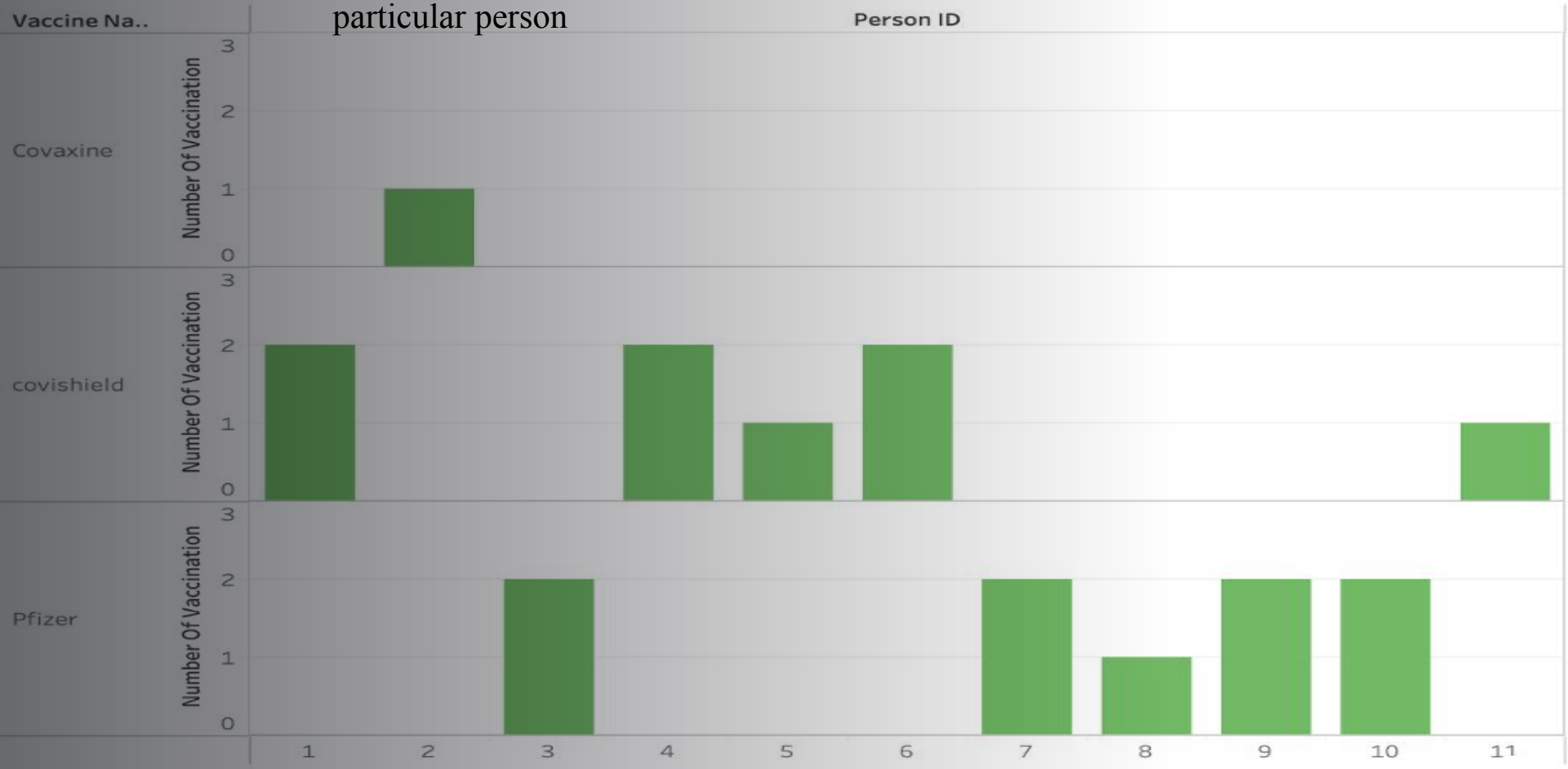
The following graph shows total number of people vaccinated by particular vaccine.



Vaccine Name (color) and count of Person ID (size). The view is filtered on Vaccine Name, which excludes Covershield.

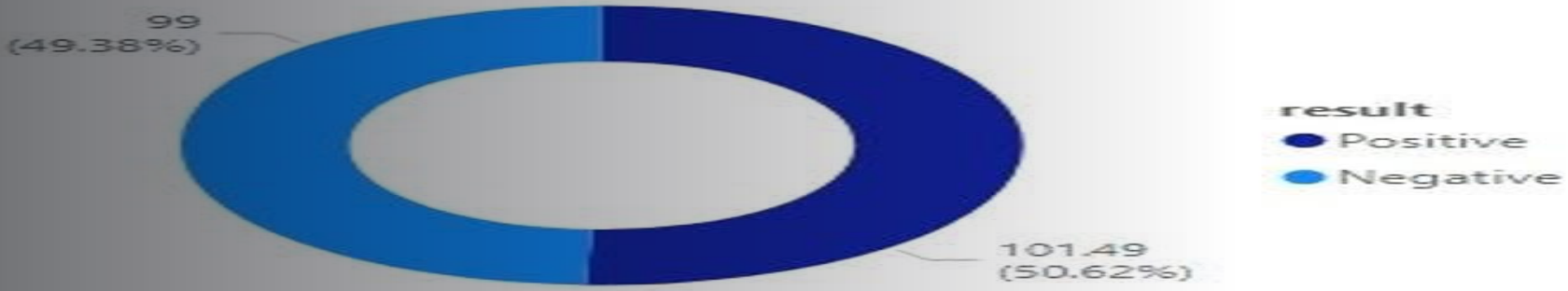
Sheet 1

The following graph shows vaccine name number of shots taken by a particular person

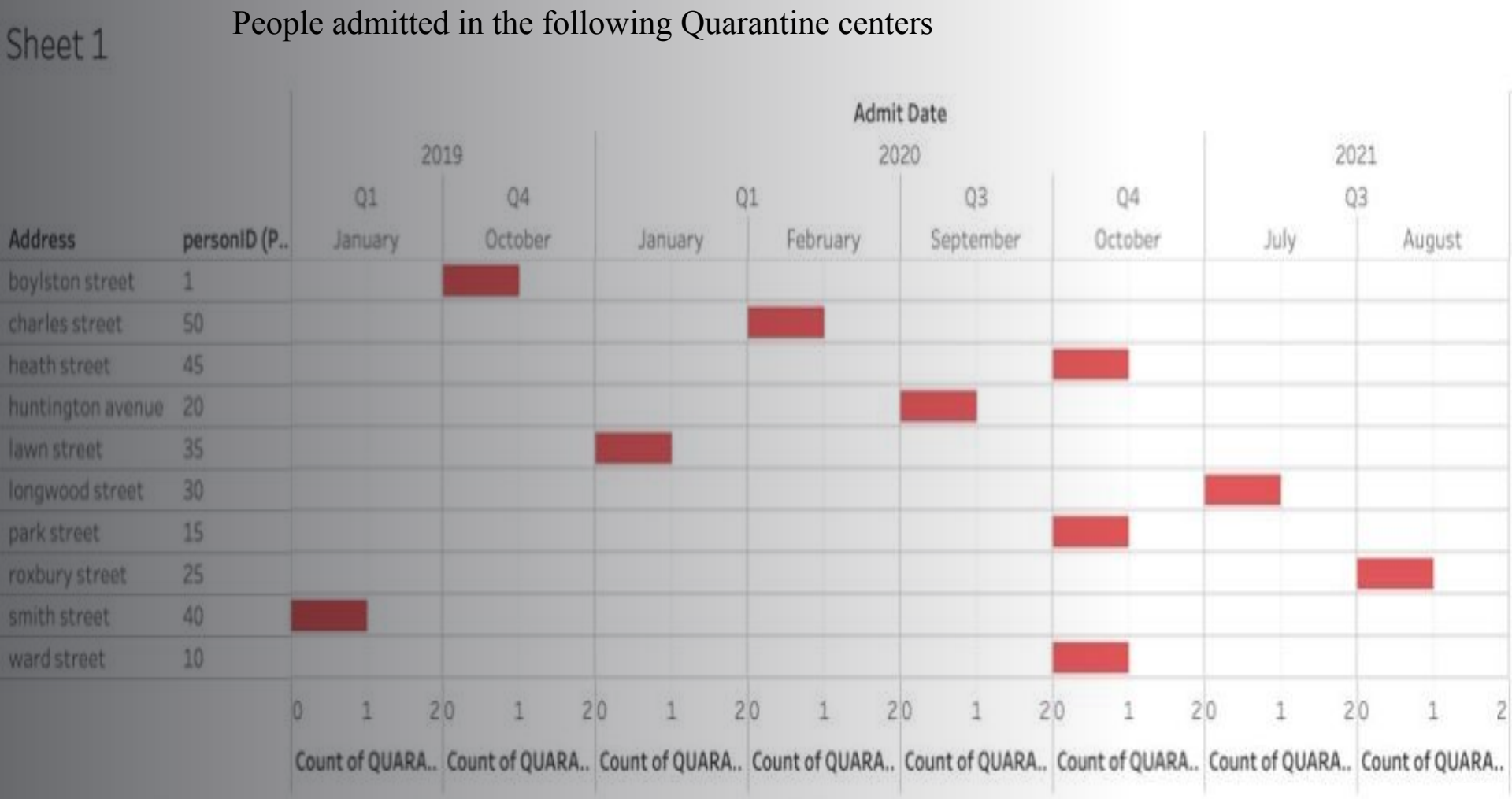


Sum of Number Of Vaccination for each Person ID broken down by Vaccine Name. The view is filtered on Vaccine Name, which excludes Covershield.

Average body temperature and oxygen level of people with results



People admitted in the following Quarantine centers



CONTRIBUTION

RONIT

- DDL
- DML
- Encryption
- Procedure
- ERD

SHREYA

- DDL
- DML
- Visualization Graphs
- Procedure
- Triggers
- ERD

VINI

- DDL
- DML
- Procedure
- Views
- Visualization Document
- ERD

SMRUTI

- DDL
- DML
- Visualization Graphs
- Procedure
- Views
- ERD

THANK YOU!