

Stage 3 – Database Implementation

CIS552 – Database Design

Prepared By: **Priyanka Kokkula & Rohan Gonjari**

Student ID: **01991135 & 01987448**

Investigator: **Professor Jiawei Yuan**

University: **University of Massachusetts, Dartmouth, MA - 02747**

Abstract

The following document provides the SQL scripts for creating the table in our database, inserting entries into said tables, and also for the queries mentioned in Stage 2.

SQL scripts for tables

Table 'patients':

```
CREATE TABLE patients (  
    Pat_ID INT AUTO_INCREMENT,  
    FName VARCHAR(255) NOT NULL,  
    LName VARCHAR(255) NOT NULL,  
    Email VARCHAR(255) NOT NULL UNIQUE,  
    Gender VARCHAR(255) NOT NULL,  
    Address VARCHAR(255) NOT NULL,  
    Age INT NOT NULL,  
    Password VARCHAR(255) NOT NULL,  
    PRIMARY Key (Pat_ID)  
)
```

Table 'med_history':

```
CREATE TABLE med_history (  
    Med_ID INT AUTO_INCREMENT,  
    Pat_ID INT,  
    Med_condition VARCHAR(255) NOT NULL,  
    Medication VARCHAR(255),  
    Surgery VARCHAR(255),  
    Mod_date DATE NOT NULL,  
    PRIMARY Key (Med_ID),  
    FOREIGN KEY(Pat_ID) REFERENCES patients(Pat_ID)  
)
```

Table 'schedules':

```
CREATE TABLE schedules (  
    Sc_ID INT PRIMARY KEY,  
    Start_time TIME NOT NULL,  
    End_time TIME NOT NULL  
)
```

Table 'doctors':

```
CREATE TABLE doctors (  
    Doc_ID INT AUTO_INCREMENT,  
    Email VARCHAR(255) NOT NULL UNIQUE,  
    FName VARCHAR(255) NOT NULL,  
    LName VARCHAR(255) NOT NULL,  
    Password VARCHAR(255) NOT NULL,  
    Sc_ID INT,  
    PRIMARY Key (Doc_ID),  
    FOREIGN KEY(Sc_ID) REFERENCES schedules(Sc_ID)  
)
```

Table 'appointments':

```
CREATE TABLE appointments (  
    App_ID INT AUTO_INCREMENT,  
    Pat_ID INT,  
    Doc_ID INT,  
    Diagnosis VARCHAR(255) NOT NULL,  
    Start_time TIME NOT NULL,  
    End_time TIME NOT NULL,  
    Symptoms VARCHAR(255) NOT NULL,  
    Prescription VARCHAR(255) NOT NULL,  
    App_date DATE NOT NULL,  
    PRIMARY KEY (App_ID),  
    FOREIGN KEY(Pat_ID) REFERENCES patients(Pat_ID),  
    FOREIGN KEY(Doc_ID) REFERENCES doctors(Doc_ID)  
)
```

Table 'invoice':

```
CREATE TABLE invoice (  
    In_ID INT AUTO_INCREMENT,  
    App_ID INT,  
    Log_date DATE NOT NULL,  
    Prod_price FLOAT NOT NULL,  
    Tax FLOAT,  
    Total FLOAT,  
    PRIMARY KEY (In_ID, App_ID),  
    FOREIGN KEY(App_ID) REFERENCES appointments(App_ID)  
)
```

Table 'drugs':

```
CREATE TABLE drugs (  
    Drug_ID INT AUTO_INCREMENT,  
    unit_price FLOAT NOT NULL,  
    Description VARCHAR(255) NOT NULL,
```

```

PRIMARY KEY (Drug_ID)
)

Table 'in_lines':
CREATE TABLE in_lines (
    Line_ID INT AUTO_INCREMENT,
    In_ID INT,
    Drug_ID INT,
    Quantity INT NOT NULL,
    PRIMARY KEY (Line_ID, In_ID),
    FOREIGN KEY(In_ID) REFERENCES invoice(In_ID),
    FOREIGN KEY(Drug_ID) REFERENCES drugs(Drug_ID)
)

```

SQL scripts for entries

Entries for 'patients':

```

INSERT INTO patients
VALUES
(1,"Priyanka","Kokkula","pkokkula@gmail.com","F","Boston",25,"password25"),
(2,"Rohan","Gonjari","rgonjari@gmail.com","M","New Bedford",24,"password24"),
(3,"John","Barea","bjhon@gmail.com","M","Quincy",21,"Jhon24"),
(4,"James","Casel","cjames@gmail.com","M","New Bedford",34,"james34"),
(5,"David","Silvia","sdavid@gmail.com","M","Boston",28,"david28"),
(6,"Anyelly","Mercedes","manyelly@gmail.com","F","Providence",29,"anyelly22"),
(7,"Esmeralda","Barea","besmeralda@gmail.com","F","New Bedford",31,"esmeralda21"),
(8,"Corely","Bravo","bcorely@gmail.com","F","New Bedford",24,"password24"),
(9,"Elsy","Sosy","esosy@gmail.com","F","Northend",32,"northend24"),
(10,"Estefeny","Sosa","esosy1@gmail.com","F","New Bedford",37,"sosy24"),
(11,"Marco","Ramirez","mramirez@gmail.com","M","New Bedford",36,"sosy24");

```

Entries for 'med_history':

```

INSERT INTO med_history
VALUES
(1,1,"disease12","drug1, drug4","N","2020-01-09"),
(2,2,"disease72","drug2","Y","2020-05-19"),
(3,3,"disease56","drug5, drug7, drug13","Y","2020-11-09"),
(4,4,"disease28","drug9","Y","2020-12-13"),
(5,5,"disease18","drug3, drug4","N","2021-03-05"),
(6,6,"disease48","drug8, drug14","Y","2021-08-12"),
(7,7,"disease52","drug7","Y","2021-12-19"),
(8,8,"disease82","drug4, drug 12","N","2022-02-05"),
(9,9,"disease87","drug6","N","2022-02-09"),
(10,10,"disease43","drug10, drug13","N","2022-05-09"),
(11,11,"disease29","drug11, drug15","Y","2022-10-13");

```

Entries for 'schedules':

```
INSERT INTO schedules
VALUES
(1,"00:00:00","06:00:00"),
(2,"06:00:00","12:00:00"),
(3,"12:00:00","18:00:00"),
(4,"18:00:00","24:00:00"),
(5,"03:00:00","09:00:00"),
(6,"09:00:00","15:00:00"),
(7,"15:00:00","21:00:00"),
(8,"21:00:00","03:00:00");
```

Entries for 'doctors':

```
INSERT INTO doctors
VALUES
(1,"doctor1@gmail.com","Roberto","Rea","rea123",4),
(2,"doctor2@gmail.com","Sandra","Rivera","sandra1",5),
(3,"doctor3@gmail.com","Bharat","Rao","rao23",1),
(4,"doctor4@gmail.com","Naveen","Naidu","naidu55",6),
(5,"doctor5@gmail.com","Shelly","Xhang","xang12",8),
(6,"doctor6@gmail.com","Haiping","Xu","xu4561123",2);
```

Entries for 'appointments':

```
INSERT INTO appointments
VALUES
(1,1,4,"diag_01","09:30:00","09:59:00","cold","drug2","2020-01-09"),
(2,1,4,"diag_02","10:30:00","10:59:00","cold","drug4","2020-04-20"),
(3,2,5,"diag_03","01:30:00","02:30:00","fever","drug6","2020-05-19"),
(4,2,1,"diag_04","22:30:00","24:00:00","high fever","drug7","2020-08-26"),
(5,3,2,"diag_05","04:00:00","04:15:00","body pain","drug13","2020-11-09"),
(6,4,6,"diag_06","06:00:00","06:15:00","body pain","drug15","2020-12-13"),
(7,5,3,"diag_07","05:00:00","05:45:00","cough","drug5","2021-03-05"),
(8,6,2,"diag_08","07:00:00","07:30:00","sore throat","drug3","2021-08-12"),
(9,7,5,"diag_09","21:00:00","22:00:00","fracture","drug10","2021-12-19"),
(10,8,6,"diag_10","08:00:00","08:30:00","back pain","drug1","2022-02-05"),
(11,9,3,"diag_11","00:00:00","01:30:00","arthritis","drug9","2022-02-09"),
(12,10,4,"diag_12","13:30:00","14:00:00","headache","drug12","2022-05-09"),
(13,11,3,"diag_13","00:30:00","01:00:00","stomach ache","drug14","2022-10-13");
```

Entries for 'invoice':

```
INSERT INTO invoice(In_ID, App_ID, Log_date, Prod_price)
VALUES
(1,1,"2020-01-09",8.00),
(2,2,"2020-04-20",13.00),
(3,3,"2020-05-19",12.00),
```

(4,4,"2020-08-26",45.00),
(5,5,"2020-11-09",4.00),
(6,6,"2020-12-13",25.00),
(7,7,"2021-03-05",5.00),
(8,8,"2021-08-12",37.00),
(9,9,"2021-12-19",20.00),
(10,10,"2022-02-05",15.00),
(11,11,"2022-02-09",7.00),
(12,12,"2022-05-09",14.00),
(13,13,"2022-10-13",16.00);

Entries for 'drugs':

INSERT INTO drugs

VALUES

(1,15.00,"drug1"),
(2,8.00,"drug2"),
(3,37.00,"drug3"),
(4,13.00,"drug4"),
(5,5.00,"drug5"),
(6,12.00,"drug6"),
(7,45.00,"drug7"),
(8,10.00,"drug8"),
(9,7.00,"drug9"),
(10,20.00,"drug10"),
(11,6.00,"drug11"),
(12,14.00,"drug12"),
(13,4.00,"drug13"),
(14,16.00,"drug14"),
(15,25.00,"drug15");

Entries for 'in_lines':

INSERT INTO in_lines

VALUES

(1,1,2,1),
(2,2,4,1),
(3,3,6,1),
(4,4,7,1),
(5,5,13,1),
(6,6,15,1),
(7,7,5,1),
(8,8,3,1),
(9,9,10,1),
(10,10,1,1),
(11,11,9,1),
(12,12,12,1),

(13,13,14,1);

Scripts for Queries

We have executed the following queries in MySQL and provided the SQL script below:

1. Write a SQL query to count the number of patients admitted to the hospital from Jan 2020 to Jan 2021.

Ans:

```
SELECT COUNT(App_date) AS 'Admitted' FROM appointments AS a, patients AS p WHERE  
a.Pat_ID=p.Pat_ID and a.App_date BETWEEN '2020-01-01' AND '2021-01-01';
```

2. Write a SQL query to display the patient id, first name, and drug name where the cost of the drug is more than 30\$, sorted by the drug name.

Ans:

```
Select pat.Pat_ID, pat.FName, pat.LName, drugs.unit_price AS 'Drug Unit Price', drugs.Description  
AS 'Drug Name'  
FROM patients as pat INNER JOIN appointments AS app INNER JOIN invoice AS inv INNER  
JOIN in_lines AS in_e INNER JOIN drugs  
ON pat.Pat_ID = app.Pat_ID AND app.App_ID = inv.App_ID AND inv.In_ID = in_e.In_ID AND  
in_e.Drug_ID = drugs.Drug_ID  
WHERE drugs.unit_price>=30  
ORDER BY drugs.Description
```

3. Write a SQL query to display the count of the number of appointments attended by each patient after 2021. Sort the results by the number of appointments in desc order.

Ans:

```
SELECT COUNT(Pat_ID) AS 'Number of Appointments' FROM appointments WHERE  
App_date>2021  
GROUP BY Pat_ID  
ORDER BY COUNT(Pat_ID) DESC;
```

4. Write a SQL query to display the doctor's id, name, and email whose end time is after 4 pm. Sort the results by doctor's name.

Ans:

```
SELECT d.Doc_ID, d.FName, d.Email  
FROM doctors AS d, schedules AS s  
WHERE s.Sc_ID=d.SC_ID AND s.End_time > '16:00:00'  
ORDER BY d.FName;
```

5. Write a SQL query to display the patient id, first name, and last name of all patients who have not had surgery yet. Sort the results by the patient's last name.

Ans:

```
SELECT DISTINCT m.Pat_Id, p.FName, p.LName from  
med_history AS m ,patients AS p WHERE p.Pat_ID=m.Pat_ID AND  
m.Surgery='N'  
ORDER BY p.FName;
```

6. Write a SQL query to display patients whose gender is female and, the tax they paid on medication after 2021.

Ans:

```
SELECT Pat_ID, FName, LName, Email, Gender, Age, tot_tax AS 'Total Tax'
```

```

FROM patients AS pat, (SELECT Pat_ID AS ID, SUM(Tax) AS tot_tax
                        FROM appointments ap INNER JOIN invoice inv
                        ON ap.App_ID=inv.App_ID
                        GROUP BY Pat_ID) AS tab
WHERE pat.Gender='F' AND pat.Pat_ID = tab.ID

```

7. Write a SQL query to list patients' first name, last name, and age, whose symptoms include "cold". Sort the results of patients by their last name and then by age in ascending order.

Ans:

```

SELECT DISTINCT a.Pat_ID, p.FName,P.LName,P.Age FROM appointments AS a, patients AS p
WHERE p.Pat_ID=a.Pat_ID AND a.Symptoms='cold'
ORDER BY P.LName,P.Age ASC;

```

8. Write a SQL query to display the patient id, last name, age, and count of their appointments that have lasted for more than 40 mins. Sort by the count of appointments in descending order.

Ans:

```

SELECT Pat_ID, FName, LName, Email, Gender, Age, count_app AS 'Appointments more than or
equal to 40 mins'
FROM patients AS pat, (SELECT Pat_ID AS ID, COUNT(Pat_ID) AS count_app
                        FROM appointments AS ap
                        WHERE (End_time - Start_time) >='4000'
                        GROUP BY ID) AS tab

WHERE pat.Pat_ID = tab.ID
ORDER BY count_app DESC

```

9. Write a SQL query to count the total number of invoices generated by each patient aged above 25 between 2021 and 2022.

Ans:

```

SELECT Pat_ID, FName, LName, Email, Gender, Age, inv_count AS 'Count of Invoices'
FROM patients AS pat, (SELECT Pat_ID AS ID, COUNT(Pat_ID) AS inv_count
                        FROM appointments ap INNER JOIN invoice inv
                        ON ap.App_ID=inv.App_ID
                        WHERE ap.App_date>='2021-01-01' AND
                        ap.App_date<='2021-12-31'
                        GROUP BY Pat_ID) AS tab
WHERE pat.Age>=25 AND pat.Pat_ID = tab.ID

```

10. Write a SQL query to display the doctor id, name, and the count of appointments they have had where the duration of the appointment is less than 15 mins. Sort by the count of appointments.

Ans:

```

SELECT Doc_ID, FName, LName, Email, count_app AS 'Appointments less than or equal to 15
mins'
FROM doctors AS doc, (SELECT Doc_ID AS ID, COUNT(Doc_ID) AS count_app
                       FROM appointments AS ap
                       WHERE (End_time - Start_time) <='1500'
                       GROUP BY ID) AS tab

WHERE doc.Doc_ID = tab.ID

```