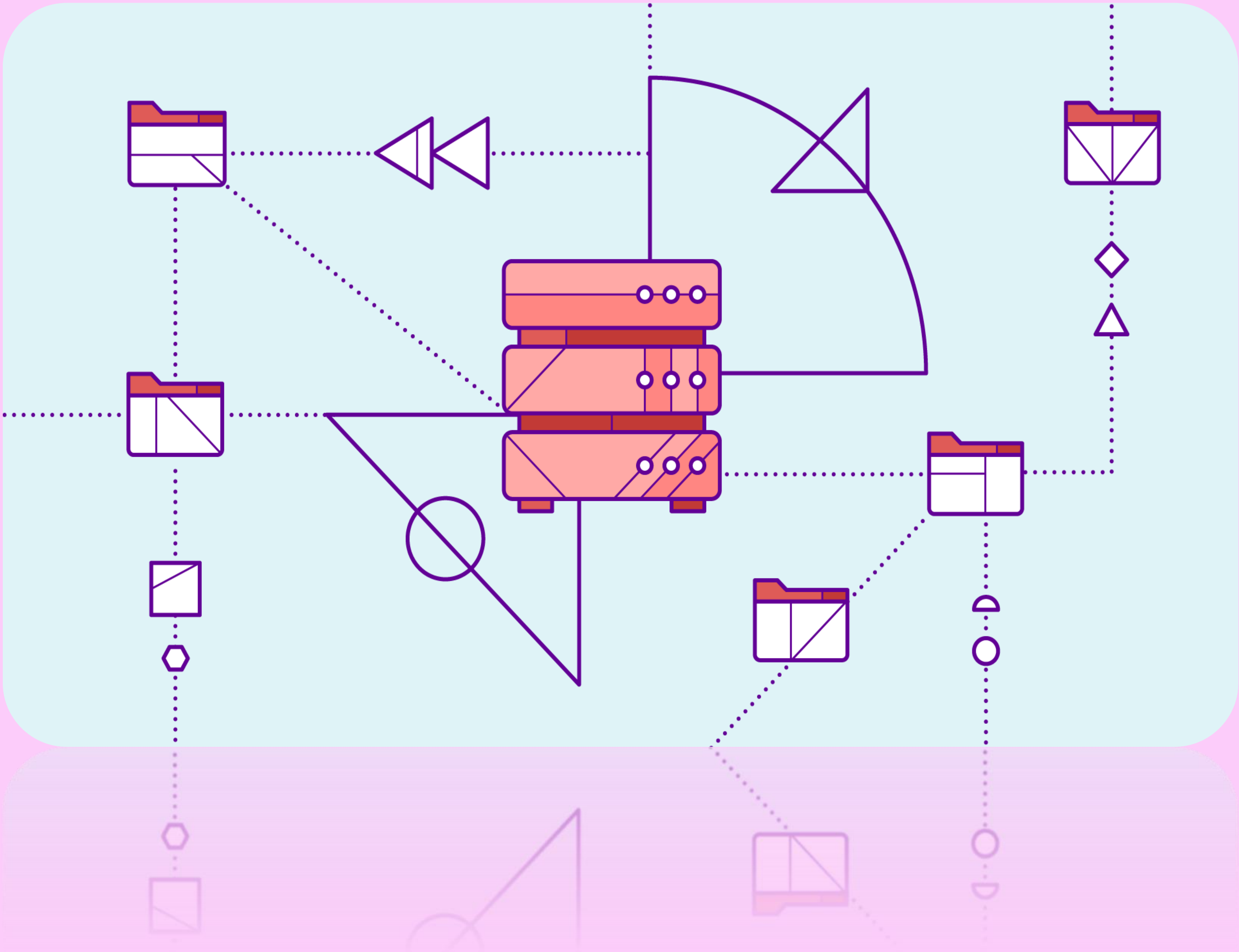


DATABASE

Final Project

Niloufar Babaahmadi

610398103

SCHOOL OF MATHEMATICS,

STATISTICS AND COMPUTER SCIENCE

The environment chosen for this project is the hospital. In order to design a practical management system, accurate data and real-life problems of a hospital are needed. Therefore, we need to research and gather data from a hospital. In order to do that, I interviewed some hospital doctors, a nurse and a secretary. Then I began to imagine different scenarios that can occur and organized them to form a system description. By studying my system description, I was able to determine different entities and their attributes and then according to these entities I formed relations between them. To take the next step, I needed to draw the EER diagram of the recognized entities, attributes and relations. Types of relationships and entity participation was decided next. After finding the tables in the EER diagram, foreign and primary keys were recognized. Now that the tables are found, we begin normalizing. Then, five views and ten queries were considered. The Project used mysql as the coding tool, and tables were built in that environment. The data was inserted and the queries were tested on them. In the final step, I defined a pair of functions, Triggers and stored procedure.

1) The hospital environment

The hospital contains different departments for different purposes; each department contains Nursing unit, Doctors' offices, rooms for patients, housekeeping unit and security office.

There are some special departments as well such as Dietary facilities, Linen services, Engineering services and equipment areas, Operation services facilities (e.g., operating rooms) and Diagnostic facilities (e.g., radiographic rooms and the laboratory) that doesn't contain rooms for patients to be hospitalized.

For simplicity, these departments were grouped together.

Various jobs exist in hospitals besides the healthcare related ones; people with jobs other can healthcare services were grouped together as employees.

We will review the management system of hospitals and the steps patients go through plus the scenarios that can occur in the next section.

2) Environment description

I. The hospital has different departments with a specific name. The department is consistent of Physicians, Nurses and employees and is controlled by the head of each group. Each department has rooms for the patients to stay in.

II. The patients are recognized by their unique National ID and their full name, gender, address, phone number and date of birth is recorded and their age will get calculated. Also, the hospital will record a brief medical background which includes the patient's weight and height to calculate their BMI, blood type and any underlying medical conditions and finally, if they own an insurance, their insurance ID is recorded. Each patient has a secretary register them and treatments that treat them.

III. The employees are recognized by their unique National ID and their full name is recorded along with their phone number, address and the year they started working in the hospital. The monthly salary of the employees is kept track by recording their normal pay rate which depends on their position plus the additional hours of work minus the absence count. The employees belong to a certain department and/or are in charge of one. They are divided into different groups such as security, secretary (in charge of registration), drivers, engineers, housekeeping, chiefs and office and management. Each group has a different pay rate.

IV. The physicians are recognized by their unique medical ID and their full name is recorded along with their phone number, address and the year they started working in the hospital. The monthly salary of the physicians is kept track by recording their normal pay rate which depends on their position plus the additional hours of work plus their share of the procedure's cost minus the absence count. The physicians belong to a certain department and/or are in charge of one. Their main task is to examine and treat the patients.

V. The nurses and technicians are recognized by their unique National ID and their full name is recorded along with their phone number, address and the year they started working in the hospital and their current position. Their monthly salary is kept track by recording their normal pay rate which depends on their position plus the additional hours of work minus the absence count. They belong to a certain department and/or are in charge of one. Their main task is to assist in procedures and aid the patients.

VI. The pay rate is a monthly salary given to different staff of the hospital considering their position.

VII. Each patient has to undergo (a) treatment(s) which is either a prescribed treatment and/or involves a certain procedure or neither of them. The treatment is recognized by the patient's ID and the date of treatment. The cost of treatment depends on the kind of treatment taken and is calculated by the formula: final cost = Checkup cost + prescription cost + undergo cost. It's recognized by physician's ID and patient's ID and the date it was suggested in.

VIII. Prescription is a set of prescribed medication by the physician which includes an explanation. We can have more than one medication prescribed and the final cost is the sum of medications' cost. It's recognized by physician's ID and patient's ID and the date it was suggested in.

IX. Medication is recognized by its code and also records the name of the medication, brand, description and the cost.

X. Undergo is a treatment process by the physician that is recognized by physician's ID and patient's ID and the date it was suggested in. It requires the aid of nurses and technicians. It includes different procedures and/or patients have to stay in hospital for a period of time. Its cost is calculated by the sum of stay's cost and procedure's cost.

XI**.** Procedure is recognized by its unique code. It has a name and a cost that is shared between the hospital and the physician.

XII. Room belongs to a department and is recognized by the Room ID. It has a Boolean called available that shows the availability of the room and it has different Types (VIP, normal, special needs) that determines the final cost.

3) Entities and attributes

I. Entity: Department

Attributes: Name

II. Entity: Patients

Attributes: Full Name, National ID, Address, Phone Number, Insurance ID, Gender, Birth Date, Age, Weight, Height, BMI, Blood Type, Underlying Medical Condition

III. Entity: Employees

Attributes: Full Name, National ID, Address, Phone Number, Working Since, Position, Additional Hours, Absence Count, Salary

IV. Entity: Physician

Attributes: Full Name, Medical ID, Address, Phone Number, Working Since, Position, Additional Hours, Absence Count, Salary

V. Entity: Nurses and Technicians

Attributes: Full Name, National ID, Address, Phone Number, Working Since, Position, Additional Hours, Absence Count, Salary

VI. Entity: Pay Rate

Attributes: Position, Pay

VII. Entity: Treatment

Attributes: Treatment ID, Date, Cost

VIII. As mentioned before, treatment falls into two categories of prescription and undergoing a certain procedure. Since these two categories have few different attributes, we will consider two different entities for them. Please take note that they can overlap; meaning a patient can undergo both of them.

Entity: Prescription

Attributes: Medications IDs, Explanation, Cost

IX. Entity: Medication

Attributes: Name, Code, Brand, Description, Cost

X. Entity: Undergo Treatment

Attributes: Cost

XI. Entity: Procedure

Attributes: Cost, Code, Name, Physician’s share

XII. Entity: Room

Attributes: Room Type, Room ID, Available, Cost

4) Relations

I. Department and Employees: (1:N)

a. Employees belong to a certain Department.

II. Department and Nurses and Technicians:

a. Nurses and Technicians belong to a certain Department
b. Nurses and Technicians can be in charge of some Departments

III. Department and Physicians: (1:K)

a. Physicians belong to a certain Department
b. Physicians can be in charge of some Departments

IV. Department and Room: Rooms belong to a certain Department.(1:M)

V. Pay Rate and Employees: Employees have a certain Pay Rate.(N:P)(employees should have a pay rate.)

VI. Pay Rate and Nurses and Technicians: Nurses and Technicians have a certain Pay Rate.(N:K)(nurses should have a pay rate.)

VII. Pay Rate and Physicians: Physicians have a certain Pay Rate.(M:N) physicians should have a pay rate.)

VIII. Employees and Patients: Employees register the patients.(N:M)(every patient should have an employee register them.)

IX. Treatment and Patients: Treatment treats the Patient.(M:N)(each patient should have a treatment.)

X. Treatment and Physicians: Physician suggests a Treatment.(M:N)(each treatment has a physician who does it.)

XI. Prescription and Medication: Prescription cites Medications.(N:M)(each prescription should have a medication.)

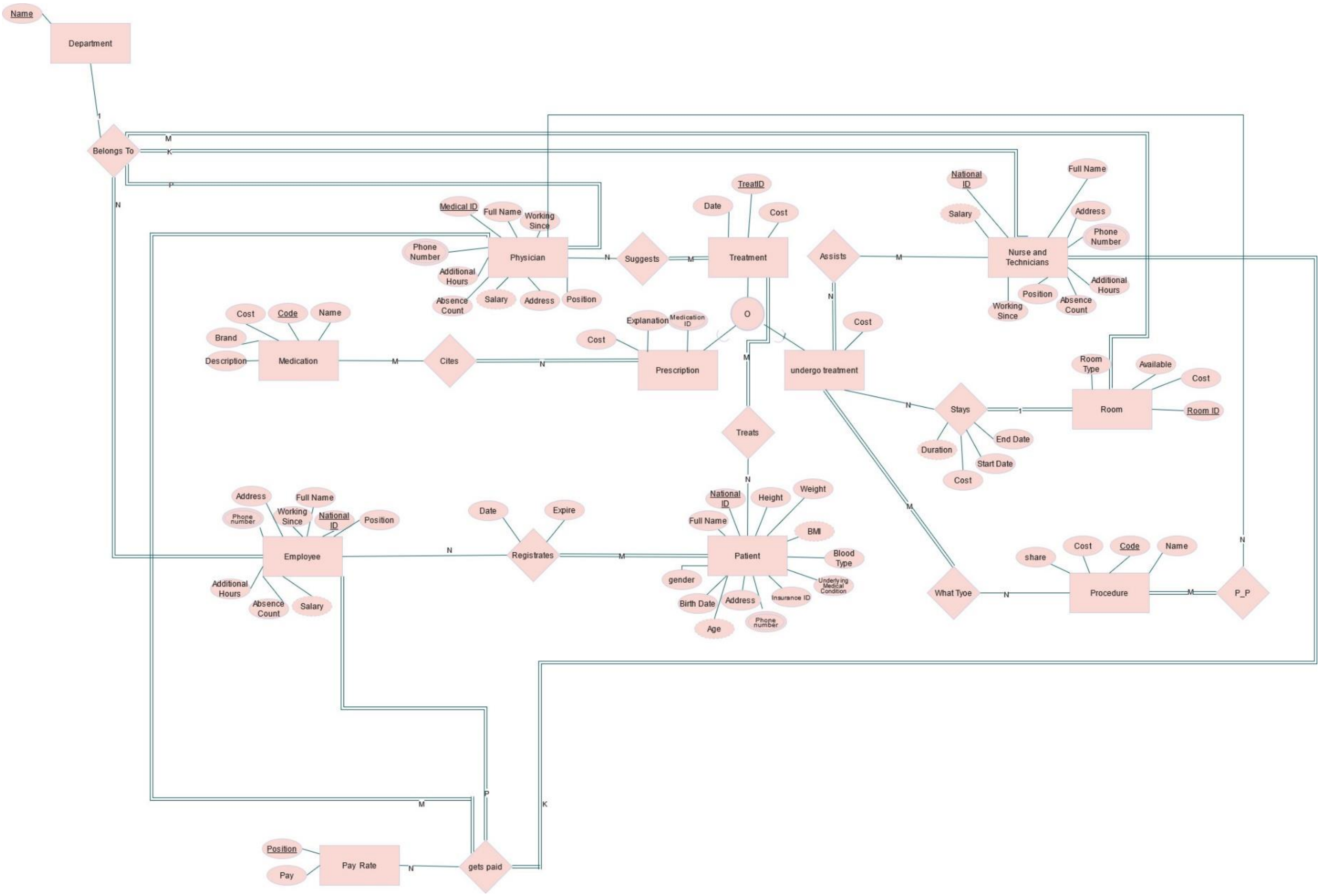
XII. Undergo Treatment and Nurses and Technicians: Nurses and Technicians assist in an Undergo process.(N:M)(in each Undergo treatment there needs to be an assist.)

XIII. Undergo Treatment and Rooms: Patients might stay in a room after Undergoing a Treatment.(N:1)(if you stay in a room, it means that you have had an undergo treatment.)

XIV. Undergo Treatment and Procedure: Patients undergo a certain Procedure.(M:N)(each undergo treatment should be due to a procedure.)

XV. Procedure and Physician: a share of the Procedure money belongs to the Physician.(M:N)(each procedure should be done by a physician.)

6) The EER diagram



6) EER diagram’s tables and foreign and primary keys

I. Department (Name, MedicalID, NationalID_Nurse, NationalID_Employee)

II. Nurses_Technicians (NationalID, Full_Name, Address, Phone_Number, Working_Since, Position, Additional_Hours, Absence_Count, Salary, Department)

III. Physicians (Full_Name, MedicalID, Address, Phone_Number, Working_Since, Position, Additional_Hours, Absence_Count, Salary, Department)

IV. Employees (NationalID, Full_Name, Address, Phone_Number, Working_Since, Position, Additional_Hours, Absence_Count, Salary, Department)

V. Pay_Rate (Position, Pay)

VI. Gets_Paid (Position, MedicalID, NationalID_Nurse, NationalID_Employee)

VII. Registers (NationalID_Patient, NationalID_Employee, Date, Expires)

VIII. Treatment (TreatID, Date, Cost)

IX. Suggests (MedicalID, TreatID)

X. Prescription (TreatID, MedID, Explanation, Cost)

XI. Medication (Name, Code, Brand, Description, Cost)

XII. Cite (Code, TreatID)

XIII. Undergo_Treatment (TreatID, Cost)

XIV. Room (Room_Type, RoomID, Available, Cost, Department)

XV. Assist (NationalID_Nurse, TreatID)

XVI. Stay (TreatID, RoomID, Start_Date, End_Date, Duration)

XVII. Procedure (Cost, Code, Name, Physician_Share)

XVIII. P_P (TreatID, Code, MedicalID)

XIV. Patient (Full_Name, NationalID_Patient, Address, Phone_Number, InsuranceID, Gender, Birth_Date, Age, Weight, Height, BMI, Blood_Type, Underlying_Medical_Condition)

XV. Treats (NationalID_Patient, TreatID)

XVI. Type (TreatID, Code)

6) Normalizing the Tables

I. Department (Name)

- It is in 1NF because attributes do not contain multiple values.
- The primary key, Name, contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.

- There is one candidate key so, this table is in BCNF.

II. Nurses_Technicians (NationalID, Full_Name, Address, Phone_Number, Working_Since, Position, Additional_Hours, Absence_Count, Salary, Department)

- It is not in 1NF because Phone_Number attribute contains multiple values. So, we must fix it:

Nurses_Technicians (NationalID, Full_Name, Address, Phone_Number1, Phone_Number2, Working_Since, Position, Additional_Hours, Absence_Count, Salary, Department)

- The primary key, NationalID, contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There are more than one candidate keys but since they do not share a common subset, this table is in BCNF.

III. Physicians (Full_Name, MedicalID, Address, Phone_Number, Working_Since, Position,

Additional_Hours, Absence_Count, Salary, Department)

- It is not in 1NF because Phone_Number attribute contains multiple values. So, we must fix it:

Physicians (Full_Name, MedicalID, Address, Phone_Number1, Phone_Number2, Working_Since, Position, Additional_Hours, Absence_Count, Salary, Department)

- The primary key, MedicalID, contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There are more than one candidate keys but since they do not share a common subset, this table is in BCNF.

IV. Employees (NationalID, Full_Name, Address, Phone_Number, Working_Since, Position, Additional_Hours, Absence_Count, Salary, Department)

- It is not in 1NF because Phone_Number attribute contains multiple values. So, we must fix it:

Employees (NationalID, Full_Name, Address, Phone_Number1, Phone_Number2, Working_Since, Position, Additional_Hours, Absence_Count, Salary, Department)

- The primary key, NationalID, contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.

- There are more than one candidate keys but since they do not share a common subset, this table is in BCNF.

V. Pay_Rate (Position, Pay)

- It is in 1NF because attributes do not contain multiple values.
- The primary key, Position, contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There is one candidate key so, this table is in BCNF.

VI. Gets_Paid (Position, MedicalID, NationalID_Nurse, NationalID_Employee)

- It is in 1NF because attributes do not contain multiple values.
- The primary key contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There is one candidate key so, this table is in BCNF.

VII. Registers (NationalID_Patient, NationalID_Employee, Date, Expires)

- It is in 1NF because attributes do not contain multiple values.
- The primary key contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There is one candidate key so, this table is in BCNF.

VIII. Treatment (TreatID, Date, Cost)

- It is in 1NF because attributes do not contain multiple values.
- The primary key, TreatID, contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There is one candidate key subset so, this table is in BCNF.

IX. Suggests (MedicalID, TreatID)

- It is in 1NF because attributes do not contain multiple values.
- The primary key contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There is one candidate key so, this table is in BCNF.

X. Prescription (TreatID, MedID, Explanation, Cost)

- It is not in 1NF because MedID attribute contains multiple values. So, we must fix it:

Prescription (TreatID, MedID1, MedID2, MedID3, Explanation, Cost)

- The primary key, TreatID, contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There is one candidate key so, this table is in BCNF.

XI. Medication (Name, Code, Brand, Description, Cost)

- It is in 1NF because attributes do not contain multiple values.
- The primary key, Code, contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There is one candidate key so, this table is in BCNF.

XII. Cite (Code, TreatID)

- It is in 1NF because attributes do not contain multiple values.
- The primary key contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There is one candidate key so, this table is in BCNF.

XIII. Undergo_Treatment (TreatID, Cost)

- It is in 1NF because attributes do not contain multiple values.
- The primary key, TreatID, contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There is one candidate key so, this table is in BCNF.

XIV. Room (Room_Type, RoomID, Available, Cost, Department)

- It is in 1NF because attributes do not contain multiple values.
- The primary key contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There are more than one candidate keys but since they do not share a common subset, this table is in BCNF.

XV. Assist (NationalID_Nurse, TreatID)

- It is in 1NF because attributes do not contain multiple values.
- The primary key contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There is one candidate key so, this table is in BCNF.

XVI. Stay (TreatID, RoomID, Start_Date, End_Date, Duration)

- It is in 1NF because attributes do not contain multiple values.
- The primary key contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There is one candidate key so, this table is in BCNF.

XVII. Procedure (Cost, TreatID, Code, Name, Physician_Share)

- It is in 1NF because attributes do not contain multiple values.
- The primary key contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There are more than one candidate keys but since they do not share a common subset, this table is in BCNF.

XVIII. P_P (TreatID, Code, MedicalID)

- It is in 1NF because attributes do not contain multiple values.
- The primary key contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There is one candidate key so, this table is in BCNF.

XIV. Patient (Full_Name, NationalID_Patient, Address, Phone_Number, InsuranceID, Gender, Birth_Date, Age, Weight, Height, BMI, Blood_Type, Underlying_Medical_Condition)

- It is not in 1NF because Phone_Number attribute contains multiple values. So, we must fix it:

Patient (Full_Name, NationalID_Patient, Address, Phone_Number1, Phone_Number2, InsuranceID, Gender, Birth_Date, Age, Weight, Height, BMI, Blood_Type, Underlying_Medical_Condition)

- The primary key, NationalID_Patient, contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There are more than one candidate keys but since they do not share a common subset, this table is in BCNF.

XV. Treats (NationalID_Patient, TreatID)

- It is in 1NF because attributes do not contain multiple values.
- The primary key contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.

- There is one candidate key so, this table is in BCNF.

XVI. Type (TreatID, Code)

- It is in 1NF because attributes do not contain multiple values.
- The primary key contains one attribute so this relation is in 2NF.
- There are no transitive functional dependencies, and hence our table is in 3NF.
- There is one candidate key so, this table is in BCNF.

7) Views

I. This is a view for showing the rooms taken by patients

```
/*Shows which rooms are taken by which patients*/
CREATE VIEW Full_room AS SELECT Room.Room_Type, Room.RoomID, Stay.Duration,
Treats.NationalID_Patient
FROM Room INNER JOIN Stay ON Stay.RoomID = Room.RoomID
INNER JOIN Treats on Stay.TreatID = Treats.TreatID;
```

II. This is a view for showing that which physicians treated a certain patient

```
/*Shows which physician treated which patient*/
CREATE VIEW Patients_physicians AS SELECT Patient.NationalID_Patient, Treats.TreatID,
Suggests.MedicalID, Physicians.Full_Name, Treatment.Date
FROM Patient INNER JOIN Treats ON Treats.NationalID_Patient = Patient.NationalID_Patient
INNER JOIN Suggests on Treats.TreatID = Suggests.TreatID
INNER JOIN Physicians on Suggests.MedicalID = Physicians.MedicalID
INNER JOIN treatment on Suggests.TreatID = Treatment.TreatID;
```

III. This is a view for showing the health stats of a patient which makes it easier for nurses to assist patients.

```
/*Health_profile_for_assist_nurse*/
CREATE VIEW Health_profile AS SELECT Patient.Full_Name, Patient.Blood_Type, Patient.Height,
Patient.Weight, Patient.Gender, Patient.Underlying_Medical_Condition,
                    Patient.Birth_Date, Prescription.MedID1, Medication.Name,
Medication.Description, Prescription.Explanation
FROM Patient INNER JOIN Treats ON Treats.NationalID_Patient = Patient.NationalID_Patient
INNER JOIN Prescription on Treats.TreatID = Prescription.TreatID
INNER JOIN Medication ON  Prescription.MedID1 = Medication.Code;
```

IV. This is a view that shows the stats of a procedure so we can easily take track of them.

```
/*Procedure's status*/
CREATE VIEW Procedure_stat AS SELECT Patient.NationalID_Patient, Treats.TreatID,
Suggests.MedicalID, Physicians.Full_Name,
                        Assist.NationalID_Nurse, Pro.Name
FROM Patient INNER JOIN Treats ON Treats.NationalID_Patient = Patient.NationalID_Patient
INNER JOIN Suggests on Treats.TreatID = Suggests.TreatID
INNER JOIN Physicians on Suggests.MedicalID = Physicians.MedicalID
INNER JOIN Assist ON  Assist.TreatID = Treats.TreatID
INNER JOIN Type ON Type.TreatID = Treats.TreatID
INNER JOIN Pro ON Pro.Code = Type.Code;
```

V. This is a view to all members of a certain department

```
/*Department status*/
CREATE VIEW Department_stats AS SELECT Department.Name,
Nurses_Technicians.NationalID_Nurse, Physicians.MedicalID,
                         Employees.NationalID_Employee, Room.Department
FROM Department INNER JOIN Nurses_Technicians ON Nurses_Technicians.Department =
Department.Name
INNER JOIN Physicians on Physicians.Department = Department.Name
INNER JOIN Employees on Employees.Department = Department.Name
INNER JOIN Room  ON  Room.Department = Department.Name;
```

8) Queries

I. Finds the underweight patients

```
/*under weight patients*/
SELECT Full_Name, (Weight / (Height * Height / 10000)) as bmi FROM Patient where (Weight /
(Height * Height / 10000)) < 18;
```

II. Finds the overweight patients

```
/*over weight patients*/
SELECT Full_Name, (Weight / (Height * Height / 10000)) as bmi FROM Patient where (Weight /
(Height * Height / 10000)) > 25;
```

III. Finds the final salary of the nurses and technicians

```
/*salary nurses*/
SELECT Nurses_Technicians.Full_Name, SUM(((Nurses_Technicians.Additional_Hours -
Nurses_Technicians.Absence_Count) * 100) + Pay_Rate.Pay) Total From Nurses_Technicians,
Pay_Rate WHERE Nurses_Technicians.Position = Pay_Rate.Position group by
Nurses_Technicians.Full_Name order by Total;
```

IV. Finds the final salary of the employees

```
/*salary employee*/
SELECT Employees.Full_Name, SUM(((Employees.Additional_Hours - Employees.Absence_Count)
* 100) + Pay_Rate.Pay) Total From Employees, Pay_Rate WHERE Employees.Position =
Pay_Rate.Position group by Employees.Full_Name order by Total;
```

V. Finds the final salary of the physicians minus their share of the procedures

```
/*salary physicians without their share of the procedure*/
SELECT Physicians.Full_Name, SUM(((Physicians.Additional_Hours - Physicians.Absence_Count) *
100) + Pay_Rate.Pay) Total From physicians, Pay_Rate WHERE Physicians.Position =
Pay_Rate.Position group by Physicians.Full_Name order by Total;
```

VI. Finds physician’s share of the procedures so far

```
/*share of the procedure for each physician*/
SELECT Physicians.MedicalID, SUM(P_P.Share) Total From physicians, P_P WHERE
Physicians.MedicalID = P_P.MedicalID group by Physicians.MedicalID order by Total;
```

VII. Some queries for finding the final payment of the physicians

```
/*salary physicians considering their share of the procedure*/
SELECT Physicians.Full_Name, SUM(((Physicians.Additional_Hours - Physicians.Absence_Count) *
100) + Pay_Rate.Pay + 500) Total From physicians, Pay_Rate WHERE Physicians.MedicalID =
640398018 group by Physicians.Full_Name order by Total;
SELECT Physicians.Full_Name, SUM(((Physicians.Additional_Hours - Physicians.Absence_Count) *
100) + Pay_Rate.Pay + 850) Total From physicians, Pay_Rate WHERE Physicians.MedicalID =
640398016 group by Physicians.Full_Name order by Total;
SELECT Physicians.Full_Name, SUM(((Physicians.Additional_Hours - Physicians.Absence_Count) *
100) + Pay_Rate.Pay + 1400) Total From physicians, Pay_Rate WHERE Physicians.MedicalID =
640398015 group by Physicians.Full_Name order by Total;
SELECT Physicians.Full_Name, SUM(((Physicians.Additional_Hours - Physicians.Absence_Count) *
100) + Pay_Rate.Pay + 2000) Total From physicians, Pay_Rate WHERE Physicians.MedicalID =
640398020 group by Physicians.Full_Name order by Total;
SELECT Physicians.Full_Name, SUM(((Physicians.Additional_Hours - Physicians.Absence_Count) *
100) + Pay_Rate.Pay + 8000) Total From physicians, Pay_Rate WHERE Physicians.MedicalID =
640398019 group by Physicians.Full_Name order by Total;
```

VIII. Adding a new medication into our records

```
/*adding a new medication*/
SELECT * FROM Medication;
INSERT INTO Medication VALUES (510, 'vitamin c', 'vitamin', 'konoha', 5);
SELECT * FROM Medication;
```

9) Using mysql, tables were created

```
CREATE DATABASE PROJECT_DB;
USE PROJECT_DB;
/* drop database PROJECT_DB; */
CREATE TABLE Department
  (
    Name varchar(50),
    primary key (Name)
  );

CREATE TABLE Nurses_Technicians
  (
    Department varchar(50) not null ,
    NationalID_Nurse int not null ,
    Full_Name varchar(50) not null ,
    Address varchar(100) not null ,
    Phone_Number1 int not null ,
    Phone_Number2 int,
    Working_Since date not null ,
    Position varchar(50) not null ,
    Additional_Hours int,
    Absence_Count int,
    primary key (NationalID_Nurse),
    foreign key (Department) references Department (Name)
  );

CREATE TABLE Physicians
  (
    Department varchar(50) not null ,
    MedicalID int not null ,
    Full_Name varchar(50) not null ,
    Address varchar(100) not null ,
    Phone_Number1 int not null ,
    Phone_Number2 int,
    Working_Since date not null ,
    Position varchar(50) not null ,
    Additional_Hours int,
    Absence_Count int,
    primary key (MedicalID),
    foreign key (Department) references Department (Name)
  );

CREATE TABLE Employees
  (
    Department varchar(50) not null ,
    NationalID_Employee int not null ,
    Full_Name varchar(50) not null ,
    Address varchar(100) not null ,
    Phone_Number1 int not null ,
    Phone_Number2 int,
    Working_Since date not null ,
    Position varchar(50) not null ,
    Additional_Hours int,
    Absence_Count int,
    primary key (NationalID_Employee),
    foreign key (Department) references Department (Name)
  );

CREATE TABLE Patient
  (
    NationalID_Patient int not null ,
    Full_Name varchar(50) not null ,
    Address varchar(100) not null ,
    Phone_Number1 int not null ,
    Phone_Number2 int,
    Birth_Date date not null ,
    Gender varchar(15) not null ,
    InsuranceID int,
    Weight int not null ,
    Height int not null ,
    Blood_Type varchar(50),
    Underlying_Medical_Condition varchar(200),
    primary key (NationalID_Patient)
  );

CREATE TABLE Pay_Rate
  (
    Position varchar(50) not null ,
    Pay int not null ,
    primary key (Position)
  );

CREATE TABLE Gets_Paid
  (
    Position varchar(50),
    PayID int,
    MedicalID int,
    NationalID_Nurse int,
    NationalID_Employee int,
    foreign key  (Position) references Pay_Rate (Position),
    foreign key (MedicalID) references Physicians (MedicalID),
    foreign key (NationalID_Nurse) references Nurses_Technicians (NationalID_Nurse),
    foreign key (NationalID_Employee) references Employees (NationalID_Employee),
    primary key (PayID)
  );

CREATE TABLE Registers
  (
    NationalID_Employee int,
    NationalID_Patient int,
    Date date,
    Expires date,
    primary key (NationalID_Patient, NationalID_Employee),
    foreign key (NationalID_Patient) references Patient (NationalID_Patient),
    foreign key (NationalID_Employee) references Employees (NationalID_Employee)
  );

CREATE TABLE Treatment
  (
    TreatID int,
    Date date not null ,
    primary key (TreatID)
  );

CREATE TABLE Suggests
  (
    TreatID int,
    MedicalID int,
    foreign key (MedicalID) references Physicians (MedicalID),
    foreign key (TreatID) references Treatment (TreatID),
    primary key (MedicalID, TreatID)
  );

CREATE TABLE Medication
  (
    Code int,
    Name varchar(50),
    Description varchar(200),
    Brand varchar(50),
    Cost int,
    primary key (Code)
  );

CREATE TABLE Prescription
  (
    TreatID int,
    MedID1 int,
    Explanation varchar(200),
    foreign key (MedID1) references Medication (code) ,
    foreign key (TreatID) references Treatment (TreatID),
    primary key (TreatID)
  );

CREATE TABLE Cite
  (
    Code int,
    TreatID int,
    foreign key (Code) references Medication (Code) ,
    foreign key (TreatID) references Treatment (TreatID),
    primary key (Code, TreatID)
  );

CREATE TABLE Undergo_Treatment
  (
    TreatID int,
    foreign key (TreatID) references Treatment (TreatID),
    primary key (TreatID)
  );

CREATE TABLE Room
  (
    Room_Type varchar(50),
    RoomID int,
    Available bool,
    Cost int,
    Department varchar(50),
    foreign key (Department) references Department (Name),
    primary key (RoomID)
  );

CREATE TABLE Assist
  (
    NationalID_Nurse int,
    TreatID int,
    foreign key (NationalID_Nurse) references Nurses_Technicians (NationalID_Nurse),
    foreign key (TreatID) references Treatment (TreatID),
    primary key (NationalID_Nurse, TreatID)
  );

CREATE TABLE Stay
  (
    RoomID int,
    TreatID int,
    Duration int,
    foreign key (RoomID) references Room (RoomID),
    foreign key (TreatID) references Treatment (TreatID),
    primary key (TreatID, RoomID)
  );

CREATE TABLE Pro
  (
    Code int,
    cost int,
    name varchar(50),
    primary key (Code)
  );

CREATE TABLE Treats
  (
    TreatID int,
    NationalID_Patient int,
    foreign key (TreatID) references Treatment (TreatID),
    foreign key (NationalID_Patient) references Patient (NationalID_Patient),
    primary key (NationalID_Patient, TreatID)
  );

CREATE TABLE Type
  (
    TreatID int,
    Code int,
    foreign key (TreatID) references Treatment (TreatID),
    foreign key (Code) references Pro (Code),
    primary key (TreatID, Code)
  );

CREATE TABLE P_P
  (
    Code int,
    TreatID int,
    MedicalID int,
    Share int,
    foreign key (Code) references Pro (code) ,
    foreign key (TreatID) references Treatment (TreatID),
    foreign key (MedicalID) references Physicians (MedicalID),
    primary key (TreatID, Code, MedicalID)
);
```

10) Inserting data into the database

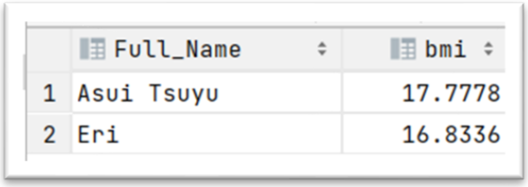
```
INSERT INTO Department VALUES ('surgery');
INSERT INTO Department VALUES ('gynaecology');
INSERT INTO Department VALUES ('obstetrics');
INSERT INTO Department VALUES ('paediatrics');
INSERT INTO Department VALUES ('orthopaedics');
INSERT INTO Department VALUES ('neurology');
INSERT INTO Department VALUES ('ENT');
INSERT INTO Department VALUES ('laboratory');
INSERT INTO Department VALUES ('Ambulance');
INSERT INTO Department VALUES ('ICU');
INSERT INTO Department VALUES ('Tech');
INSERT INTO Department VALUES ('Business Office');
INSERT INTO Department VALUES ('Maintenance Department');
INSERT INTO Nurses_Technicians VALUES ('surgery', 640398001, 'Naruto Uzumaki', 'Konoha village Land of Fire', 5597501, NULL, '2002-09-20', 'senior nurse', 0, 12);
INSERT INTO Nurses_Technicians VALUES ('obstetrics', 640398002, 'Sasuke Uchiha', 'Konoha village Land of Fire', 5597502, 5597510, '2002-09-20', 'senior nurse', 12, 0);
INSERT INTO Nurses_Technicians VALUES ('gynaecology', 640398003, 'Sakura Haruno', 'Konoha village Land of Fire', 5597503, 5597509, '2002-09-20', 'senior nurse', 24, 0);
INSERT INTO Nurses_Technicians VALUES ('paediatrics', 640398004, 'Hinata Hyuga', 'Konoha village Land of Fire', 5597504, NULL, '2002-09-20', 'senior nurse', 6, 0);
INSERT INTO Nurses_Technicians VALUES ('orthopaedics', 640398005, 'Kakashi Hatake', 'Konoha village Land of Fire', 5597505, 5597508, '2002-09-20', 'senior nurse', 6, 0);
INSERT INTO Nurses_Technicians VALUES ('neurology', 640398006, 'Iruka Umino', 'Konoha village Land of Fire', 5597506, 5597507, '2002-09-20', 'junior nurse', 6, 0);
INSERT INTO Nurses_Technicians VALUES ('ENT', 640398007, 'Konohamaru Sarutobi', 'Konoha village Land of Fire', 5597511, 5597518, '2002-09-20', 'junior nurse', 0, 6);
INSERT INTO Nurses_Technicians VALUES ('laboratory', 640398008, 'Asuma Sarutobi', 'Konoha village Land of Fire', 55975012, 55975013, '2002-09-20', 'lab technician', 0, 0);
INSERT INTO Nurses_Technicians VALUES ('ICU', 640398009, 'Kurenai Yuhi', 'Konoha village Land of Fire', 55975014, 55975015, '2002-09-20', 'senior nurse', 0, 0);
INSERT INTO Nurses_Technicians VALUES ('surgery', 640398010, 'Shino Aburame', 'Konoha village Land of Fire', 5597516, 5597517, '2002-09-20', 'junior nurse', 0, 0);
INSERT INTO Physicians VALUES ('surgery', 640398011, 'Shikamaru Nara', 'Konoha village Land of Fire', 5597518, 5597519, '2002-09-20', 'surgeon', 1, 2);
INSERT INTO Physicians VALUES ('obstetrics', 640398012, 'Neji Hyuga', 'Konoha village Land of Fire', 5597520, 5597521, '2002-09-20', 'obstetrician', 1, 3);
INSERT INTO Physicians VALUES ('gynaecology', 640398013, 'Rock Lee', 'Konoha village Land of Fire', 5597522, 5597523, '2002-09-20', 'gynaecologist', 0, 2);
INSERT INTO Physicians VALUES ('paediatrics', 640398014, 'Orochimaru', 'Konoha village Land of Fire', 5597524, 5597525, '2002-09-20', 'paediatrician', 1, 1);
INSERT INTO Physicians VALUES ('orthopaedics', 640398015, 'Jiraiya', 'Konoha village Land of Fire', 5597526, 5597527, '2002-09-20', 'orthopedic', 0, 1);
INSERT INTO Physicians VALUES ('neurology', 640398016, 'Tsunade', 'Konoha village Land of Fire', 5597528, 5597529, '2002-09-20', 'neurologist', 0, 0);
INSERT INTO Physicians VALUES ('ENT', 640398017, 'Itachi Uchiha', 'Konoha village Land of Fire', 5597530, 5597531, '2002-09-20', 'surgical specialty', 0, 0);
INSERT INTO Physicians VALUES ('ICU', 640398018, 'Ino Yamanaka', 'Konoha village Land of Fire', 5597532, 5597533, '2002-09-20', 'surgeon', 0, 0);
INSERT INTO Physicians VALUES ('Ambulance', 640398019, 'Sai', 'Konoha village Land of Fire', 5597534, 5597535, '2002-09-20', 'emergency doctor', 0, 0);
INSERT INTO Physicians VALUES ('laboratory', 640398020, 'Tenten', 'Konoha village Land of Fire', 5597536, 5597537, '2002-09-20', 'pathologist', 24, 0);
INSERT INTO Employees VALUES ('Maintenance Department', 640398021, 'Eren Yeager', 'Shinganshina district paradise island', 5597540, 5597541, '2002-09-20', 'mechanical eng', 0, 0);
INSERT INTO Employees VALUES ('Maintenance Department', 640398022, 'Armin Arlert', 'Shinganshina district paradise island', 5597538, 5597539, '2002-09-20', 'biomedical eng', 0, 0);
INSERT INTO Employees VALUES ('Business Office', 640398023, 'Mikasa Ackerman', 'Shinganshina district paradise island', 5597542, 5597543, '2002-09-20', 'secratery', 0, 0);
INSERT INTO Employees VALUES ('Business Office', 640398024, 'Jean Krisein', 'Wall rose paradise island', 5597544, 5597545, '2002-09-20', 'manager', 0, 0);
INSERT INTO Employees VALUES ('Tech', 640398025, 'Connie springer', 'Wall rose paradise island', 5597546, 5597547, '2002-09-20', 'junior computer eng', 0, 0);
INSERT INTO Employees VALUES ('Tech', 640398026, 'Sasha Brus', 'Dauper village Wall rose paradise island', 5597548, NULL, '2002-09-20', 'senior computer eng', 0, 0);
INSERT INTO Employees VALUES ('surgery', 640398027, 'Levi', 'underground city paradise island', 5597549, NULL, '2002-09-20', 'housekeeping', 0, 0);
INSERT INTO Employees VALUES ('gynaecology', 640398028, 'Reiner Braun', 'Eldian district Marley', 5597550, 5597551, '2002-09-20', 'security', 0, 0);
INSERT INTO Employees VALUES ('neurology', 640398029, 'Annie Leohart', 'Eldian district Marley', 5597552, 5597553, '2002-09-20', 'security', 0, 0);
INSERT INTO Employees VALUES ('Ambulance', 640398030, 'Bertolt Hoover', 'Eldian district Marley', 5597554, 5597555, '2002-09-20', 'driver', 0, 0);
INSERT INTO Patient VALUES (640398031, 'Aizawa Shota', 'Musutafu Shizuoka prefecture japan', 5597556, 5597557, '2000-11-08', 'm', NULL, 80, 183, 'b', 'sensitive eyes');
INSERT INTO Patient VALUES (640398032, 'All Might', 'Musutafu Shizuoka prefecture japan', 5597558, 5597559, '2000-06-10', 'm', 101,100, 220, 'a', 'lung cancer');
INSERT INTO Patient VALUES (640398033, 'Asui Tsuyu', 'Musutafu Shizuoka prefecture japan', 5597560, 5597561, '2004-03-12', 'f', 102,40, 150, 'b', NULL);
INSERT INTO Patient VALUES (640398034, 'Ida Tenya', 'Musutafu Shizuoka prefecture japan', 5597562, 5597563, '2003-08-22', 'm', 103,80, 179, 'a', 'left leg has permanant injuries');
INSERT INTO Patient VALUES (640398035, 'Urakaka Ochako', 'Musutafu Shizuoka prefecture japan', 5597564, 5597565, '2004-11-08', 'f', 104,50, 156, 'b', NULL);
INSERT INTO Patient VALUES (640398036, 'Kaminari Denki', 'Musutafu Shizuoka prefecture japan', 5597566, 5597567, '2004-07-29', 'm', 105,163, 163, 'o', 'anxiety');
INSERT INTO Patient VALUES (640398037, 'Todoroki Shoto', 'Musutafu Shizuoka prefecture japan', 5597568, NULL, '2000-01-11', 'm', 106,70, 176, 'o', 'burnt marks on left eye');
INSERT INTO Patient VALUES (640398038, 'Bakugo Katsuki', 'Musutafu Shizuoka prefecture japan', 5597569, 5597570, '2000-02-20', 'm', 107,60, 173, 'z', NULL);
INSERT INTO Patient VALUES (640398039, 'Midoriya Izuku', 'Musutafu Shizuoka prefecture
japan', 5597571, 5597572, '2000-04-15', 'm', 108,60, 166, 'o', 'severed left leg');
INSERT INTO Patient VALUES (640398040, 'Eri', 'Musutafu Shizuoka prefecture japan', 5597573,
5597574, '2016-12-21', 'f', '0009',20, 109, 'o', 'migraine headaches');
INSERT INTO Pay_Rate VALUES ('senior nurse', 3000);
INSERT INTO Pay_Rate VALUES ('junior nurse', 2500);
INSERT INTO Pay_Rate VALUES ('lab technician', 3500);
INSERT INTO Pay_Rate VALUES ('surgeon', 5000);
INSERT INTO Pay_Rate VALUES ('obstetrician', 5500);
INSERT INTO Pay_Rate VALUES ('gynaecologist', 5500);
INSERT INTO Pay_Rate VALUES ('paediatrician', 5500);
INSERT INTO Pay_Rate VALUES ('orthopedic', 5000);
INSERT INTO Pay_Rate VALUES ('neurologist', 6000);
INSERT INTO Pay_Rate VALUES ('surgical specialty', 5500);
INSERT INTO Pay_Rate VALUES ('emergency doctor', 5000);
INSERT INTO Pay_Rate VALUES ('pathologist', 4000);
INSERT INTO Pay_Rate VALUES ('mechanical eng', 4000);
INSERT INTO Pay_Rate VALUES ('biomedical eng', 4000);
INSERT INTO Pay_Rate VALUES ('secratery', 2000);
INSERT INTO Pay_Rate VALUES ('manager', 2500);
INSERT INTO Pay_Rate VALUES ('junior computer eng', 3500);
INSERT INTO Pay_Rate VALUES ('senior computer eng', 4500);
INSERT INTO Pay_Rate VALUES ('housekeeping', 1000);
INSERT INTO Pay_Rate VALUES ('security', 1500);
INSERT INTO Pay_Rate VALUES ('driver', 1000);
INSERT INTO Gets_Paid VALUES ('junior nurse', 300, NULL, 640398006, NULL);
INSERT INTO Gets_Paid VALUES ('junior nurse', 301, NULL, 640398007, NULL);
INSERT INTO Gets_Paid VALUES ('junior nurse', 302, NULL, 640398010, NULL);
INSERT INTO Gets_Paid VALUES ('senior nurse', 303, NULL, 640398001, NULL);
INSERT INTO Gets_Paid VALUES ('senior nurse', 304, NULL, 640398002, NULL);
INSERT INTO Gets_Paid VALUES ('senior nurse', 305, NULL, 640398003, NULL);
INSERT INTO Gets_Paid VALUES ('senior nurse', 306, NULL, 640398004, NULL);
INSERT INTO Gets_Paid VALUES ('senior nurse', 307, NULL, 640398005, NULL);
INSERT INTO Gets_Paid VALUES ('senior nurse', 308, NULL, 640398009, NULL);
INSERT INTO Gets_Paid VALUES ('surgeon', 309, 640398011, NULL, NULL);
INSERT INTO Gets_Paid VALUES ('surgeon', 310, 640398018, NULL, NULL);
INSERT INTO Gets_Paid VALUES ('obstetrician', 311, 640398012, NULL, NULL);
INSERT INTO Gets_Paid VALUES ('gynaecologist', 312, 640398013, NULL, NULL);
INSERT INTO Gets_Paid VALUES ('paediatrician', 313, 640398014, NULL, NULL);
INSERT INTO Gets_Paid VALUES ('orthopedic', 314, 640398015, NULL, NULL);
INSERT INTO Gets_Paid VALUES ('neurologist', 315, 640398016, NULL, NULL);
INSERT INTO Gets_Paid VALUES ('surgical specialty', 316, 640398017, NULL, NULL);
INSERT INTO Gets_Paid VALUES ('emergency doctor', 317, 640398019, NULL, NULL);
INSERT INTO Gets_Paid VALUES ('pathologist', 318, 640398020, NULL, NULL);
INSERT INTO Gets_Paid VALUES ('mechanical eng', 319, NULL, NULL, 640398021);
INSERT INTO Gets_Paid VALUES ('biomedical eng', 320, NULL, NULL, 640398022);
INSERT INTO Gets_Paid VALUES ('secratery', 321, NULL, NULL, 640398023);
INSERT INTO Gets_Paid VALUES ('manager', 322, NULL, NULL, 640398024);
INSERT INTO Gets_Paid VALUES ('junior computer eng', 323, NULL, NULL, 640398025);
INSERT INTO Gets_Paid VALUES ('senior computer eng', 324, NULL, NULL, 640398026);
INSERT INTO Gets_Paid VALUES ('housekeeping', 325, NULL, NULL, 640398027);
INSERT INTO Gets_Paid VALUES ('security', 326, NULL, NULL, 640398028);
INSERT INTO Gets_Paid VALUES ('security', 327, NULL, NULL, 640398029);
INSERT INTO Gets_Paid VALUES ('driver', 328, NULL, NULL, 640398030);
INSERT INTO Registers VALUES (640398023, 640398031, '2021-09-10', '2022-09-10');
INSERT INTO Registers VALUES (640398023, 640398032, '2021-09-10', '2022-09-10');
INSERT INTO Registers VALUES (640398023, 640398033, '2021-09-10', '2022-09-10');
INSERT INTO Registers VALUES (640398023, 640398034, '2021-09-10', '2022-09-10');
INSERT INTO Registers VALUES (640398023, 640398035, '2021-09-10', '2022-09-10');
INSERT INTO Registers VALUES (640398023, 640398036, '2021-09-10', '2022-09-10');
INSERT INTO Registers VALUES (640398023, 640398037, '2021-09-10', '2022-09-10');
INSERT INTO Registers VALUES (640398023, 640398038, '2021-09-10', '2022-09-10');
INSERT INTO Registers VALUES (640398023, 640398039, '2021-09-10', '2022-09-10');
INSERT INTO Registers VALUES (640398023, 640398040, '2021-09-10', '2022-09-10');
INSERT INTO Treatment VALUES (0, '2021-09-10');
INSERT INTO Treatment VALUES (1, '2021-09-10');
INSERT INTO Treatment VALUES (2, '2021-09-10');
INSERT INTO Treatment VALUES (3, '2021-09-10');
INSERT INTO Treatment VALUES (4, '2021-09-10');
INSERT INTO Treatment VALUES (5, '2021-09-10');
INSERT INTO Treatment VALUES (6, '2021-09-10');
INSERT INTO Treatment VALUES (7, '2021-09-10');
INSERT INTO Treatment VALUES (8, '2021-09-10');
INSERT INTO Treatment VALUES (9, '2021-09-10');
INSERT INTO Suggests VALUES (0, 640398018);
INSERT INTO Suggests VALUES (1, 640398015);
INSERT INTO Suggests VALUES (2, 640398015);
INSERT INTO Suggests VALUES (3, 640398015);
INSERT INTO Suggests VALUES (4, 640398016);
INSERT INTO Suggests VALUES (5, 640398016);
INSERT INTO Suggests VALUES (6, 640398020);
INSERT INTO Suggests VALUES (7, 640398020);
INSERT INTO Suggests VALUES (8, 640398019);
INSERT INTO Suggests VALUES (9, 640398019);
INSERT INTO Medication VALUES (500, 'fire', 'muscle pain', 'todoroki', 10);
INSERT INTO Medication VALUES (501, 'ice', 'joint pain', 'todoroki', 10);
INSERT INTO Medication VALUES (502, 'pain', 'm/s', 'rain', 100);
INSERT INTO Medication VALUES (503, 'wind', 'pain relief', 'uzumaki', 10);
INSERT INTO Medication VALUES (504, 'earth', 'stomachache', 'shouji', 10);
INSERT INTO Medication VALUES (505, 'lightening', 'revives dead people', 'uchiha', 10);
INSERT INTO Medication VALUES (506, 'numo', 'does something', 'todoroki', 10);
INSERT INTO Medication VALUES (507, 'feather', 'weight loss', 'todoroki', 10);
INSERT INTO Medication VALUES (508, 'painplus', 'reduces pain', 'todoroki', 10);
INSERT INTO Medication VALUES (509, 'healer', 'heals up', 'todoroki', 10);
INSERT INTO Prescription VALUES (0, 500,  'everyday for a week');
INSERT INTO Prescription VALUES (1, 500,  'everyday for a week');
INSERT INTO Prescription VALUES (2, 502,  'everyday for a week');
INSERT INTO Prescription VALUES (3, 501,  'everyday for a week');
INSERT INTO Prescription VALUES (4, 500, 'everyday for a week');
INSERT INTO Prescription VALUES (5, 500,  'everyday for a week');
INSERT INTO Prescription VALUES (6, 500,  'everyday for a week');
INSERT INTO Prescription VALUES (7, 500,  'everyday for a week');
INSERT INTO Prescription VALUES (8, 500,  'everyday for a week');
```

```sql
INSERT INTO Prescription VALUES (9, 505, 'everyday for a week');
INSERT INTO Cite VALUES (500, 0);
INSERT INTO Cite VALUES (500, 1);
INSERT INTO Cite VALUES (502, 2);
INSERT INTO Cite VALUES (501, 3);
INSERT INTO Cite VALUES (500, 4);
INSERT INTO Cite VALUES (500, 5);
INSERT INTO Cite VALUES (500, 6);
INSERT INTO Cite VALUES (500, 7);
INSERT INTO Cite VALUES (500, 8);
INSERT INTO Cite VALUES (500, 9);
INSERT INTO Cite VALUES (505, 9);
INSERT INTO Undergo_Treatment VALUES (0);
INSERT INTO Undergo_Treatment VALUES (1);
INSERT INTO Undergo_Treatment VALUES (2);
INSERT INTO Undergo_Treatment VALUES (3);
INSERT INTO Undergo_Treatment VALUES (4);
INSERT INTO Undergo_Treatment VALUES (5);
INSERT INTO Undergo_Treatment VALUES (6);
INSERT INTO Undergo_Treatment VALUES (7);
INSERT INTO Undergo_Treatment VALUES (8);
INSERT INTO Undergo_Treatment VALUES (9);
INSERT INTO Room VALUES ('vip', 700, true, 50, 'surgery');
INSERT INTO Room VALUES ('vip', 701, true, 50, 'surgery');
INSERT INTO Room VALUES ('vip', 702, true, 50, 'surgery');
INSERT INTO Room VALUES ('vip', 703, true, 50, 'surgery');
INSERT INTO Room VALUES ('vip', 704, true, 50, 'surgery');
INSERT INTO Room VALUES ('normal', 705, true, 25, 'ICU');
INSERT INTO Room VALUES ('normal', 706, true, 25, 'ICU');
INSERT INTO Room VALUES ('normal', 707, true, 25, 'ICU');
INSERT INTO Room VALUES ('normal', 708, true, 25, 'ICU');
INSERT INTO Room VALUES ('normal', 709, true, 25, 'neurology');
INSERT INTO Assist VALUES (640398003, 0);
INSERT INTO Assist VALUES (640398003, 1);
INSERT INTO Assist VALUES (640398001, 2);
INSERT INTO Assist VALUES (640398001, 3);
INSERT INTO Assist VALUES (640398003, 4);
INSERT INTO Assist VALUES (640398003, 5);
INSERT INTO Assist VALUES (640398006, 6);
INSERT INTO Assist VALUES (640398006, 7);
INSERT INTO Assist VALUES (640398010, 8);
INSERT INTO Assist VALUES (640398010, 9);
INSERT INTO Stay VALUES (700, 0, 1);
INSERT INTO Stay VALUES (701, 1, 1);
INSERT INTO Stay VALUES (702, 2, 1);
INSERT INTO Stay VALUES (703, 3, 1);
INSERT INTO Stay VALUES (704, 4, 5);
INSERT INTO Stay VALUES (705, 5, 5);
INSERT INTO Stay VALUES (706, 6, 1);
INSERT INTO Stay VALUES (707, 7, 1);
INSERT INTO Stay VALUES (708, 8, 2);
INSERT INTO Stay VALUES (709, 9, 2);
INSERT INTO Pro VALUES (200, 1000, 'broken leg surgery');
INSERT INTO Pro VALUES (201, 1000,  'broken hand surgery');
INSERT INTO Pro VALUES (202, 2000,  'blood test');
INSERT INTO Pro VALUES (203, 1000,  'laser treatment');
INSERT INTO Pro VALUES (204, 1000,  'x ray');
INSERT INTO Pro VALUES (205, 1000,  'radiology');
INSERT INTO Pro VALUES (206, 1000,  'ct scan');
INSERT INTO Pro VALUES (207, 3000,  'mri');
INSERT INTO Pro VALUES (208, 6000,  'surgery1');
INSERT INTO Pro VALUES (209, 10000,  'surgery2');
INSERT INTO Treats VALUES (0, 640398031);
INSERT INTO Treats VALUES (1, 640398032);
INSERT INTO Treats VALUES (2, 640398033);
INSERT INTO Treats VALUES (3, 640398034);
INSERT INTO Treats VALUES (4, 640398035);
INSERT INTO Treats VALUES (5, 640398036);
INSERT INTO Treats VALUES (6, 640398037);
INSERT INTO Treats VALUES (7, 640398038);
INSERT INTO Treats VALUES (8, 640398039);
INSERT INTO Treats VALUES (9, 640398040);
INSERT INTO Type VALUES ( 0, 200);
INSERT INTO Type VALUES ( 1, 201);
INSERT INTO Type VALUES ( 2, 202);
INSERT INTO Type VALUES ( 3, 203);
INSERT INTO Type VALUES ( 4, 204);
INSERT INTO Type VALUES ( 5, 205);
INSERT INTO Type VALUES ( 6, 206);
INSERT INTO Type VALUES ( 7, 201);
INSERT INTO Type VALUES ( 8, 201);
INSERT INTO Type VALUES ( 9, 201);
INSERT INTO P_P VALUES (200, 0,  640398018, 500);
INSERT INTO P_P VALUES (201, 1,  640398015, 500);
INSERT INTO P_P VALUES (202, 2,  640398015, 500);
INSERT INTO P_P VALUES (203, 3,  640398015, 400);
INSERT INTO P_P VALUES (204, 4,  640398016, 400);
INSERT INTO P_P VALUES (205, 5,  640398016, 450);
INSERT INTO P_P VALUES (206, 6,  640398020, 500);
INSERT INTO P_P VALUES (207, 7,  640398020, 1500);
INSERT INTO P_P VALUES (208, 8,  640398019, 3000);
INSERT INTO P_P VALUES (209, 9,  640398019, 5000);
```

11) Implementing the queries on our data

I. Finds the underweight patients

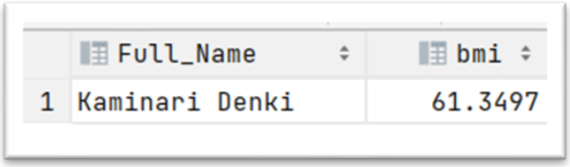
```
/*under weight patients*/
SELECT Full_Name, (Weight / (Height * Height / 10000)) as bmi FROM Patient where (Weight /
(Height * Height / 10000)) < 18;
```



	Full_Name	bmi
1	Asui Tsuyu	17.7778
2	Eri	16.8336

II. Finds the overweight patients

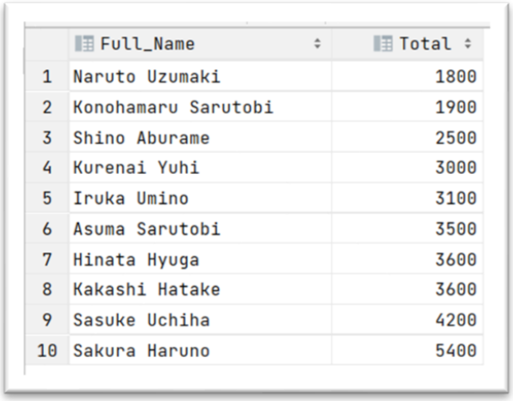
```
/*over weight patients*/
SELECT Full_Name, (Weight / (Height * Height / 10000)) as bmi FROM Patient where (Weight /
(Height * Height / 10000)) > 25;
```



	Full_Name	bmi
1	Kaminari Denki	61.3497

III. Finds the final salary of the nurses and technicians

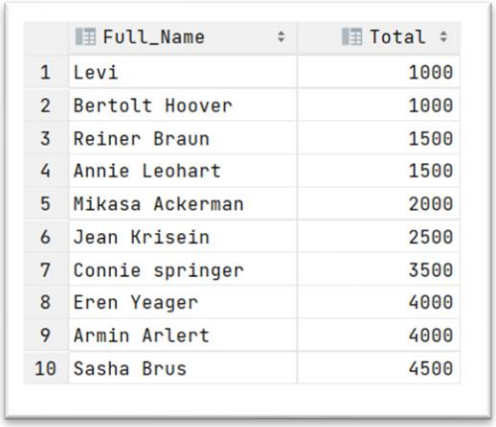
```
/*salary nurses*/
SELECT Nurses_Technicians.Full_Name, SUM(((Nurses_Technicians.Additional_Hours -
Nurses_Technicians.Absence_Count) * 100) + Pay_Rate.Pay) Total From Nurses_Technicians,
Pay_Rate WHERE Nurses_Technicians.Position = Pay_Rate.Position group by
Nurses_Technicians.Full_Name order by Total;
```



	Full_Name	Total
1	Naruto Uzumaki	1800
2	Konohamaru Sarutobi	1900
3	Shino Aburame	2500
4	Kurenai Yuhi	3000
5	Iruka Umino	3100
6	Asuma Sarutobi	3500
7	Hinata Hyuga	3600
8	Kakashi Hatake	3600
9	Sasuke Uchiha	4200
10	Sakura Haruno	5400

IV. Finds the final salary of the employees

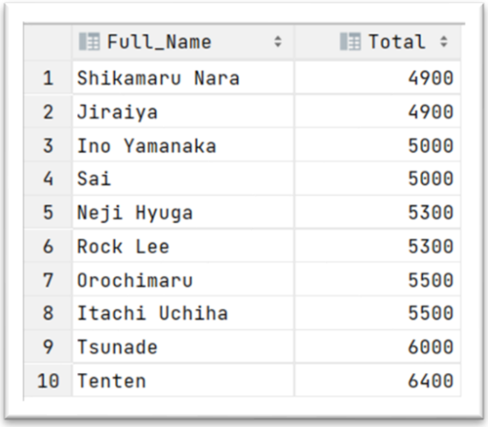
```
/*salary employee*/
SELECT Employees.Full_Name, SUM(((Employees.Additional_Hours - Employees.Absence_Count)
* 100) + Pay_Rate.Pay) Total From Employees, Pay_Rate WHERE Employees.Position =
Pay_Rate.Position group by Employees.Full_Name order by Total;
```



	Full_Name	Total
1	Levi	1000
2	Bertolt Hoover	1000
3	Reiner Braun	1500
4	Annie Leohart	1500
5	Mikasa Ackerman	2000
6	Jean Krisein	2500
7	Connie springer	3500
8	Eren Yeager	4000
9	Armin Arlert	4000
10	Sasha Brus	4500

V. Finds the final salary of the physicians minus their share of the procedures

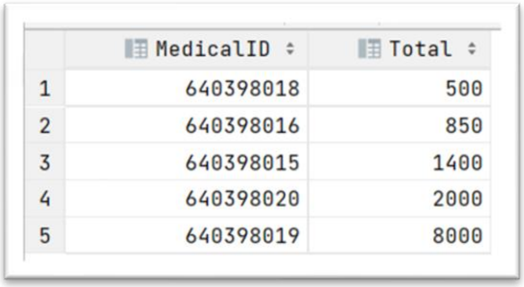
```
/*salary physicians without their share of the procedure*/
SELECT Physicians.Full_Name, SUM(((Physicians.Additional_Hours - Physicians.Absence_Count) *
100) + Pay_Rate.Pay) Total From physicians, Pay_Rate WHERE Physicians.Position =
Pay_Rate.Position group by Physicians.Full_Name order by Total;
```



	Full_Name	Total
1	Shikamaru Nara	4900
2	Jiraiya	4900
3	Ino Yamanaka	5000
4	Sai	5000
5	Neji Hyuga	5300
6	Rock Lee	5300
7	Orochimaru	5500
8	Itachi Uchiha	5500
9	Tsunade	6000
10	Tenten	6400

VI. Finds physician's share of the procedures so far

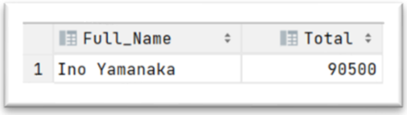
```
/*share of the procedure for each physician*/
SELECT Physicians.MedicalID, SUM(P_P.Share) Total From physicians, P_P WHERE
Physicians.MedicalID = P_P.MedicalID group by Physicians.MedicalID order by Total;
```



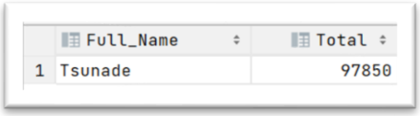
	MedicalID	Total
1	640398018	500
2	640398016	850
3	640398015	1400
4	640398020	2000
5	640398019	8000

VII. Some queries for finding the final payment of the physicians

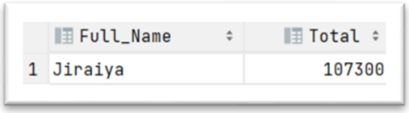
```
/*salary physicians considering their share of the procedure*/
SELECT Physicians.Full_Name, SUM(((Physicians.Additional_Hours - Physicians.Absence_Count) * 100) + Pay_Rate.Pay + 500) Total From physicians, Pay_Rate WHERE Physicians.MedicalID = 640398018 group by Physicians.Full_Name order by Total;
SELECT Physicians.Full_Name, SUM(((Physicians.Additional_Hours - Physicians.Absence_Count) * 100) + Pay_Rate.Pay + 850) Total From physicians, Pay_Rate WHERE Physicians.MedicalID = 640398016 group by Physicians.Full_Name order by Total;
SELECT Physicians.Full_Name, SUM(((Physicians.Additional_Hours - Physicians.Absence_Count) * 100) + Pay_Rate.Pay + 1400) Total From physicians, Pay_Rate WHERE Physicians.MedicalID = 640398015 group by Physicians.Full_Name order by Total;
SELECT Physicians.Full_Name, SUM(((Physicians.Additional_Hours - Physicians.Absence_Count) * 100) + Pay_Rate.Pay + 2000) Total From physicians, Pay_Rate WHERE Physicians.MedicalID = 640398020 group by Physicians.Full_Name order by Total;
SELECT Physicians.Full_Name, SUM(((Physicians.Additional_Hours - Physicians.Absence_Count) * 100) + Pay_Rate.Pay + 8000) Total From physicians, Pay_Rate WHERE Physicians.MedicalID = 640398019 group by Physicians.Full_Name order by Total;
```



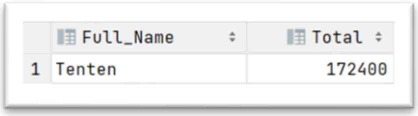
	Full_Name	Total
1	Ino Yamanaka	90500



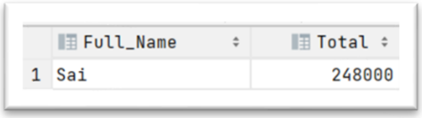
	Full_Name	Total
1	Tsunade	97850



	Full_Name	Total
1	Jiraiya	107300



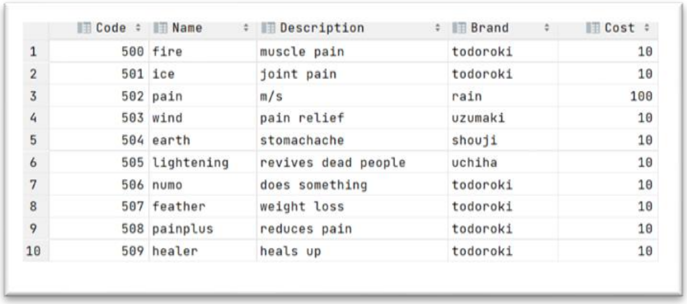
	Full_Name	Total
1	Tenten	172400



	Full_Name	Total
1	Sai	248000

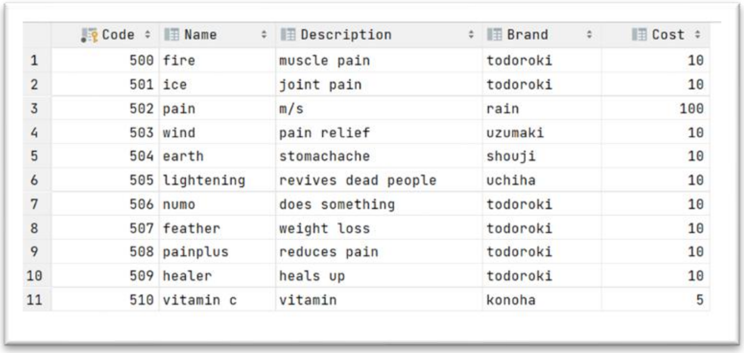
VIII. Adding a new medication into our records

```
/*adding a new medication*/
SELECT * FROM Medication;
INSERT INTO Medication VALUES (510, 'vitamin c', 'vitamin', 'konoha', 5);
SELECT * FROM Medication;
```



	Code	Name	Description	Brand	Cost
1	500	fire	muscle pain	todoroki	10
2	501	ice	joint pain	todoroki	10
3	502	pain	m/s	rain	100
4	503	wind	pain relief	uzumaki	10
5	504	earth	stomachache	shouji	10
6	505	lightening	revives dead people	uchiha	10
7	506	numo	does something	todoroki	10
8	507	feather	weight loss	todoroki	10
9	508	painplus	reduces pain	todoroki	10
10	509	healer	heals up	todoroki	10

Before insert



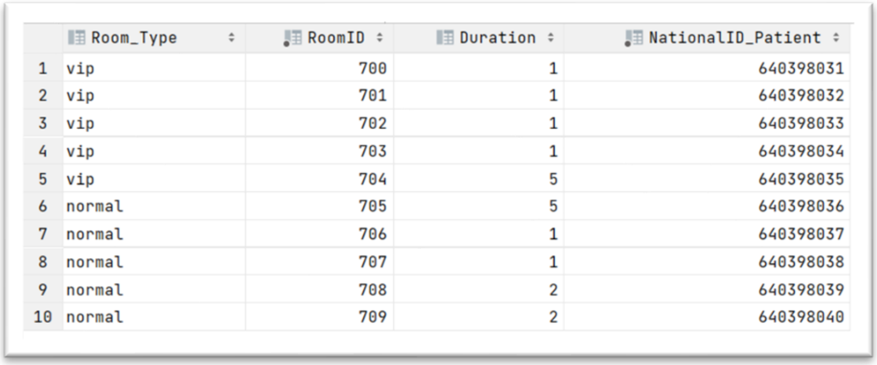
	Code	Name	Description	Brand	Cost
1	500	fire	muscle pain	todoroki	10
2	501	ice	joint pain	todoroki	10
3	502	pain	m/s	rain	100
4	503	wind	pain relief	uzumaki	10
5	504	earth	stomachache	shouji	10
6	505	lightening	revives dead people	uchiha	10
7	506	numo	does something	todoroki	10
8	507	feather	weight loss	todoroki	10
9	508	painplus	reduces pain	todoroki	10
10	509	healer	heals up	todoroki	10
11	510	vitamin c	vitamin	konoha	5

After insert

12) Views

I. This is a view for showing the rooms taken by patients

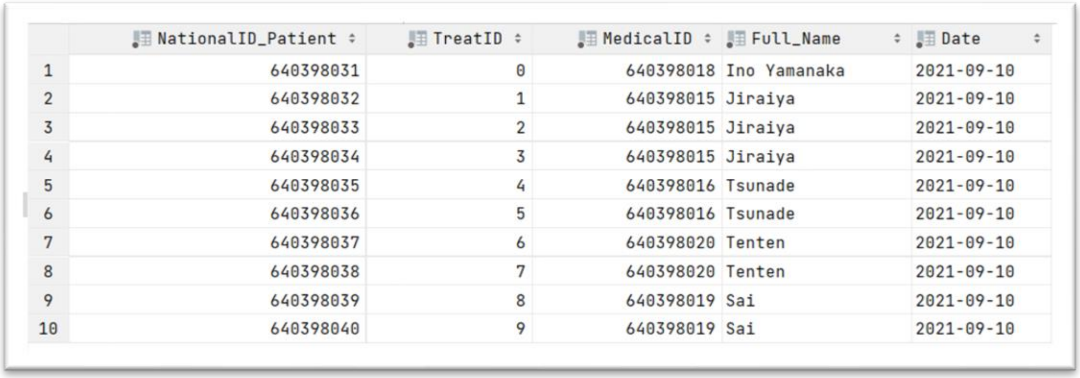
```
/*Shows which rooms are taken by which patients*/
CREATE VIEW Full_room AS SELECT Room.Room_Type, Room.RoomID, Stay.Duration,
Treats.NationalID_Patient
FROM Room INNER JOIN Stay ON Stay.RoomID = Room.RoomID
INNER JOIN Treats on Stay.TreatID = Treats.TreatID;
```



	Room_Type	RoomID	Duration	NationalID_Patient
1	vip	700	1	640398031
2	vip	701	1	640398032
3	vip	702	1	640398033
4	vip	703	1	640398034
5	vip	704	5	640398035
6	normal	705	5	640398036
7	normal	706	1	640398037
8	normal	707	1	640398038
9	normal	708	2	640398039
10	normal	709	2	640398040

II. This is a view for showing that which physicians treated a certain patient

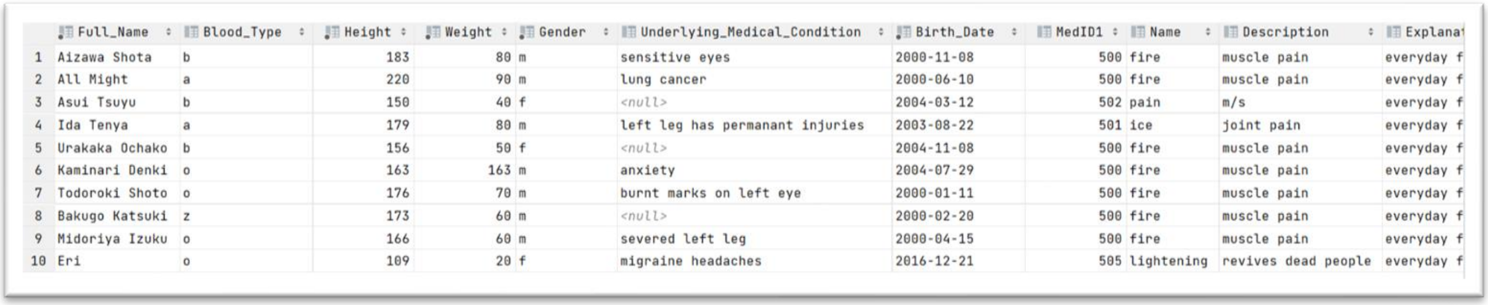
```
/*Shows which physician treated which patient*/
CREATE VIEW Patients_physicians AS SELECT Patient.NationalID_Patient, Treats.TreatID,
Suggests.MedicalID, Physicians.Full_Name, Treatment.Date
FROM Patient INNER JOIN Treats ON Treats.NationalID_Patient = Patient.NationalID_Patient
INNER JOIN Suggests on Treats.TreatID = Suggests.TreatID
INNER JOIN Physicians on Suggests.MedicalID = Physicians.MedicalID
INNER JOIN treatment on Suggests.TreatID = Treatment.TreatID;
```



	NationalID_Patient	TreatID	MedicalID	Full_Name	Date
1	640398031	0	640398018	Ino Yamanaka	2021-09-10
2	640398032	1	640398015	Jiraiya	2021-09-10
3	640398033	2	640398015	Jiraiya	2021-09-10
4	640398034	3	640398015	Jiraiya	2021-09-10
5	640398035	4	640398016	Tsunade	2021-09-10
6	640398036	5	640398016	Tsunade	2021-09-10
7	640398037	6	640398020	Tenten	2021-09-10
8	640398038	7	640398020	Tenten	2021-09-10
9	640398039	8	640398019	Sai	2021-09-10
10	640398040	9	640398019	Sai	2021-09-10

III. This is a view for showing the health stats of a patient which makes it easier for nurses to assist patients.

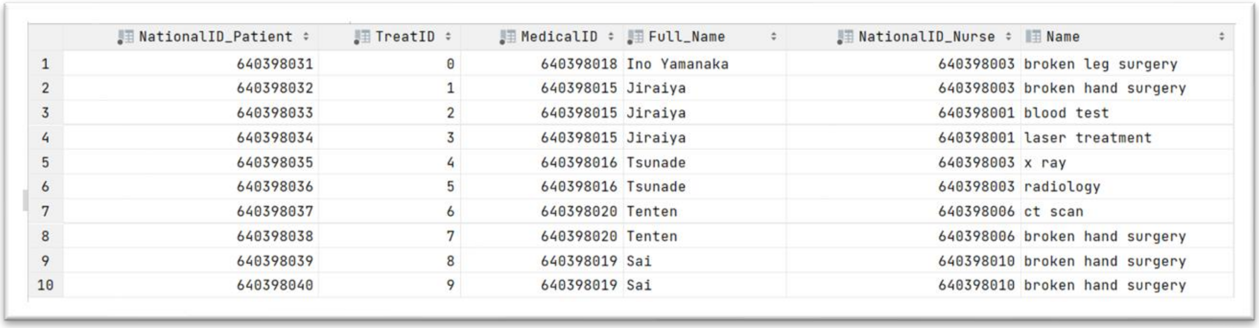
```
/*Health_profile_for_assist_nurse*/
CREATE VIEW Health_profile AS SELECT Patient.Full_Name, Patient.Blood_Type, Patient.Height,
Patient.Weight, Patient.Gender, Patient.Underlying_Medical_Condition,
                              Patient.Birth_Date, Prescription.MedID1, Medication.Name,
Medication.Description, Prescription.Explanation
FROM Patient INNER JOIN Treats ON Treats.NationalID_Patient = Patient.NationalID_Patient
INNER JOIN Prescription on Treats.TreatID = Prescription.TreatID
INNER JOIN Medication ON  Prescription.MedID1 = Medication.Code;
```



	Full_Name	Blood_Type	Height	Weight	Gender	Underlying_Medical_Condition	Birth_Date	MedID1	Name	Description	Explana
1	Aizawa Shota	b	183	80	m	sensitive eyes	2000-11-08	500	fire	muscle pain	everyday f
2	All Might	a	220	90	m	lung cancer	2000-06-10	500	fire	muscle pain	everyday f
3	Asui Tsuyu	b	150	40	f	<null>	2004-03-12	502	pain	m/s	everyday f
4	Ida Tenya	a	179	80	m	left leg has permanant injuries	2003-08-22	501	ice	joint pain	everyday f
5	Urakaka Ochako	b	156	50	f	<null>	2004-11-08	500	fire	muscle pain	everyday f
6	Kaminari Denki	o	163	163	m	anxiety	2004-07-29	500	fire	muscle pain	everyday f
7	Todoroki Shoto	o	176	70	m	burnt marks on left eye	2000-01-11	500	fire	muscle pain	everyday f
8	Bakugo Katsuki	z	173	60	m	<null>	2000-02-20	500	fire	muscle pain	everyday f
9	Midoriya Izuku	o	166	60	m	severed left leg	2000-04-15	500	fire	muscle pain	everyday f
10	Eri	o	109	20	f	migraine headaches	2016-12-21	505	lightening	revives dead people	everyday f

IV. This is a view that shows the stats of a procedure so we can easily take track of them.

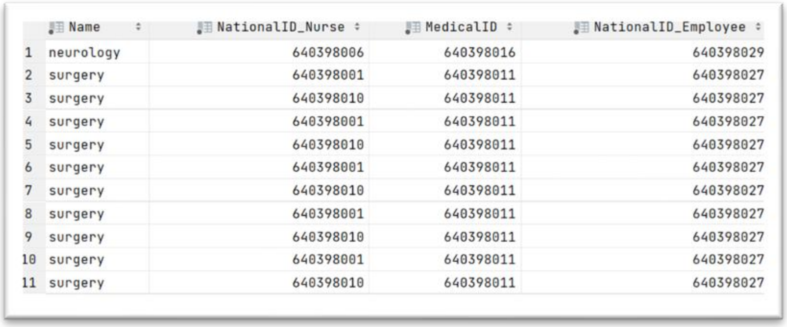
```
/*Procedure's status*/
CREATE VIEW Procedure_stat AS SELECT Patient.NationalID_Patient, Treats.TreatID,
Suggests.MedicalID, Physicians.Full_Name,
                    Assist.NationalID_Nurse, Pro.Name
FROM Patient INNER JOIN Treats ON Treats.NationalID_Patient = Patient.NationalID_Patient
INNER JOIN Suggests on Treats.TreatID = Suggests.TreatID
INNER JOIN Physicians on Suggests.MedicalID = Physicians.MedicalID
INNER JOIN Assist ON  Assist.TreatID = Treats.TreatID
INNER JOIN Type ON Type.TreatID = Treats.TreatID
INNER JOIN Pro ON Pro.Code = Type.Code;
```



	NationalID_Patient	TreatID	MedicalID	Full_Name	NationalID_Nurse	Name
1	640398031	0	640398018	Ino Yamanaka	640398003	broken leg surgery
2	640398032	1	640398015	Jiraiya	640398003	broken hand surgery
3	640398033	2	640398015	Jiraiya	640398001	blood test
4	640398034	3	640398015	Jiraiya	640398001	laser treatment
5	640398035	4	640398016	Tsunade	640398003	x ray
6	640398036	5	640398016	Tsunade	640398003	radiology
7	640398037	6	640398020	Tenten	640398006	ct scan
8	640398038	7	640398020	Tenten	640398006	broken hand surgery
9	640398039	8	640398019	Sai	640398010	broken hand surgery
10	640398040	9	640398019	Sai	640398010	broken hand surgery

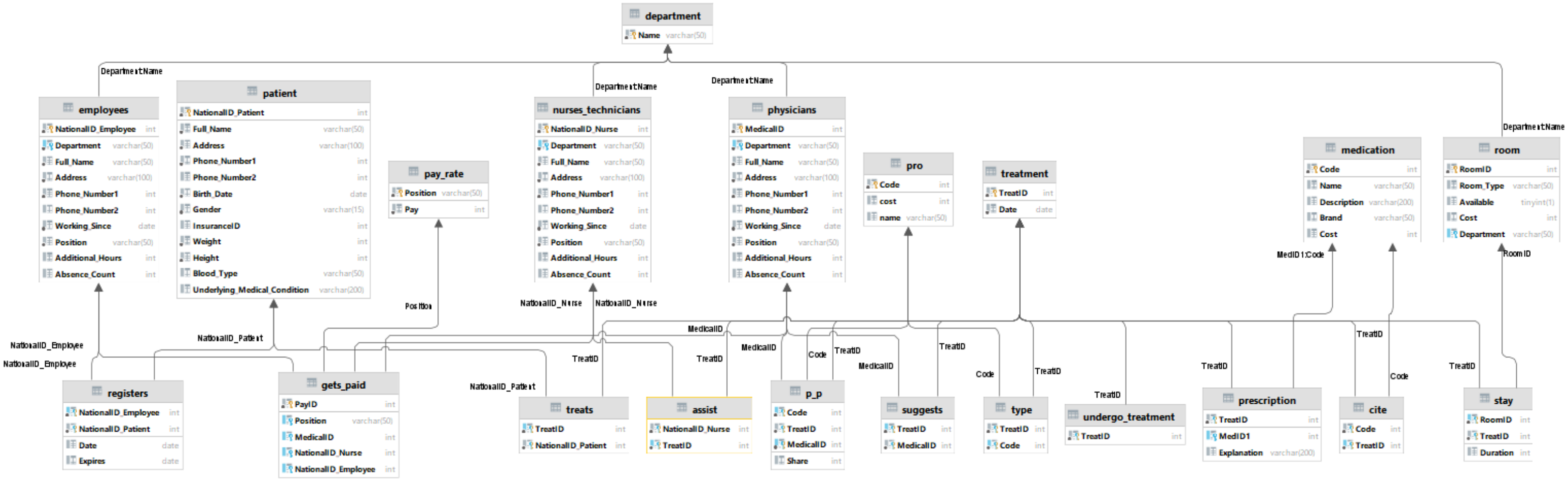
V. This is a view to all members of a certain department

```
/*Department status*/
CREATE VIEW Department_stats AS SELECT Department.Name,
Nurses_Technicians.NationalID_Nurse, Physicians.MedicalID,
                    Employees.NationalID_Employee, Room.Department
FROM Department INNER JOIN Nurses_Technicians ON Nurses_Technicians.Department =
Department.Name
INNER JOIN Physicians on Physicians.Department = Department.Name
INNER JOIN Employees on Employees.Department = Department.Name
INNER JOIN Room  ON  Room.Department = Department.Name;
```



	Name	NationalID_Nurse	MedicalID	NationalID_Employee
1	neurology	640398006	640398016	640398029
2	surgery	640398001	640398011	640398027
3	surgery	640398010	640398011	640398027
4	surgery	640398001	640398011	640398027
5	surgery	640398010	640398011	640398027
6	surgery	640398001	640398011	640398027
7	surgery	640398010	640398011	640398027
8	surgery	640398001	640398011	640398027
9	surgery	640398010	640398011	640398027
10	surgery	640398001	640398011	640398027
11	surgery	640398010	640398011	640398027

13)

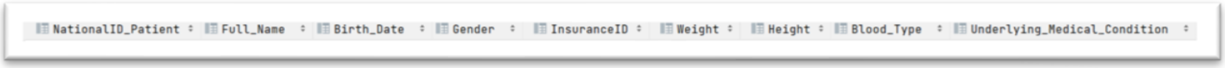


14) Triggers

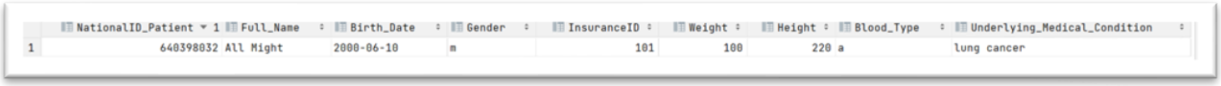
I. My first trigger is set in a way to stop my database from losing the data it has on patients, in other words some information of the patients are never deleted since they are invaluable to hospitals.

```
CREATE TABLE data_keep
  (
    NationalID_Patient int,
    Full_Name varchar(50),
    Birth_Date date,
    Gender varchar(15),
    InsuranceID int,
    Weight int,
    Height int,
    Blood_Type varchar(50),
    Underlying_Medical_Condition varchar(200),
    foreign key (NationalID_Patient) references patient(NationalID_Patient) on DELETE
cascade
  );

delimiter //
CREATE TRIGGER patient_data_keep
  before update on patient
  for each row
  begin
```

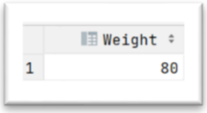
```
    insert into data_keep values (OLD.NationalID_Patient, OLD.Full_Name, OLD.Birth_Date,
OLD.Gender, OLD.InsuranceID
     , OLD.Weight, OLD.Height, OLD.Blood_Type, OLD.Underlying_Medical_Condition);
    end //

select * from data_keep;

update patient set Weight = 80 where NationalID_Patient = 640398032;

select * from data_keep;
select Weight from patient where NationalID_Patient = 640398032;

update patient set Weight = 90 where NationalID_Patient = 640398032;

select * from data_keep;
select Weight from patient where NationalID_Patient = 640398032;
```



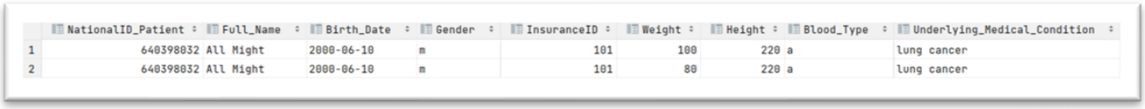
NationalID_Patient	Full_Name	Birth_Date	Gender	InsuranceID	Weight	Height	Blood_Type	Underlying_Medical_Condition



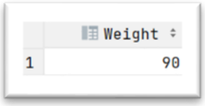
	NationalID_Patient ▾ 1	Full_Name	Birth_Date	Gender	InsuranceID	Weight	Height	Blood_Type	Underlying_Medical_Condition
1	640398032	All Might	2000-06-10	m	101	100	220	a	lung cancer



	Weight
1	80



	NationalID_Patient	Full_Name	Birth_Date	Gender	InsuranceID	Weight	Height	Blood_Type	Underlying_Medical_Condition
1	640398032	All Might	2000-06-10	m	101	100	220	a	lung cancer
2	640398032	All Might	2000-06-10	m	101	80	220	a	lung cancer

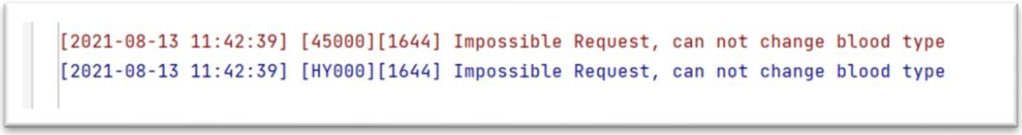


	Weight
1	90

II. This Trigger stops anyone from changing the blood types of the patients since it is impossible for a person to have a change in blood type.

```
delimiter //
create trigger unchangeable
   after update
   on patient for each row
   begin
```

```
    if OLD.Blood_Type <> NEW.Blood_Type then
    signal sqlstate '45000' set message_text = 'Impossible Request, can not change blood type';
    end if;
  END //
delimiter ;
update patient set Blood_Type = 'ab' where NationalID_Patient = 640398032;
```

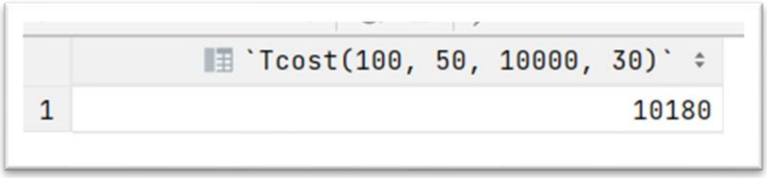


```
[2021-08-13 11:42:39] [45000][1644] Impossible Request, can not change blood type
[2021-08-13 11:42:39] [HY000][1644] Impossible Request, can not change blood type
```

15) Functions

I. The final payment the patients must make can be calculated by the function bellow

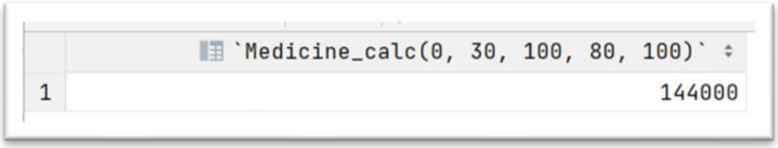
```
delimiter //
create function Tcost(prescription_cost int, stay_cost int, procedure_cost int, visit_cost int) returns
int deterministic
begin
  return (prescription_cost + stay_cost + procedure_cost + visit_cost);
end //
delimiter ;
select Tcost(100, 50, 10000, 30);
```



```
     `Tcost(100, 50, 10000, 30)`
1                          10180
```

II. The function bellow calculated the speed of serum that nurses must adjust

```
delimiter //
create function Medicine_calc(ML_med_per_hour int, dosage int, V_microset int, patient_weight int,
med_entity_microset int)
returns float deterministic
begin
  set Ml_med_per_hour = (dosage * v_microset * patient_weight * 60)/med_entity_microset;
  return (ML_med_per_hour);
end //
delimiter ;
select Medicine_calc(0, 30, 100, 80, 100);
```

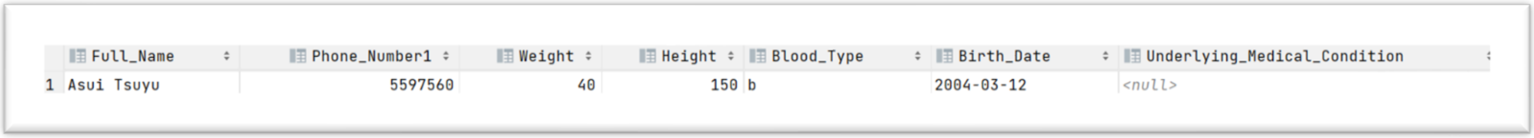


```
     `Medicine_calc(0, 30, 100, 80, 100)`
1                                 144000
```

16) Stored procedure

I. Finds the patients and their important attributes by their national ID

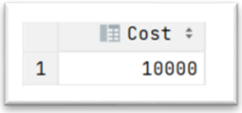
```
delimiter //
create procedure National_ID_finder (National_ID int)
begin
   select Full_Name, Phone_Number1, Weight, Height, Blood_Type, Birth_Date,
Underlying_Medical_Condition
   from patient
   where NationalID_Patient = National_ID;
end //
delimiter ;

call National_ID_finder(640398033);
```



	Full_Name	Phone_Number1	Weight	Height	Blood_Type	Birth_Date	Underlying_Medical_Condition
1	Asui Tsuyu	5597560	40	150	b	2004-03-12	<null>

II. Finds the price od procedures using their code

```
delimiter //
create procedure Procedure_price_finder (Procedure_code int)
begin
   select Cost
   from Pro
   where pro.Code = Procedure_code;
end //
delimiter ;

call Procedure_price_finder(209);
```



	Cost
1	10000