2020/2021

DATABASE PROJECT

Submitted in fulfillment of the requirements for the

DATABASE PARTIAL GRADE FROM THE LEBANESE UNIVERSITY
FACULTY OF ENGINEERING – BRANCH III

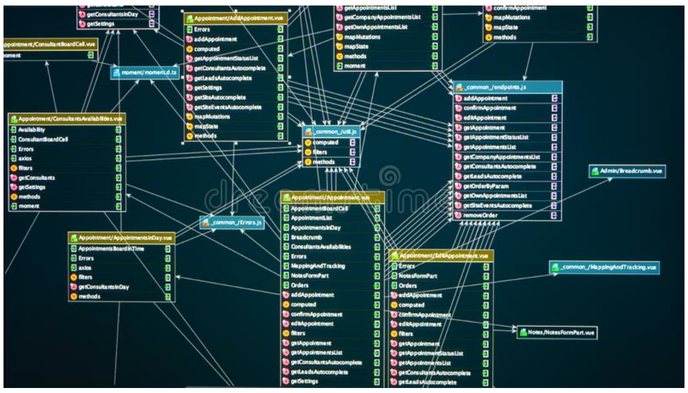
Major: Electrical and Telecommunication Engineering

Prepared By:

Dani Zeineddine 5485

Hussein Kazem 5667

COVID-19 Vaccination process database management system implementation



Supervised by:

Dr. Ahmad Fadlallah

Presented on **28** June 2021.

Table of Contents

Abstract

The number of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2)-infected patients keeps rising in most of the European countries despite the pandemic precaution measures. The current antiviral and anti-inflammatory therapeutic approaches are only supportive, have limited efficacy, and the prevention in reducing the transmission of SARS-CoV-2 virus is the best hope for public health. It is presumed that an effective vaccination against SARS-CoV-2 infection could mobilize the innate and adaptive immune responses and provide protection against severe forms of coronavirus disease 2019 (COVID-19) disease.

The main purpose of this project is implementing a database management system to automate the vaccination process in a country.

Introduction

Vaccination process of the COVID-19 virus depends on many different criteria, including age, whether we are infected or not, etc. Collecting data from different cases allows us to calculate the efficacy of a vaccine.

Our main purpose of this project is to provide a database management system to automate this process. This is done by explaining first the principal entities and relationships then implementing it using dB fiddle (with MySQL v5.7) and phpmyadmin.

In our project, to design the database , we will have 7 main relations:

- Citizen
- Health_workers
- Engineers
- Professors
- Journalists
- Available_Vaccines
- Vaccinated_people

The "Citizen" entity is the table including information about all the citizens of a country and wants to be vaccinated,collected using an official platform.

"Health_workers" , "Engineers" , "Professors" , "Journalists" represent the people that have different priorities of being vaccinated corresponding to other types of jobs. A citizen must send his data using the official vaccination platform and then his information is sent by the company he works for to the government to get the approval of being vaccinated early, this is why we should have a separate entity for each of the listed jobs.

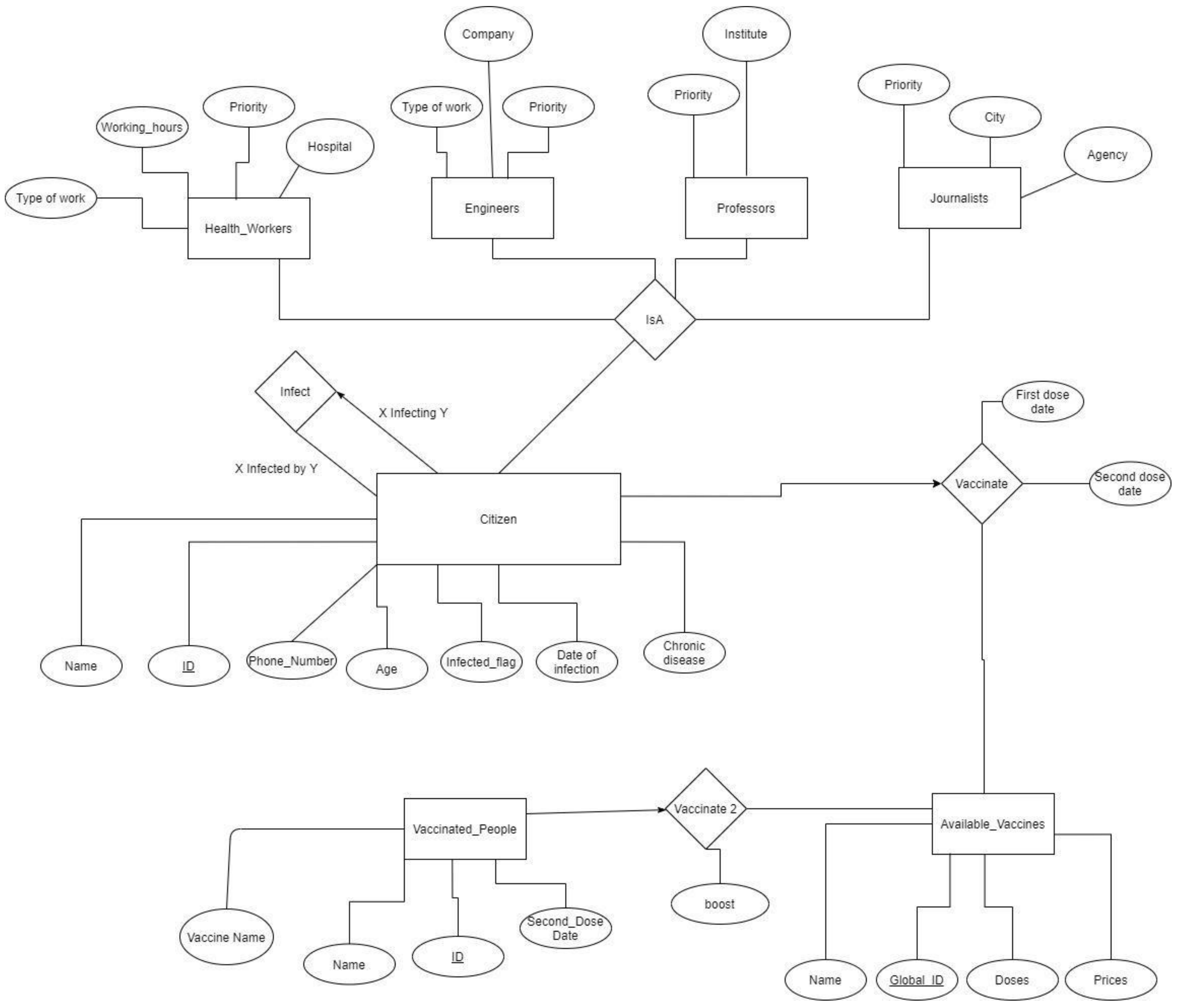
"Availabe_Vaccines" entity is a table including all the types of available vaccines, their prices and the number of available doses in a country.

"Vaccinated_people" is a table including all the people that have taken the 2 doses of a double dose vaccine (Pfizer, AstraZeneca,etc …) or 1 dose of a single dose vaccine (Johnson & Johnson).

These entities are related to each other using different relationships that we will explain in the next part.

Business Rules

Our implementation of the project is based on the following ER diagram:



Each citizen will have a Name, ID , age , a flag INFECTED indicating if the citizen was infected (1) with the virus before or not (0), the date of infection (NULL if INFECTED flag is 0) , information if he has chronic diseases or not (using a Boolean variable).

Citizen(name [varchar] , <u>id</u> [int] , age [int] , INFECTED [boolean] , infection_date [date] , chronic_disease [Boolean]).

“Health_workers” , “Engineers” , “Professors” , “Journalists” will be structured after collecting the required data, they represent an IsA relationship from Citizen entity and structured using the basic ER model.

Health_worker(id[int], Type of work [varchar], hours of work[int], priority[int]).
Engineer (Engineer_ID int, Company varchar(20),Type_of_work varchar(10), Priority int).
Journalist (Journalist_ID int, Agency varchar(20),City varchar(25),Priority int).
Professor (Professor_ID int(11), Institute varchar(30), Priority int).

The priority value is based on the type of work of each worker (example, a doctor working in the COVID-19 departement of a hospital has a higher priority value than a dentist).

In the country there exists a specified number of doses of each vaccine
(pfizer, moderna, AstraZeneca , Sinopharm, Sputnik V, J&J), each vaccine has a corresponding global ID referring to it.

Available_Vaccines(name [varchar], global_id [int] , doses [int], prices [int]).

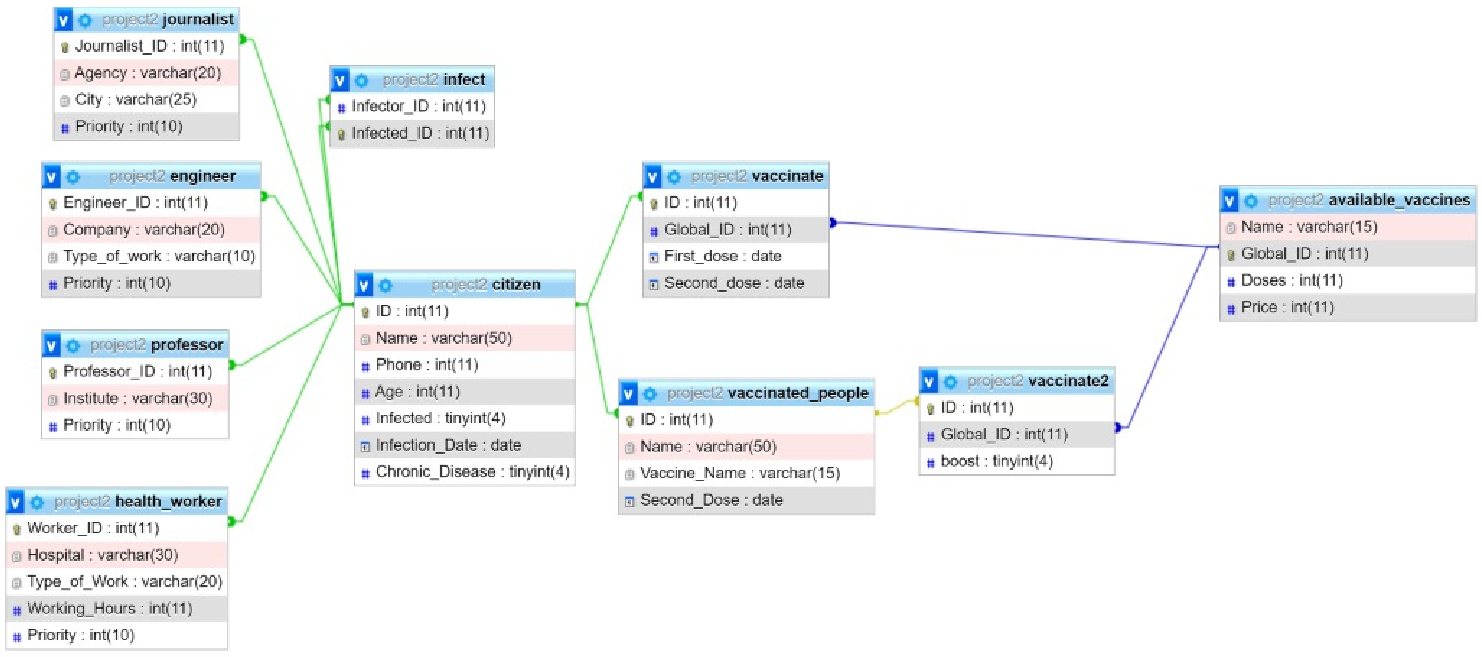
Each citizen that has taken the 2 doses of a vaccine is noted in the table **Vaccinated_People** with noting the date of taking the second dose.

Infect(Infector_ID int,Infected_ID int) : a one to many unary relationship between 2 citizens that collects information about infections in the goal of calculating some correlations between vaccination and infections.

Vaccinate(ID int(11) , Global_ID int(11), First_dose date, Second_dose date) : a one to many binary relationship between a citizen and available_vaccines. Each citizen can take only one type of vaccine.This entity includes the date of the first and second dose.

Vaccinate2(ID int, Global_ID int ,boost bool) : a one to many binary relationship between vaccinated_people and available_vaccines. A fully vaccinated person must take a boost dose after 6 months of taking the second dose. The boost variable indicates if the citizen has taken a boost (1) or no (0).

After modeling the ER diagram into relational database entities, and assigning the correct primary and foreign keys for each entity, we obtain the following physical diagram (using ***phpMyAdmin***) that summarizes our next step.



In each table is noted the primary key (PK) and different foreign keys and the relationships between them.

*The schema can be built from the file **Project.sql** using DB fiddle,sql fiddle or any online database schema builder and selecting MySQL v.5.7 or 5.6.*

*Data samples are written in the file **DataSamples.sql***

*Queries are listed in the next section as well as written in the file **Queries.sql***

Implementation and queries

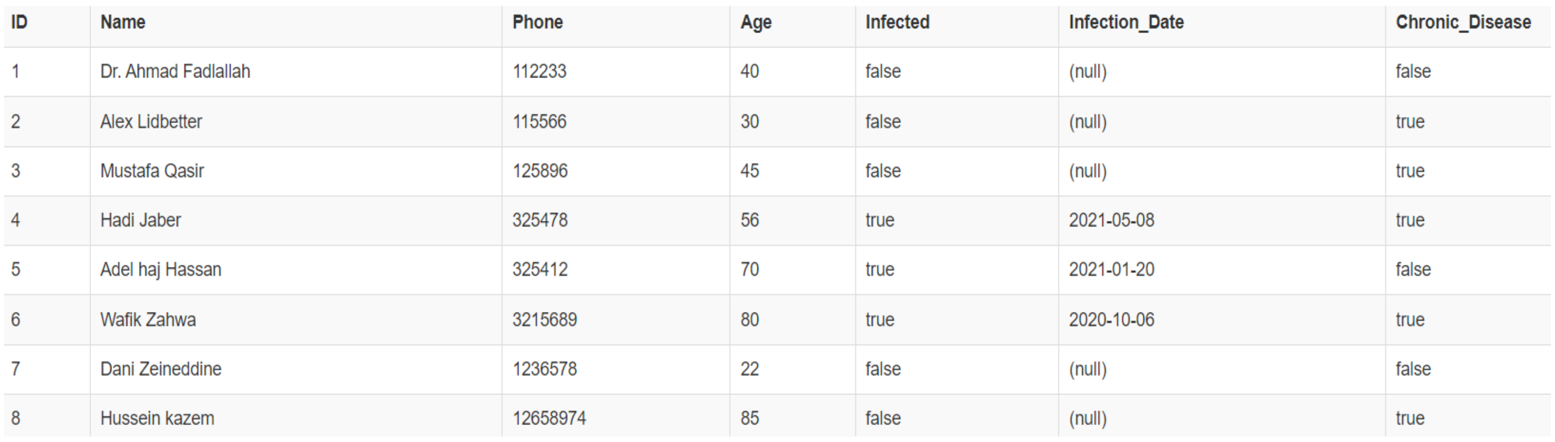
SQL queries are executed using DB Fiddle before creating the python application.
Inserting a citizen:

1)Normal citizen:
INSERT INTO citizen (id, name, phone, age, infected, infection_date, chronic_disease) values (8, 'Hussein kazem', '71001320', 85, false, null, true);

2)Worker citizen: example on professors:
INSERT INTO citizen (id, name, phone, age, infected, infection_date, chronic_disease) values (1, 'Dr. Ahmad Fadlallah', '70860230', 40, false, null, false);

INSERT INTO professor (Professor_ID,Institute,Priority) values (1,'ULFG 3',10);

Display all the citizens registered in the official platform:
*SELECT * FROM Citizen;*

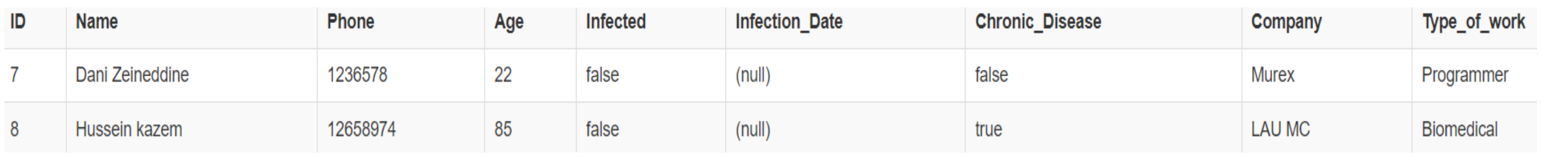


ID	Name	Phone	Age	Infected	Infection_Date	Chronic_Disease
1	Dr. Ahmad Fadlallah	112233	40	false	(null)	false
2	Alex Lidbetter	115566	30	false	(null)	true
3	Mustafa Qasir	125896	45	false	(null)	true
4	Hadi Jaber	325478	56	true	2021-05-08	true
5	Adel haj Hassan	325412	70	true	2021-01-20	false
6	Wafik Zahwa	3215689	80	true	2020-10-06	true
7	Dani Zeineddine	1236578	22	false	(null)	false
8	Hussein kazem	12658974	85	false	(null)	true

Display journalists, health workers, professors,engineers:
*SELECT * FROM citizen where id in (select professor_id from professor);*
*SELECT * FROM citizen where id in (select worker_id from health_worker);*
*SELECT * FROM citizen where id in (select journalist_id from journalist);*
*SELECT * FROM citizen where id in (select engineer_id from engineer);*

If we want to include the type of work and place:
SELECT DISTINCT c.,e.Company,e.Type_of_work FROM citizen c,engineer e where c.id=e.engineer_id;*

Example:



ID	Name	Phone	Age	Infected	Infection_Date	Chronic_Disease	Company	Type_of_work
7	Dani Zeineddine	1236578	22	false	(null)	false	Murex	Programmer
8	Hussein kazem	12658974	85	false	(null)	true	LAU MC	Biomedical

Display all the targeted citizens to be vaccinated depending on a minimum age(here is 60) and approved to be vaccinated:

```
SELECT * FROM citizen c where age>60 and (infected <> true OR datediff('2021/06/28',Infection_Date)>90) and c.id not in (Select v.id from vaccinate v );
```

datediff() function returns the answer in days of the difference of the 2 inputs. The first input is our current day.
To be vaccinated, a citizen must :
Be at least at a specified age (can be changed in the queries).
Must be recovered from the infection (if he was infected) from at least 3 months ago.

Vaccinating the upper targeted citizens with the pfizer vaccine (for example),we just need to insert the information in the vaccinate entity after decrementing the number of available doses in the pfizer vaccine:

```
UPDATE available_vaccines a SET a.doses=a.doses-(select count(c.id) FROM citizen c where age>60 and (infected <> true OR datediff('2021/06/28',Infection_Date)>90) and c.id not in (Select v.id from vaccinate v )) where a.global_id=1 ;

INSERT INTO vaccinate SELECT c.id,1,'2021/06/25',null FROM citizen c where age>60 and (infected <> true OR datediff('2021/06/28',Infection_Date)>90) and c.id not in (Select v.id from vaccinate v ) ;
```

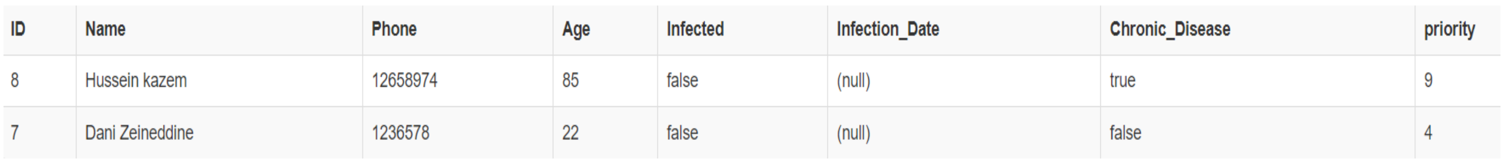
Display all the targeted citizens to be vaccinated depending on the type of work after combining and comparing the citizen entity with a job entity and collecting a successful match, and listing them depending on a specified priority:

```
SELECT c.*,j.priority FROM citizen c,professor j where c.id in (select professor_id from professor) and j.professor_id=c.id ORDER BY j.priority DESC;

SELECT c.*,j.priority FROM citizen c,health_worker j where c.id in (select worker_id from health_worker) and j.worker_id=c.id ORDER BY j.priority DESC;

SELECT c.*,j.priority FROM citizen c, journalist j where c.id in (select journalist_id from journalist) and j.journalist_id=c.id ORDER BY j.priority DESC;

SELECT c.*,j.priority FROM citizen c,engineer j where id in (select engineer_id from engineer) and j.engineer_id=c.id ORDER BY j.priority DESC;
```



ID	Name	Phone	Age	Infected	Infection_Date	Chronic_Disease	priority
8	Hussein kazem	12658974	85	false	(null)	true	9
7	Dani Zeineddine	1236578	22	false	(null)	false	4

Changing the data of the Citizen entity for a citizen after taking the first dose (marking the date of taking the 1st dose):

```
INSERT INTO vaccinate (id, global_id, first_dose,second_dose) values (5, 1, '2021/05/26',null);
```

Changing the data of a citizen after taking the second dose(marking the date of the second dose): let's take an example:

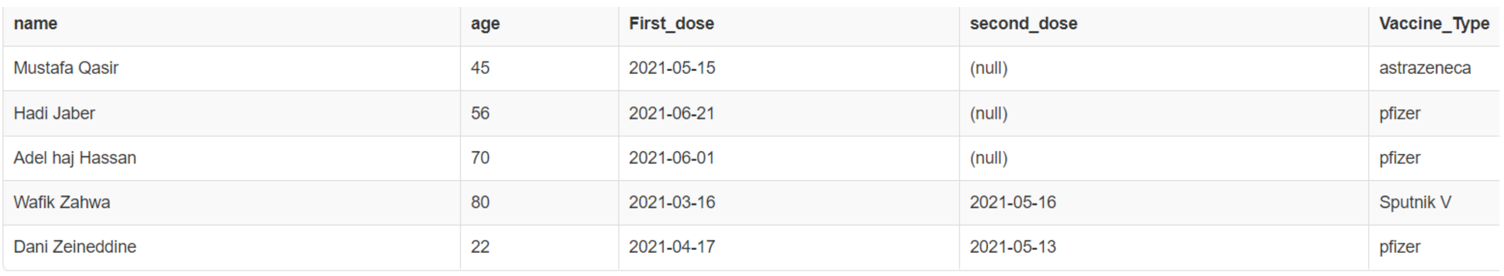
```
UPDATE vaccinate v SET v.second_dose='2021/06/1' where v.id=3;
```

Display all the targeted citizens to be vaccinated with a second dose after 21 days of the first dose:

```
SELECT * FROM citizen c,vaccinate v where c.id=v.id and v.second_dose is null and v.first_dose is not null and  datediff('2021/06/24',v.first_dose)>20;
```

Displaying name,age, date of first and second dose and vaccine type taken by citizens:

```
SELECT c.name,c.age,k.First_dose,k.second_dose,l.name as Vaccine_Type FROM citizen c,vaccinate k,available_vaccines l where c.id=k.id and l.Global_ID=k.Global_ID;
```



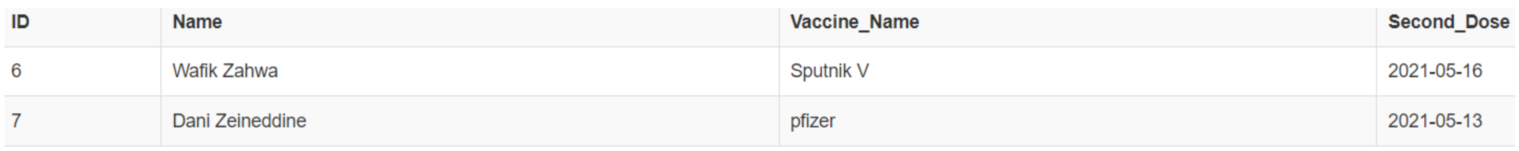
name	age	First_dose	second_dose	Vaccine_Type
Mustafa Qasir	45	2021-05-15	(null)	astrazeneca
Hadi Jaber	56	2021-06-21	(null)	pfizer
Adel haj Hassan	70	2021-06-01	(null)	pfizer
Wafik Zahwa	80	2021-03-16	2021-05-16	Sputnik V
Dani Zeineddine	22	2021-04-17	2021-05-13	pfizer

Each citizen that has taken 2 doses must be added to the table vaccinated_people after 15 days of the second dose:

```
INSERT INTO vaccinated_people SELECT c.id,c.name as citizen_name,a.name,v.second_dose FROM citizen c,available_vaccines a,vaccinate v WHERE c.id=v.id and v.global_id=a.global_id and v.second_dose is not null and datediff('2021/06/24',v.second_dose)>15;
```

Viewing fully vaccinated people:

```
SELECT * FROM vaccinated_people;
```



ID	Name	Vaccine_Name	Second_Dose
6	Wafik Zahwa	Sputnik V	2021-05-16
7	Dani Zeineddine	pfizer	2021-05-13

Adding information to vaccinate2:

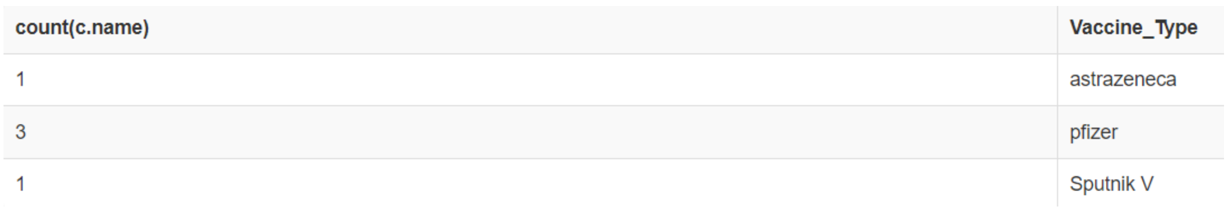
```
INSERT INTO vaccinate2 SELECT a.id,null,null from vaccinated_people a;
```

Checking citizens that need a boost after 6 months of the second dose:

```
SELECT p.* from vaccinated_people p,vaccinate2 v where v.global_id is null and v.boost is null and p.id=v.id;
```

Counting the number of doses taken by each vaccine :

```
SELECT count(c.name),l.name as Vaccine_Type FROM citizen c,vaccinate
k,available_vaccines l WHERE c.id=k.id AND l.Global_ID=k.Global_ID GROUP BY
l.name;
```



count(c.name)	Vaccine_Type
1	astrazeneca
3	pfizer
1	Sputnik V

Calculating the % of vaccination efficacy after taking 2 doses:(probability of being infected meanwhile we have taken 2 doses)

```
SELECT (count(infected_id)/(select (count(vaccinate.id)) from vaccinate))*100 as
vaccination_efficacy_in_percent FROM infect,vaccinate where
infect.Infected_ID=vaccinate.id and (vaccinate.first_dose is not null and
vaccinate.second_dose is not null);
```

Calculating the % of vaccination efficacy after taking 1 dose: (probability of being infected meanwhile we have taken 1 dose)

```
SELECT (count(infected_id)/(select (count(vaccinate.id)) from vaccinate))*100 as
vaccination_efficacy_in_percent FROM infect,vaccinate where
infect.Infected_ID=vaccinate.id and (vaccinate.first_dose is not null and
vaccinate.second_dose is null);
```

Calculating the % of covid spreading after taking 2 doses (probability of a person to infect others meanwhile this person has taken 2 doses)

```
select (count(infector_id)/(select (count(vaccinate.id)) from vaccinate))*100 as
vaccination_spreding_efficacy_in_percent from infect,vaccinate where
infect.Infector_ID=vaccinate.id and (vaccinate.first_dose is not null and
vaccinate.second_dose is not null);
```

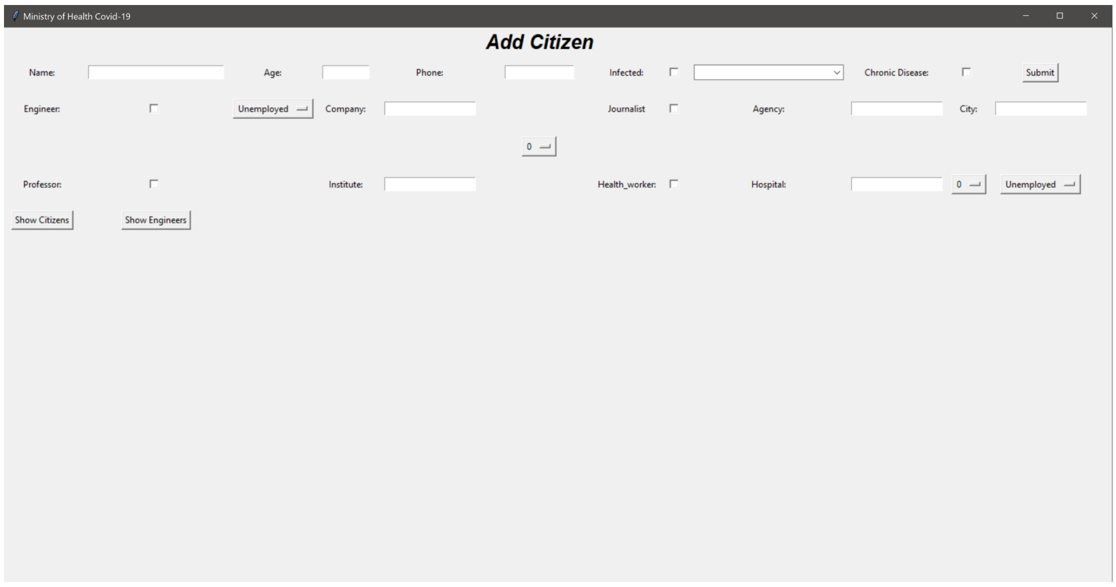
Calculating the % of covid spreading after taking 1 dose (probability of a person to infect others meanwhile this person has taken 1 dose)

```
SELECT (count(infector_id)/(select (count(vaccinate.id)) from vaccinate))*100 as
vaccination_spreding_efficacy_in_percent FROM infect,vaccinate where
infect.Infector_ID=vaccinate.id and (vaccinate.first_dose is not null and
vaccinate.second_dose is null);
```

Conclusion and Future considerations: Python GUI application

In conclusion, we all know how big of an effect covid-19 has had on our daily lives. The discovery of the vaccines has made our lives easier and brought us a few steps closer to going back to our normal lives. This project has been a great learning experience as it has offered us the opportunity to learn more about covid and its vaccines, in addition to a more hands on learning experience when it comes to database management systems.

For future considerations, perhaps building a fully operational GUI for the registration process in addition to a GUI for the hospital or health ministry employees to keep track of registrations and vaccination processes. A basic registration GUI using python and tkinter package could look like:


Ministry of Health Covid-19
Add Citizen
Name:
Age:
Phone:
Infected:
Chronic Disease:
Submit
Engineer:
Unemployed
Company:
Journalist
Agency:
City:
0
Professor:
Institute:
Health_worker:
Hospital:
0
Unemployed
Show Citizens
Show Engineers