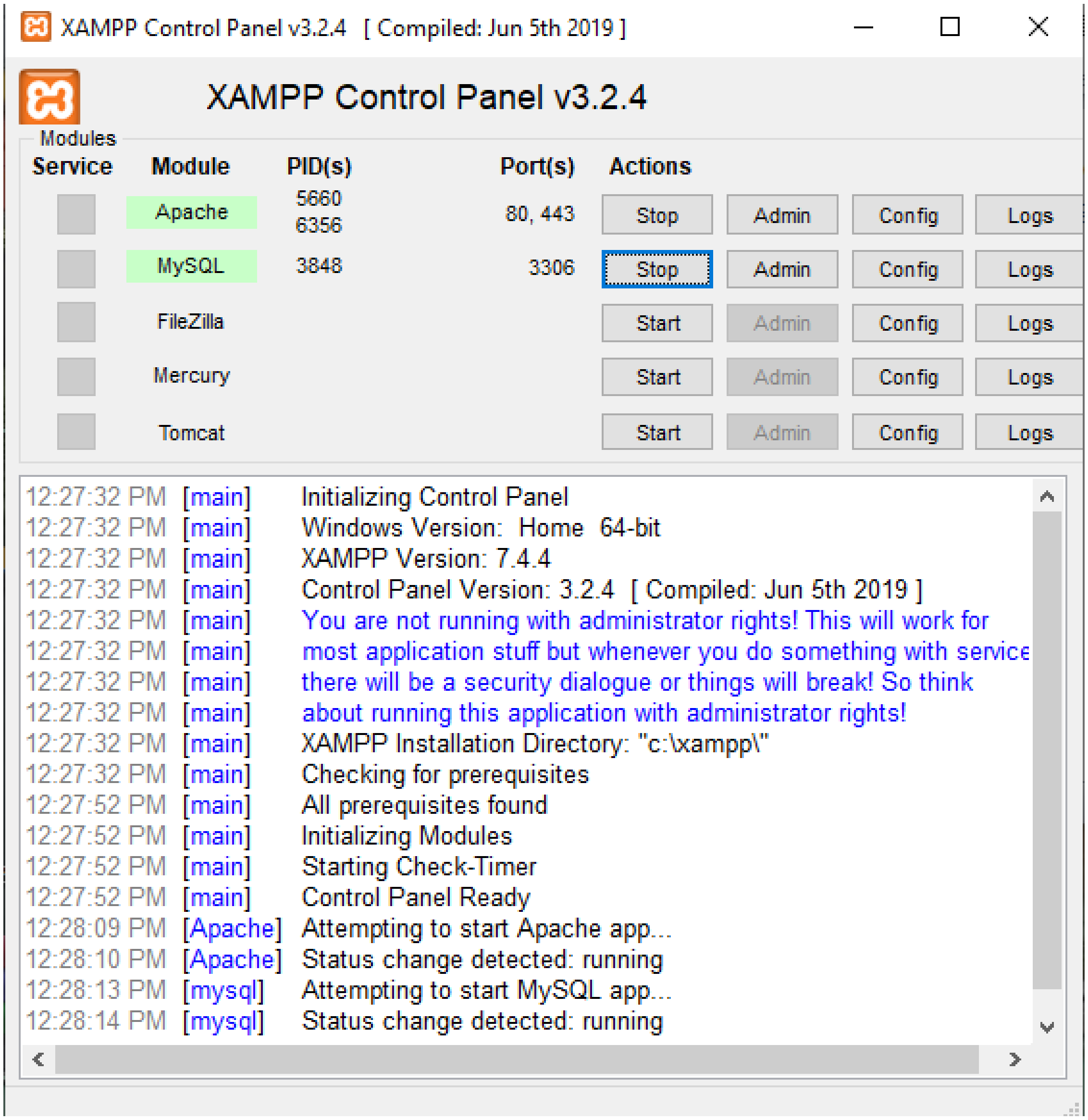
DATABASE PROJECT

HOSPITAL MANAGEMENT SYSTEM

91815855E109 & 91815855E107

Introduction

This is a Hospital Management System which was designed to hold data information about doctors, the patients, and employees   like nurses, ward boys, ambulance carriers which are considered as entities in the design system. All of this is setup in a website which is hosted by a PHP- server through a XAMP Software created by APACHE and FRIENDS: apachefriends.org/index.html


XAMPP Control Panel v3.2.4   [Compiled: Jun 5th 2019]
XAMPP Control Panel v3.2.4
Modules

Service	Module	PID(s)	Port(s)	Actions			
□	Apache	5660 6356	80, 443	Stop	Admin	Config	Logs
□	MySQL	3848	3306	Stop	Admin	Config	Logs
□	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
□	Tomcat			Start	Admin	Config	Logs

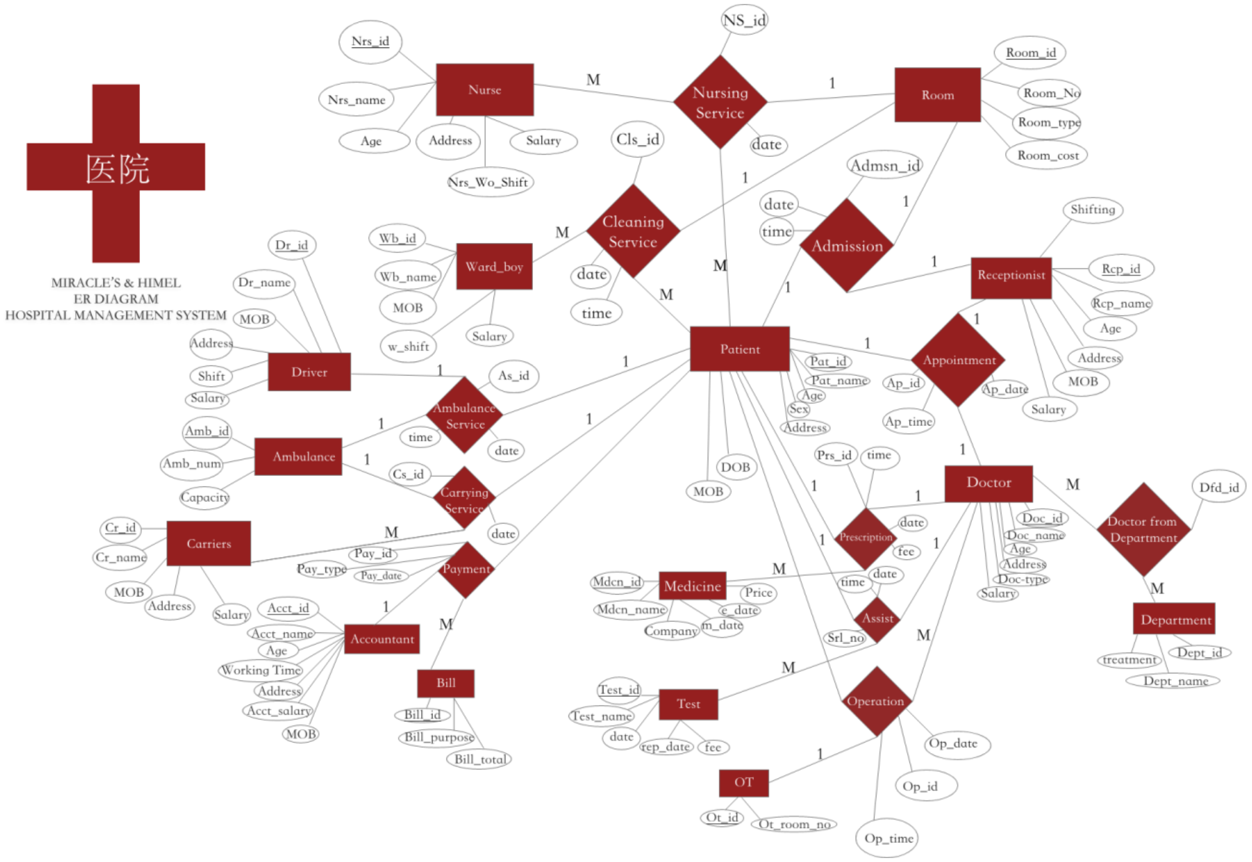
```
12:27:32 PM  [main]    Initializing Control Panel
12:27:32 PM  [main]    Windows Version:  Home  64-bit
12:27:32 PM  [main]    XAMPP Version: 7.4.4
12:27:32 PM  [main]    Control Panel Version: 3.2.4  [ Compiled: Jun 5th 2019 ]
12:27:32 PM  [main]    You are not running with administrator rights! This will work for
12:27:32 PM  [main]    most application stuff but whenever you do something with service
12:27:32 PM  [main]    there will be a security dialogue or things will break! So think
12:27:32 PM  [main]    about running this application with administrator rights!
12:27:32 PM  [main]    XAMPP Installation Directory: "c:\xampp\"
12:27:32 PM  [main]    Checking for prerequisites
12:27:52 PM  [main]    All prerequisites found
12:27:52 PM  [main]    Initializing Modules
12:27:52 PM  [main]    Starting Check-Timer
12:27:52 PM  [main]    Control Panel Ready
12:28:09 PM  [Apache]  Attempting to start Apache app...
12:28:10 PM  [Apache]  Status change detected: running
12:28:13 PM  [mysql]   Attempting to start MySQL app...
12:28:14 PM  [mysql]   Status change detected: running
```

Fig 1 the User Interface of the Software

With this Software, I was able to link My SQLite, PHP 7, CSS 3, and HTML5. To create a good website that has animation. NOTE: This is my first website!!!

THE ER-DIAGRAM

In the ER diagram, we can view the entities- Patient, Doctor, Receptionist, Department, Medicine, Test, OT (Operation Theater), Room, Nurse, Ward_boy, Driver, Ambulance, Carriers, Accountant and Bill. Among these entities, relationships exist which connect all the entities in the diagram. For example, Patient, Doctor and Receptionist are connected via the relationship Appointment. In other words, a receptionist will set up a doctor's appointment for a patient. Similarly, Doctor, Patient and Medicine are connected via the relationship Prescription. Here, a doctor may prescribe one or more medicine to a patient. In a similar way, other entities are connected via relationships in a meaningful way.



NOTE: This ER-diagram Comprises of both **Binary** and **Ternary** relationships.

FROM ER TO RELATIONS

RELATIONS

The entire ER diagram can be converted to a Relational Model as shown below. The attribute(s) of a relation which serve as a primary key of the table are underlined. Those attribute(s) which represent foreign keys are highlighted.

i) Patient (Pat_id, Pat_name, age, sex, Address, DOB, MOB)

Here DOB refers to patient's date of birth and MOB refers to mobile number.

ii) Room (Room_id, Room_No, Room_type, Room_cost)

iii) Admission (admsn_id, **Pat_id, Room_id**, **Rcp_id**, admsn_date, admsn_time)

This is a relationship table among Patient, Room and Receptionist tables. Primary key of the Patient table goes to Admission table as foreign key. Primary key of the Room table goes to Admission table as foreign key. Primary key of the Receptionist table goes to Admission table as foreign key. Admsn_id is a primary key in the Admission table. Since the relationship between Patient to Receptionist to Room is 1 to 1 to 1, Pat_Id from Patient table, Room_Id from Room table and Rcp_Id from Receptionist table become foreign keys in the Admission relationship table.

iv) Receptionist(rcp_id, rcp_name, Age, Address,Salary, MOB, Shifting)

Here Shifting refers to morning, afternoon or night shifts.

v) Doctor(Doc_id, Doc_name, Age, Address, Salary, MOB)

vi) Appointment (Ap_id, **Pat_id**, **Doc_id**, **Rcp_id**, Ap_date,Ap_time)

This is a Relationship table between Patient, Receptionist & Doctor tables.

vii) Bill(Bill_id, Bill_purpose, Bill_total)

Here Bill_purpose refers to the cause e.g blood test for which the bill is paid.

viii) Accountant(Acct_id, Acct_name, Age, Address, MOB, Working_time, Acct_salary)

ix)Payment (Pay_id, Bill_id, **Pat_id, Acct_id**, Pay_type, Pay_date)

This is a relationship table between Patient, Bill & Accountant tables. Primary key of the Patient Table goes to Payment table as foreign key. Primary key of the Accountant Table goes to

Payment table as foreign key. Primary Key of the Bill Table becomes a part of the primary key in Payment table. (Pay_id, Bill_id) is the primary key in the Payment table. Since the relationship between Patient to Accountant to Bill is 1 to 1 to M (which stands for many), Pat_Id from Patient table and Acct_id from Accountant table become foreign Keys in the Payment table while Bill_id from Bill table becomes a part of the primary key in the Payment table. In this table, pay_type refers to the type of payment such as cash, pay order, check or credit card.

x) Medicine(Mdcn_id, Mdcn_name, company, m_date, e_date, price)

m_date refers to manufacture date. e_date refers to expiry date.

xi) Prescription (Prs_id, **Doc_id**, Mdcn_id, **Pat_id**, date, fee)

This is a relationship table between Patient, Doctor & Medicine tables.

xii)Test(Test_id, Test_name, date, rep_date, fee)

Here rep-date refers to the date the report of the test will be given.

xiii) Assist(Srl_no, **Pat_id**, **Doc_id**, Test_id,Ass_ time, Ass_ date)

This is a relationship table between Patient, Doctor & Test tables. Srl_no corresponds to serial number of conducted test.

xiv) OT (Ot_id, Ot-room_no)

Here OT refers to Operation Theater.

xv) Operation(Op_id, Doc_id, **Pat_id**, **Ot_id**, Op_date, Op_time)

This is a relationship table between Patient, Doctor & OT tables.

xvi) Department(Dept_id, Dept_name, treatment)

xvii) Doctor_from_Department(Dfd_id, **Doc_id**, Dept_id)

This is a relationship table between Doctor & Department tables.

xviii) Nurse(Nrs_id, Nrs_name, Age, Address, MoB, Nrs_wo_shift,salary)

Here Nrs_wo_shift refers to the working shift of the nurse such as morning, afternoon or night shifts.

xix) Nursing_Service(Ns_id, Pat_id , Nrs_id, **Room_id**)

This is a relationship table between Patient, Room & Nurse tables.

xx) Ward_Boy(Wb_id, Wb_name, MoB, w_shift, Salary)

Here w_shift refers to working shift of a ward boy.

xxi) Cleaning _Service(Cls_id, Pat_id, Wb_id, **Room_id**)

This is a relationship table between Patient , Room & Ward_Boy tables.

xxii) Driver(Dr_id, Dr_name, MoB, Address, Shift, Salary)

xxiii) Ambulance(Amb_id, Amb_num, Capacity)

xxiv) Ambulance_Service(As_id, **Pat_id**, **Dr_id**, **Amb_id**)

This is a relationship table between Patient, Driver & Ambulance tables.

xxv) Carriers(Cr_id, Cr_name, MOB, Address, Salary)

Carriers are those persons who carry patients from the ambulance to the hospital premises.
xxvi) Carrying_Service (Cs_id, Cr_id, **Amb_id**, **Pat_id**)

This is a relationship table between Patient, Ambulance & Carriers tables.

FUNCTIONAL DEPENDENCIES

Most of the Functional Dependencies are of keys to their attributes with exemption of those below:

Room Table:

Room(Room_id, Room_No, Room_type, Room_cost)

Room_id -> Room_No

Here Room-id corresponds to serial numbers like 1, 2, 3, 4 etc. Room_No corresponds to designated numbers of rooms like 206, 307, 508 etc. A Functional Dependency exists from Room_No to Room_id because two different Room_Nos cannot correspond to the same Room_id. Similarly, the following functional dependencies exist:

Room_id ->Room_type

Room_id -> Room_cost

Note: Both relations are in Boyce Codd Normal Form.

Violation of Normal forms

Payment Table:

This is a relationship table between Patient, Bill & Accountant tables.
Payment (Pay_id, Pat_id, Bill_id, Acct_id, Pay_type,Pay_date)
For the above relation, the following functional dependencies exist:
Pay_id -> Pay_type, Pay_date, Pat_id
Two different patient ids, payment dates and payment types cannot correspond to the same payment id. So Pay_type, Paydate and Pat_id are fully functionally dependent on Pay_id.
Bill_id -> Acct_id , Pat_id
Similarly two different accountant ids and patient ids cannot correspond to the same bill id. So Acct_Id and Pat_id are fully functionally dependent on Bill_id.

Based on the above functional dependencies, I took the liberty to split the relation.
Payment1 (Pay_id, Pay_Type, Pay_date, **Pat_id**)
Payment2 (BP_id, **Bill_id, Acct_id**, **Pat_id**)
BP_id stands for Bill Pay id and corresponds to serial numbers of all bills such as 1, 2, 3 etc. in ascending order. Bill_id on the other hand corresponds to the bill code numbers.
The relations are in 3NF. The relations are in BCNF.

BOYCE CODD NORMAL FORM (BCNF):

With the correction made above in the functional dependencies. It is safe to say they are all in Boyce's Codd Normal Form.

SQL SCHEMA AND DATABASE CONTROL FILES

Schema Definition:

This Schema entails all the relations and attribute in respect to their data definition. They are written below. **Foreign keys will be identified by highlight.**

Patient(Pat_id:string,Pat_name:string,age:integer,sex:char,Address:string, DOB:date,MOB:integer);

Room(Room_id:string,Room_no:integer,Room_type:string,Room_cost:decimal);

Admission(admsn_id:string,**Pat_id**:string,**Room_id**:string,**Rcp_id**:string, admsn_date:date,admsn_time:time);

Receptionist(Rcp_id:string,Rcp_name:string,age:integer,Address:string, Salary:decimal,MOB:integer,Shifting:string);

Doctor(Doc_id:string,Doc_name:string,age:integer,Address:string, Salary:decimal,MOB:integer);

Appointment(Ap_id:string,**Pat_id**:string,**Doc_id**:string,**Rcp_id**:string,Ap_date:date,Ap_time:time);

Bill(Bill_id:string,Bill_purposes:string,Bill_total:decimal);

Accountant(Acct_id:string,Acct_name:string,Age:integer,Address:string,MOB:integer,Working_time:time,Acct_Salary:decimal);

Payment1(Pay_id:string,Pay_type:string,Pay_date:date,**Pat_id**:string);

Payment2(BP_id:string,**Bill_id**:string,**Acct_id**:string,**Pat_id**:string);

Medicine(Mdcn_id:string,Mdcn_name:string,company:string,m_date:date,e_date:date,price:decimal);

Prescription(Prs-id:string, **Doc_id:string**, Mdcn_id:string, **Pat_id**:string, fee:decimal);

Test(Test_id:string,Test_name:string,Test_date:date,rep_date:date,fee:decimal);

Assist(Srl_no:integer,**Pat_id**:string,**Doc_id**:string,Test_id:string,Ass_time:time,Ass_date:date);

OT(Ot_id:string,Ot_room_no:integer);

Operation(Op_id:string,Doc_id:string,**Pat_id**:string,**Ot_id:**string,Op_date:date, Op_time:time);

Department(Dept_id:string,Dept_name:string,treatment:string);

Doctor_from_Department(Dfd_id:string,**Doc_id**:string,Dept_id:string);

Nurse(Nrs_id:string,Nrs_name:string,age:int,Address:string,Salary:decimal,MOB:integer,Nrs_wo_shift:string);

Nursing_Service(Ns_id:string,Nrs_id:string,Pat_id:string,**Room_id**:string);

Ward_Boy(Wb_id:string,Wb_name:string,MOB:integer,w_shift:string);

Cleaning_Service(Cls_id:string,Pat_id:string,Wb_id:string,**Room_id**:string);

Driver(Dr_id:string,Dr_name:string,Address:string,Salary:decimal,MOB:integer, Shift:string);

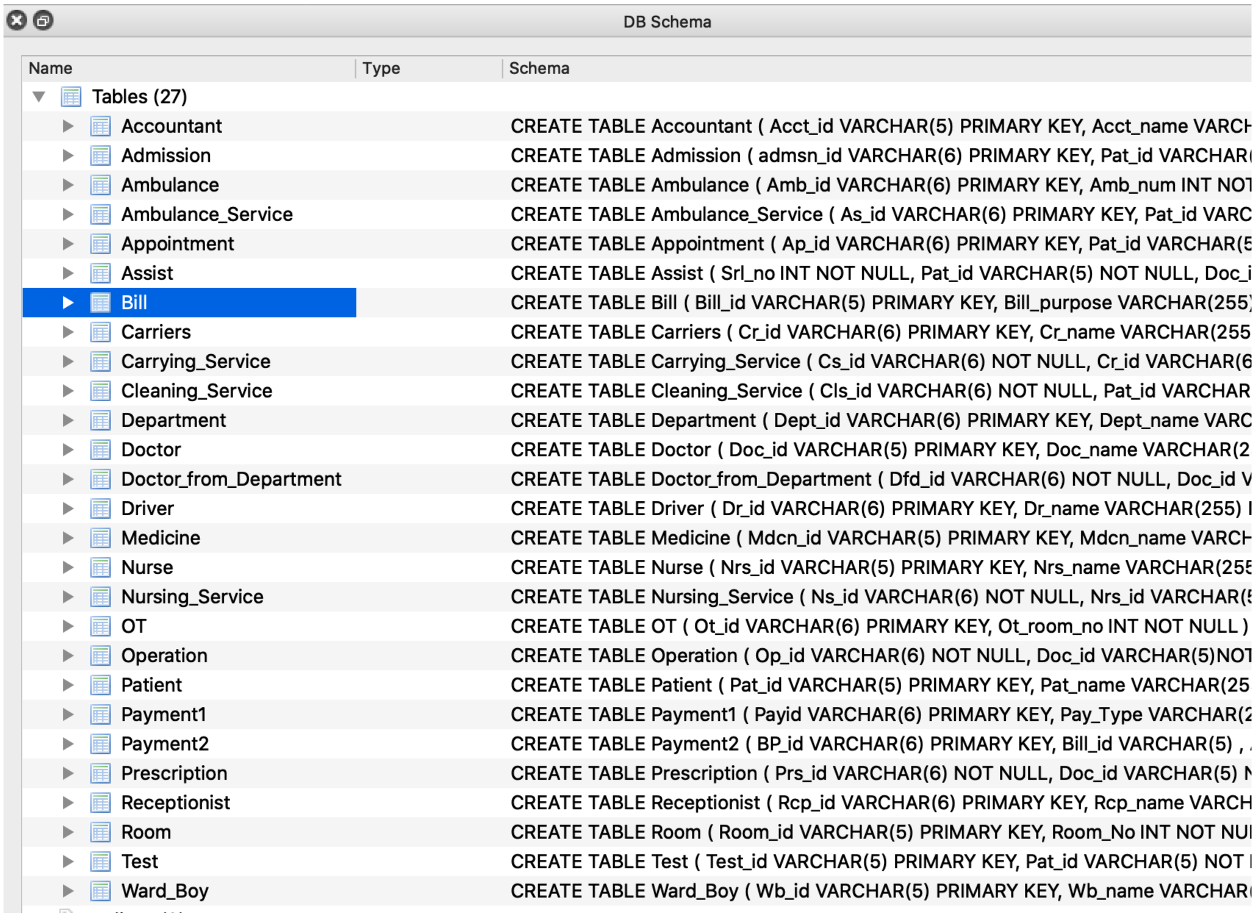
Ambulance(Amb_id:string,Amb_num:integer,Capacity:integer);

Ambulance_Service(As_id:string,**Pat_id**:string,**Dr_id**:string,**Amb_id**:string);

Carriers(Cr_id:integer,Cr_name:string,Address:string,Salary:decimal,MOB:integr);

Carrying_Service(Cs_id:string,Cr_id:string,**Amb_id**:integer,**Pat_id**:string);

THE DIGITAL VERSION IS SHOWN BELOW



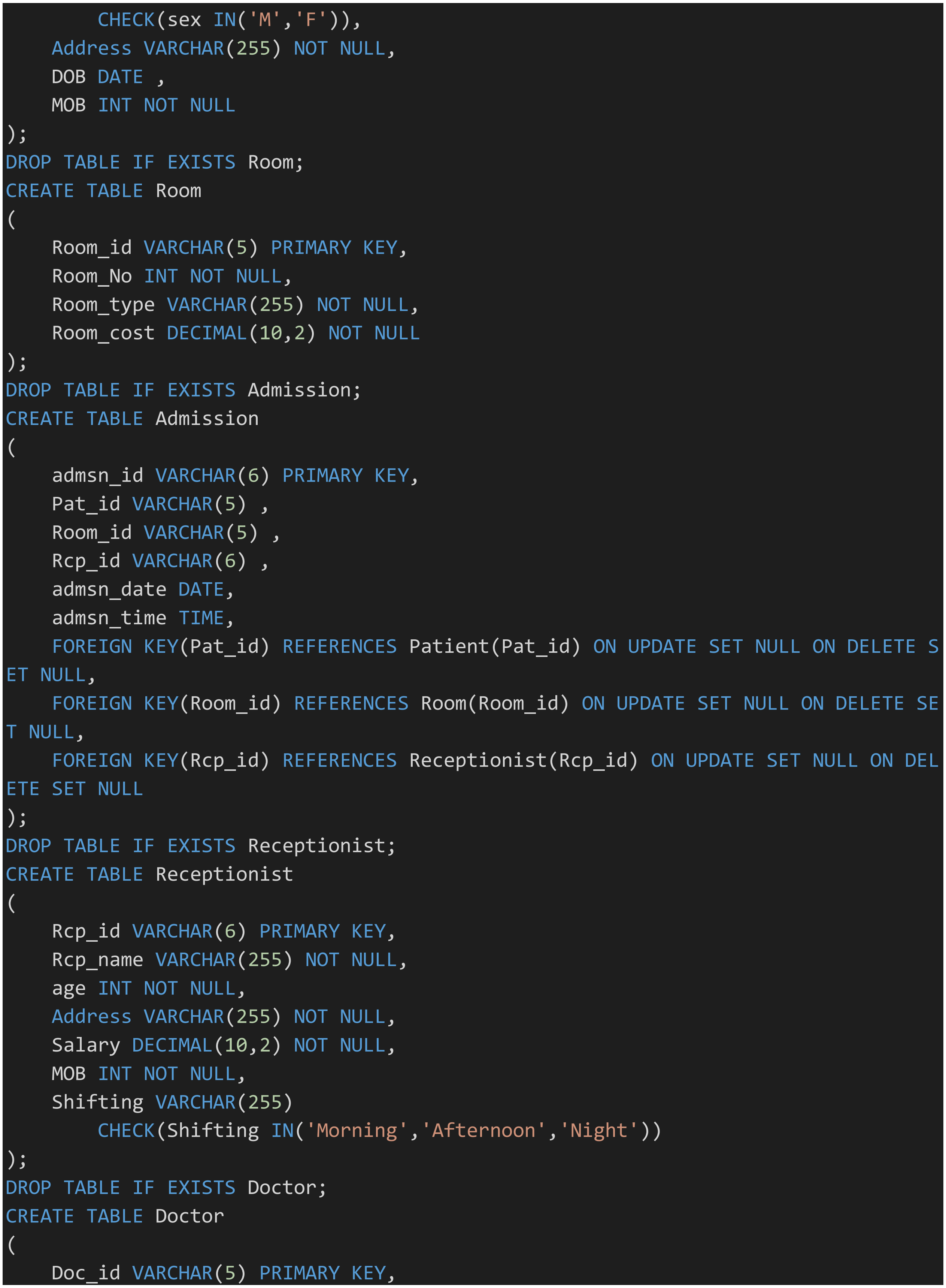
DB Schema

Name	Type	Schema
Tables (27)		
Accountant		CREATE TABLE Accountant (Acct_id VARCHAR(5) PRIMARY KEY, Acct_name VARCH
Admission		CREATE TABLE Admission (admsn_id VARCHAR(6) PRIMARY KEY, Pat_id VARCHAR
Ambulance		CREATE TABLE Ambulance (Amb_id VARCHAR(6) PRIMARY KEY, Amb_num INT NOT
Ambulance_Service		CREATE TABLE Ambulance_Service (As_id VARCHAR(6) PRIMARY KEY, Pat_id VARC
Appointment		CREATE TABLE Appointment (Ap_id VARCHAR(6) PRIMARY KEY, Pat_id VARCHAR(
Assist		CREATE TABLE Assist (Srl_no INT NOT NULL, Pat_id VARCHAR(5) NOT NULL, Doc_i
Bill		CREATE TABLE Bill (Bill_id VARCHAR(5) PRIMARY KEY, Bill_purpose VARCHAR(255
Carriers		CREATE TABLE Carriers (Cr_id VARCHAR(6) PRIMARY KEY, Cr_name VARCHAR(255
Carrying_Service		CREATE TABLE Carrying_Service (Cs_id VARCHAR(6) NOT NULL, Cr_id VARCHAR(
Cleaning_Service		CREATE TABLE Cleaning_Service (Cls_id VARCHAR(6) NOT NULL, Pat_id VARCHAR
Department		CREATE TABLE Department (Dept_id VARCHAR(6) PRIMARY KEY, Dept_name VARC
Doctor		CREATE TABLE Doctor (Doc_id VARCHAR(5) PRIMARY KEY, Doc_name VARCHAR(2
Doctor_from_Department		CREATE TABLE Doctor_from_Department (Dfd_id VARCHAR(6) NOT NULL, Doc_id V
Driver		CREATE TABLE Driver (Dr_id VARCHAR(6) PRIMARY KEY, Dr_name VARCHAR(255)
Medicine		CREATE TABLE Medicine (Mdcn_id VARCHAR(5) PRIMARY KEY, Mdcn_name VARCH
Nurse		CREATE TABLE Nurse (Nrs_id VARCHAR(5) PRIMARY KEY, Nrs_name VARCHAR(255
Nursing_Service		CREATE TABLE Nursing_Service (Ns_id VARCHAR(6) NOT NULL, Nrs_id VARCHAR(
OT		CREATE TABLE OT (Ot_id VARCHAR(6) PRIMARY KEY, Ot_room_no INT NOT NULL)
Operation		CREATE TABLE Operation (Op_id VARCHAR(6) NOT NULL, Doc_id VARCHAR(5)NOT
Patient		CREATE TABLE Patient (Pat_id VARCHAR(5) PRIMARY KEY, Pat_name VARCHAR(25
Payment1		CREATE TABLE Payment1 (Payid VARCHAR(6) PRIMARY KEY, Pay_Type VARCHAR(
Payment2		CREATE TABLE Payment2 (BP_id VARCHAR(6) PRIMARY KEY, Bill_id VARCHAR(5) ,
Prescription		CREATE TABLE Prescription (Prs_id VARCHAR(6) NOT NULL, Doc_id VARCHAR(5)
Receptionist		CREATE TABLE Receptionist (Rcp_id VARCHAR(6) PRIMARY KEY, Rcp_name VARCH
Room		CREATE TABLE Room (Room_id VARCHAR(5) PRIMARY KEY, Room_No INT NOT NU
Test		CREATE TABLE Test (Test_id VARCHAR(5) PRIMARY KEY, Pat_id VARCHAR(5) NOT
Ward_Boy		CREATE TABLE Ward_Boy (Wb_id VARCHAR(5) PRIMARY KEY, Wb_name VARCHAR

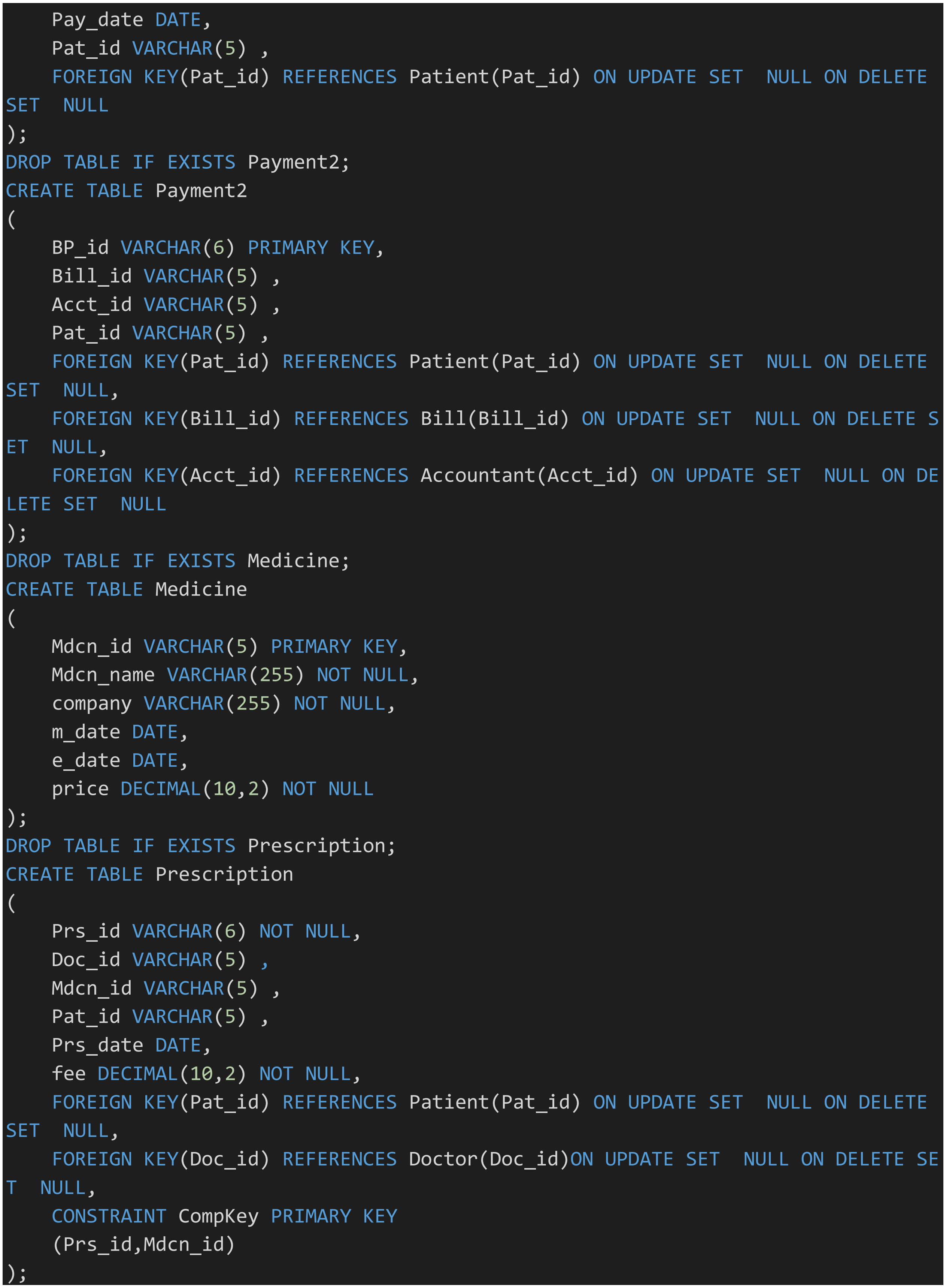
Control File

This is based upon the ER-diagram. The control file i.e. the source code for this database is thus shown below:

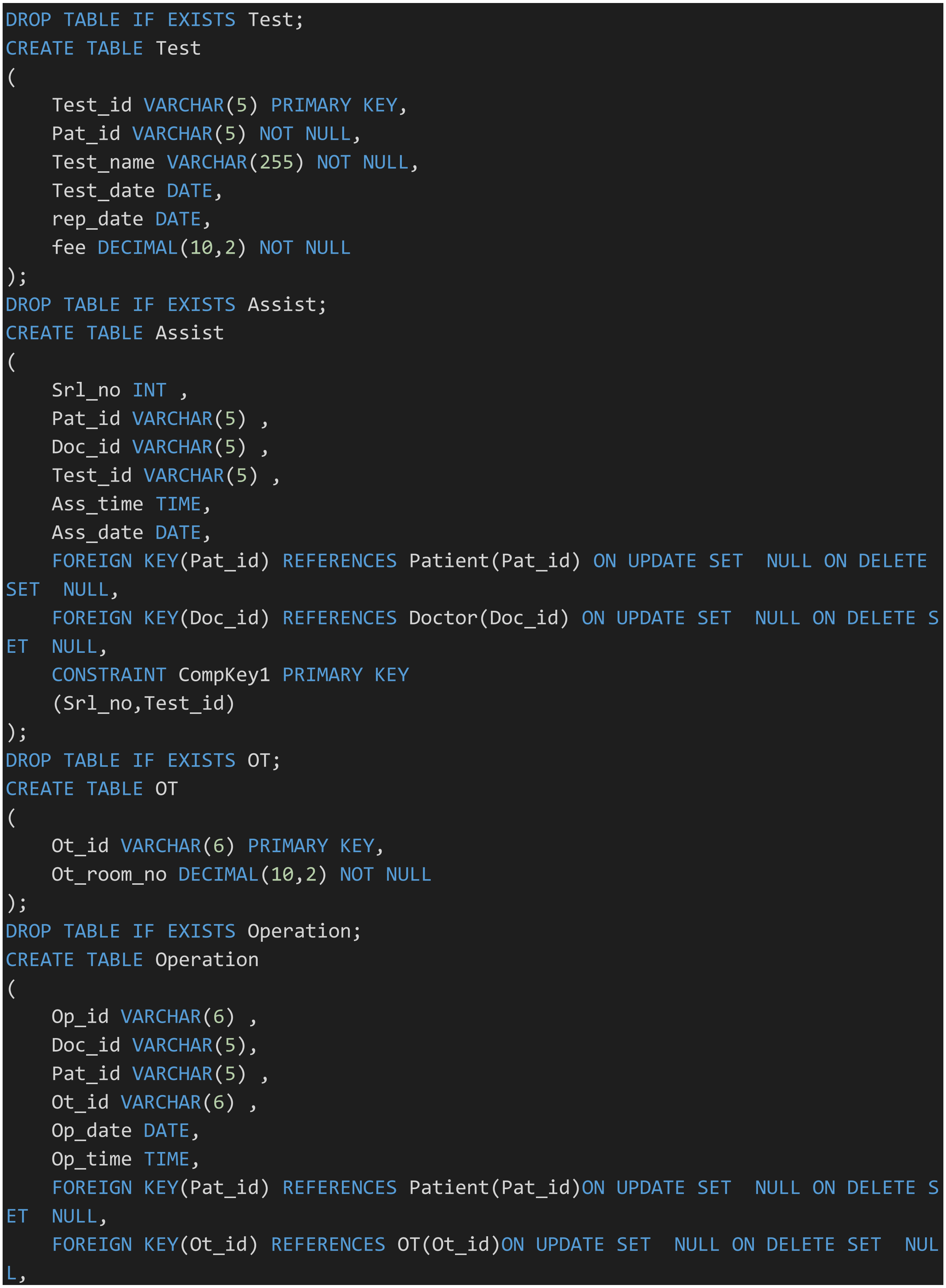
```
DROP TABLE IF EXISTS Patient;
CREATE TABLE Patient
(
    Pat_id VARCHAR(5) PRIMARY KEY,
    Pat_name VARCHAR(255) NOT NULL,
    age INT NOT NULL,
    sex VARCHAR(1)
```



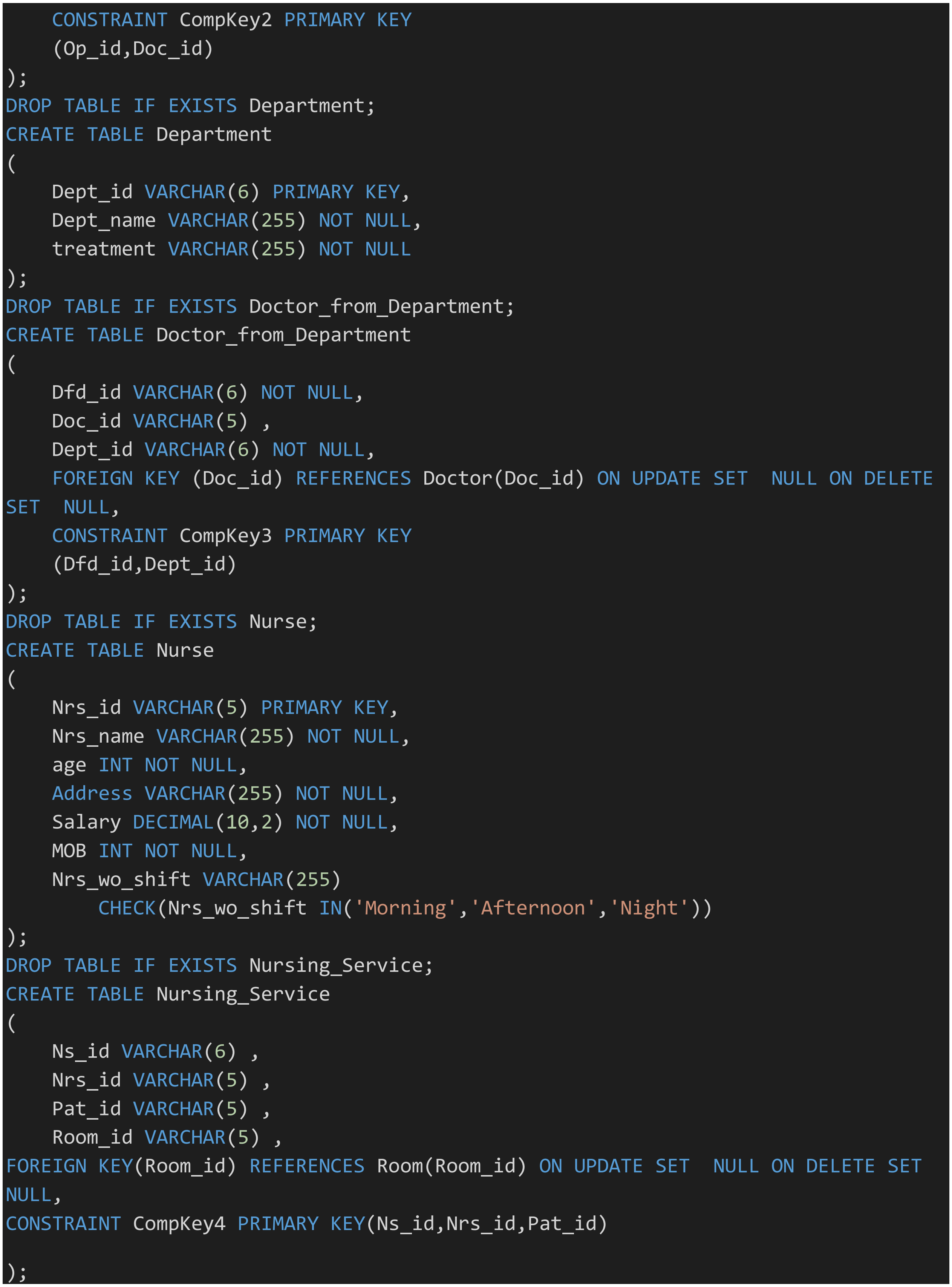
```
        CHECK(sex IN('M','F')),
    Address VARCHAR(255) NOT NULL,
    DOB DATE ,
    MOB INT NOT NULL
);
DROP TABLE IF EXISTS Room;
CREATE TABLE Room
(
    Room_id VARCHAR(5) PRIMARY KEY,
    Room_No INT NOT NULL,
    Room_type VARCHAR(255) NOT NULL,
    Room_cost DECIMAL(10,2) NOT NULL
);
DROP TABLE IF EXISTS Admission;
CREATE TABLE Admission
(
    admsn_id VARCHAR(6) PRIMARY KEY,
    Pat_id VARCHAR(5) ,
    Room_id VARCHAR(5) ,
    Rcp_id VARCHAR(6) ,
    admsn_date DATE,
    admsn_time TIME,
    FOREIGN KEY(Pat_id) REFERENCES Patient(Pat_id) ON UPDATE SET NULL ON DELETE SET NULL,
    FOREIGN KEY(Room_id) REFERENCES Room(Room_id) ON UPDATE SET NULL ON DELETE SET NULL,
    FOREIGN KEY(Rcp_id) REFERENCES Receptionist(Rcp_id) ON UPDATE SET NULL ON DELETE SET NULL
);
DROP TABLE IF EXISTS Receptionist;
CREATE TABLE Receptionist
(
    Rcp_id VARCHAR(6) PRIMARY KEY,
    Rcp_name VARCHAR(255) NOT NULL,
    age INT NOT NULL,
    Address VARCHAR(255) NOT NULL,
    Salary DECIMAL(10,2) NOT NULL,
    MOB INT NOT NULL,
    Shifting VARCHAR(255)
        CHECK(Shifting IN('Morning','Afternoon','Night'))
);
DROP TABLE IF EXISTS Doctor;
CREATE TABLE Doctor
(
    Doc_id VARCHAR(5) PRIMARY KEY,
    Doc_name VARCHAR(255),
    Age INT NOT NULL,
    Address VARCHAR(255) NOT NULL,
    Salary DECIMAL(10,2) NOT NULL ,
    MOB INT NOT NULL
);
DROP TABLE IF EXISTS Appointment;
CREATE TABLE Appointment
(
    Ap_id VARCHAR(6) PRIMARY KEY,
    Pat_id VARCHAR(5),
    Doc_id VARCHAR(5) ,
    Rcp_id VARCHAR(6) ,
    Ap_date DATE,
    Ap_time TIME,
    FOREIGN KEY(Pat_id) REFERENCES Patient(Pat_id) ON UPDATE SET  NULL ON DELETE
SET  NULL,
    FOREIGN KEY(Doc_id) REFERENCES Doctor(Doc_id) ON UPDATE SET  NULL ON DELETE S
ET  NULL,
    FOREIGN KEY(Rcp_id) REFERENCES Receptionist(Rcp_id)ON UPDATE SET  NULL ON DEL
ETE SET  NULL
);
DROP TABLE IF EXISTS Bill;
CREATE TABLE Bill
(
    Bill_id VARCHAR(5) PRIMARY KEY,
    Bill_purpose VARCHAR(255) NOT NULL,
    Bill_total DECIMAL(10,2) NOT NULL
);
DROP TABLE IF EXISTS Accountant;
CREATE TABLE Accountant
(
    Acct_id VARCHAR(5) PRIMARY KEY,
    Acct_name VARCHAR(255) NOT NULL,
    Age INT NOT NULL,
    Address VARCHAR(255) NOT NULL,
    MOB INT NOT NULL,
    Working_time TIME,
    Acct_Salary DECIMAL(10,2) NOT NULL
);
DROP TABLE IF EXISTS Payment1;
CREATE TABLE Payment1
(
    Payid VARCHAR(6) PRIMARY KEY,
    Pay_Type VARCHAR(255) NOT NULL,
```



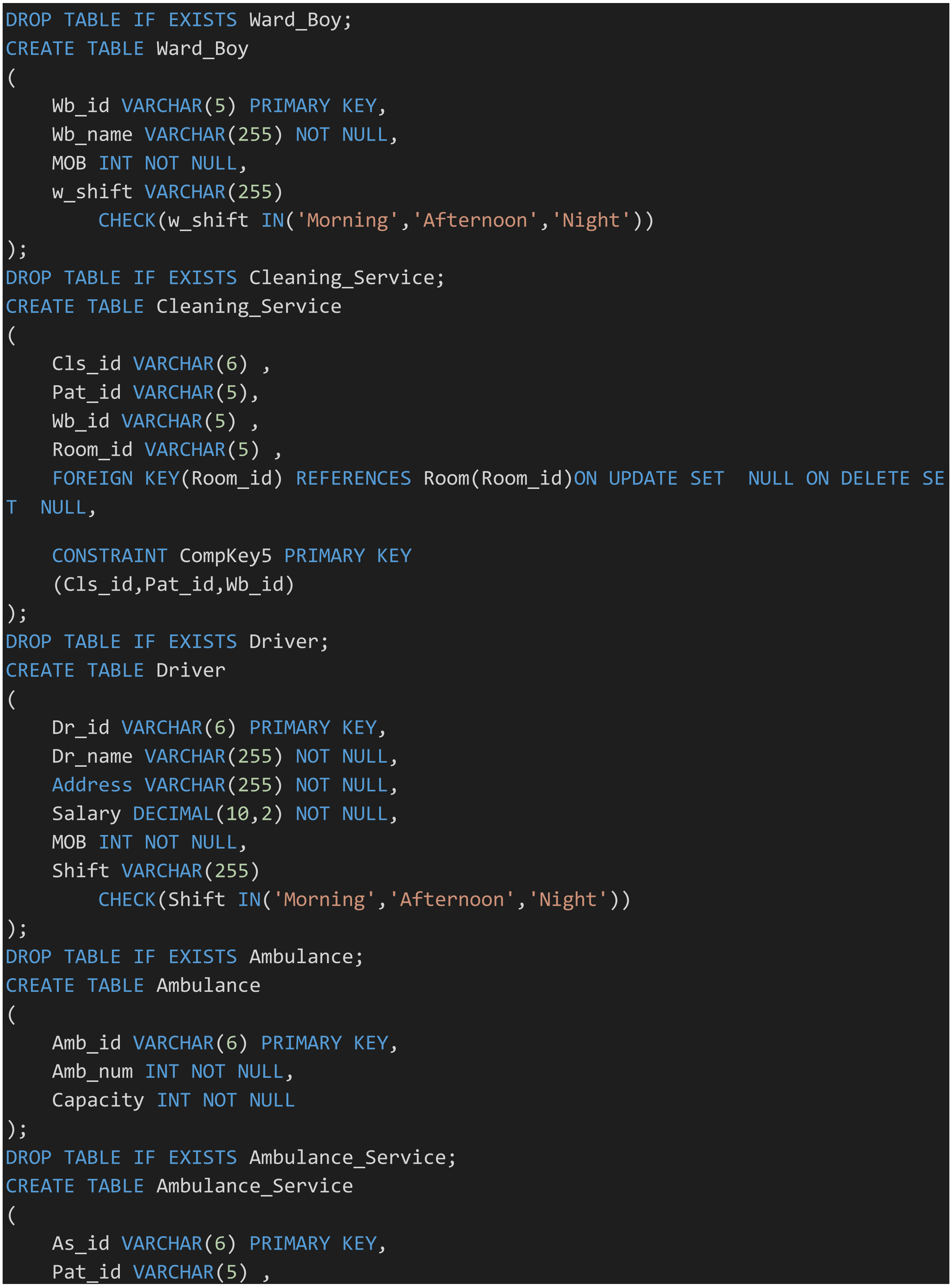
```
    Pay_date DATE,
    Pat_id VARCHAR(5) ,
    FOREIGN KEY(Pat_id) REFERENCES Patient(Pat_id) ON UPDATE SET  NULL ON DELETE
SET  NULL
);
DROP TABLE IF EXISTS Payment2;
CREATE TABLE Payment2
(
    BP_id VARCHAR(6) PRIMARY KEY,
    Bill_id VARCHAR(5) ,
    Acct_id VARCHAR(5) ,
    Pat_id VARCHAR(5) ,
    FOREIGN KEY(Pat_id) REFERENCES Patient(Pat_id) ON UPDATE SET  NULL ON DELETE
SET  NULL,
    FOREIGN KEY(Bill_id) REFERENCES Bill(Bill_id) ON UPDATE SET  NULL ON DELETE S
ET  NULL,
    FOREIGN KEY(Acct_id) REFERENCES Accountant(Acct_id) ON UPDATE SET  NULL ON DE
LETE SET  NULL
);
DROP TABLE IF EXISTS Medicine;
CREATE TABLE Medicine
(
    Mdcn_id VARCHAR(5) PRIMARY KEY,
    Mdcn_name VARCHAR(255) NOT NULL,
    company VARCHAR(255) NOT NULL,
    m_date DATE,
    e_date DATE,
    price DECIMAL(10,2) NOT NULL
);
DROP TABLE IF EXISTS Prescription;
CREATE TABLE Prescription
(
    Prs_id VARCHAR(6) NOT NULL,
    Doc_id VARCHAR(5) ,
    Mdcn_id VARCHAR(5) ,
    Pat_id VARCHAR(5) ,
    Prs_date DATE,
    fee DECIMAL(10,2) NOT NULL,
    FOREIGN KEY(Pat_id) REFERENCES Patient(Pat_id) ON UPDATE SET  NULL ON DELETE
SET  NULL,
    FOREIGN KEY(Doc_id) REFERENCES Doctor(Doc_id)ON UPDATE SET  NULL ON DELETE SE
T  NULL,
    CONSTRAINT CompKey PRIMARY KEY
    (Prs_id,Mdcn_id)
);
```



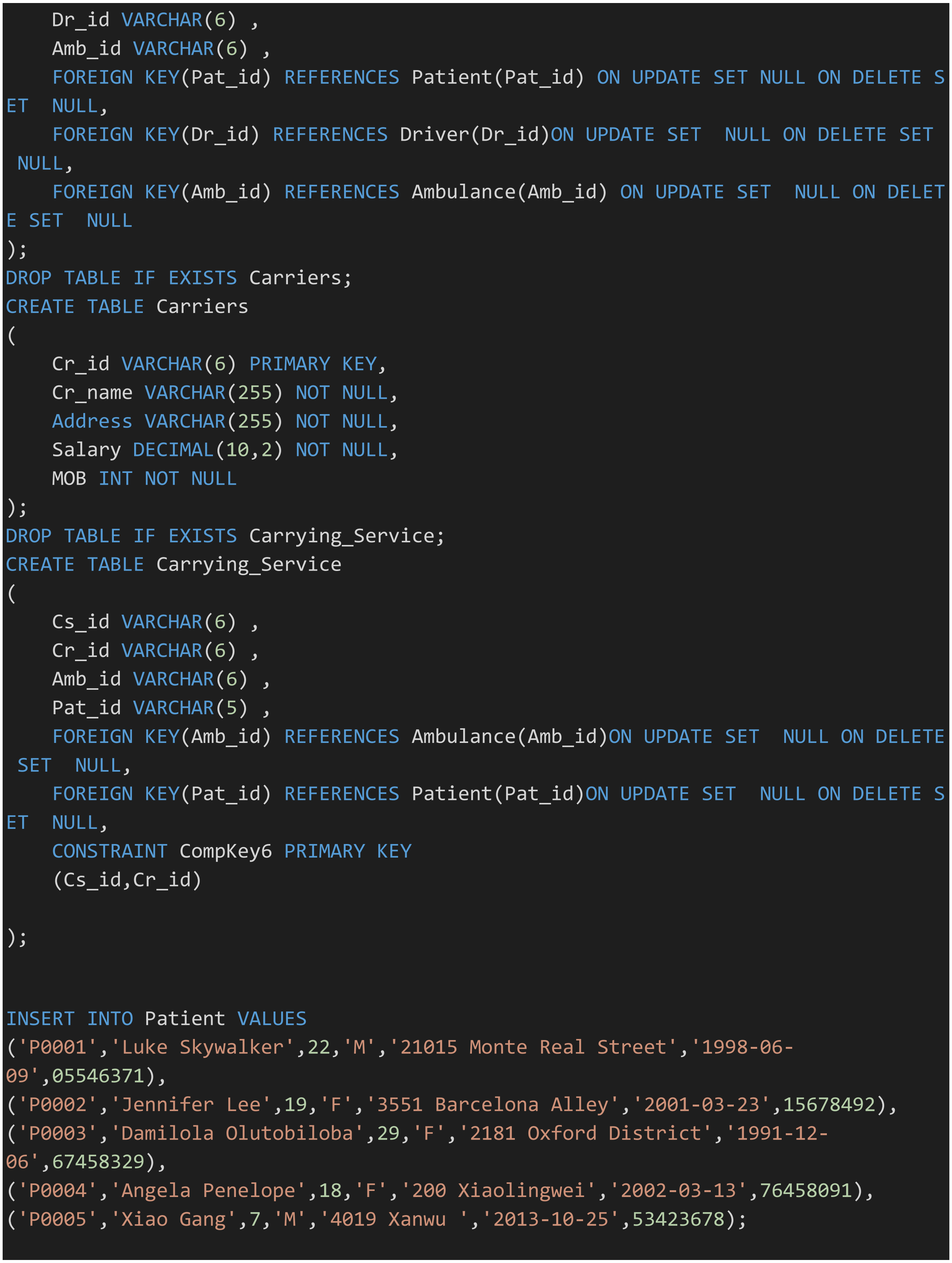
```sql
DROP TABLE IF EXISTS Test;
CREATE TABLE Test
(
    Test_id VARCHAR(5) PRIMARY KEY,
    Pat_id VARCHAR(5) NOT NULL,
    Test_name VARCHAR(255) NOT NULL,
    Test_date DATE,
    rep_date DATE,
    fee DECIMAL(10,2) NOT NULL
);
DROP TABLE IF EXISTS Assist;
CREATE TABLE Assist
(
    Srl_no INT ,
    Pat_id VARCHAR(5) ,
    Doc_id VARCHAR(5) ,
    Test_id VARCHAR(5) ,
    Ass_time TIME,
    Ass_date DATE,
    FOREIGN KEY(Pat_id) REFERENCES Patient(Pat_id) ON UPDATE SET  NULL ON DELETE SET  NULL,
    FOREIGN KEY(Doc_id) REFERENCES Doctor(Doc_id) ON UPDATE SET  NULL ON DELETE SET  NULL,
    CONSTRAINT CompKey1 PRIMARY KEY
    (Srl_no,Test_id)
);
DROP TABLE IF EXISTS OT;
CREATE TABLE OT
(
    Ot_id VARCHAR(6) PRIMARY KEY,
    Ot_room_no DECIMAL(10,2) NOT NULL
);
DROP TABLE IF EXISTS Operation;
CREATE TABLE Operation
(
    Op_id VARCHAR(6) ,
    Doc_id VARCHAR(5),
    Pat_id VARCHAR(5) ,
    Ot_id VARCHAR(6) ,
    Op_date DATE,
    Op_time TIME,
    FOREIGN KEY(Pat_id) REFERENCES Patient(Pat_id)ON UPDATE SET  NULL ON DELETE SET  NULL,
    FOREIGN KEY(Ot_id) REFERENCES OT(Ot_id)ON UPDATE SET  NULL ON DELETE SET  NULL,
```



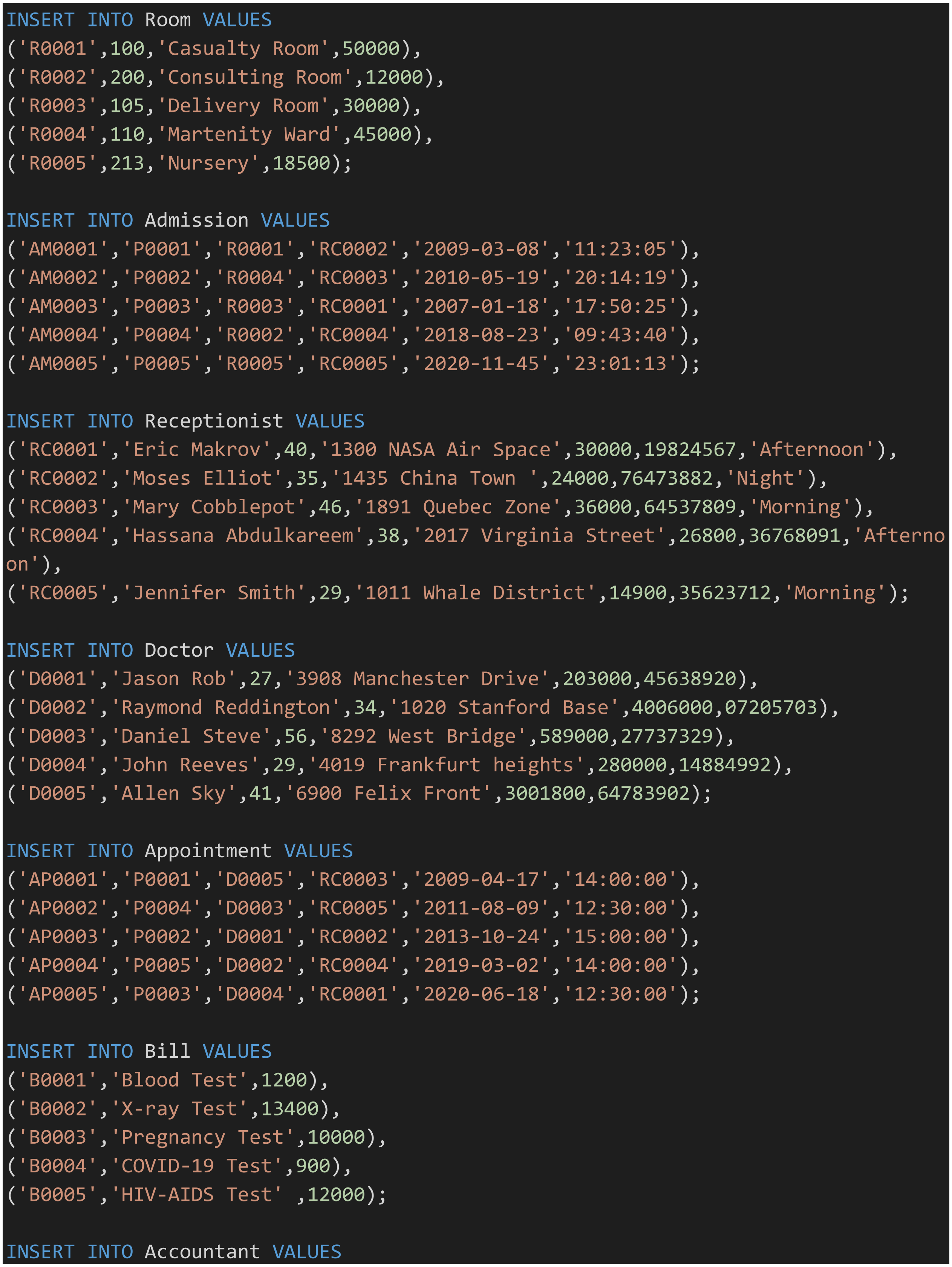
```
    CONSTRAINT CompKey2 PRIMARY KEY
    (Op_id,Doc_id)
);
DROP TABLE IF EXISTS Department;
CREATE TABLE Department
(
    Dept_id VARCHAR(6) PRIMARY KEY,
    Dept_name VARCHAR(255) NOT NULL,
    treatment VARCHAR(255) NOT NULL
);
DROP TABLE IF EXISTS Doctor_from_Department;
CREATE TABLE Doctor_from_Department
(
    Dfd_id VARCHAR(6) NOT NULL,
    Doc_id VARCHAR(5) ,
    Dept_id VARCHAR(6) NOT NULL,
    FOREIGN KEY (Doc_id) REFERENCES Doctor(Doc_id) ON UPDATE SET  NULL ON DELETE
SET  NULL,
    CONSTRAINT CompKey3 PRIMARY KEY
    (Dfd_id,Dept_id)
);
DROP TABLE IF EXISTS Nurse;
CREATE TABLE Nurse
(
    Nrs_id VARCHAR(5) PRIMARY KEY,
    Nrs_name VARCHAR(255) NOT NULL,
    age INT NOT NULL,
    Address VARCHAR(255) NOT NULL,
    Salary DECIMAL(10,2) NOT NULL,
    MOB INT NOT NULL,
    Nrs_wo_shift VARCHAR(255)
        CHECK(Nrs_wo_shift IN('Morning','Afternoon','Night'))
);
DROP TABLE IF EXISTS Nursing_Service;
CREATE TABLE Nursing_Service
(
    Ns_id VARCHAR(6) ,
    Nrs_id VARCHAR(5) ,
    Pat_id VARCHAR(5) ,
    Room_id VARCHAR(5) ,
FOREIGN KEY(Room_id) REFERENCES Room(Room_id) ON UPDATE SET  NULL ON DELETE SET
NULL,
CONSTRAINT CompKey4 PRIMARY KEY(Ns_id,Nrs_id,Pat_id)

);
```



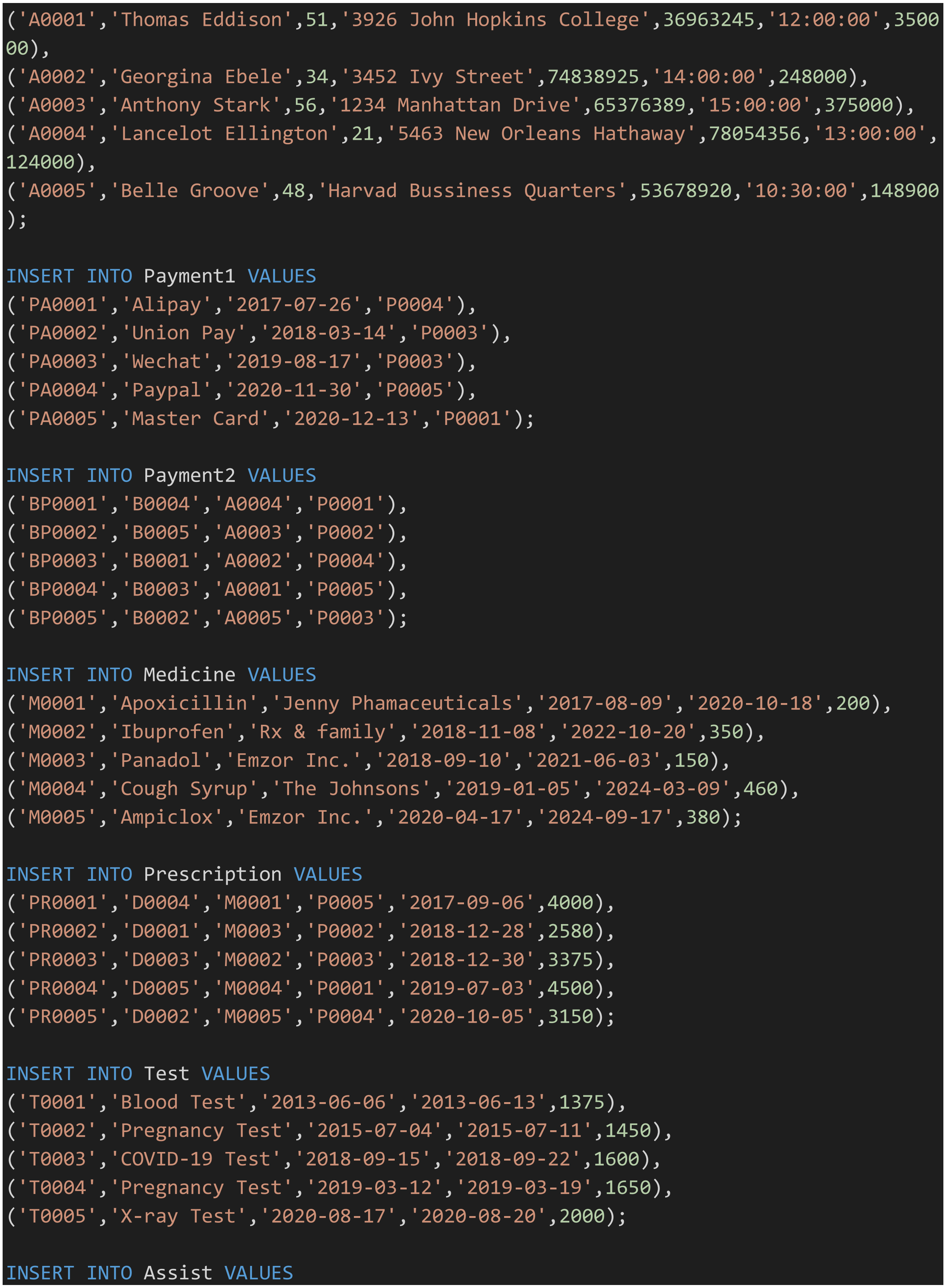
```sql
DROP TABLE IF EXISTS Ward_Boy;
CREATE TABLE Ward_Boy
(
    Wb_id VARCHAR(5) PRIMARY KEY,
    Wb_name VARCHAR(255) NOT NULL,
    MOB INT NOT NULL,
    w_shift VARCHAR(255)
        CHECK(w_shift IN('Morning','Afternoon','Night'))
);
DROP TABLE IF EXISTS Cleaning_Service;
CREATE TABLE Cleaning_Service
(
    Cls_id VARCHAR(6) ,
    Pat_id VARCHAR(5),
    Wb_id VARCHAR(5) ,
    Room_id VARCHAR(5) ,
    FOREIGN KEY(Room_id) REFERENCES Room(Room_id)ON UPDATE SET  NULL ON DELETE SE
T  NULL,

    CONSTRAINT CompKey5 PRIMARY KEY
    (Cls_id,Pat_id,Wb_id)
);
DROP TABLE IF EXISTS Driver;
CREATE TABLE Driver
(
    Dr_id VARCHAR(6) PRIMARY KEY,
    Dr_name VARCHAR(255) NOT NULL,
    Address VARCHAR(255) NOT NULL,
    Salary DECIMAL(10,2) NOT NULL,
    MOB INT NOT NULL,
    Shift VARCHAR(255)
        CHECK(Shift IN('Morning','Afternoon','Night'))
);
DROP TABLE IF EXISTS Ambulance;
CREATE TABLE Ambulance
(
    Amb_id VARCHAR(6) PRIMARY KEY,
    Amb_num INT NOT NULL,
    Capacity INT NOT NULL
);
DROP TABLE IF EXISTS Ambulance_Service;
CREATE TABLE Ambulance_Service
(
    As_id VARCHAR(6) PRIMARY KEY,
    Pat_id VARCHAR(5) ,
```



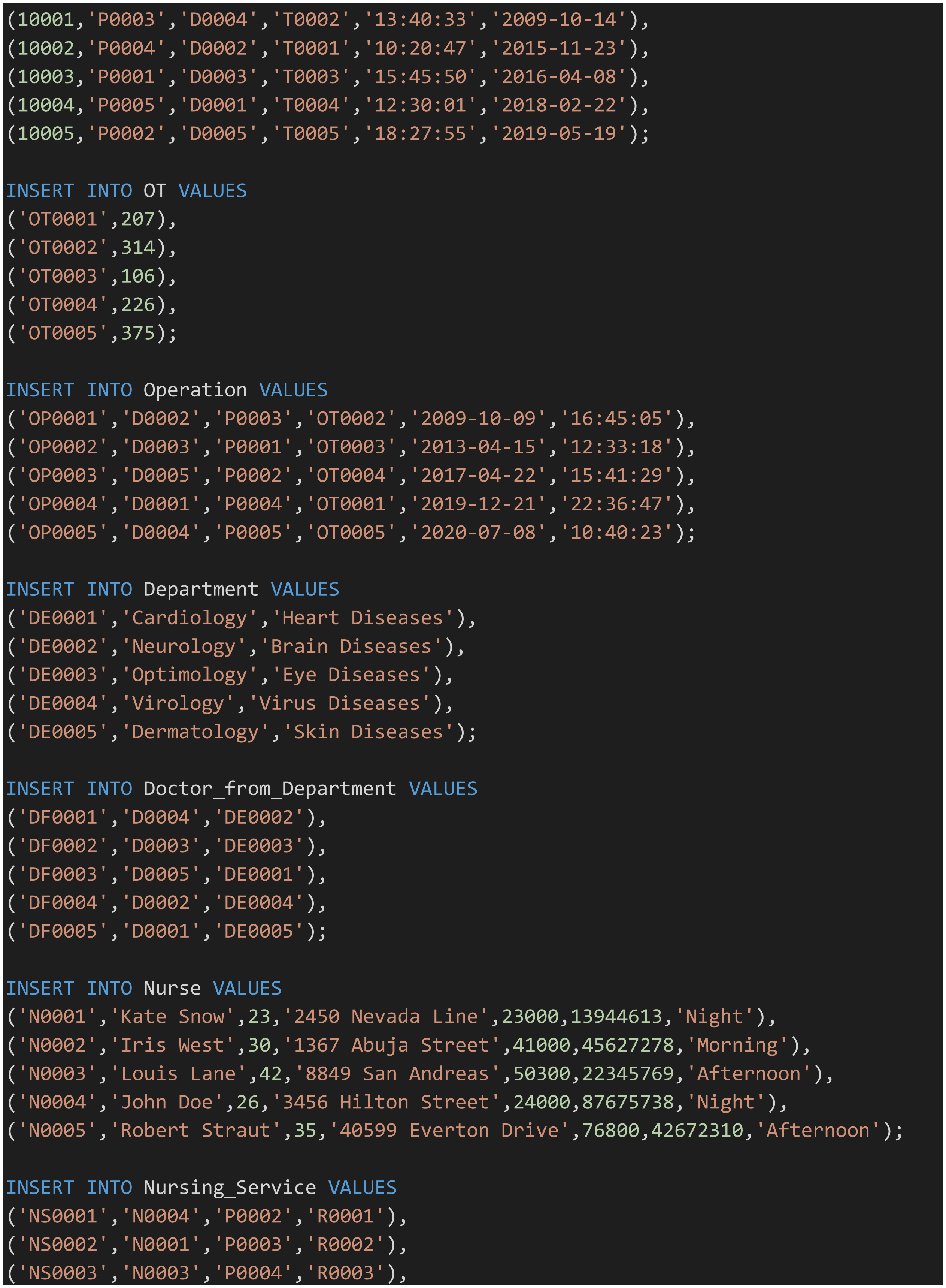
```
    Dr_id VARCHAR(6) ,
    Amb_id VARCHAR(6) ,
    FOREIGN KEY(Pat_id) REFERENCES Patient(Pat_id) ON UPDATE SET NULL ON DELETE S
ET  NULL,
    FOREIGN KEY(Dr_id) REFERENCES Driver(Dr_id)ON UPDATE SET  NULL ON DELETE SET
 NULL,
    FOREIGN KEY(Amb_id) REFERENCES Ambulance(Amb_id) ON UPDATE SET  NULL ON DELET
E SET  NULL
);
DROP TABLE IF EXISTS Carriers;
CREATE TABLE Carriers
(
    Cr_id VARCHAR(6) PRIMARY KEY,
    Cr_name VARCHAR(255) NOT NULL,
    Address VARCHAR(255) NOT NULL,
    Salary DECIMAL(10,2) NOT NULL,
    MOB INT NOT NULL
);
DROP TABLE IF EXISTS Carrying_Service;
CREATE TABLE Carrying_Service
(
    Cs_id VARCHAR(6) ,
    Cr_id VARCHAR(6) ,
    Amb_id VARCHAR(6) ,
    Pat_id VARCHAR(5) ,
    FOREIGN KEY(Amb_id) REFERENCES Ambulance(Amb_id)ON UPDATE SET  NULL ON DELETE
 SET  NULL,
    FOREIGN KEY(Pat_id) REFERENCES Patient(Pat_id)ON UPDATE SET  NULL ON DELETE S
ET  NULL,
    CONSTRAINT CompKey6 PRIMARY KEY
    (Cs_id,Cr_id)

);


INSERT INTO Patient VALUES
('P0001','Luke Skywalker',22,'M','21015 Monte Real Street','1998-06-
09',05546371),
('P0002','Jennifer Lee',19,'F','3551 Barcelona Alley','2001-03-23',15678492),
('P0003','Damilola Olutobiloba',29,'F','2181 Oxford District','1991-12-
06',67458329),
('P0004','Angela Penelope',18,'F','200 Xiaolingwei','2002-03-13',76458091),
('P0005','Xiao Gang',7,'M','4019 Xanwu ','2013-10-25',53423678);
```



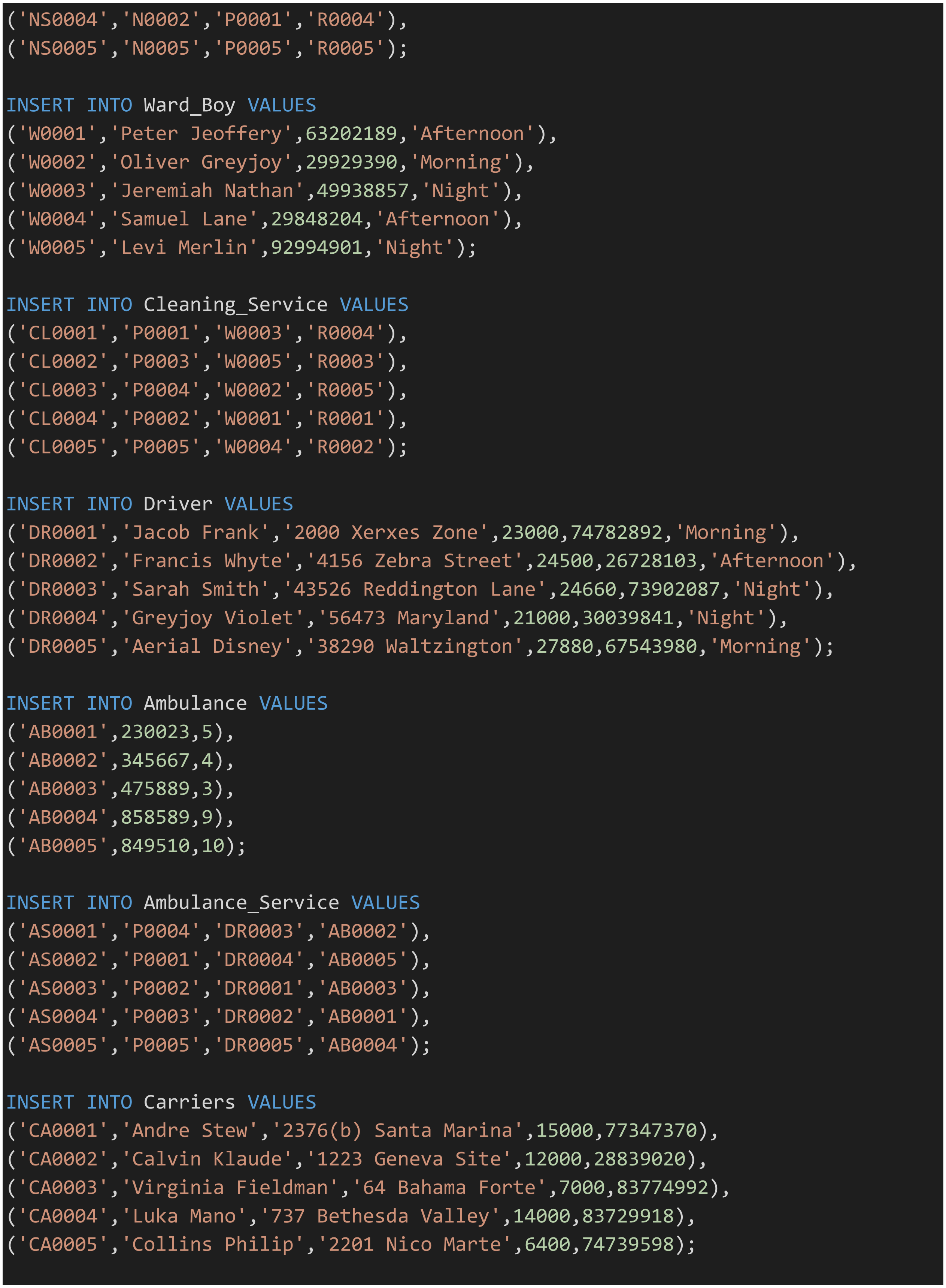
```sql
INSERT INTO Room VALUES
('R0001',100,'Casualty Room',50000),
('R0002',200,'Consulting Room',12000),
('R0003',105,'Delivery Room',30000),
('R0004',110,'Martenity Ward',45000),
('R0005',213,'Nursery',18500);

INSERT INTO Admission VALUES
('AM0001','P0001','R0001','RC0002','2009-03-08','11:23:05'),
('AM0002','P0002','R0004','RC0003','2010-05-19','20:14:19'),
('AM0003','P0003','R0003','RC0001','2007-01-18','17:50:25'),
('AM0004','P0004','R0002','RC0004','2018-08-23','09:43:40'),
('AM0005','P0005','R0005','RC0005','2020-11-45','23:01:13');

INSERT INTO Receptionist VALUES
('RC0001','Eric Makrov',40,'1300 NASA Air Space',30000,19824567,'Afternoon'),
('RC0002','Moses Elliot',35,'1435 China Town ',24000,76473882,'Night'),
('RC0003','Mary Cobblepot',46,'1891 Quebec Zone',36000,64537809,'Morning'),
('RC0004','Hassana Abdulkareem',38,'2017 Virginia Street',26800,36768091,'Afterno
on'),
('RC0005','Jennifer Smith',29,'1011 Whale District',14900,35623712,'Morning');

INSERT INTO Doctor VALUES
('D0001','Jason Rob',27,'3908 Manchester Drive',203000,45638920),
('D0002','Raymond Reddington',34,'1020 Stanford Base',4006000,07205703),
('D0003','Daniel Steve',56,'8292 West Bridge',589000,27737329),
('D0004','John Reeves',29,'4019 Frankfurt heights',280000,14884992),
('D0005','Allen Sky',41,'6900 Felix Front',3001800,64783902);

INSERT INTO Appointment VALUES
('AP0001','P0001','D0005','RC0003','2009-04-17','14:00:00'),
('AP0002','P0004','D0003','RC0005','2011-08-09','12:30:00'),
('AP0003','P0002','D0001','RC0002','2013-10-24','15:00:00'),
('AP0004','P0005','D0002','RC0004','2019-03-02','14:00:00'),
('AP0005','P0003','D0004','RC0001','2020-06-18','12:30:00');

INSERT INTO Bill VALUES
('B0001','Blood Test',1200),
('B0002','X-ray Test',13400),
('B0003','Pregnancy Test',10000),
('B0004','COVID-19 Test',900),
('B0005','HIV-AIDS Test' ,12000);

INSERT INTO Accountant VALUES
```



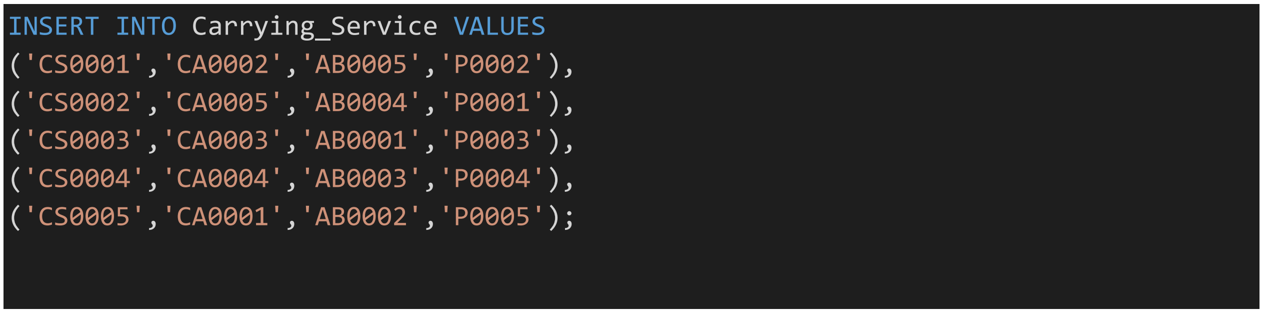
```sql
('A0001','Thomas Eddison',51,'3926 John Hopkins College',36963245,'12:00:00',3500
00),
('A0002','Georgina Ebele',34,'3452 Ivy Street',74838925,'14:00:00',248000),
('A0003','Anthony Stark',56,'1234 Manhattan Drive',65376389,'15:00:00',375000),
('A0004','Lancelot Ellington',21,'5463 New Orleans Hathaway',78054356,'13:00:00',
124000),
('A0005','Belle Groove',48,'Harvad Bussiness Quarters',53678920,'10:30:00',148900
);

INSERT INTO Payment1 VALUES
('PA0001','Alipay','2017-07-26','P0004'),
('PA0002','Union Pay','2018-03-14','P0003'),
('PA0003','Wechat','2019-08-17','P0003'),
('PA0004','Paypal','2020-11-30','P0005'),
('PA0005','Master Card','2020-12-13','P0001');

INSERT INTO Payment2 VALUES
('BP0001','B0004','A0004','P0001'),
('BP0002','B0005','A0003','P0002'),
('BP0003','B0001','A0002','P0004'),
('BP0004','B0003','A0001','P0005'),
('BP0005','B0002','A0005','P0003');

INSERT INTO Medicine VALUES
('M0001','Apoxicillin','Jenny Phamaceuticals','2017-08-09','2020-10-18',200),
('M0002','Ibuprofen','Rx & family','2018-11-08','2022-10-20',350),
('M0003','Panadol','Emzor Inc.','2018-09-10','2021-06-03',150),
('M0004','Cough Syrup','The Johnsons','2019-01-05','2024-03-09',460),
('M0005','Ampiclox','Emzor Inc.','2020-04-17','2024-09-17',380);

INSERT INTO Prescription VALUES
('PR0001','D0004','M0001','P0005','2017-09-06',4000),
('PR0002','D0001','M0003','P0002','2018-12-28',2580),
('PR0003','D0003','M0002','P0003','2018-12-30',3375),
('PR0004','D0005','M0004','P0001','2019-07-03',4500),
('PR0005','D0002','M0005','P0004','2020-10-05',3150);

INSERT INTO Test VALUES
('T0001','Blood Test','2013-06-06','2013-06-13',1375),
('T0002','Pregnancy Test','2015-07-04','2015-07-11',1450),
('T0003','COVID-19 Test','2018-09-15','2018-09-22',1600),
('T0004','Pregnancy Test','2019-03-12','2019-03-19',1650),
('T0005','X-ray Test','2020-08-17','2020-08-20',2000);

INSERT INTO Assist VALUES
```



```
(10001,'P0003','D0004','T0002','13:40:33','2009-10-14'),
(10002,'P0004','D0002','T0001','10:20:47','2015-11-23'),
(10003,'P0001','D0003','T0003','15:45:50','2016-04-08'),
(10004,'P0005','D0001','T0004','12:30:01','2018-02-22'),
(10005,'P0002','D0005','T0005','18:27:55','2019-05-19');

INSERT INTO OT VALUES
('OT0001',207),
('OT0002',314),
('OT0003',106),
('OT0004',226),
('OT0005',375);

INSERT INTO Operation VALUES
('OP0001','D0002','P0003','OT0002','2009-10-09','16:45:05'),
('OP0002','D0003','P0001','OT0003','2013-04-15','12:33:18'),
('OP0003','D0005','P0002','OT0004','2017-04-22','15:41:29'),
('OP0004','D0001','P0004','OT0001','2019-12-21','22:36:47'),
('OP0005','D0004','P0005','OT0005','2020-07-08','10:40:23');

INSERT INTO Department VALUES
('DE0001','Cardiology','Heart Diseases'),
('DE0002','Neurology','Brain Diseases'),
('DE0003','Optimology','Eye Diseases'),
('DE0004','Virology','Virus Diseases'),
('DE0005','Dermatology','Skin Diseases');

INSERT INTO Doctor_from_Department VALUES
('DF0001','D0004','DE0002'),
('DF0002','D0003','DE0003'),
('DF0003','D0005','DE0001'),
('DF0004','D0002','DE0004'),
('DF0005','D0001','DE0005');

INSERT INTO Nurse VALUES
('N0001','Kate Snow',23,'2450 Nevada Line',23000,13944613,'Night'),
('N0002','Iris West',30,'1367 Abuja Street',41000,45627278,'Morning'),
('N0003','Louis Lane',42,'8849 San Andreas',50300,22345769,'Afternoon'),
('N0004','John Doe',26,'3456 Hilton Street',24000,87675738,'Night'),
('N0005','Robert Straut',35,'40599 Everton Drive',76800,42672310,'Afternoon');

INSERT INTO Nursing_Service VALUES
('NS0001','N0004','P0002','R0001'),
('NS0002','N0001','P0003','R0002'),
('NS0003','N0003','P0004','R0003'),
```



```
('NS0004','N0002','P0001','R0004'),
('NS0005','N0005','P0005','R0005');

INSERT INTO Ward_Boy VALUES
('W0001','Peter Jeoffery',63202189,'Afternoon'),
('W0002','Oliver Greyjoy',29929390,'Morning'),
('W0003','Jeremiah Nathan',49938857,'Night'),
('W0004','Samuel Lane',29848204,'Afternoon'),
('W0005','Levi Merlin',92994901,'Night');

INSERT INTO Cleaning_Service VALUES
('CL0001','P0001','W0003','R0004'),
('CL0002','P0003','W0005','R0003'),
('CL0003','P0004','W0002','R0005'),
('CL0004','P0002','W0001','R0001'),
('CL0005','P0005','W0004','R0002');

INSERT INTO Driver VALUES
('DR0001','Jacob Frank','2000 Xerxes Zone',23000,74782892,'Morning'),
('DR0002','Francis Whyte','4156 Zebra Street',24500,26728103,'Afternoon'),
('DR0003','Sarah Smith','43526 Reddington Lane',24660,73902087,'Night'),
('DR0004','Greyjoy Violet','56473 Maryland',21000,30039841,'Night'),
('DR0005','Aerial Disney','38290 Waltzington',27880,67543980,'Morning');

INSERT INTO Ambulance VALUES
('AB0001',230023,5),
('AB0002',345667,4),
('AB0003',475889,3),
('AB0004',858589,9),
('AB0005',849510,10);

INSERT INTO Ambulance_Service VALUES
('AS0001','P0004','DR0003','AB0002'),
('AS0002','P0001','DR0004','AB0005'),
('AS0003','P0002','DR0001','AB0003'),
('AS0004','P0003','DR0002','AB0001'),
('AS0005','P0005','DR0005','AB0004');

INSERT INTO Carriers VALUES
('CA0001','Andre Stew','2376(b) Santa Marina',15000,77347370),
('CA0002','Calvin Klaude','1223 Geneva Site',12000,28839020),
('CA0003','Virginia Fieldman','64 Bahama Forte',7000,83774992),
('CA0004','Luka Mano','737 Bethesda Valley',14000,83729918),
('CA0005','Collins Philip','2201 Nico Marte',6400,74739598);
```



```sql
INSERT INTO Carrying_Service VALUES
('CS0001','CA0002','AB0005','P0002'),
('CS0002','CA0005','AB0004','P0001'),
('CS0003','CA0003','AB0001','P0003'),
('CS0004','CA0004','AB0003','P0004'),
('CS0005','CA0001','AB0002','P0005');
```

QUERIES,INSERTIONS,DELETIONS,AND UPDATES

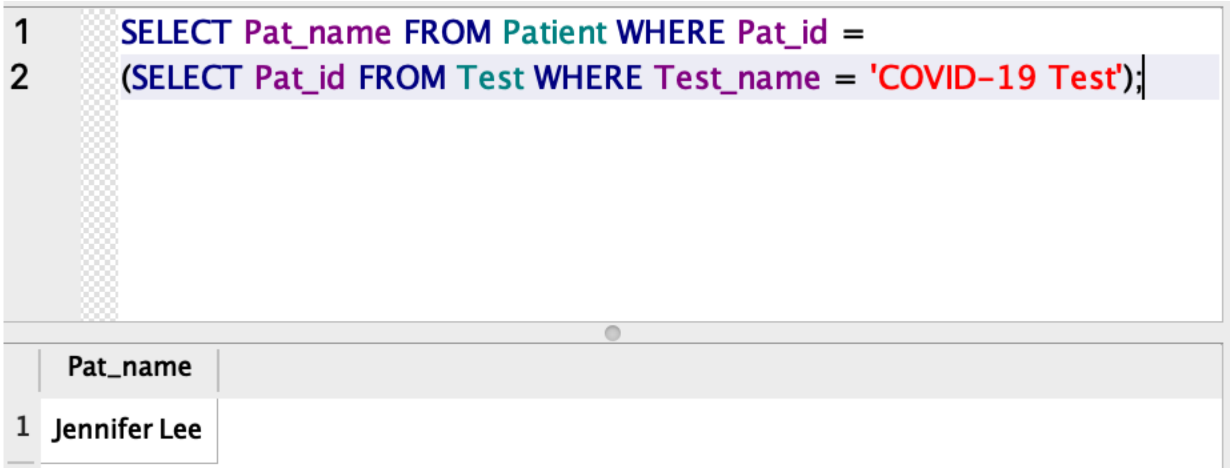
Queries

In this section I am going to come up with different real life problems to address the idea of this section.

1. **There has been a release of a virus namely the *Corona-Virus* and it is fast spreading. Our hospital is willing to offer samples of the Virus to scientists to analyze and come up with a drug for it from a patient who has tested positive .What query should be written to select the patient with the virus that tested positive?**

   Solution

   Since the sample can be located at the **Test** relation, we therefore write down the following query:

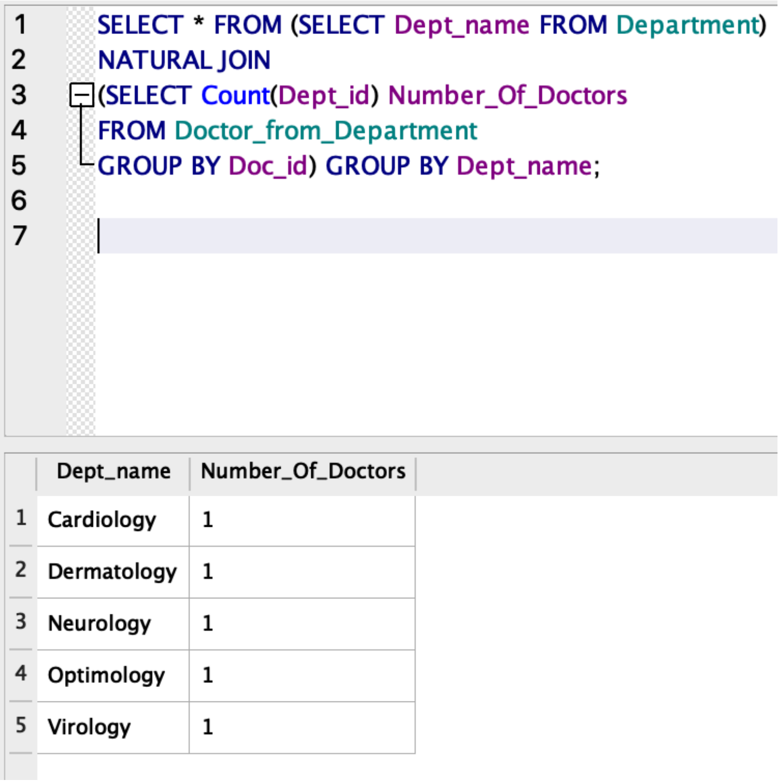


```
SELECT Pat_name FROM Patient WHERE Pat_id =
(SELECT Pat_id FROM Test WHERE Test_name = 'COVID-19 Test');
```

```
  Pat_name
1 Jennifer Lee
```

2. There needs to be a consensus of all the doctors for each depart. Can you list the number of doctors from each department?

Solution:

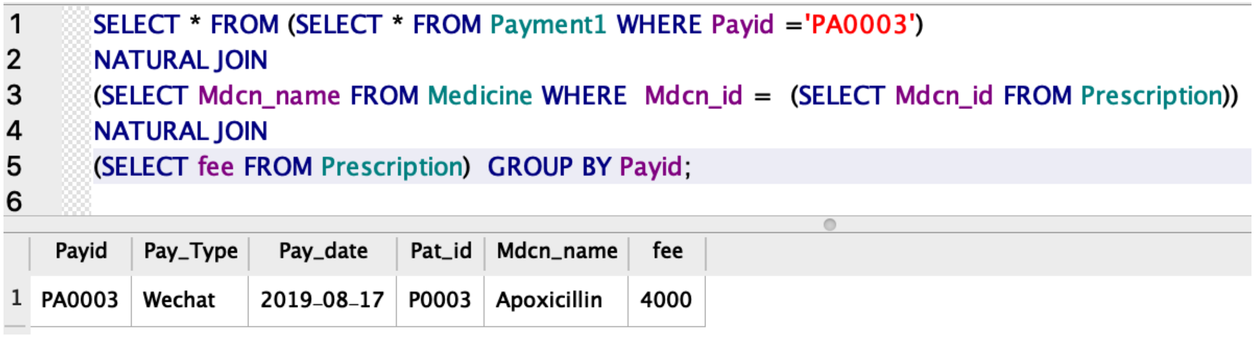


```
SELECT * FROM (SELECT Dept_name FROM Department)
NATURAL JOIN
(SELECT Count(Dept_id) Number_Of_Doctors
FROM Doctor_from_Department
GROUP BY Doc_id) GROUP BY Dept_name;
```

	Dept_name	Number_Of_Doctors
1	Cardiology	1
2	Dermatology	1
3	Neurology	1
4	Optimology	1
5	Virology	1

3. An individual is being suspected of stealing a drug at the expense of the pharmacist attending to another person. The individual is asked to give his Payment Identification number. To prove his innocence, he quickly gives the pharmacist his Payment Identification number (which is “PA0003”). Using the Payment Identification number (represented as Payid), write a query to output the row or tuple that shows that the individual paid for the drug and give the name of the drug.

Solution

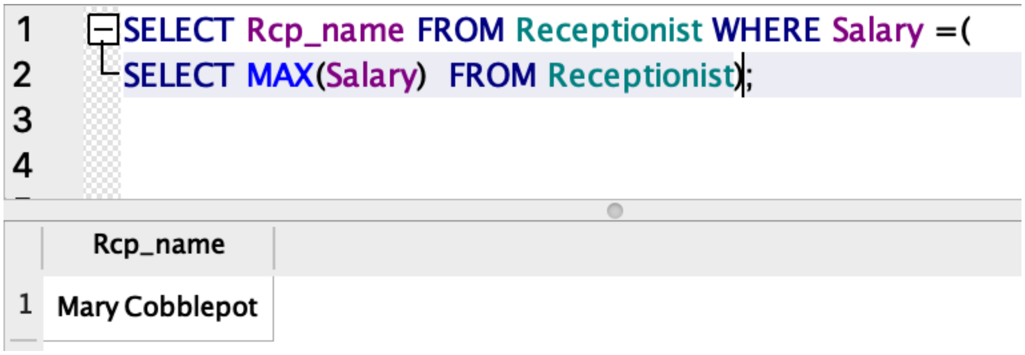


```
SELECT * FROM (SELECT * FROM Payment1 WHERE Payid ='PA0003')
NATURAL JOIN
(SELECT Mdcn_name FROM Medicine WHERE  Mdcn_id =  (SELECT Mdcn_id FROM Prescription))
NATURAL JOIN
(SELECT fee FROM Prescription)  GROUP BY Payid;
```

	Payid	Pay_Type	Pay_date	Pat_id	Mdcn_name	fee
1	PA0003	Wechat	2019-08-17	P0003	Apoxicillin	4000

4. We want you to help one of the Accountants find the name of the Receptionist with maximum salary. Write a query for this task.

Solution

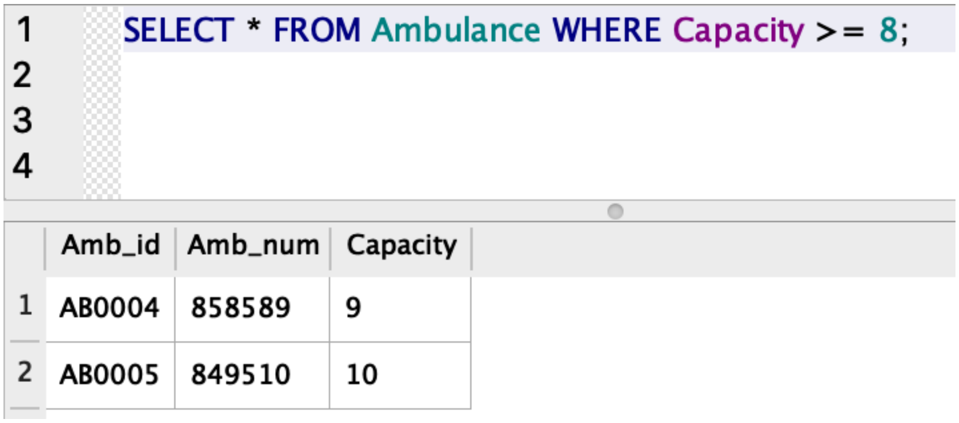


```
SELECT Rcp_name FROM Receptionist WHERE Salary =(
SELECT MAX(Salary)  FROM Receptionist);
```

	Rcp_name
1	Mary Cobblepot

5. There has been an Accident at the high way road near Elm Street. The witnesses have counted about 8 victims .Quickly, write a query to identify which Ambulances have ability to carry up to 8 victims so we can take the victims to our hospital.

Solution



```
SELECT * FROM Ambulance WHERE Capacity >= 8;
```

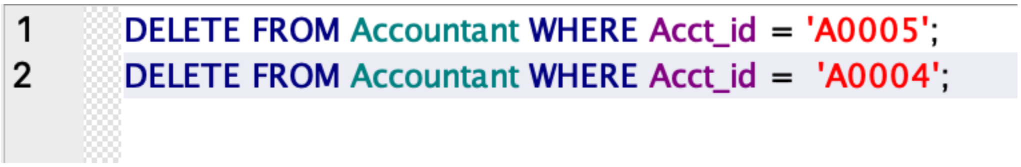
	Amb_id	Amb_num	Capacity
1	AB0004	858589	9
2	AB0005	849510	10

DELETIONS

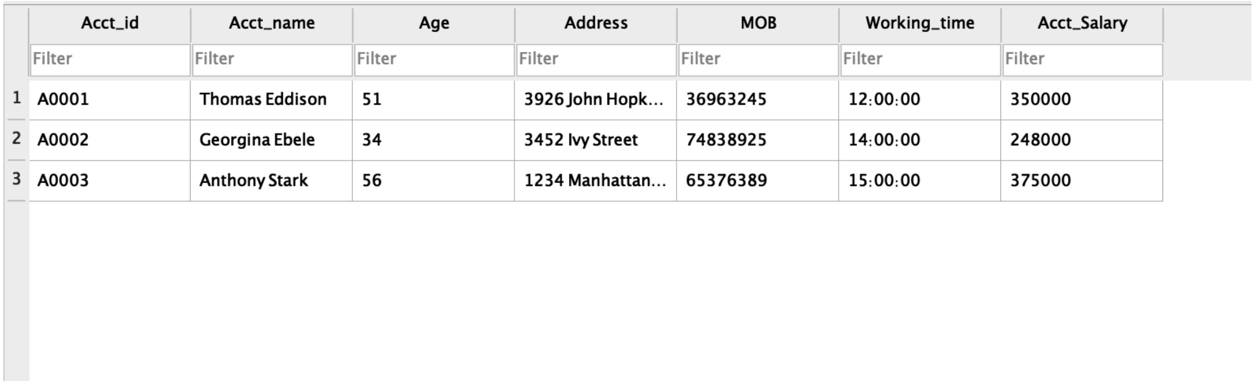
This section also involves life Scenarios

1. **'A THIEF IN OUR MIST'. An accountant was caught stealing money from the hospitals savings. After severe interrogation, it turns out the accountant had an accomplice. The owner of our hospital has decided to fire both accountants. Write a query to delete their names from the database.**

   Solution



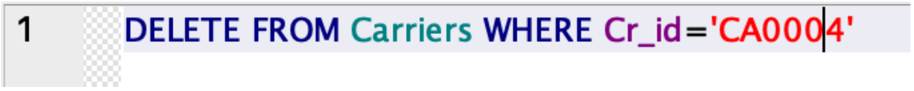
```
DELETE FROM Accountant WHERE Acct_id = 'A0005';
DELETE FROM Accountant WHERE Acct_id =  'A0004';
```



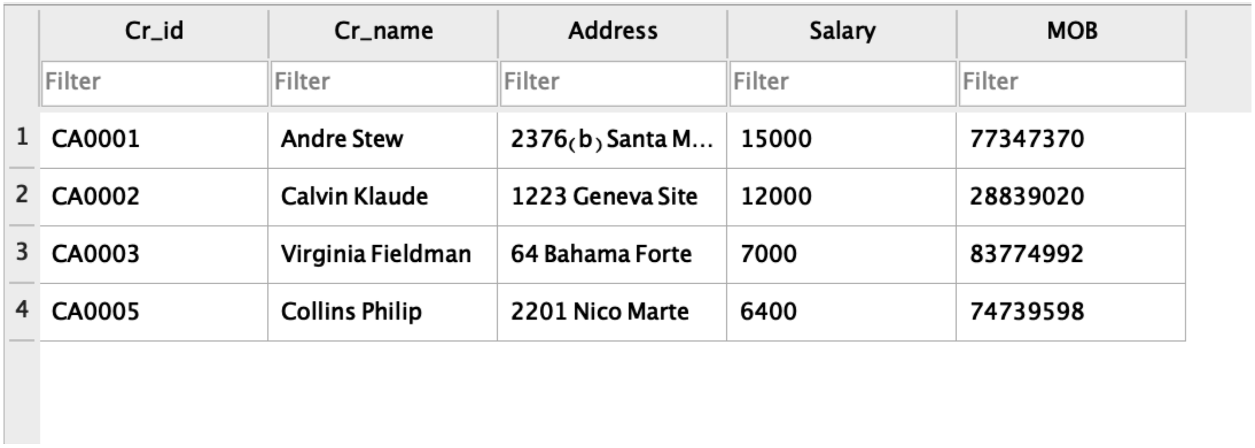
	Acct_id	Acct_name	Age	Address	MOB	Working_time	Acct_Salary
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	A0001	Thomas Eddison	51	3926 John Hopk...	36963245	12:00:00	350000
2	A0002	Georgina Ebele	34	3452 Ivy Street	74838925	14:00:00	248000
3	A0003	Anthony Stark	56	1234 Manhattan...	65376389	15:00:00	375000

2. **MOVING TO NEW HEIGHTS. One of the people who carry patients from the ambulance into the hospital has found a new job as the manager of a hotel. Write a query to delete her information from the database.**

   Solution



```
DELETE FROM Carriers WHERE Cr_id='CA0004'
```

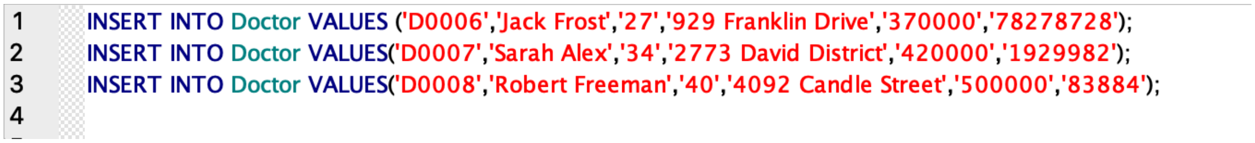


	Cr_id	Cr_name	Address	Salary	MOB
	Filter	Filter	Filter	Filter	Filter
1	CA0001	Andre Stew	2376(b) Santa M...	15000	77347370
2	CA0002	Calvin Klaude	1223 Geneva Site	12000	28839020
3	CA0003	Virginia Fieldman	64 Bahama Forte	7000	83774992
4	CA0005	Collins Philip	2201 Nico Marte	6400	74739598

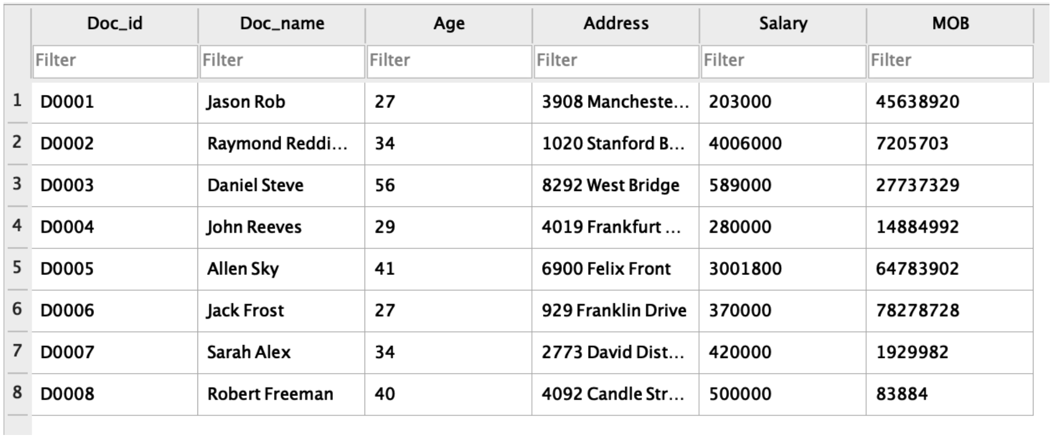
INSERTIONS

This section will also involve real life scenarios.

1. **Our hospital is the only hospital in the area. Therefore she has decided to employ more Doctors and Nurses to help aid needs of the incoming patients .She has put up an attractive offer for the Salaries of those who care to be employed. So far three Doctors have been employed. Write a query to input their names in the Hospital's Database.**
   Solution

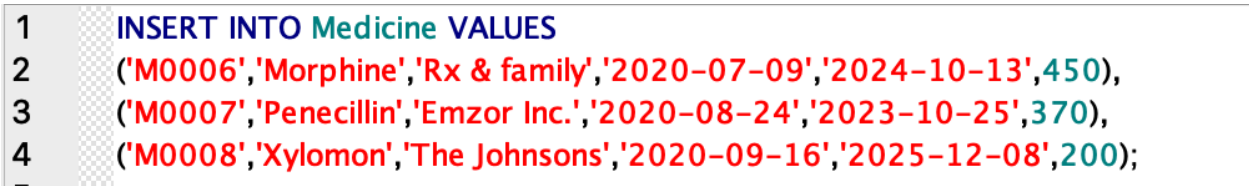


```
INSERT INTO Doctor VALUES ('D0006','Jack Frost','27','929 Franklin Drive','370000','78278728');
INSERT INTO Doctor VALUES('D0007','Sarah Alex','34','2773 David District','420000','1929982');
INSERT INTO Doctor VALUES('D0008','Robert Freeman','40','4092 Candle Street','500000','83884');
```

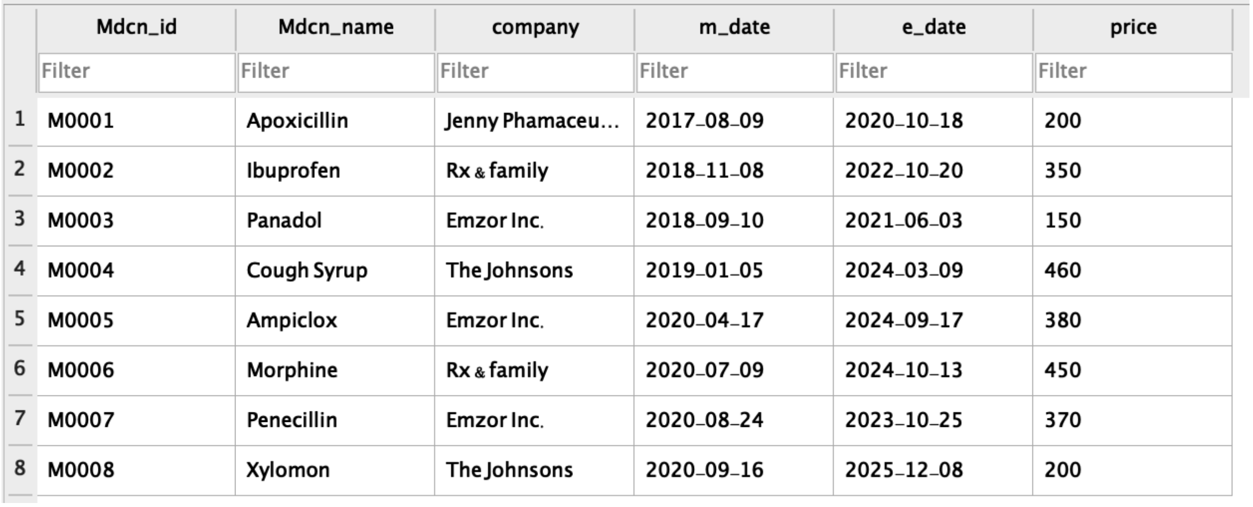


	Doc_id	Doc_name	Age	Address	Salary	MOB
	Filter	Filter	Filter	Filter	Filter	Filter
1	D0001	Jason Rob	27	3908 Mancheste...	203000	45638920
2	D0002	Raymond Reddi...	34	1020 Stanford B...	4006000	7205703
3	D0003	Daniel Steve	56	8292 West Bridge	589000	27737329
4	D0004	John Reeves	29	4019 Frankfurt ...	280000	14884992
5	D0005	Allen Sky	41	6900 Felix Front	3001800	64783902
6	D0006	Jack Frost	27	929 Franklin Drive	370000	78278728
7	D0007	Sarah Alex	34	2773 David Dist...	420000	1929982
8	D0008	Robert Freeman	40	4092 Candle Str...	500000	83884

2. **The next shipment of drugs has arrived namely Morphine,Penecillin, and Xylomon. Write a query to insert the drugs into the database.**
   Solution



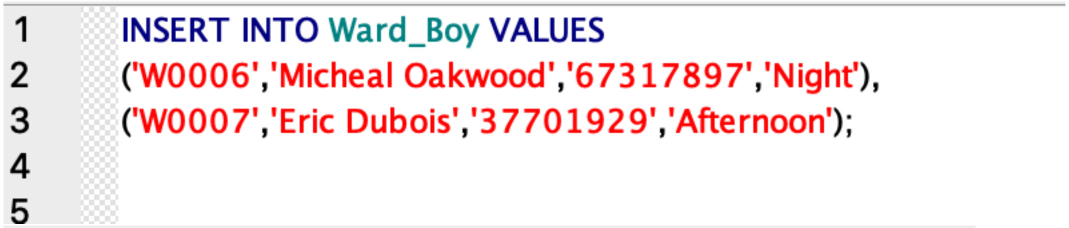
```sql
INSERT INTO Medicine VALUES
('M0006','Morphine','Rx & family','2020-07-09','2024-10-13',450),
('M0007','Penecillin','Emzor Inc.','2020-08-24','2023-10-25',370),
('M0008','Xylomon','The Johnsons','2020-09-16','2025-12-08',200);
```



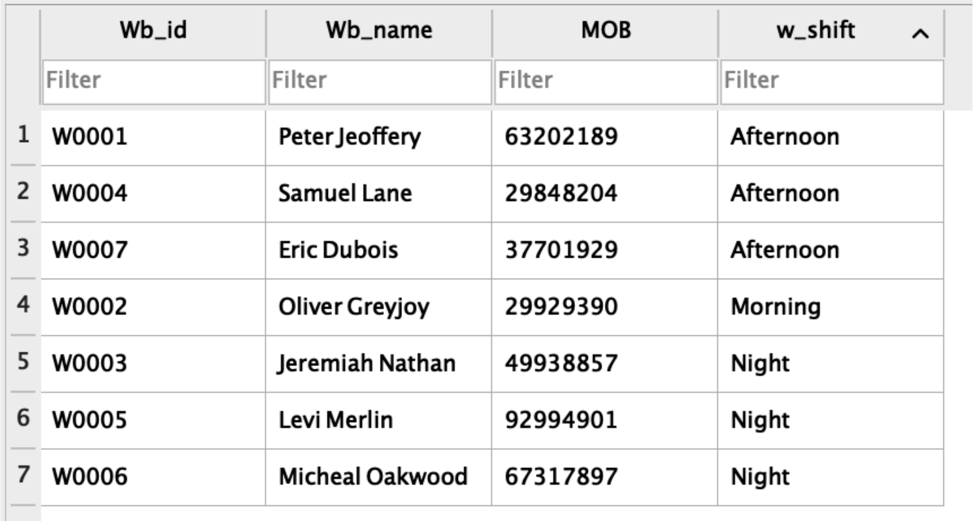
	Mdcn_id	Mdcn_name	company	m_date	e_date	price
1	M0001	Apoxicillin	Jenny Phamaceu…	2017-08-09	2020-10-18	200
2	M0002	Ibuprofen	Rx & family	2018-11-08	2022-10-20	350
3	M0003	Panadol	Emzor Inc.	2018-09-10	2021-06-03	150
4	M0004	Cough Syrup	The Johnsons	2019-01-05	2024-03-09	460
5	M0005	Ampiclox	Emzor Inc.	2020-04-17	2024-09-17	380
6	M0006	Morphine	Rx & family	2020-07-09	2024-10-13	450
7	M0007	Penecillin	Emzor Inc.	2020-08-24	2023-10-25	370
8	M0008	Xylomon	The Johnsons	2020-09-16	2025-12-08	200

3. Ever since the State Government's speech people have volunteered for Ward Boy in many hospitals as a means of community service. For our hospital, there have been two people who have volunteered. Write a query to include them into our database.

Solution



```
INSERT INTO Ward_Boy VALUES
('W0006','Micheal Oakwood','67317897','Night'),
('W0007','Eric Dubois','37701929','Afternoon');
```

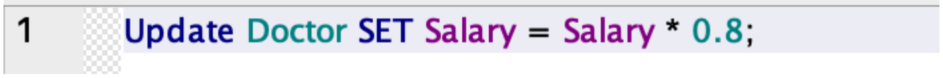


	Wb_id	Wb_name	MOB	w_shift
	Filter	Filter	Filter	Filter
1	W0001	Peter Jeoffery	63202189	Afternoon
2	W0004	Samuel Lane	29848204	Afternoon
3	W0007	Eric Dubois	37701929	Afternoon
4	W0002	Oliver Greyjoy	29929390	Morning
5	W0003	Jeremiah Nathan	49938857	Night
6	W0005	Levi Merlin	92994901	Night
7	W0006	Micheal Oakwood	67317897	Night

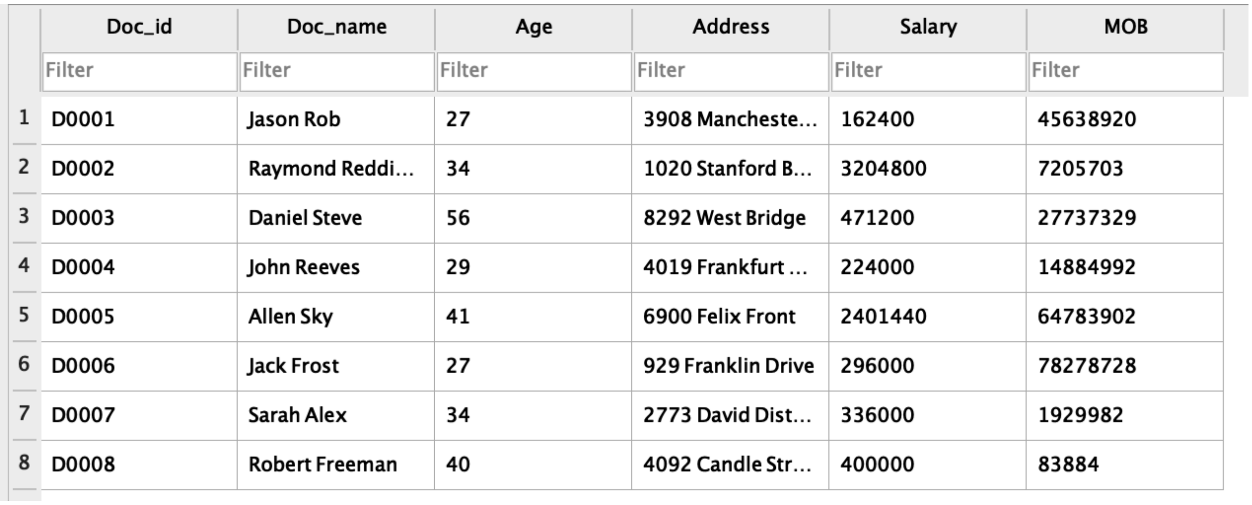
UPDATES

This one will be just like others.

1. **The owner of the hospital has decided to cut costs by reducing the salary of the Doctors by 20% .Write a query to change this in the database.**
   Solution

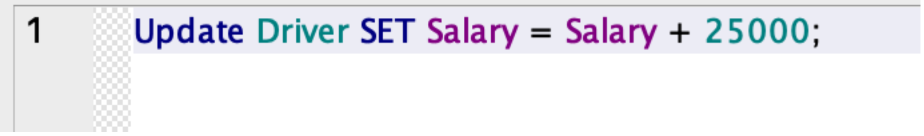


```
Update Doctor SET Salary = Salary * 0.8;
```

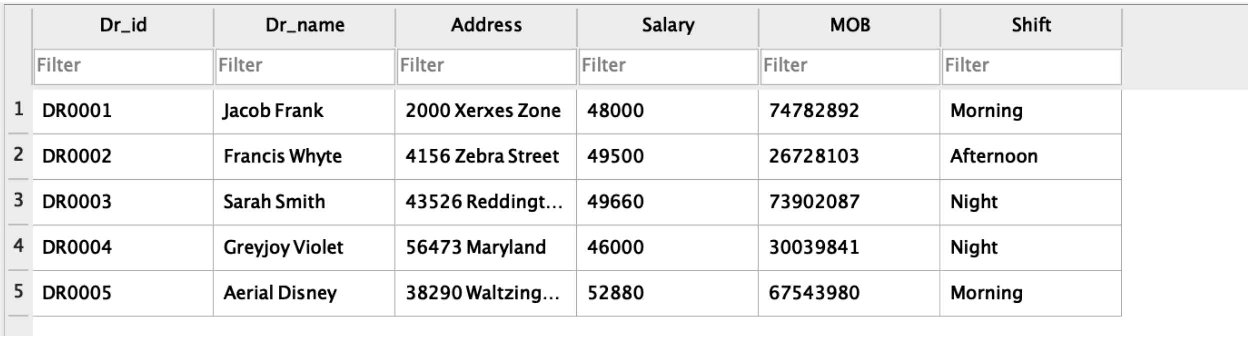


	Doc_id	Doc_name	Age	Address	Salary	MOB
	Filter	Filter	Filter	Filter	Filter	Filter
1	D0001	Jason Rob	27	3908 Mancheste...	162400	45638920
2	D0002	Raymond Reddi...	34	1020 Stanford B...	3204800	7205703
3	D0003	Daniel Steve	56	8292 West Bridge	471200	27737329
4	D0004	John Reeves	29	4019 Frankfurt ...	224000	14884992
5	D0005	Allen Sky	41	6900 Felix Front	2401440	64783902
6	D0006	Jack Frost	27	929 Franklin Drive	296000	78278728
7	D0007	Sarah Alex	34	2773 David Dist...	336000	1929982
8	D0008	Robert Freeman	40	4092 Candle Str...	400000	83884

2. **'A Generous man indeed'. The owner of the hospital has decided to increase the salary of the drivers after considering how much they risk their life on the highway just to get to a patient so he has decided to give them a raise of #25,000.Write a query to show this**
   Solution

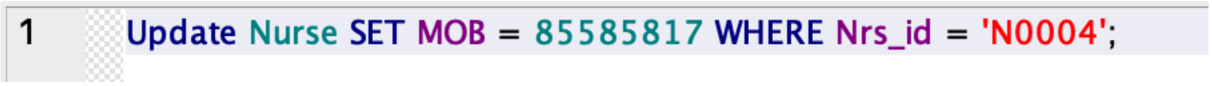


```
Update Driver SET Salary = Salary + 25000;
```

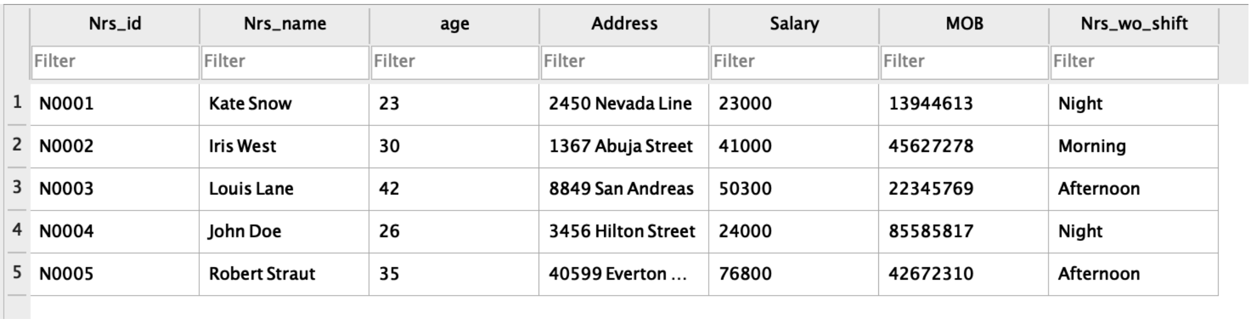


	Dr_id	Dr_name	Address	Salary	MOB	Shift
	Filter	Filter	Filter	Filter	Filter	Filter
1	DR0001	Jacob Frank	2000 Xerxes Zone	48000	74782892	Morning
2	DR0002	Francis Whyte	4156 Zebra Street	49500	26728103	Afternoon
3	DR0003	Sarah Smith	43526 Reddingt...	49660	73902087	Night
4	DR0004	Greyjoy Violet	56473 Maryland	46000	30039841	Night
5	DR0005	Aerial Disney	38290 Waltzing...	52880	67543980	Morning

3. **Mr. John Doe lost his phone on his way home. So he got a new one with a new number. Write a query to update his mob number.**
Solution



```
Update Nurse SET MOB = 85585817 WHERE Nrs_id = 'N0004';
```



	Nrs_id	Nrs_name	age	Address	Salary	MOB	Nrs_wo_shift
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	N0001	Kate Snow	23	2450 Nevada Line	23000	13944613	Night
2	N0002	Iris West	30	1367 Abuja Street	41000	45627278	Morning
3	N0003	Louis Lane	42	8849 San Andreas	50300	22345769	Afternoon
4	N0004	John Doe	26	3456 Hilton Street	24000	85585817	Night
5	N0005	Robert Straut	35	40599 Everton ...	76800	42672310	Afternoon

TRIGGERS AND CONSTRAINTS

CONSTRAINTS

We have four kinds of constraints in this Database namely:

- **PRIMARY KEY-A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table. For Example**

```
DROP TABLE IF EXISTS Carriers;
CREATE TABLE Carriers
(
    Cr_id VARCHAR(6) PRIMARY KEY,
    Cr_name VARCHAR(255) NOT NULL,
    Address VARCHAR(255) NOT NULL,
    Salary DECIMAL(10,2) NOT NULL,
    MOB INT NOT NULL
);
```

- **NOT NULL – Ensures that a column cannot have a NULL value. For example**

```
DROP TABLE IF EXISTS Carriers;
CREATE TABLE Carriers
(
    Cr_id VARCHAR(6) PRIMARY KEY,
    Cr_name VARCHAR(255) NOT NULL,
    Address VARCHAR(255) NOT NULL,
    Salary DECIMAL(10,2) NOT NULL,
    MOB INT NOT NULL
);
```

- **FOREIGN KEY-Uniquely identifies a row/record in another table. For example**

```
DROP TABLE IF EXISTS Ambulance_Service;
CREATE TABLE Ambulance_Service
(
    As_id VARCHAR(6) PRIMARY KEY,
    Pat_id VARCHAR(5) ,
    Dr_id VARCHAR(6) ,
    Amb_id VARCHAR(6) ,
 FOREIGN KEY(Pat_id) REFERENCES Patient(Pat_id) ON UPDATE SET   NULL ON DELETE SET   NULL,
FOREIGN KEY(Dr_id) REFERENCES Driver(Dr_id)ON UPDATE SET   NULL ON DELETE SET   NULL,
FOREIGN KEY(Amb_id) REFERENCES Ambulance(Amb_id) ON UPDATE SET   NULL ON DELETE SET   NULL
);
```

- **CHECK- Ensures that all values in a column satisfies a specific condition. For example**

```sql
DROP TABLE IF EXISTS Driver;
CREATE TABLE Driver
(
    Dr_id VARCHAR(6) PRIMARY KEY,
    Dr_name VARCHAR(255) NOT NULL,
    Address VARCHAR(255) NOT NULL,
    Salary DECIMAL(10,2) NOT NULL,
    MOB INT NOT NULL,
    Shift VARCHAR(255)
        CHECK(Shift IN('Morning','Afternoon','Night'))
);
```

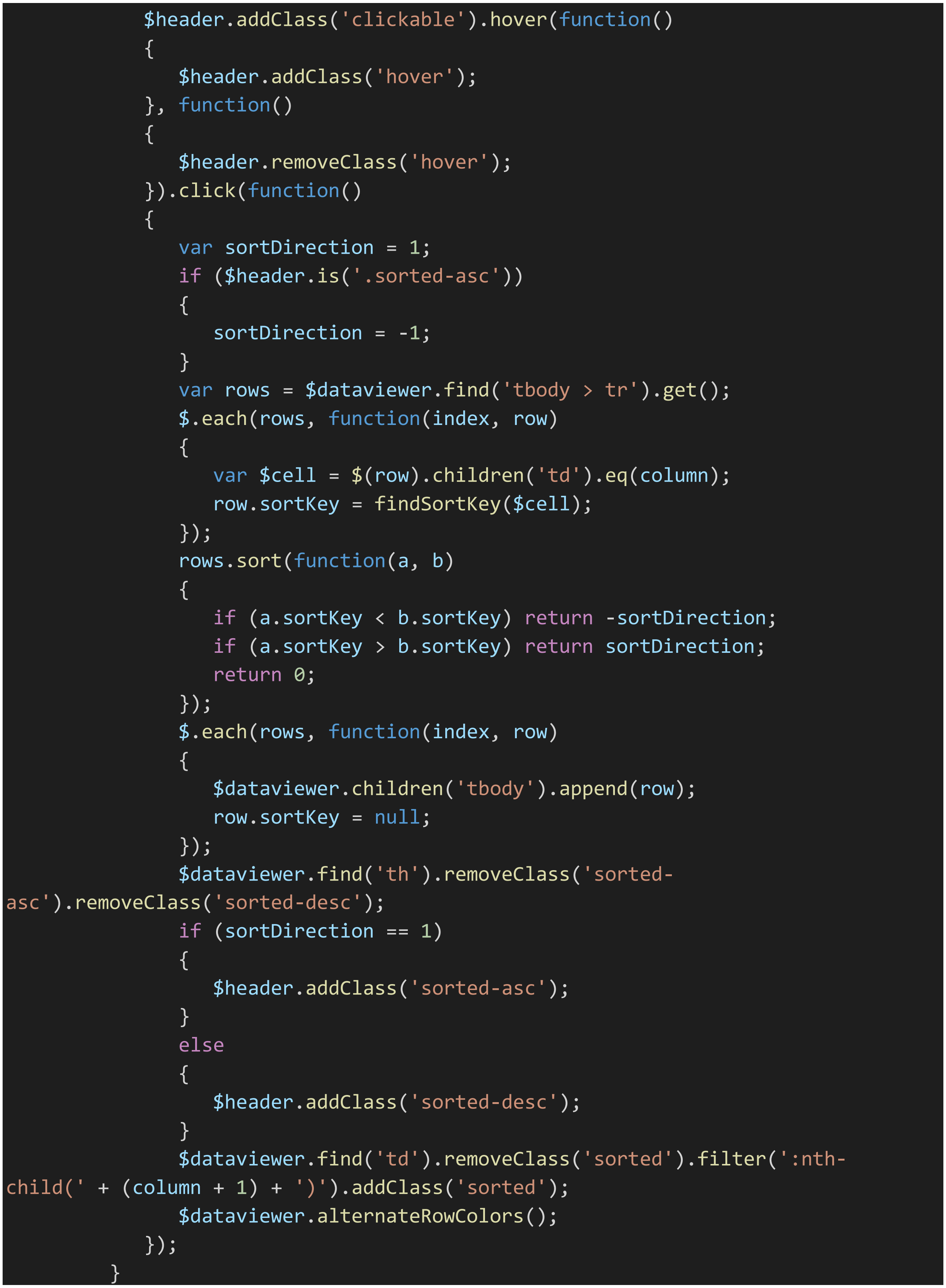
NOTE: There are no triggers.

WEBSITE

This website is not really much but we just wanted to prove that we could link the database to a website. We will only give PHP source code since HTML and CSS are quite large.

PHP -Source code
For Doctors

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Page</title>
<meta name="generator" content="WYSIWYG Web Builder 15 -
 http://www.wysiwygwebbuilder.com">
<link href="HMS.css" rel="stylesheet">
<link href="viewer1.css" rel="stylesheet">
<script src="jquery-1.12.4.min.js"></script>
<script>
$(document).ready(function()
{
   $.fn.alternateRowColors = function()
   {
      $('tbody tr:odd', this).removeClass('even').addClass('odd');
      $('tbody tr:even', this).removeClass('odd').addClass('even');
      return this;
   };
   $('table.sortable').each(function()
   {
      var $dataviewer = $(this);
      $dataviewer.alternateRowColors();
      $('th', $dataviewer).each(function(column)
      {
         var $header = $(this);
         var findSortKey;
         findSortKey = function($cell)
         {
            return $cell.find('.sort-
key').text().toUpperCase() + ' ' + $cell.text().toUpperCase();
         };
         if (findSortKey)
         {
```



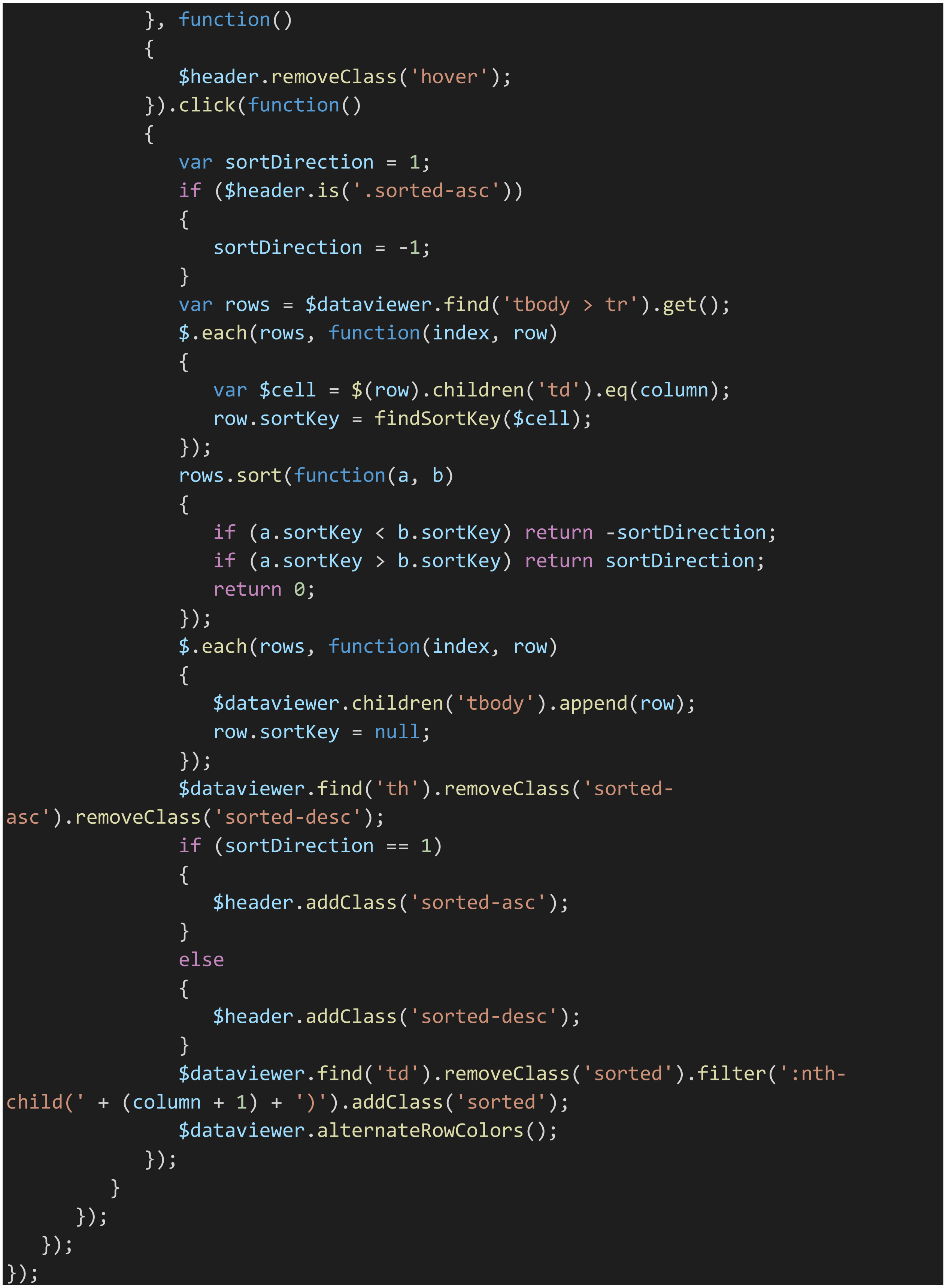
```
            $header.addClass('clickable').hover(function()
            {
               $header.addClass('hover');
            }, function()
            {
               $header.removeClass('hover');
            }).click(function()
            {
               var sortDirection = 1;
               if ($header.is('.sorted-asc'))
               {
                  sortDirection = -1;
               }
               var rows = $dataviewer.find('tbody > tr').get();
               $.each(rows, function(index, row)
               {
                  var $cell = $(row).children('td').eq(column);
                  row.sortKey = findSortKey($cell);
               });
               rows.sort(function(a, b)
               {
                  if (a.sortKey < b.sortKey) return -sortDirection;
                  if (a.sortKey > b.sortKey) return sortDirection;
                  return 0;
               });
               $.each(rows, function(index, row)
               {
                  $dataviewer.children('tbody').append(row);
                  row.sortKey = null;
               });
               $dataviewer.find('th').removeClass('sorted-
asc').removeClass('sorted-desc');
               if (sortDirection == 1)
               {
                  $header.addClass('sorted-asc');
               }
               else
               {
                  $header.addClass('sorted-desc');
               }
               $dataviewer.find('td').removeClass('sorted').filter(':nth-
child(' + (column + 1) + ')').addClass('sorted');
               $dataviewer.alternateRowColors();
            });
         }
```

```
      });
   });
});
</script>
</head>
<body>
<div id="dataviewer" style="position:absolute;overflow:auto;left:0px;top:0px;width:970px;height:861px;z-index:0">
<?php
$mysql_host = 'localhost';
$mysql_user = 'root';
$mysql_password = '';
$mysql_database = 'hms';
$mysql_table = 'doctor';
$db = mysqli_connect($mysql_host, $mysql_user, $mysql_password);
mysqli_select_db($db,$mysql_database);
$sql = "SELECT * FROM ".$mysql_table;
$result = mysqli_query($db,$sql);
?>
<table cellpadding="0" cellspacing="0" width="100%" class="sortable paginated">
<thead>
   <tr>
<?php
$fields_num = mysqli_num_fields($result);
for ($i=0; $i<$fields_num; $i++)
{
   $field = mysqli_fetch_field($result);
   echo "      <th>" . $field->name . "</th>\n";
}
?>
   </tr>
</thead>
<tbody>
<?php
while ($row = mysqli_fetch_row($result))
{
   echo "   <tr>\n";
   foreach ($row as $cell)
   {
      echo "      <td>" . $cell . "</td>\n";
   }
   echo "   </tr>\n";
}
?>
</tbody>
```

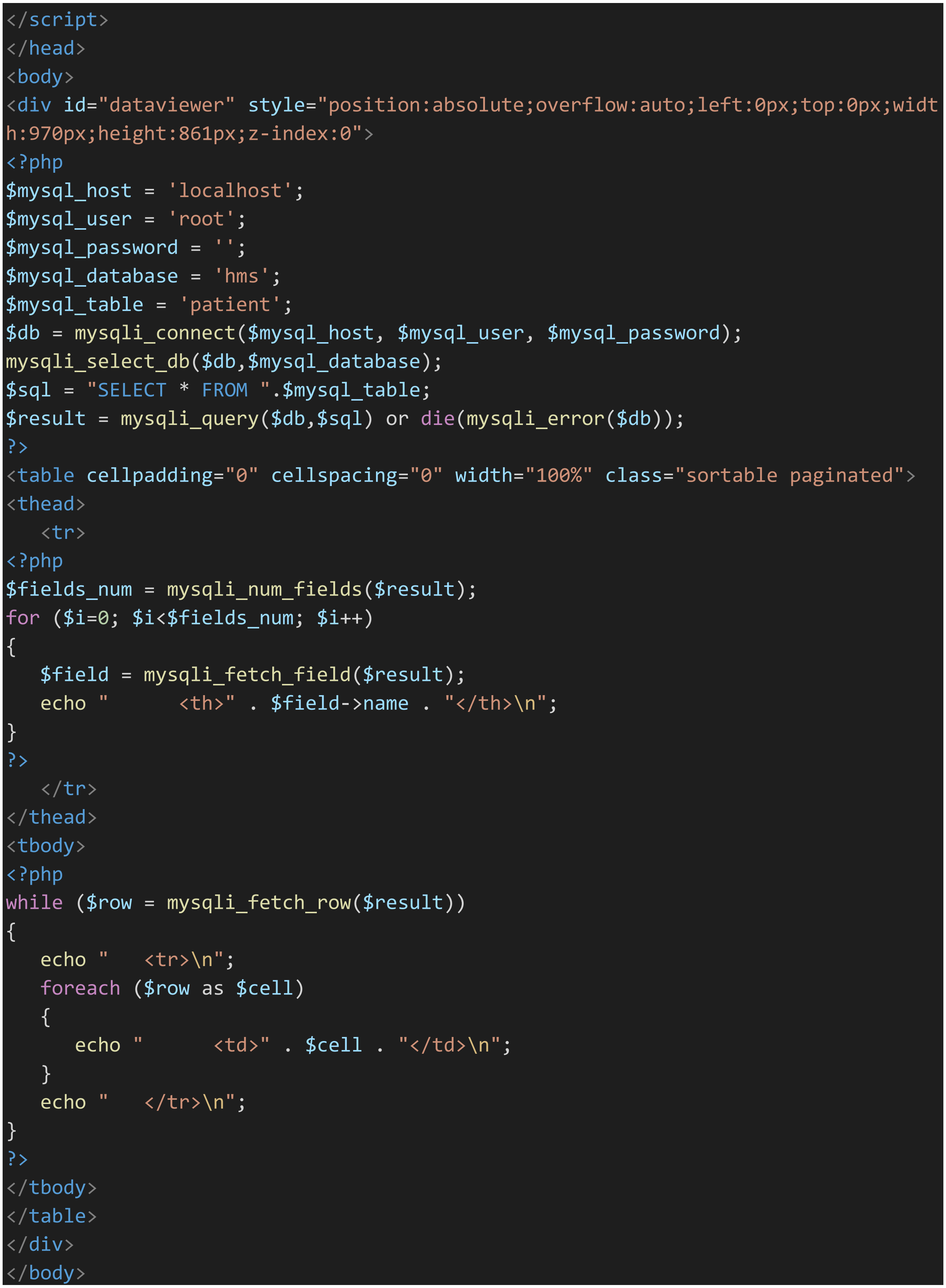
```
</table>
</div>
</body>
</html>
```

For Patient

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Page</title>
<meta name="generator" content="WYSIWYG Web Builder 15 -
 http://www.wysiwygwebbuilder.com">
<link href="HMS.css" rel="stylesheet">
<link href="viewer2.css" rel="stylesheet">
<script src="jquery-1.12.4.min.js"></script>
<script>
$(document).ready(function()
{
   $.fn.alternateRowColors = function()
   {
      $('tbody tr:odd', this).removeClass('even').addClass('odd');
      $('tbody tr:even', this).removeClass('odd').addClass('even');
      return this;
   };
   $('table.sortable').each(function()
   {
      var $dataviewer = $(this);
      $dataviewer.alternateRowColors();
      $('th', $dataviewer).each(function(column)
      {
         var $header = $(this);
         var findSortKey;
         findSortKey = function($cell)
         {
            return $cell.find('.sort-
key').text().toUpperCase() + ' ' + $cell.text().toUpperCase();
         };
         if (findSortKey)
         {
            $header.addClass('clickable').hover(function()
            {
               $header.addClass('hover');
```



```
            }, function()
            {
               $header.removeClass('hover');
            }).click(function()
            {
               var sortDirection = 1;
               if ($header.is('.sorted-asc'))
               {
                  sortDirection = -1;
               }
               var rows = $dataviewer.find('tbody > tr').get();
               $.each(rows, function(index, row)
               {
                  var $cell = $(row).children('td').eq(column);
                  row.sortKey = findSortKey($cell);
               });
               rows.sort(function(a, b)
               {
                  if (a.sortKey < b.sortKey) return -sortDirection;
                  if (a.sortKey > b.sortKey) return sortDirection;
                  return 0;
               });
               $.each(rows, function(index, row)
               {
                  $dataviewer.children('tbody').append(row);
                  row.sortKey = null;
               });
               $dataviewer.find('th').removeClass('sorted-
asc').removeClass('sorted-desc');
               if (sortDirection == 1)
               {
                  $header.addClass('sorted-asc');
               }
               else
               {
                  $header.addClass('sorted-desc');
               }
               $dataviewer.find('td').removeClass('sorted').filter(':nth-
child(' + (column + 1) + ')').addClass('sorted');
               $dataviewer.alternateRowColors();
            });
         }
      });
   });
});
```

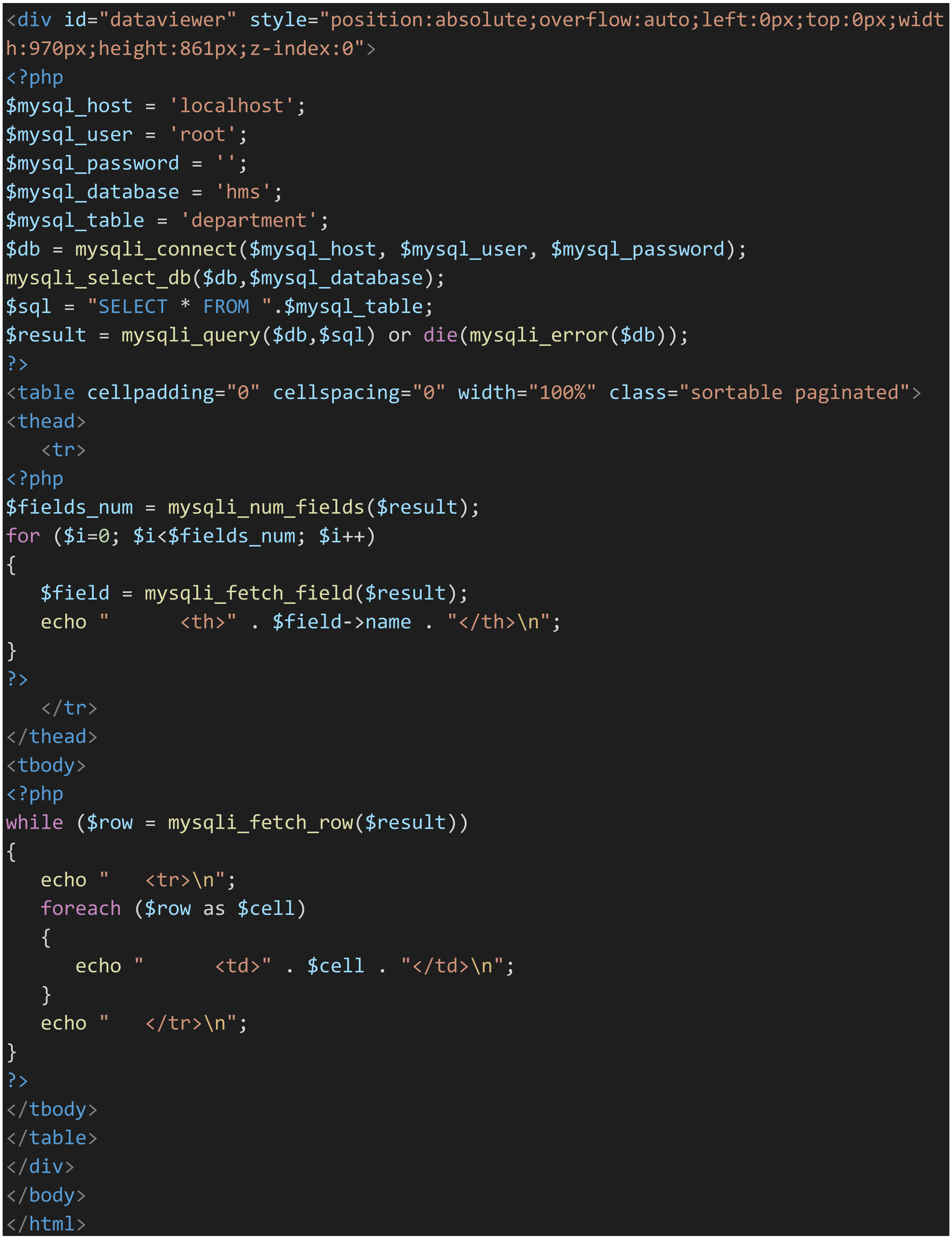


```
</script>
</head>
<body>
<div id="dataviewer" style="position:absolute;overflow:auto;left:0px;top:0px;width:970px;height:861px;z-index:0">
<?php
$mysql_host = 'localhost';
$mysql_user = 'root';
$mysql_password = '';
$mysql_database = 'hms';
$mysql_table = 'patient';
$db = mysqli_connect($mysql_host, $mysql_user, $mysql_password);
mysqli_select_db($db,$mysql_database);
$sql = "SELECT * FROM ".$mysql_table;
$result = mysqli_query($db,$sql) or die(mysqli_error($db));
?>
<table cellpadding="0" cellspacing="0" width="100%" class="sortable paginated">
<thead>
   <tr>
<?php
$fields_num = mysqli_num_fields($result);
for ($i=0; $i<$fields_num; $i++)
{
   $field = mysqli_fetch_field($result);
   echo "      <th>" . $field->name . "</th>\n";
}
?>
   </tr>
</thead>
<tbody>
<?php
while ($row = mysqli_fetch_row($result))
{
   echo "   <tr>\n";
   foreach ($row as $cell)
   {
      echo "      <td>" . $cell . "</td>\n";
   }
   echo "   </tr>\n";
}
?>
</tbody>
</table>
</div>
</body>
```

```
</html>
```

For Department

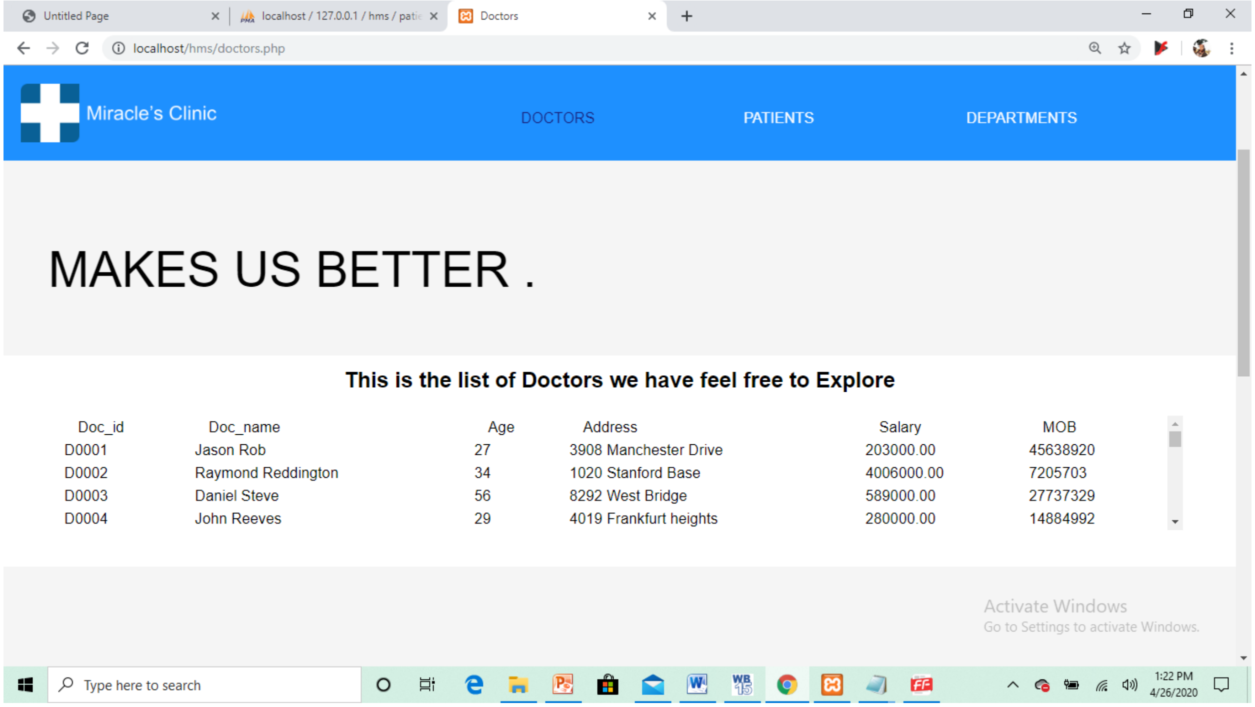
```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Page</title>
<meta name="generator" content="WYSIWYG Web Builder 15 -
 http://www.wysiwygwebbuilder.com">
<link href="HMS.css" rel="stylesheet">
<link href="page1.css" rel="stylesheet">
<script src="jquery-1.12.4.min.js"></script>
<script>
$(document).ready(function()
{
   $.fn.alternateRowColors = function()
   {
      $('tbody tr:odd', this).removeClass('even').addClass('odd');
      $('tbody tr:even', this).removeClass('odd').addClass('even');
      return this;
   };
   $('table.sortable').each(function()
   {
      var $dataviewer = $(this);
      $dataviewer.alternateRowColors();
      $('th', $dataviewer).each(function(column)
      {
         var $header = $(this);
         var findSortKey;
         findSortKey = function($cell)
         {
            return $cell.find('.sort-
key').text().toUpperCase() + ' ' + $cell.text().toUpperCase();
         };
         if (findSortKey)
         {
            $header.addClass('clickable').hover(function()
            {
               $header.addClass('hover');
            }, function()
            {
               $header.removeClass('hover');
            }).click(function()
            {
               var sortDirection = 1;
               if ($header.is('.sorted-asc'))
               {
                  sortDirection = -1;
               }
               var rows = $dataviewer.find('tbody > tr').get();
               $.each(rows, function(index, row)
               {
                  var $cell = $(row).children('td').eq(column);
                  row.sortKey = findSortKey($cell);
               });
               rows.sort(function(a, b)
               {
                  if (a.sortKey < b.sortKey) return -sortDirection;
                  if (a.sortKey > b.sortKey) return sortDirection;
                  return 0;
               });
               $.each(rows, function(index, row)
               {
                  $dataviewer.children('tbody').append(row);
                  row.sortKey = null;
               });
               $dataviewer.find('th').removeClass('sorted-
asc').removeClass('sorted-desc');
               if (sortDirection == 1)
               {
                  $header.addClass('sorted-asc');
               }
               else
               {
                  $header.addClass('sorted-desc');
               }
               $dataviewer.find('td').removeClass('sorted').filter(':nth-
child(' + (column + 1) + ')').addClass('sorted');
               $dataviewer.alternateRowColors();
            });
         }
      });
   });
});
</script>
</head>
<body>
```



```
<div id="dataviewer" style="position:absolute;overflow:auto;left:0px;top:0px;width:970px;height:861px;z-index:0">
<?php
$mysql_host = 'localhost';
$mysql_user = 'root';
$mysql_password = '';
$mysql_database = 'hms';
$mysql_table = 'department';
$db = mysqli_connect($mysql_host, $mysql_user, $mysql_password);
mysqli_select_db($db,$mysql_database);
$sql = "SELECT * FROM ".$mysql_table;
$result = mysqli_query($db,$sql) or die(mysqli_error($db));
?>
<table cellpadding="0" cellspacing="0" width="100%" class="sortable paginated">
<thead>
   <tr>
<?php
$fields_num = mysqli_num_fields($result);
for ($i=0; $i<$fields_num; $i++)
{
   $field = mysqli_fetch_field($result);
   echo "      <th>" . $field->name . "</th>\n";
}
?>
   </tr>
</thead>
<tbody>
<?php
while ($row = mysqli_fetch_row($result))
{
   echo "   <tr>\n";
   foreach ($row as $cell)
   {
      echo "      <td>" . $cell . "</td>\n";
   }
   echo "   </tr>\n";
}
?>
</tbody>
</table>
</div>
</body>
</html>
```

The Views of this page on the website:

Doctors:

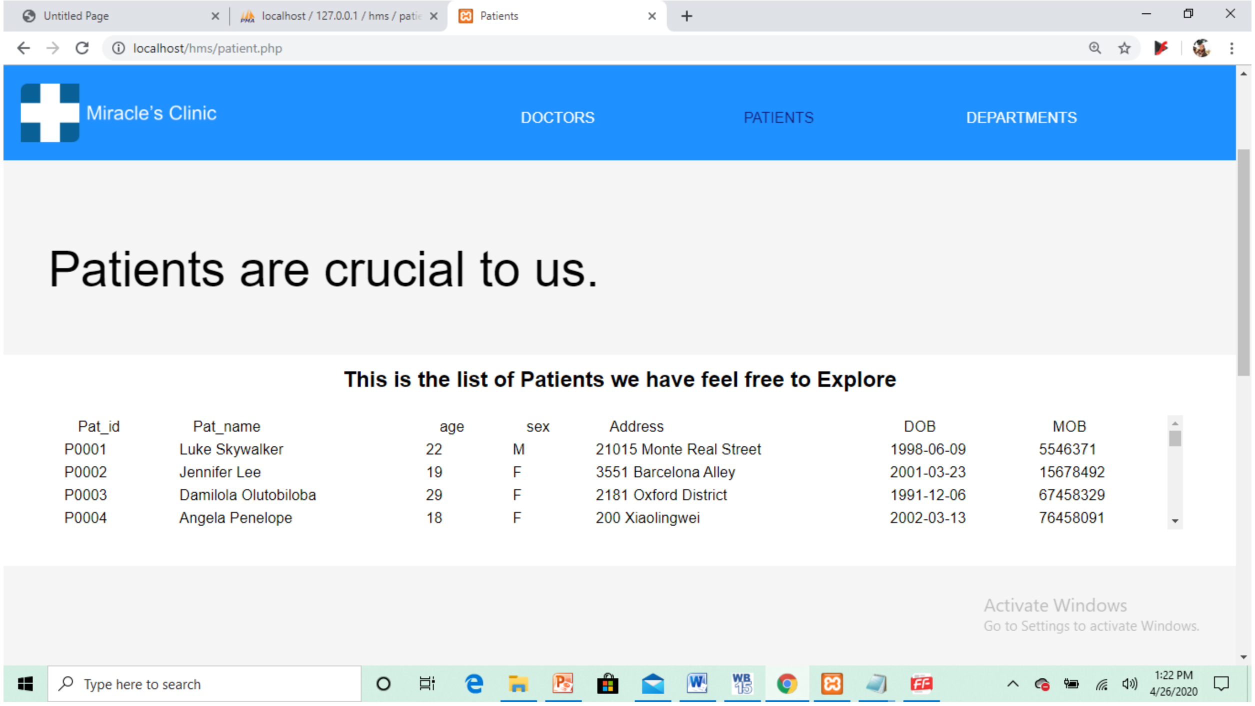


MAKES US BETTER .

This is the list of Doctors we have feel free to Explore

Doc_id	Doc_name	Age	Address	Salary	MOB
D0001	Jason Rob	27	3908 Manchester Drive	203000.00	45638920
D0002	Raymond Reddington	34	1020 Stanford Base	4006000.00	7205703
D0003	Daniel Steve	56	8292 West Bridge	589000.00	27737329
D0004	John Reeves	29	4019 Frankfurt heights	280000.00	14884992

Patients:

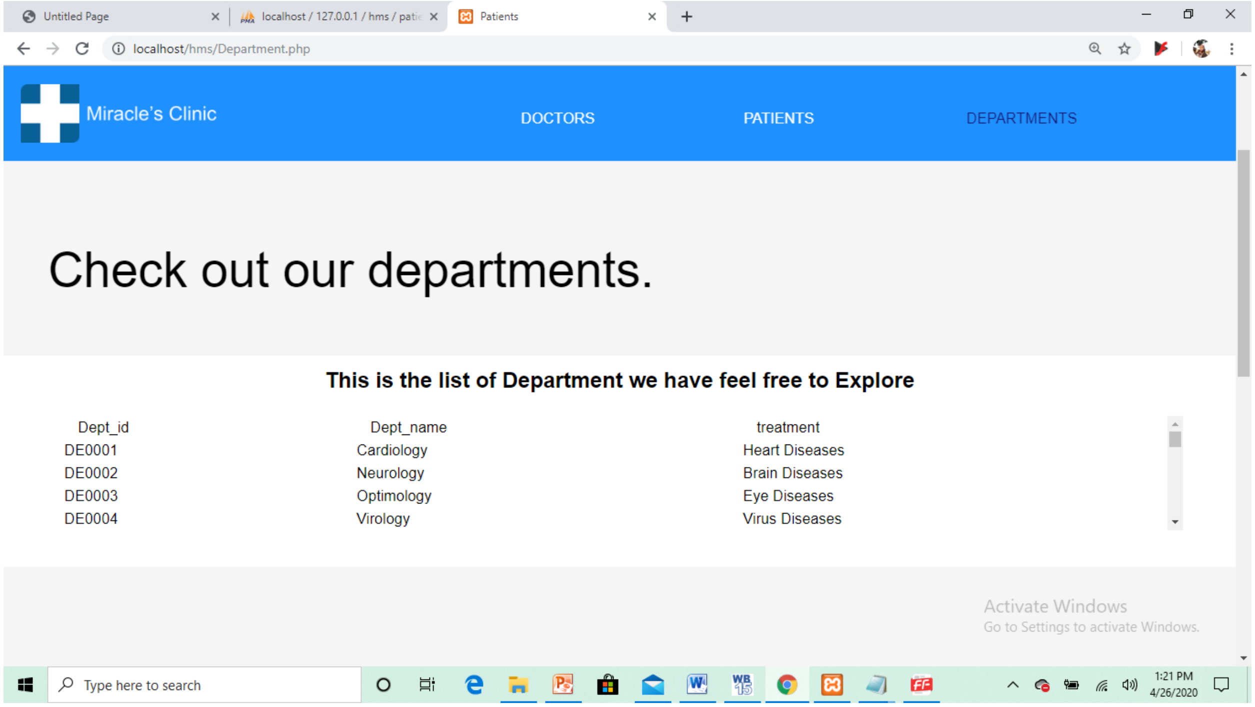


Patients are crucial to us.

This is the list of Patients we have feel free to Explore

Pat_id	Pat_name	age	sex	Address	DOB	MOB
P0001	Luke Skywalker	22	M	21015 Monte Real Street	1998-06-09	5546371
P0002	Jennifer Lee	19	F	3551 Barcelona Alley	2001-03-23	15678492
P0003	Damilola Olutobiloba	29	F	2181 Oxford District	1991-12-06	67458329
P0004	Angela Penelope	18	F	200 Xiaolingwei	2002-03-13	76458091

Department:



localhost/hms/Department.php

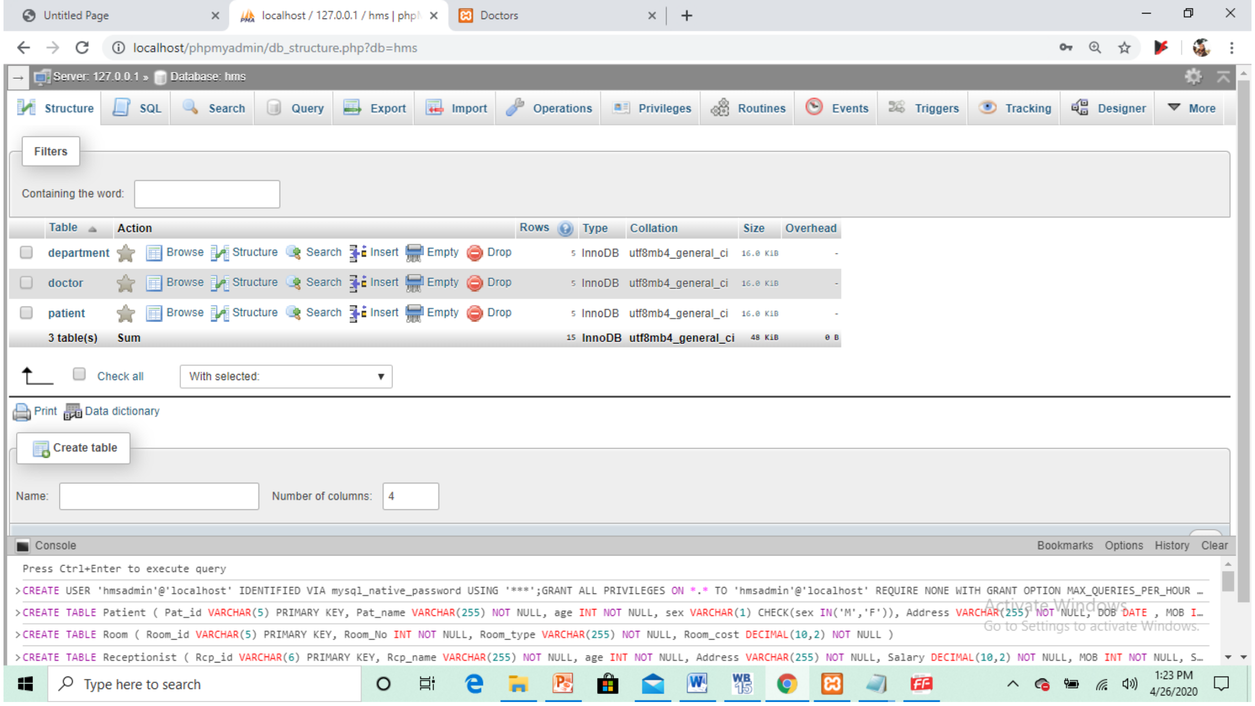
Miracle's Clinic — DOCTORS — PATIENTS — DEPARTMENTS

Check out our departments.

This is the list of Department we have feel free to Explore

Dept_id	Dept_name	treatment
DE0001	Cardiology	Heart Diseases
DE0002	Neurology	Brain Diseases
DE0003	Optimology	Eye Diseases
DE0004	Virology	Virus Diseases

This was done with the help of a local host.



localhost/phpmyadmin/db_structure.php?db=hms

Server: 127.0.0.1 » Database: hms

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
department	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
doctor	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
patient	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
3 table(s)	Sum	15	InnoDB	utf8mb4_general_ci	48 KiB	0 B

□ Check all — With selected:

Print — Data dictionary

Create table

Name: — Number of columns: 4

Console

```
Press Ctrl+Enter to execute query
>CREATE USER 'hmsadmin'@'localhost' IDENTIFIED VIA mysql_native_password USING '***';GRANT ALL PRIVILEGES ON *.* TO 'hmsadmin'@'localhost' REQUIRE NONE WITH GRANT OPTION MAX_QUERIES_PER_HOUR …
>CREATE TABLE Patient ( Pat_id VARCHAR(5) PRIMARY KEY, Pat_name VARCHAR(255) NOT NULL, age INT NOT NULL, sex VARCHAR(1) CHECK(sex IN('M','F')), Address VARCHAR(255) NOT NULL, DOB DATE , MOB I…
>CREATE TABLE Room ( Room_id VARCHAR(5) PRIMARY KEY, Room_No INT NOT NULL, Room_type VARCHAR(255) NOT NULL, Room_cost DECIMAL(10,2) NOT NULL )
>CREATE TABLE Receptionist ( Rcp_id VARCHAR(6) PRIMARY KEY, Rcp_name VARCHAR(255) NOT NULL, age INT NOT NULL, Address VARCHAR(255) NOT NULL, Salary DECIMAL(10,2) NOT NULL, MOB INT NOT NULL, S…
```