ABDULLAH GÜL
ÜNİVERSİTESİ

Comp204 Midterm Project
Phase 1,2,3(All In One)

Mehmet Anıl İrfanoğlu

Instructor
Ahmet Soran
Samet Tonyalı

Abstract

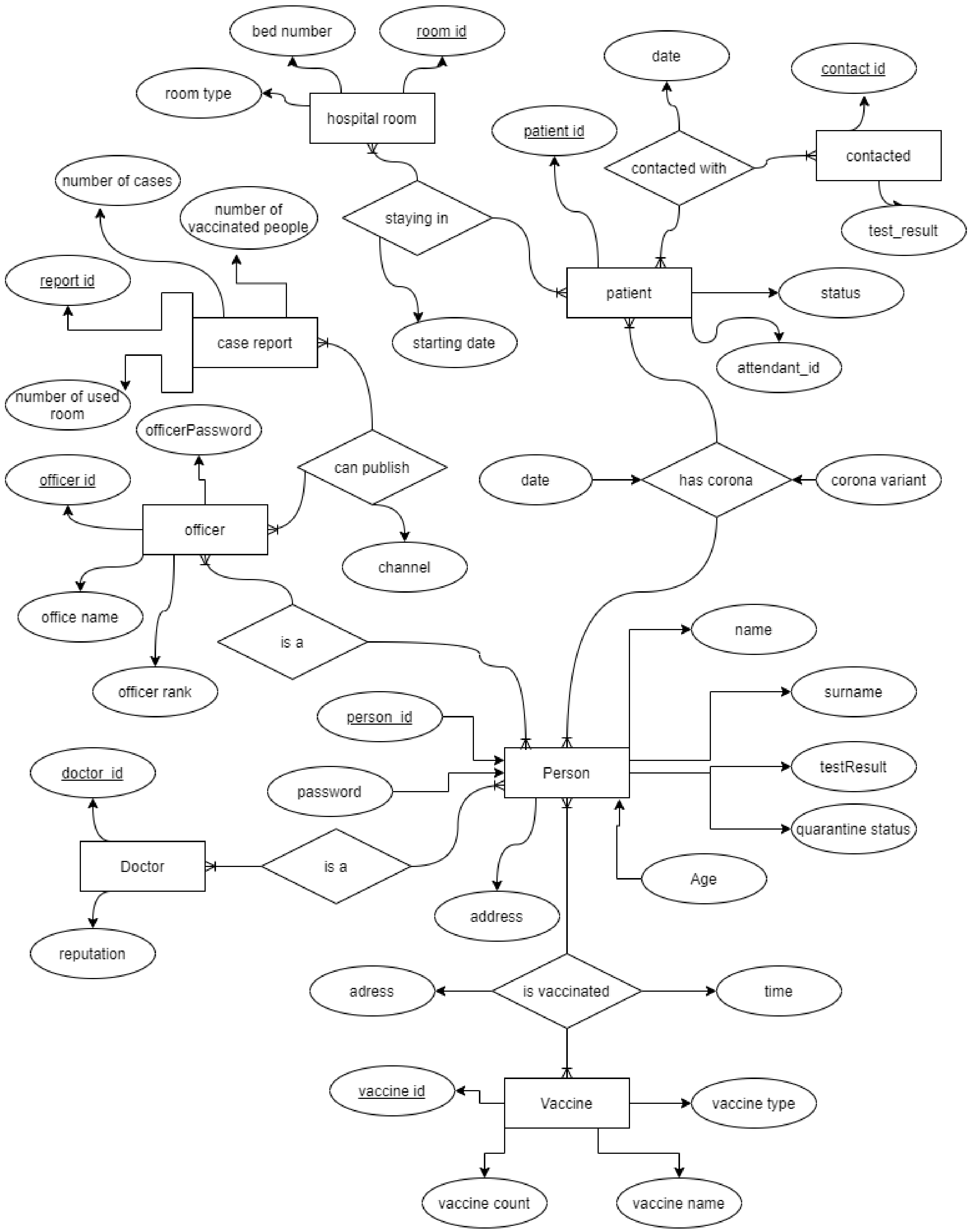
I made a database study on the first subject given in the visa in this project. In this context, I aimed to predict what might be required and establish a system accordingly. I have done more than one work for this purpose. I examined the corona epidemic, which is today's problem in general terms, and presented an application design in which the weight to contribute to the measures to be taken against this disease is given to the database.

REQUIREMENT ANALYSIS

I will write what might be needed in creating this project in this section. In my opinion, priority should be given to people in a database to be established about the disease. In this context, I have determined the person as the main starting point. but I do not think that the person alone is enough, and in this context, I have created a few more fields that depend on the person's structure and receive support from him. Chief among them is the patient class. I aim to keep a person's disease information here. In addition to this, I think that doctors who take care of patients should also be in this system. I also added government officials, who are above all else, who have access to everything, to the system. These people will be able to share the necessary parts of the collected data with the public at certain time intervals.

I built structures related to the disease as well as people. For example, I am thinking of recording the information about who the sick people are in contact with. I also aim to keep information about the rooms where the patients stay. Based on this, I think to include another important issue of vaccination into the system. In this context, I aim to keep information about which vaccine is available, which vaccine is used and vaccine. Finally, I think of designing places where there are general status statistics and where information that can be shared by government offices will be kept.

E-R DIAGRAM



DESIGN PHILOSOPHY

ER Diagram:

- **Address**

- **Person**
- **Doctor**
- **Patient**
- **Contacted**
- **HospitalRoom**
- **Officer**
- **CaseReport**
- **Vaccine**

In this section, I will talk about the elements in the design I created and what they do. First of all, I want to talk about the **person** being. person as the main starting point; It contains general information such as **person_id, name**, **surname**, **age**. This information is important in identifying individuals. Based on these, the person's **password**, **test result** and whether he is in **quarantine** are also kept in person.

Another structure that depends on the person is the **patient.** With the **hasCorona** relation linked to the patient, I keep track of **date** the person is sick and the **type of corona**. As I am currently keeping general information in the person, I did not specify them again in the patient section.

This patient entity is linked to other tables with 2 different relations. For example, **contactedWith** relaiton and **contacted** person was kept. In contactedWith, I also record when the contact occurred. On the contacted table, I keep the **ID** and **test result** of the person in contact.

The other relation to which the patient is connected is **hospitalRoom**. Here, I connect which patient is staying in which **room** with **stayingIn** relation and I keep the information on which **date** the patient started to stay in that room.

On the **hospitalRoom** table, I keep important information about the hospital room. For example, I keep the **numberOfBeds** in the hospital room and what **type of ro**om this room is. By room type I mean, for example, a room can be **intensive care**, and since not all equipment in intensive care is in normal rooms, keeping this information will be very important in the future.

It is also important to know how many beds are in the room because if the room is full, no more patients can be accommodated in that room. I wanted to determine which **doctor** was looking at which patient by adding the doctor table. This information is connected to the **patient** with **attendant_id**.

Speaking of the **officer**, these people can be thought of as high-level users. To give an example, these people can also be considered as officials working in the infrastructure of the ministry of health. The purpose of these officers is to share situation **reports** at regular intervals, as stated in **caseReport**, by looking at the data collected in general.

In these reports they share, the information about how many people are sick with **NumberOfCases**, how many rooms are used now with **numberOfUsedRoom**, and lastly, how many people have been vaccinated with **numberOfVaccinatedPeople** are kept.

These shared reports are shared by the officer by adding the information on **canPublish** relation and the **channel** that will be published on which platform. these reports can only be shared by the officer.

CARDINALITIES

MANY TO MANY

Patient,Contacted

Patient,HospitalRoom

Person,Patient

Person,Vaccine

Person,Doctor

Person,Officer

Officer,CaseReport

NORMALIZATION

1NF,2NF,3NF EXPLANATIONS

In order for my system to be in 1nf, it had to be an element in each cell. To achieve this, contact list, published reports, vaccine event can be considered. Because one person may have contact with more than one person and we cannot keep all contacts in one cell, and therefore, the information can be kept in a table for contact and the information can be stored in this way, and at the same time, multiple reports can be put into 1nf form with the same logic and set up a published table. Another case is that a person may have been vaccinated more than once, and to store it, the information that should be kept in a cell can be stored via a table by using the isVaccinated table. I will give an example of the things I have done to make my design in 2nf. The most important example is the person, doctor, patient structure. With this structure, I linked the tables that are independent from each other, but also the tables that store the data of the same person, with foreign keys. For example, a person can also be an officer, as well as a doctor. But no matter what, they are the same people all connected to each other through the foreign keys that I have defined. With this logic, a person who is not person cannot be a doctor, and these features are the requirements for 2nf. Finally, I will talk about what I have done for the 3nf form. Address information and vaccine information can be given as examples of what I do to catch this form. Because the vaccine information is linked to vaccine_id and if I open a vaccine table and keep it there rather than keeping this information in the vaccine's table, I can access this information from the person person through vaccine_id, but there is another important point here if I want to keep vaccine information about a person, also isvaccinated table. You should also check it out because someone who does not have vaccine does not have to have a vaccine_id. Another situation is related to the address. As for the address, there is the following situation, the city and district information depends on addree_id, which we can think of as the postal code because postal code is a set of numbers used to define a region and if I keep the address_id in person rather than keeping each city district information in person The address_id is actually the postal code specifying city and district, in this case, I set up the address table and keep the city district information for each person through the postal code address id, and I have provided 3nf because in the definition of the sought condition in 3nf, "A relation that is in First and Second Normal Form and in which no non-primary-key attribute is transitively dependent on the primary key, then it is in Third Normal Form (3NF) "and this definition is provided through these operations.

ADDITIONAL NOTES ABOUT NORMALIZATION

Normalization is normally done after the er diagram is designed, but while I was designing the er diagram, I made some changes that should be made in the future, namely the normalization process while designing ER.

First of all, the attribute of the address information in the er diagram causes an data redundancy use so I transferred the address information to a table. And in this way, if many people are streaming at the same address, there will be no data redundancy in the database. Another issue is the vaccine status.

Normally, whether or not to be vaccinated could be stated directly in person, but in this case it would take up unnecessary space because more than one user can get the same vaccine. The same is true for hospital rooms, because it makes more sense to create a separate table rather than giving the hospital room information on the patient table. In this way, if more than one person stays in the same room, the database will not keep extra information for nothing.

There is another situation related to the contact list. As we know, a patient may have been in contact with many people, so I think it would be much more logical to open a table called contact and link them to the patient rather than adding an attribute to the patient table. In fact, it is the state person structure that reduces data redundancy the most. Because the most basic information of the person is stored in the person table, and whether a person is a doctor, an officer or a patient, they can connect to the person table and retrieve their information from there. In this way, instead of storing extra name and surname type information for each type of person, these are stored once and when necessary, the requested information is used by accessing them.

Another situation is related to the case report shared by the officers. As it is known, a report can be shared by more than one person on more than one platform, so it would be very logical to keep them as a separate table and in this way, data redundancy will be reduced.

Another situation is that if a person has been vaccinated more than once, we are saved from keeping the same information for the second time by keeping it in the relation table. Since the system minimizes this type of data redundancy, a person can be a doctor, a patient and a person at the same time, and while doing this, the information of the same person is not recorded repeatedly because different information is stored in each table.

ER TO RELATIONAL MAPPING

Normal Entitities:

Address(Address_Id,City,District)

Person(Person_id , Name ,Surname ,Password, Age ,TestResult, Address_Id ,Quarantine_Status)

Doctor(Doctor_Id , Reputation)

Patient(Patient_Id,Status,Attendant_Id)

Contacted(Contact_Id,TestResult)

HospitalRoom(Room_Id,BedNumber,RoomType)

Officer(Officer_Id,OfficeName,OfficerPassword,OfficerRank)

CaseReport(Report_Id,NumberOfCases,NumberOfUsedRooms,NumberOfVaccinatedPeople)

Vaccine(Vaccine_Id,VaccineCount,VaccineName,VaccineType)

Relations:

ContactedWith(Carrier_Id,Patient_Id,Contact_Id,date)

HasCorona(Case_Id,Person_Id,Patient_Id,Date,CoronaVariant)

IsVaccinated(Vaccination_Id,Person_Id,Vaccine_Id,Address_Id,Date)

CanPublish(Publish_Id,Officer_Id,Report_Id,Channel)

StayingIn(Staying_Id,Room_Id,Patient_Id,StartingDate)

FUNCTIONAL DEPENDENCIES

Address(Address_Id,City,District)

Address_Id->City, Address_Id->District so,

Address_Id $^{+}$ = Address_Id City District ;

so **Address_Id** is **candidate key.**

Person(Person_id , Name ,Surname ,Password, Age ,TestResult, Address_Id ,Quarantine_Status)

Person_id ->Name, Person_id ->Surname, Person_id ->Password, Person_id ->Age, Person_id ->TestResult, Person_id ->Address_Id, Person_id ->Quarantine_Status

Person_id $^{+}$ = Person_id Name Surname Password Age TestResult Address_Id Quarantine_Status ;

so **Person_id** is **candidate key.**

Doctor(Doctor_Id , Reputation)

Doctor_Id -> Reputation so,

Doctor_Id $^{+}$ = Doctor_Id Reputation ;

so **Doctor_Id** is **candidate key.**

Patient(Patient_Id,Status,Attendant_Id)

Patient_Id -> Status, Patient_Id -> Attendant_Id so,

Patient_Id $^{+}$ = Patient_Id Status Attendant_Id;

so **Patient_Id** is **candidate key.**

Contacted(Contact_Id,TestResult)

Contact_Id -> TestResult so,

Contact_Id $^{+}$ = Contact_Id TestResult ;

so **Contact_Id** is **candidate key.**

HospitalRoom(Room_Id,BedNumber,RoomType)

Room_Id -> BedNumber, Room_Id -> RoomType so,

Room_Id $^{+}$ = Room_Id BedNumber RoomType;

so **Room_Id** is **candidate key.**

Officer(Officer_Id,OfficeName,OfficerPassword,OfficerRank)

Officer_Id -> OfficeName, Officer_Id -> OfficerPassword, Officer_Id -> OfficerRank so,

Officer_Id $^{+}$ = Officer_Id OfficeName OfficerPassword OfficerRank;

so **Officer_Id** is **candidate key.**

CaseReport(Report_Id,NumberOfCases,NumberOfUsedRooms,NumberOfVaccinatedPeople)

Report_Id -> NumberOfCases, Report_Id -> NumberOfUsedRooms, Report_Id -> NumberOfVaccinatedPeople so,

Report_Id $^{+}$ = Report_Id NumberOfCases NumberOfUsedRooms NumberOfVaccinatedPeople;

so **Report_Id** is **candidate key.**

Vaccine(Vaccine_Id,VaccineCount,VaccineName,VaccineType)

Vaccine_Id -> VaccineCount, Vaccine_Id -> VaccineName, Vaccine_Id -> VaccineType so,

Vaccine_Id $^{+}$ = Vaccine_Id VaccineCount VaccineName VaccineType;

so **Vaccine_Id** is **candidate key.**

ContactedWith(Carrier_Id,Patient_Id,Contact_Id,date)

Carrier_Id -> Patient_Id, Carrier_Id -> Contact_Id, Carrier_Id -> date so,

Carrier_Id $^{+}$ = Carrier_Id Patient_Id Contact_Id date;

so **Carrier_Id** is **candidate key.**

HasCorona(Case_Id,Person_Id,Patient_Id,Date,CoronaVariant)

Case_Id -> Person_Id, Case_Id -> Patient_Id, Case_Id -> Date , Case_Id -> CoronaVariant so,

Case_Id $^{+}$ = Case_Id Person_Id Patient_Id Date CoronaVariant;

so **Case_Id** is **candidate key.**

IsVaccinated(Vaccination_Id,Person_Id,Vaccine_Id,Address_Id,Date)

Vaccination_Id -> Person_Id, Vaccination_Id -> Vaccine_Id, Vaccination_Id -> Address_Id , Vaccination_Id -> Date so,

Vaccination_Id $^{+}$ = Vaccination_Id Person_Id Vaccine_Id Address_Id Date;

so **Vaccination_Id** is **candidate key.**

CanPublish(Publish_Id,Officer_Id,Report_Id,Channel)

Publish_Id -> Officer_Id, Publish_Id -> Report_Id, Publish_Id -> Channel so,
Publish_Id $^{+}$ = Publish_Id Officer_Id Report_Id Channel;
so **Publish_Id** is **candidate key.**

StayingIn(Staying_Id,Room_Id,Patient_Id,StartingDate)
Staying_Id -> Room_Id, Staying_Id -> Patient_Id, Staying_Id -> StartingDate so,
Staying_Id $^{+}$ = Staying_Id Patient_Id Report_Id StartingDate;
so **Staying_Id** is **candidate key.**

SCRIPTS

DDL CODES: (important note: please create schema with name of `midterm` **because foreign keys depends on it)**

```
CREATE TABLE `Address` (
`Address_Id` numeric NOT NULL,
`City` varchar (20) NOT NULL,
`District` varchar (20) NOT NULL,
PRIMARY KEY (`Address_Id`)
);
CREATE TABLE `Person` (
`Person_Id` numeric NOT NULL,
`Name` varchar(45) NOT NULL,
`Surname` varchar(45) NOT NULL,
`Password` varchar(45) NOT NULL,
`Age` numeric NOT NULL,
`TestResult` varchar(20) NOT NULL,
`Address_Id` numeric NOT NULL,
`Quarantine_Status` varchar(20) NOT NULL,
PRIMARY KEY (`Person_Id`),
FOREIGN KEY (`Address_Id`)
REFERENCES  `Address` (`Address_Id`)
);
CREATE TABLE `Doctor` (`Doctor_Id` numeric NOT NULL,
`Reputation` varchar (45) NOT NULL,
PRIMARY KEY (`Doctor_Id`)
);
```

```
CREATE TABLE `Patient` (
`Patient_Id` numeric NOT NULL,
`Status` varchar(45) NOT NULL,
`Attendant_Id` numeric NOT NULL,
PRIMARY KEY (`Patient_Id`),
FOREIGN KEY (`Patient_Id`)
REFERENCES `midterm`. `Person` (`Person_Id`),
FOREIGN KEY (`Attendant_Id`)
REFERENCES `Doctor` (`Doctor_Id`)
);
CREATE TABLE `Contacted` (
`Contact_Id` numeric NOT NULL,
`TestResult` varchar(45) NOT NULL,
PRIMARY KEY (`Contact_Id`),
FOREIGN KEY (`Contact_Id`)
REFERENCES  `Person` (`Person_Id`)
);
CREATE TABLE `HospitalRoom` (
`Room_Id` numeric NOT NULL,
`BedNumber` numeric NOT NULL,
`RoomType` varchar(45) NOT NULL,
PRIMARY KEY (`Room_Id`)
);
CREATE TABLE `Officer` (
`Officer_Id` numeric NOT NULL,
`OfficeName` varchar(45) NOT NULL,
`OfficerPassword` varchar(45) NOT NULL,
`OfficerRank` varchar(45) NOT NULL,
PRIMARY KEY (`Officer_Id`),
FOREIGN KEY (`Officer_Id`)
REFERENCES  `Person` (`Person_Id`));
CREATE TABLE `CaseReport` (
`Report_Id` numeric NOT NULL,
`NumberOfCases` numeric NOT NULL,
`NumberOfUsedRooms` numeric NOT NULL,
`NumberOfVaccinatedPeople` numeric NOT NULL,
PRIMARY KEY (`Report_Id`)
);
CREATE TABLE `Vaccine` (
`Vaccine_Id` numeric NOT NULL,
`VaccineCount` numeric NOT NULL,
`VaccineName` varchar(45) NOT NULL,
`VaccineType` varchar(45) NOT NULL,
PRIMARY KEY (`Vaccine_Id`)
);
CREATE TABLE `ContactedWith` (
`Carrier_Id` int AUTO_INCREMENT ,
`Patient_Id` numeric NOT NULL,
`Contact_Id` numeric NOT NULL,
`Date` varchar (20) NOT NULL,
PRIMARY KEY (`Carrier_Id`) ,
FOREIGN KEY (`Patient_Id`)
REFERENCES  `Patient` (`Patient_Id`),
FOREIGN KEY (`Contact_Id`)
REFERENCES  `Contacted` (`Contact_Id`)
);
CREATE TABLE `HasCorona` (
`Case_Id` int AUTO_INCREMENT,
`Person_Id` numeric NOT NULL,
`Patient_Id` numeric NOT NULL,
`Date` varchar (20) NOT NULL,
`CoronaVariant` varchar (20) NOT NULL,
PRIMARY KEY (`Case_Id`),
FOREIGN KEY (`Person_Id`)REFERENCES `midterm`. `Person` (`Person_Id`),
FOREIGN KEY (`Patient_Id`)
REFERENCES  `Patient` (`Patient_Id`)
);
CREATE TABLE `IsVaccinated` (
`Vaccination_Id` int AUTO_INCREMENT,
`Person_Id` numeric NOT NULL,
`Vaccine_Id` numeric NOT NULL,
`Address_Id` numeric NOT NULL,
`Date` varchar (20) NOT NULL,
PRIMARY KEY (`Vaccination_Id`),
FOREIGN KEY (`Person_Id`)
REFERENCES  `Person` (`Person_Id`),
FOREIGN KEY (`Vaccine_Id`)
REFERENCES  `Vaccine` (`Vaccine_Id`),
FOREIGN KEY (`Address_Id`)
REFERENCES  `Address` (`Address_Id`)
);
CREATE TABLE `CanPublish` (
`Publish_Id` int AUTO_INCREMENT,
`Officer_Id` numeric NOT NULL,
`Report_Id` numeric NOT NULL,
`Channel` varchar (45) NOT NULL,
PRIMARY KEY (`Publish_Id`),
FOREIGN KEY (`Officer_Id`)
REFERENCES  `Officer` (`Officer_Id`),
FOREIGN KEY (`Report_Id`)
REFERENCES `CaseReport` (`Report_Id`)
);
CREATE TABLE `StayingIn` (
`Staying_Id` int AUTO_INCREMENT,
`Room_Id` numeric NOT NULL,
`Patient_Id` numeric NOT NULL,
`StartingDate` varchar (45) NOT NULL,
PRIMARY KEY (`Staying_Id`),
FOREIGN KEY (`Room_Id`)REFERENCES  `HospitalRoom` (`Room_Id`),
FOREIGN KEY (`Patient_Id`)
REFERENCES  `Patient` (`Patient_Id`)
);
```

DML INSERT CODES

```
INSERT INTO address VALUES (1,"Rize","Çayeli");

INSERT INTO address VALUES (2,"Kayseri","Kocasinan");

INSERT INTO address VALUES (3,"Erzurum","Oltu");

INSERT INTO address VALUES (4,"Samsun","Bafra");

INSERT INTO address VALUES (5,"Trabzon","Of");

INSERT INTO address VALUES (6,"Izmir","Bornova");

INSERT INTO address VALUES (7,"Trabzon","Sürmene");

INSERT INTO address VALUES (8,"Ankara","Mamak");

INSERT INTO address VALUES (9,"Kayseri","Talas");

INSERT INTO address VALUES (10,"Kocaeli","Gebze");
INSERT INTO address VALUES (11,"Rize","Kalkandere");
INSERT INTO address VALUES (12,"Ankara","Kızılay");

INSERT INTO person VALUES (1,"Anıl","irfanoğlu","123",20,"negative",1,"no");
INSERT INTO person VALUES (2,"Ali","Yılmaz","456",25,"negative",3,"no");
INSERT INTO person VALUES (3,"Kasım","Yıldız","pass",36,"positive",5,"yes");
INSERT INTO person VALUES (4,"Cemre","Kalender","word",22,"positive",1,"yes");
INSERT INTO person VALUES (5,"Ahmet","Sezer","asdf",44,"negative",4,"no");
INSERT INTO person VALUES (6,"Asım","Ayten","passing",75,"positive",2,"yes");
INSERT INTO person VALUES (7,"Veli","Aydın","as",21,"negative",9,"no");
INSERT INTO person VALUES (8,"Yaman","Hilal","sa",9,"negative",8,"no");
INSERT INTO person VALUES (9,"Merve","Al","123",48,"negative",7,"no");
INSERT INTO person VALUES (10,"Gamze","Beydemir","gfhdgf",32,"negative",6,"no");
INSERT INTO person VALUES (11,"Gizem","Yaman","jhgjg",56,"negative",5,"no");
INSERT INTO person VALUES (12,"Aslı","Kurt","uyrhfghg",27,"negative",4,"no");
INSERT INTO person VALUES (13,"Zeynep","Bilgin","rteytryh",16,"negative",3,"no");
INSERT INTO person VALUES (14,"Bahar","Korkmaz","jjkjhk",61,"negative",2,"no");
INSERT INTO person VALUES (15,"Harun","Peker","lkjhjhgjh",53,"negative",1,"no");
INSERT INTO person VALUES (16,"Barbaros","Demir","safdsgfds",34,"negative",2,"no");
INSERT INTO person VALUES (17,"Ali","Keskin","gkmhgj",44,"negative",3,"no");
INSERT INTO person VALUES (18,"Fatma","Bostan","csadf",78,"negative",4,"no");
INSERT INTO person VALUES (19,"Ziya","Keskin","erefd",18,"negative",5,"no");
INSERT INTO person VALUES (20,"Selenay","Bilgin","fcgdfd",64,"negative",6,"no");
INSERT INTO person VALUES (21,"Fikri","Kaya","asdsdf",29,"negative",7,"no");
INSERT INTO person VALUES (22,"Sadık","Kör","vdfsdf",30,"negative",8,"no");
INSERT INTO person VALUES (23,"Ayten","Berber","atrtd",36,"negative",9,"no");
INSERT INTO person VALUES (24,"Erol","Kısır","eawf",39,"negative",10,"no");
INSERT INTO person VALUES (25,"Eren","Pastırma","ewaf",43,"negative",11,"no");
INSERT INTO person VALUES (26,"Alper","Petek","tgfdv",46,"negative",12,"no");
INSERT INTO person VALUES (27,"Hüsnü","Peker","qwe",48,"negative",11,"no");
INSERT INTO person VALUES (28,"Batuhan","Fadıl","ret",51,"negative",10,"no");
INSERT INTO person VALUES (29,"Yahya","Terzi","fsd",64,"negative",9,"no");
INSERT INTO person VALUES (30,"Polat","Alemdar","sdfs",68,"negative",8,"no");
INSERT INTO person VALUES (31,"Memati","Bas","vbdcsds",72,"negative",7,"no");
INSERT INTO person VALUES (32,"Mümin","Ayten","vcdssd",17,"negative",6,"no");
INSERT INTO person VALUES (33,"Osman","Sofu","wead",21,"negative",5,"no");
INSERT INTO person VALUES (34,"Faik","Bulut","cvsdsc",36,"negative",4,"no");
INSERT INTO person VALUES (35,"Fazıl","Demir","fdsda",41,"negative",3,"no");
INSERT INTO person VALUES (36,"Ferhat","Ak","sadfsa",55,"positive",2,"yes");
INSERT INTO person VALUES (37,"Aybala","Kara","dfssd",65,"positive",1,"yes");
INSERT INTO person VALUES (38,"Asu","Akay","pefasdf",52,"positive",2,"yes");
INSERT INTO person VALUES (39,"Ayfer","Deli","dfsd",6,"positive",3,"yes");
INSERT INTO person VALUES (40,"Akif","Azman","eafs",48,"positive",4,"yes");
INSERT INTO person VALUES (41,"Numan","Uzun","cvvsd",50,"positive",5,"yes");
INSERT INTO person VALUES (42,"Beyza","Demirci","dsafdsc",27,"positive",6,"yes");
INSERT INTO person VALUES (43,"Kaan","Ada","passing",19,"negative",7,"no");

INSERT INTO doctor VALUES (5,"Cardiolog");
INSERT INTO doctor VALUES (20,"internal medicine");
INSERT INTO doctor VALUES (21,"neurology");
INSERT INTO doctor VALUES (22,"internal medicine");
INSERT INTO doctor VALUES (23,"Cardiolog");
INSERT INTO doctor VALUES (24,"internal medicine");
INSERT INTO doctor VALUES (25,"Cardiolog");
INSERT INTO doctor VALUES (26,"general surgeon");
INSERT INTO doctor VALUES (27,"Cardiolog");
INSERT INTO doctor VALUES (28,"chest diseases");

INSERT INTO officer VALUES (1,"Healthcare HQ","admin","Head of Department");
INSERT INTO officer VALUES (10,"Healthcare HQ","admin1","Worker");
INSERT INTO officer VALUES (11,"Healthcare HQ","admin2","Worker");
INSERT INTO officer VALUES (12,"Healthcare HQ","admin3","Worker");
INSERT INTO officer VALUES (13,"Healthcare HQ","admin4","Worker");
INSERT INTO officer VALUES (14,"Healthcare HQ","admin5","Worker");
INSERT INTO officer VALUES (15,"Healthcare HQ","admin6","Worker");
INSERT INTO officer VALUES (16,"Healthcare HQ","admin7","Worker");
INSERT INTO officer VALUES (17,"Healthcare HQ","admin8","Worker");
INSERT INTO officer VALUES (18,"Healthcare HQ","admin9","Worker");

INSERT INTO patient VALUES (3,"alive",5);
INSERT INTO patient VALUES (4,"alive",5);
INSERT INTO patient VALUES (6,"alive",5);
INSERT INTO patient VALUES (36,"alive",25);
INSERT INTO patient VALUES (37,"alive",28);
INSERT INTO patient VALUES (38,"alive",24);
INSERT INTO patient VALUES (39,"alive",22);
INSERT INTO patient VALUES (40,"alive",28);
INSERT INTO patient VALUES (41,"alive",26);
INSERT INTO patient VALUES (42,"alive",21);



INSERT INTO hascorona (Person_Id,Patient_Id,Date,CoronaVariant) VALUES
(3,3,"08/05/2021","england");
INSERT INTO hascorona (Person_Id,Patient_Id,Date,CoronaVariant) VALUES
(4,4,"06/05/2021","original");
INSERT INTO hascorona (Person_Id,Patient_Id,Date,CoronaVariant) VALUES
(6,6,"10/05/2021","england");
INSERT INTO hascorona (Person_Id,Patient_Id,Date,CoronaVariant) VALUES
(36,36,"10/05/2021","brazil");
INSERT INTO hascorona (Person_Id,Patient_Id,Date,CoronaVariant) VALUES
(37,37,"10/05/2021","england");
INSERT INTO hascorona (Person_Id,Patient_Id,Date,CoronaVariant) VALUES
(38,38,"10/05/2021","original");
INSERT INTO hascorona (Person_Id,Patient_Id,Date,CoronaVariant) VALUES
(39,39,"10/05/2021","original");
INSERT INTO hascorona (Person_Id,Patient_Id,Date,CoronaVariant) VALUES
(40,40,"10/05/2021","brazil");
INSERT INTO hascorona (Person_Id,Patient_Id,Date,CoronaVariant) VALUES
```

```
(41,41,"10/05/2021","england");
INSERT INTO hascorona (Person_Id,Patient_Id,Date,CoronaVariant) VALUES
(42,42,"10/05/2021","brazil");

INSERT INTO hospitalroom VALUES (1,1,"intensive care");
INSERT INTO hospitalroom VALUES (2,1,"normal unit");
INSERT INTO hospitalroom VALUES (3,1,"intensive care");
INSERT INTO hospitalroom VALUES (4,1,"intensive care");
INSERT INTO hospitalroom VALUES (5,1,"intensive care");
INSERT INTO hospitalroom VALUES (6,2,"normal unit");
INSERT INTO hospitalroom VALUES (7,1,"intensive care");
INSERT INTO hospitalroom VALUES (8,2,"intensive care");
INSERT INTO hospitalroom VALUES (9,2,"intensive care");
INSERT INTO hospitalroom VALUES (10,2,"intensive care");

INSERT INTO stayingin (Room_Id,Patient_Id,StartingDate) VALUES (1,3,"05/05/2021");
INSERT INTO stayingin (Room_Id,Patient_Id,StartingDate) VALUES (2,4,"04/05/2021");
INSERT INTO stayingin (Room_Id,Patient_Id,StartingDate) VALUES (3,6,"05/05/2021");
INSERT INTO stayingin (Room_Id,Patient_Id,StartingDate) VALUES (10,36,"05/05/2021");
INSERT INTO stayingin (Room_Id,Patient_Id,StartingDate) VALUES (9,37,"05/05/2021");
INSERT INTO stayingin (Room_Id,Patient_Id,StartingDate) VALUES (8,38,"05/05/2021");
INSERT INTO stayingin (Room_Id,Patient_Id,StartingDate) VALUES (7,39,"05/05/2021");
INSERT INTO stayingin (Room_Id,Patient_Id,StartingDate) VALUES (6,40,"05/05/2021");

INSERT INTO vaccine VALUES (1,100,"Sinovac","inactive");
INSERT INTO vaccine VALUES (2,100,"Biontech","vector");
INSERT INTO vaccine VALUES (3,100,"Astrazeneca","inactive");
INSERT INTO vaccine VALUES (4,100,"Moderna","inactive");
INSERT INTO vaccine VALUES (5,100,"Johnson","mRNA");
INSERT INTO vaccine VALUES (6,100,"Novavax","inactive");
INSERT INTO vaccine VALUES (7,100,"Fixme","inactive");
INSERT INTO vaccine VALUES (8,100,"Viral","vector");
INSERT INTO vaccine VALUES (9,100,"Sputnik","inactive");
INSERT INTO vaccine VALUES (10,100,"Covaxin","inactive");

INSERT INTO isvaccinated (Person_Id,Vaccine_Id,Address_Id,Date) VALUES (3,1,4,"08/05/2021");
INSERT INTO isvaccinated (Person_Id,Vaccine_Id,Address_Id,Date) VALUES (4,3,4,"08/05/2021");
INSERT INTO isvaccinated (Person_Id,Vaccine_Id,Address_Id,Date) VALUES
(11,2,10,"08/05/2021");
INSERT INTO isvaccinated (Person_Id,Vaccine_Id,Address_Id,Date) VALUES (13,6,8,"08/05/2021");

INSERT INTO isvaccinated (Person_Id,Vaccine_Id,Address_Id,Date) VALUES (15,8,9,"08/05/2021");
INSERT INTO isvaccinated (Person_Id,Vaccine_Id,Address_Id,Date) VALUES (38,5,6,"08/05/2021");
INSERT INTO isvaccinated (Person_Id,Vaccine_Id,Address_Id,Date) VALUES (39,7,4,"08/05/2021");
INSERT INTO isvaccinated (Person_Id,Vaccine_Id,Address_Id,Date) VALUES (40,9,3,"08/05/2021");
INSERT INTO isvaccinated (Person_Id,Vaccine_Id,Address_Id,Date) VALUES (27,2,7,"08/05/2021");

INSERT INTO contacted VALUES (2,"negative");
INSERT INTO contacted VALUES (10,"negative");
INSERT INTO contacted VALUES (11,"negative");
INSERT INTO contacted VALUES (12,"negative");
INSERT INTO contacted VALUES (13,"negative");
INSERT INTO contacted VALUES (14,"negative");
INSERT INTO contacted VALUES (15,"negative");
INSERT INTO contacted VALUES (20,"negative");
INSERT INTO contacted VALUES (21,"negative");
INSERT INTO contacted VALUES (22,"negative");

INSERT INTO contactedwith (Patient_Id,Contact_Id,Date) VALUES (3,2,"08/05/2021");
INSERT INTO contactedwith (Patient_Id,Contact_Id,Date) VALUES (38,10,"08/05/2021");
INSERT INTO contactedwith (Patient_Id,Contact_Id,Date) VALUES (38,11,"08/05/2021");
INSERT INTO contactedwith (Patient_Id,Contact_Id,Date) VALUES (38,12,"08/05/2021");
INSERT INTO contactedwith (Patient_Id,Contact_Id,Date) VALUES (40,13,"08/05/2021");
INSERT INTO contactedwith (Patient_Id,Contact_Id,Date) VALUES (41,14,"08/05/2021");
INSERT INTO contactedwith (Patient_Id,Contact_Id,Date) VALUES (41,15,"08/05/2021");
INSERT INTO contactedwith (Patient_Id,Contact_Id,Date) VALUES (41,20,"08/05/2021");
INSERT INTO contactedwith (Patient_Id,Contact_Id,Date) VALUES (4,21,"08/05/2021");
INSERT INTO contactedwith (Patient_Id,Contact_Id,Date) VALUES (4,22,"08/05/2021");

INSERT INTO casereport VALUES (1,2,2,3);
INSERT INTO casereport VALUES (2,4,6,8);
INSERT INTO casereport VALUES (3,10,12,14);
INSERT INTO casereport VALUES (4,16,18,20);
INSERT INTO casereport VALUES (5,22,24,26);
INSERT INTO casereport VALUES (6,27,27,30);
INSERT INTO casereport VALUES (7,28,28,32);
INSERT INTO casereport VALUES (8,34,30,33);
INSERT INTO casereport VALUES (9,36,30,34);
INSERT INTO casereport VALUES (10,40,30,35);

INSERT INTO canpublish (Officer_Id,Report_Id,Channel) VALUES (1,1,"Television");
INSERT INTO canpublish (Officer_Id,Report_Id,Channel) VALUES (18,2,"Magazine");
INSERT INTO canpublish (Officer_Id,Report_Id,Channel) VALUES (14,3,"Television");
INSERT INTO canpublish (Officer_Id,Report_Id,Channel) VALUES (12,4,"Magazine");
INSERT INTO canpublish (Officer_Id,Report_Id,Channel) VALUES (10,5,"Magazine");
INSERT INTO canpublish (Officer_Id,Report_Id,Channel) VALUES (13,6,"Television");
INSERT INTO canpublish (Officer_Id,Report_Id,Channel) VALUES (16,7,"Television");
INSERT INTO canpublish (Officer_Id,Report_Id,Channel) VALUES (1,8,"Television");
INSERT INTO canpublish (Officer_Id,Report_Id,Channel) VALUES (15,9,"Television");
INSERT INTO canpublish (Officer_Id,Report_Id,Channel) VALUES (17,10,"Television");
```

SQL QUESTIONS

1. **Who are persons over the age of 20 who have not been vaccinated**?

Ans: if we look at if a person in isvaccinated table and his age over than 20,we have the result

```
select Name,Surname,age
from person
where  age>20 and (Person_Id) not in
           (select Person_Id
   from isvaccinated
   );
```

Π{Name,Surname,Age }(σ{age>20 ∧ Person_Id —(Π{Person_Id}(σ(isvaccinated))) }(Person))

2. **How many empty beds are in which rooms in the hospital?**

Ans: if we look at all beds which are accomadated and if we took difference, we arrive answer

SQL:

```
select Room_Id,BedNumber-1 as avaliableRoomNumber ,RoomType
from hospitalroom
where  (Room_Id) in
           (select Room_Id
   from stayingin
   ) union
   select Room_Id,BedNumber as avaliableRoomNumber,RoomType
from hospitalroom
where  (Room_Id) not in
           (select Room_Id
   from stayingin
   );
```

RA:

[Π{Room_Id,BedNumber-1 }(σ{Room_Id ∩(Π{Room_Id}(σ(stayingin))) }(hospitalroom))]∪[Π{Room_Id,BedNumber}(σ{Room_Id —(Π{Room_Id}(σ(stayingin))) }(hospitalroom))]

3. **Who are the people with the mutated virus?**

Ans: mutated virus is not the original one so in that case if we check for that, we arrive result

SQL:

```
select Name,Surname,CoronaVariant from hascorona inner join person on
person.Person_Id=hascorona.Person_Id where CoronaVariant !="original";
```

RA:

Π{Name,Surname,CoronaVariant }(σ{CoronaVariant !="original"}(hascorona $\bowtie_{person.Person\_Id=hascorona.Person\_Id}$ person))

4. **What are the names and conditions of people who are not in the hospital although they have corona?**

Ans:we will choose patients which are not in hospital.

SQL:

```
select Name,Surname,age,Status
from patient inner join person on patient.Patient_Id=person.Person_Id
where (Patient_Id) not in
            (select Patient_Id
   from stayingin
   );
;
```

RA:
Π{Name,Surname,Age,Status }(σ{ Patient_Id —(Π{Patient_Id}(σ(stayingin))) }(Patient ⋈ person))

5. **What is the ratio of people who have been quarantined so far to general persons?**

Ans:we will divide the number of people who in quarantine to all people

SQL:

```
SELECT
   (CAST((SELECT COUNT(*) FROM person WHERE Quarantine_Status='yes') AS FLOAT)/
    CAST((SELECT COUNT(*) FROM person) AS FLOAT)*100)
   AS QuarantineRatio;
```

RA:
σ{CAST(Π{count(*) }(σ{Quarantine_Status='yes' }(person))) }/σ{CAST(Π{count(*) }(σ(person))) }

6. **Who are the people who have corona but are still not vaccinated and what are their conditions?**

Ans:we will look the people who have corona but couldn't have vaccine.

SQL:

```
select Name,Surname,age,Status
from patient inner join person on patient.Patient_Id=person.Person_Id
where (Person_Id) not in
            (select Person_Id
   from isvaccinated
   );
;
```

RA:
Π{Name,Surname,Age,Status }(σ{ Person_Id —(Π{Person_Id}(σ(isvaccinated))) }(Patient ⋈ person))

7. **How many people are in the system in total, how many of them are corona, how many of them are vaccinated and finally, how many hospital rooms are used?**

Ans:we will gather total information

SQL:

```
SELECT count(*) as TotalPerson,
   (SELECT Count(*) FROM isvaccinated) as TotalVaccinated, (SELECT Count(*) FROM hascorona)
as TotalCorona, (SELECT Count(*) FROM stayingin) as TotalUsedHospitalRoom
   FROM person
  ;
```

RA:
Π{Π{count(*)}(σ(person)), Π{count(*)}(σ(isvaccinated)), Π{count(*)}(σ(hacorona)),
Π{count(*)}(σ(stayingin)) }(Person)

8. how many people ill in rize city?

Ans: we will gather data to specific area

SQL:

```
select City,count(*) as NumberOfIllPeople from person inner join address on person.Address_Id
=address.Address_Id where city="Rize" and (Person_Id) in (select Person_Id from hascorona);
```

RA:
Π{City,count(*) }(σ{city="Rize" ∧ Person_Id ∩(Π{Person_Id}(σ(hacorona))) }(Address ⋈ Person))

9. how many reports did the chosen person published until now and what is the name, surname and department of this person?

Ans: we will count by looking to canpublis table

SQL:

```
select name,surname,OfficeName,OfficerRank,(SELECT Count(*) FROM canpublish inner join
officer inner join person on person.Person_Id=officer.Officer_Id and
officer.Officer_Id=canpublish.Officer_Id where Name="anıl") as SharedReports from officer inner
join person  on person.Person_Id = officer.Officer_Id  where Name="anıl";
```

RA:
Π{name,surname,OfficeName,OfficerRank}(Π{count(*)}(σ{ Name="anıl"}(canpublish ⋈ officer ⋈ person)))

10. how many people didn't get corona even they contacted with ill people?

Ans: we will count negative results from contacts

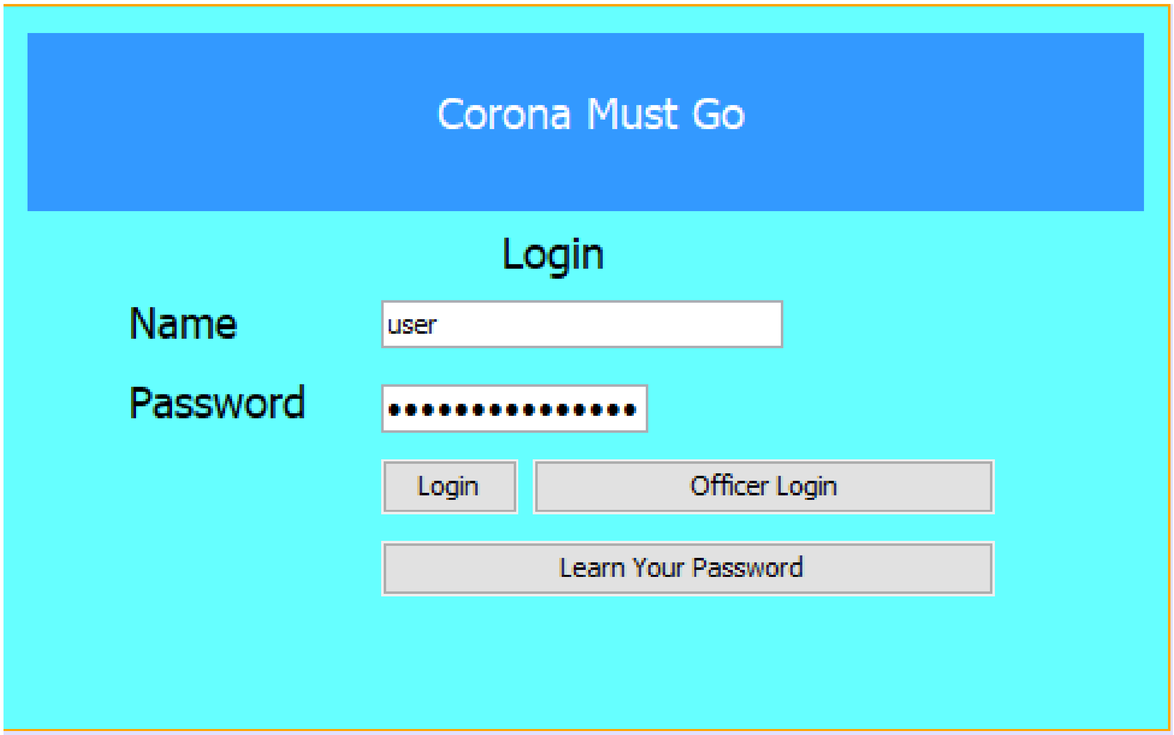
SQL:

select count(*) as DidntGet from contacted where TestResult="negative";

RA:

Π{count(*)}(σ{TestResult="negative"}(contacted))

INTERFACE PART



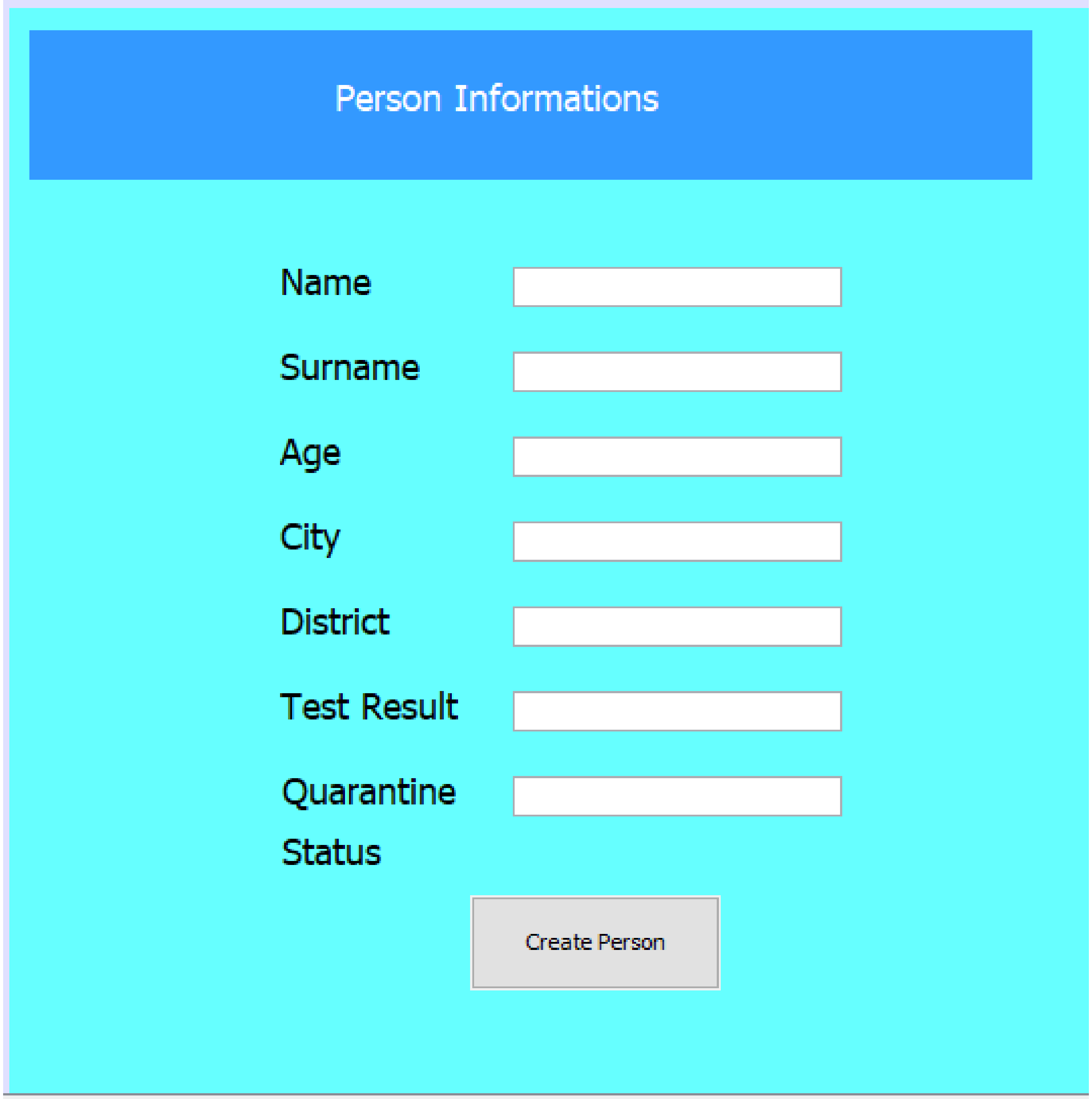
Corona Must Go

Login

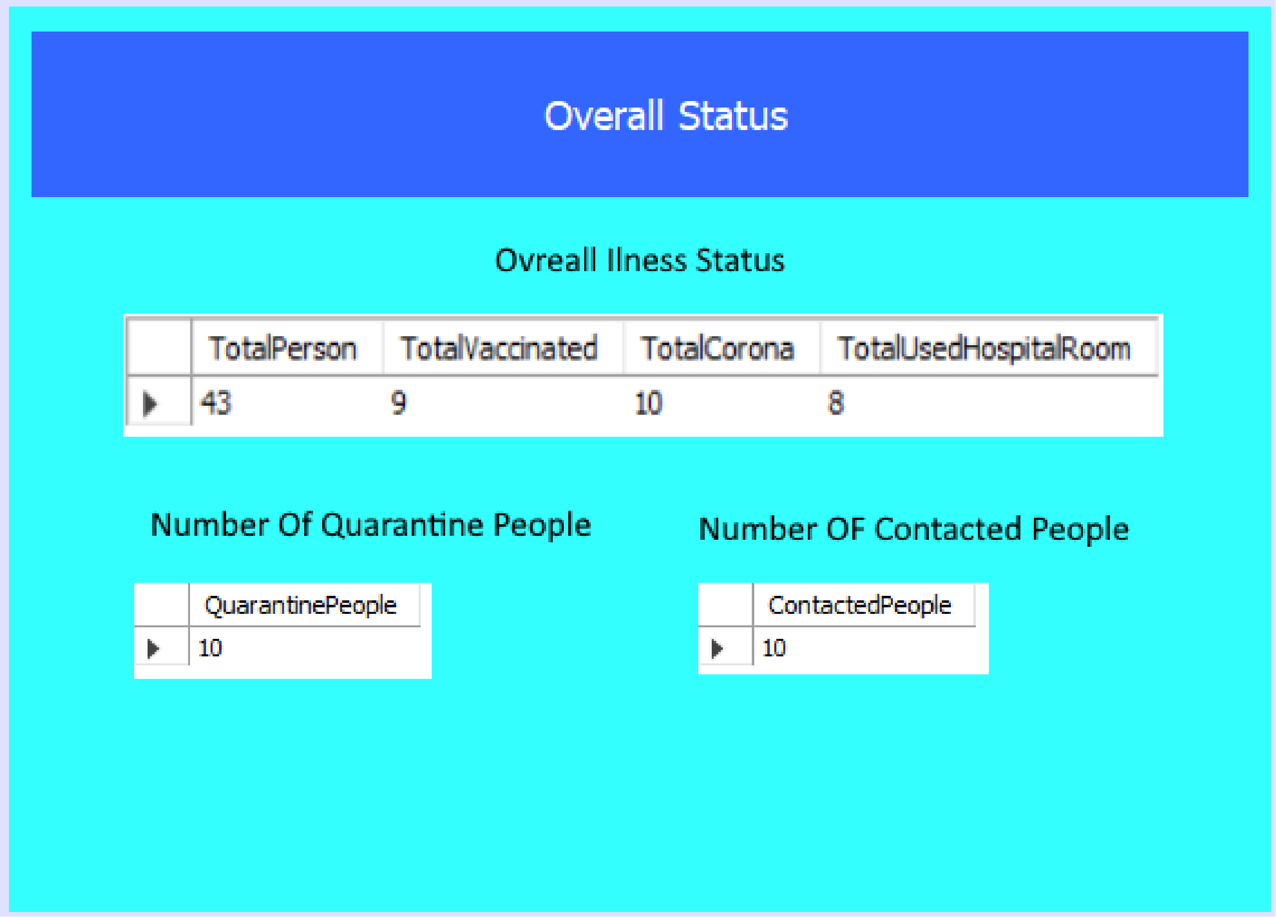
Name: user

Password: ●●●●●●●●●●●●●●●

This is the login screen. In our system person can create its own person account but in order to get an officer account, person should contact with the medicine ministry to get officer account to access all data, except that one, a person can enter to system and can see overall status.



In this part person can create his account with filling these informations which are listed in photo.



Overall Status

Ovreall Ilness Status

	TotalPerson	TotalVaccinated	TotalCorona	TotalUsedHospitalRoom
▶	43	9	10	8

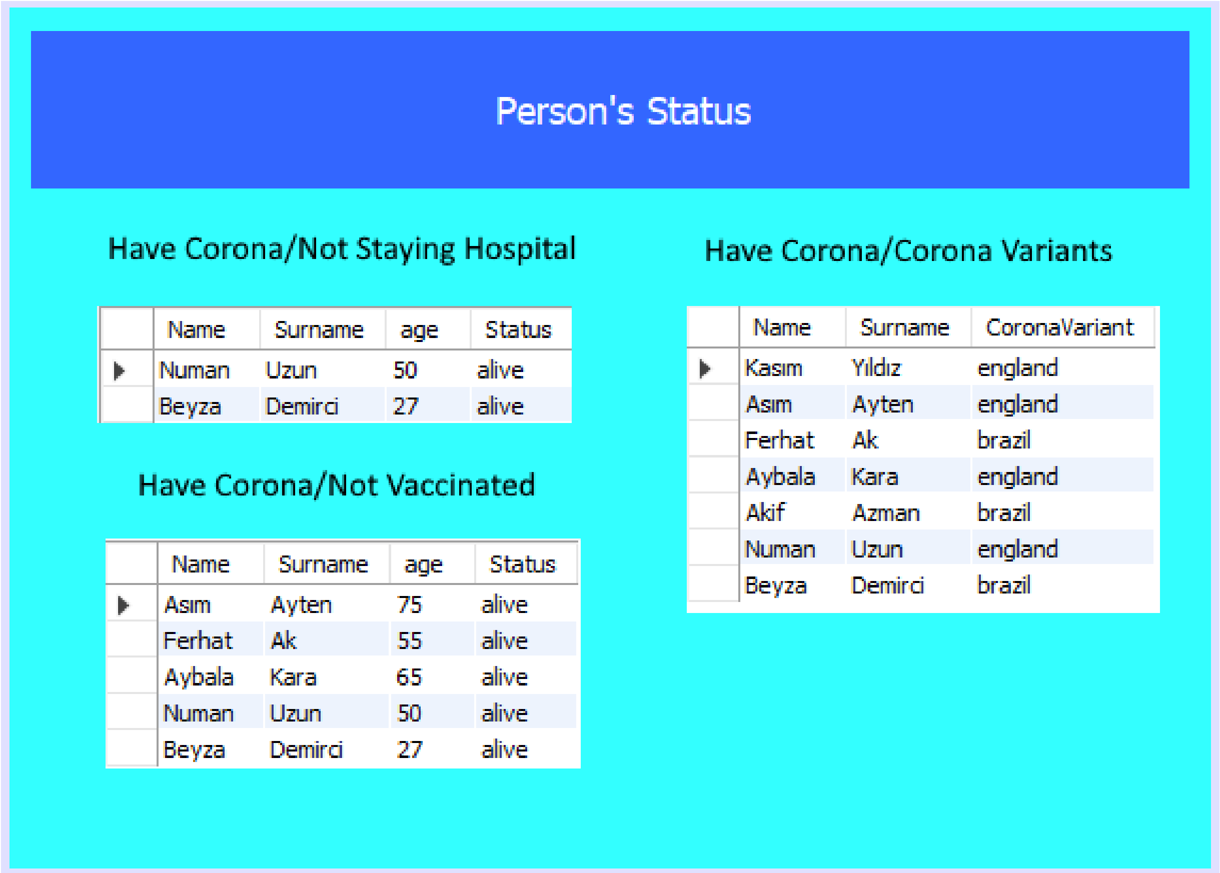
Number Of Quarantine People

	QuarantinePeople
▶	10

Number OF Contacted People

	ContactedPeople
▶	10

This screen is prvided to show everyone the overall status about coronavirus disease. This page accessible by all persons accounts and they can see overall status about disease.



Person's Status

Have Corona/Not Staying Hospital

	Name	Surname	age	Status
▶	Numan	Uzun	50	alive
	Beyza	Demirci	27	alive

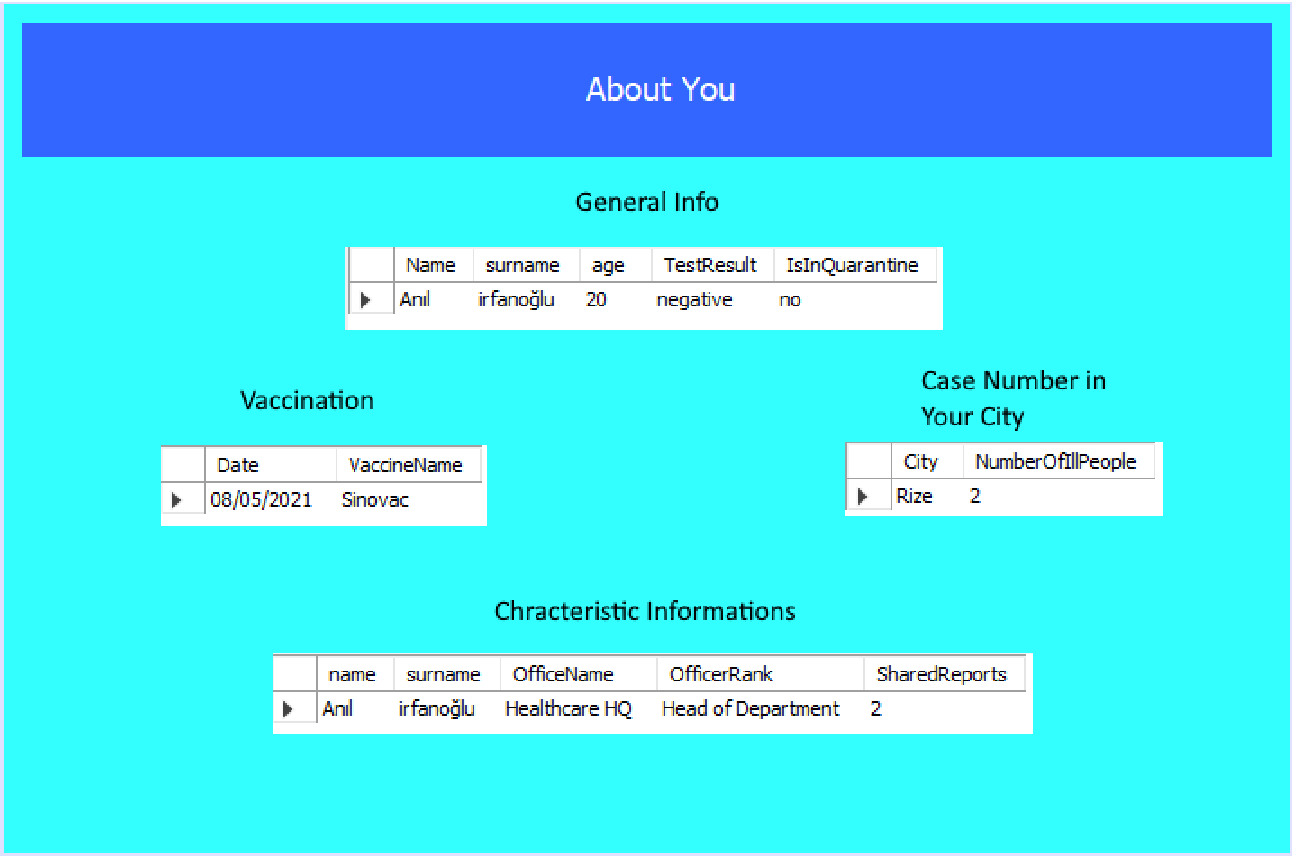
Have Corona/Not Vaccinated

	Name	Surname	age	Status
▶	Asım	Ayten	75	alive
	Ferhat	Ak	55	alive
	Aybala	Kara	65	alive
	Numan	Uzun	50	alive
	Beyza	Demirci	27	alive

Have Corona/Corona Variants

	Name	Surname	CoronaVariant
▶	Kasım	Yıldız	england
	Asım	Ayten	england
	Ferhat	Ak	brazil
	Aybala	Kara	england
	Akif	Azman	brazil
	Numan	Uzun	england
	Beyza	Demirci	brazil

This page is for the officers. Only officers can access that page and they will monitor about person’s status in that page and they will take actions according to these datas. Actually there are very important tables which states some emergency problems and they will take action immediately.



General Info

	Name	surname	age	TestResult	IsInQuarantine
▶	Anıl	irfanoğlu	20	negative	no

Vaccination

	Date	VaccineName
▶	08/05/2021	Sinovac

Case Number in Your City

	City	NumberOfIllPeople
▶	Rize	2

Chracteristic Informations

	name	surname	OfficeName	OfficerRank	SharedReports
▶	Anıl	irfanoğlu	Healthcare HQ	Head of Department	2

In this page, a person can see information about itself and about his status. Whoever you are, you can see you general informations and also you can look at your characteristic informations for example if you are an officer, you can check your officer informations and you can look at how many magazine did you publish , if you doctor, you could see the number of patients that you look and also you can access the informations about them.