CSU44D01 SQL PROJECT

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

SUBMITTED BY

NAME: VANSHIKA SINHA

TCD ID: 21355135

Application Description

I have designed an application containing the hospital management database. The system is designed to store the information related to interactions between doctors and patients, the remedies prescribed by doctors, details of diagnostic tests and room stays by patients and the final billing information. As in a hospital, my database has several entities, and each entity has several attributes:

- Patient: Patient_ID, Name, Age, Gender, Address
- Doctor: Doctor_ID, Name, Age, Gender, Address, Qualification, Specialization
- Room: Room_ID, Room_no, Room_type, Status, Location, Floor
- Lab_test: Test_ID, Date_of_test, Time_of_test, Result, Test_name
- Bill: Bill_no, Room_charge, No_of_days, Test_charge, Consultation_charge, Total_charge
- Consultation: Consultation_ID, Date_of_consultation, Time_of_consultation
- Prescription: Prescription_ID, Medicines_Treatments, Disease

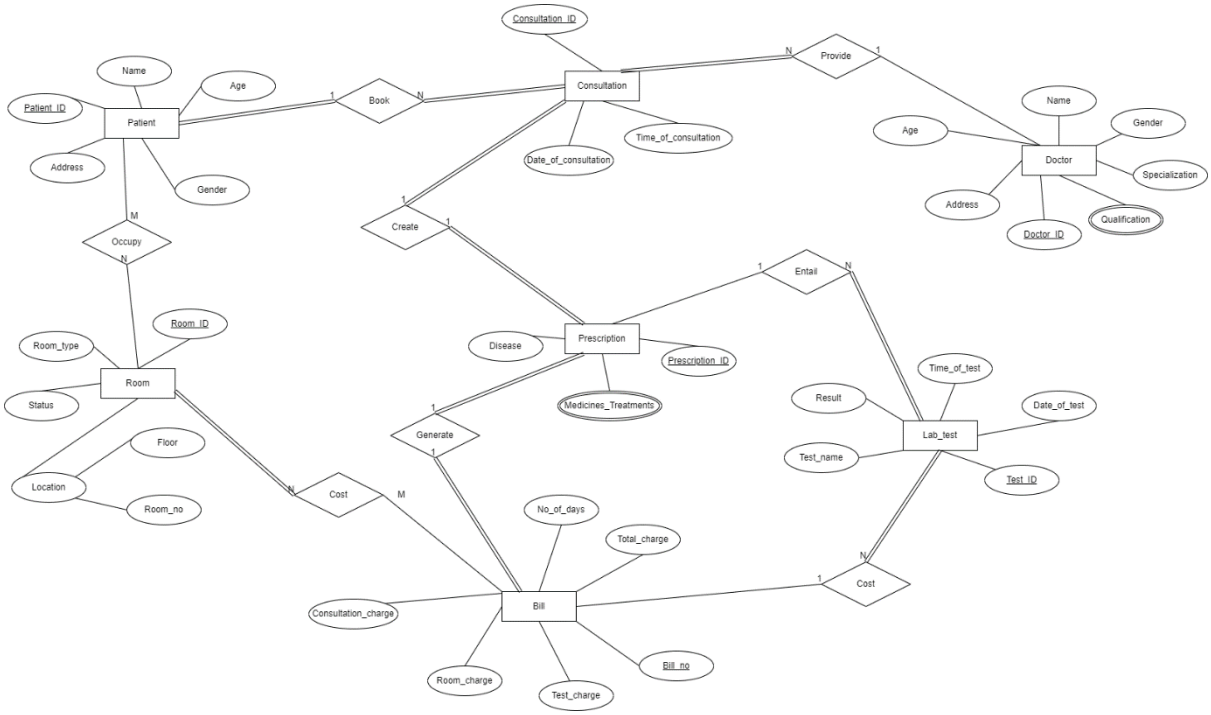
Description of Tables

Table Name	Description
PATIENT	This table stores the patient information for the patients who have availed consultation in the past.
DOCTOR	This table stores the doctors' information who work at the hospital.
CONSULTATION	This table stores the details of consultation between patients and doctors such as date, time, etc.
DOCTOR_QUALIFICATION	This table is to store the multiple qualifications that each doctor may have.
ROOM	This table is the list of all rooms in the hospital. The rooms are categorized as Normal, Premium and Intensive Care and can be occupied or not occupied at any point of time.
ROOM_BOOKING	This table is the mapping table for many-to-many relationship between patients and rooms.
PRESCRIPTION	This table stores the diagnosis given by the doctor after each consultation.
REMEDY	This table stores the medicines or other treatments suggested by the doctor as part of the prescription.

LAB_TEST	This table stores the details of diagnostic tests conducted at the hospital and their results.
BILL	This table stores the billing details for each prescription comprising of charges for rooms, tests and consultations.
ROOM_BILLING	This table is the mapping table for many-to-many relationship between rooms and bills.

Note: The ER diagram, Relational schema and functional dependency diagram have also been attached as separate files alongside the headings for easier viewing. Click on the attachment icon to view.

Entity Relationship Diagram

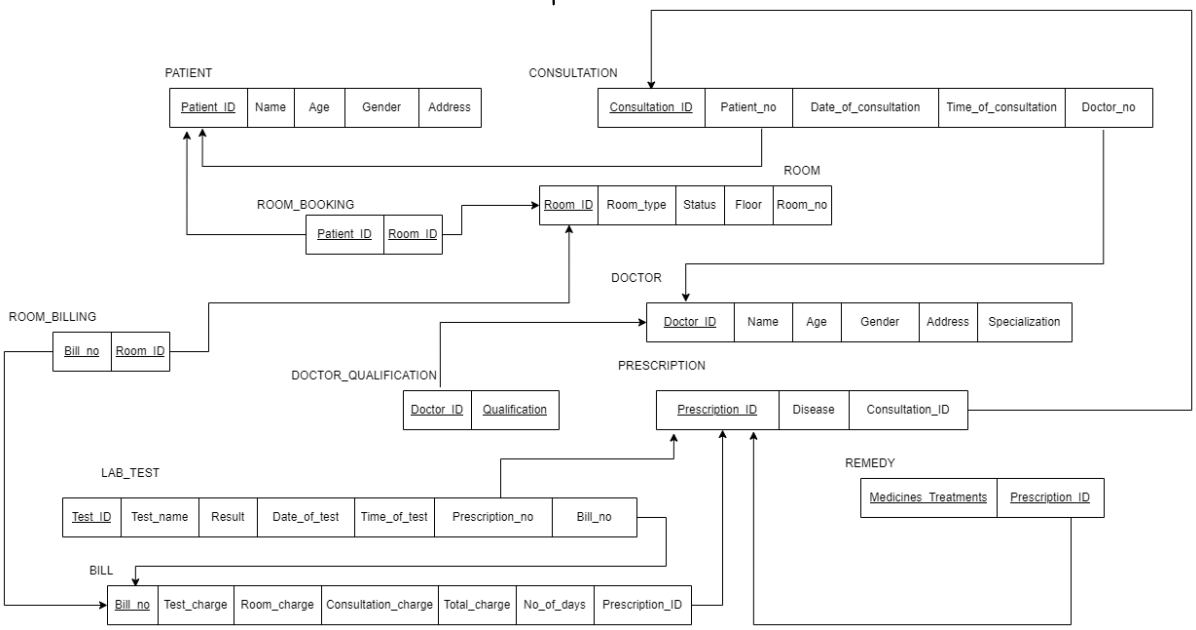


Description of Entities and their Relationships

The entities interact with each other according to the following relationships:

- Patients book a consultation which is provided by the doctors.
- A consultation leads to a creation of a prescription.
- The prescription may entail a lab test if the doctor prescribes one.
- Depending on the condition of the patient, the patient may have to occupy a room.
- The prescription generates a bill.
- The room costs and lab test costs along with the consultation costs are included in the bill.

Mapping to Relational Schema

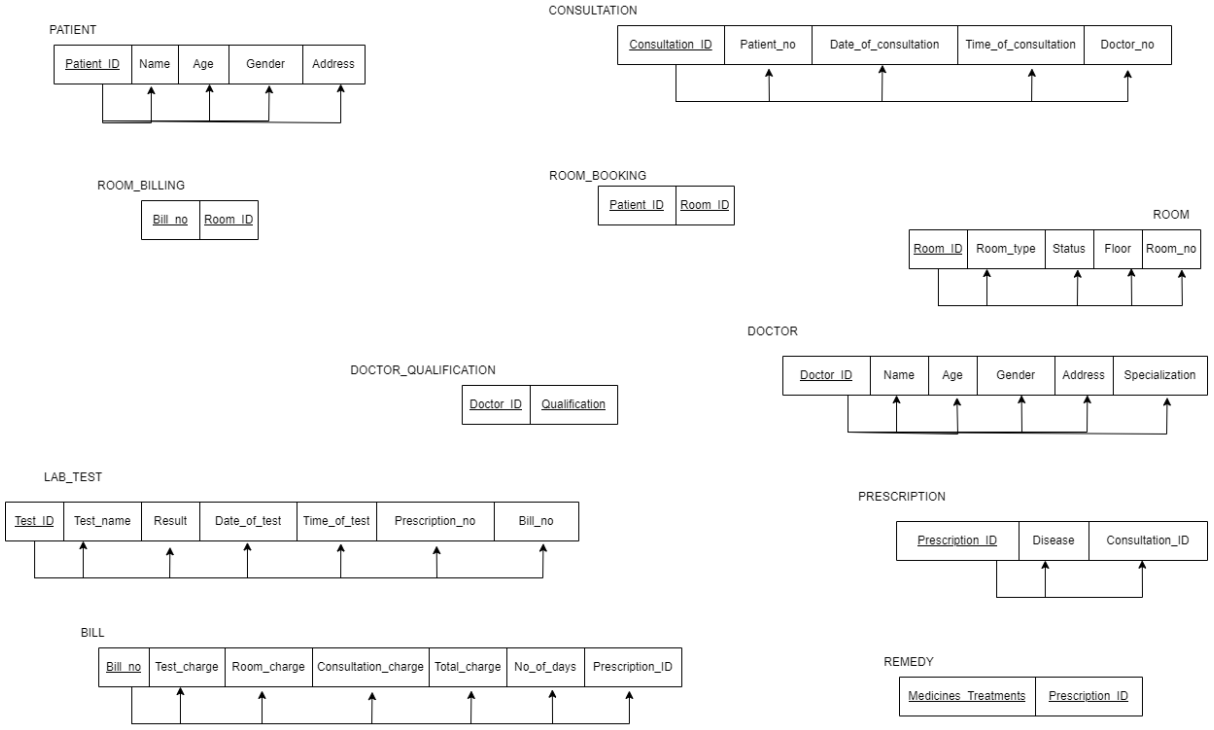


Mapping Rules

The mapping rules between tables are explained below:

- PATIENT is mapped to CONSULTATION through a foreign key to Patient_ID in CONSULTATION relation as they follow a one-to-many relationship.
- DOCTOR is mapped to CONSULTATION through a foreign key to Doctor_ID in CONSULTATION relation as they follow a one-to-many relationship.
- DOCTOR is mapped to DOCTOR_QUALIFICATION through a foreign key to Doctor_ID in DOCTOR_QUALIFICATION relation as it is a multivalued attribute of the DOCTOR entity.
- PATIENT is mapped to ROOM through a mapping table called ROOM_BOOKING as they follow a many-to-many relationship.
- CONSULTATION is mapped to a PRESCRIPTION through a foreign key to Consultation_ID in CONSULTATION relation as they follow a one-to-one relationship.
- PRESCRIPTION is mapped to REMEDY through a foreign key to Prescription_ID in REMEDY relation as it is a multivalued attribute of the PRESCRIPTION entity.
- PRESCRIPTION is mapped to BILL through a foreign key to Prescription_ID in BILL as they follow a one-to-one relationship.
- LAB_TEST is mapped to BILL through a foreign key to Bill_no in LAB_TEST relation as they have a many-to-one relationship.
- BILL is mapped to ROOM through a mapping table called ROOM_BILLING as they follow a many-to-many relationship.

Functional Dependency Diagram



The functional diagram shows that in all tables, non-key attributes are functionally dependent on the primary key in each table.

Explanation and SQL Code

Create and use database

```
Select MySQL 8.0 Command Line Client
Enter password: **********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 8.0.31 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE HOSPITAL_MANAGEMENT_SYSTEM;
Query OK, 1 row affected (0.03 sec)
```

We have created a database named HOSPITAL_MANAHGEMENT_SYSTEM.

```
mysql> USE HOSPITAL_MANAGEMENT_SYSTEM;
Database changed
mysql>
```

Creating tables

Here, we have created tables for all the entities along with the relevant table constraints, implicit constraints and explicit constraints.

PATIENT

```
mysql> CREATE TABLE PATIENT (
    -> Patient_ID INT NOT NULL PRIMARY KEY,
    -> Name VARCHAR(20) NOT NULL,
    -> Age INT NOT NULL CHECK (Age>=0 AND Age<=150),
    -> Gender VARCHAR(10) NOT NULL CHECK (Gender IN ('Male', 'Female', 'Others')),
    -> Address VARCHAR(100));
Query OK, 0 rows affected (0.06 sec)
```

DOCTOR

```
mysql> CREATE TABLE DOCTOR (
    -> Doctor_ID INT NOT NULL PRIMARY KEY,
    -> Name VARCHAR(20) NOT NULL,
    -> Age INT NOT NULL CHECK (Age>=0 AND Age<=150),
    -> Gender VARCHAR(10) NOT NULL CHECK (Gender IN ('Male', 'Female', 'Others')),
    -> Address VARCHAR(100) NOT NULL,
    -> Specialization VARCHAR(50) NOT NULL);
Query OK, 0 rows affected (0.02 sec)
```

DOCTOR_QUALIFICATION

```
mysql> CREATE TABLE DOCTOR_QUALIFICATION (
    -> Doctor_ID INT NOT NULL,
    -> FOREIGN KEY (Doctor_ID) REFERENCES DOCTOR (Doctor_ID)
    -> ON DELETE CASCADE ON UPDATE CASCADE,
    -> Qualification VARCHAR(50) NOT NULL,
    -> PRIMARY KEY (Doctor_ID, Qualification));
Query OK, 0 rows affected (0.05 sec)
```

ROOM

```
mysql> CREATE TABLE ROOM (
    -> Room_ID INT NOT NULL PRIMARY KEY,
    -> Room_type VARCHAR(30) NOT NULL CHECK (Room_type IN ('Normal', 'Premium', 'Intensive Care')),
    -> Status VARCHAR(20) NOT NULL CHECK (Status IN ('Occupied', 'Not occupied')),
    -> Floor INT NOT NULL CHECK (Floor>=1),
    -> Room_no INT NOT NULL);
Query OK, 0 rows affected (0.03 sec)
```

ROOM_BOOKING

```
mysql> CREATE TABLE ROOM_BOOKING (
    -> Patient_ID INT NOT NULL,
    -> FOREIGN KEY (Patient_ID) REFERENCES PATIENT (Patient_ID)
    -> ON DELETE CASCADE ON UPDATE CASCADE,
    -> Room_ID INT NOT NULL,
    -> FOREIGN KEY (Room_ID) REFERENCES ROOM (Room_ID)
    -> ON DELETE CASCADE ON UPDATE CASCADE,
    -> PRIMARY KEY (Patient_ID, Room_ID));
Query OK, 0 rows affected (0.06 sec)
```

CONSULTATION

```
mysql> CREATE TABLE CONSULTATION (
    -> Consultation_ID INT NOT NULL PRIMARY KEY,
    -> Patient_no INT NOT NULL,
    -> FOREIGN KEY (Patient_no) REFERENCES PATIENT (Patient_ID)
    -> ON DELETE CASCADE ON UPDATE CASCADE,
    -> Date_of_consultation DATE NOT NULL,
    -> Time_of_consultation TIME NOT NULL,
    -> Doctor_no INT NOT NULL,
    -> FOREIGN KEY (Doctor_no) REFERENCES DOCTOR (Doctor_ID)
    -> ON UPDATE CASCADE);
Query OK, 0 rows affected (0.06 sec)
```

PRESCRIPTION

```
mysql> CREATE TABLE PRESCRIPTION (
    -> Prescription_ID INT NOT NULL PRIMARY KEY,
    -> Disease VARCHAR(50) NOT NULL,
    -> Consultation_ID INT NOT NULL UNIQUE,
    -> FOREIGN KEY (Consultation_ID) REFERENCES CONSULTATION (Consultation_ID)
    -> ON DELETE CASCADE ON UPDATE CASCADE);
Query OK, 0 rows affected (0.04 sec)
```

REMEDY

```
mysql> CREATE TABLE REMEDY (
    -> Prescription_ID INT NOT NULL,
    -> FOREIGN KEY (Prescription_ID) REFERENCES PRESCRIPTION (Prescription_ID)
    -> ON DELETE CASCADE ON UPDATE CASCADE,
    -> Medicines_Treatments VARCHAR(100) NOT NULL,
    -> PRIMARY KEY (Prescription_ID, Medicines_Treatments));
Query OK, 0 rows affected (0.02 sec)
```

BILL

```
mysql> CREATE TABLE BILL (
    -> Bill_no INT NOT NULL PRIMARY KEY,
    -> Test_charge FLOAT,
    -> Room_charge FLOAT,
    -> Consultation_charge FLOAT,
    -> No_of_days INT CHECK (No_of_days>=0),
    -> Prescription_ID INT NOT NULL UNIQUE,
    -> FOREIGN KEY (Prescription_ID) REFERENCES PRESCRIPTION (Prescription_ID)
    -> ON DELETE CASCADE ON UPDATE CASCADE);
Query OK, 0 rows affected (0.06 sec)
```

ROOM_BILLING

```
mysql> CREATE TABLE ROOM_BILLING (
    -> Room_ID INT NOT NULL,
    -> FOREIGN KEY (Room_ID) REFERENCES ROOM (Room_ID)
    -> ON DELETE CASCADE ON UPDATE CASCADE,
    -> Bill_no INT NOT NULL,
    -> FOREIGN KEY (Bill_no) REFERENCES BILL (Bill_no)
    -> ON DELETE CASCADE ON UPDATE CASCADE,
    -> PRIMARY KEY (Bill_no, Room_ID));
Query OK, 0 rows affected (0.06 sec)
```

LAB_TEST

```
mysql> CREATE TABLE LAB_TEST (
    -> Test_ID INT NOT NULL PRIMARY KEY,
    -> Test_name VARCHAR(50) NOT NULL,
    -> Result VARCHAR(50) NOT NULL,
    -> Date_of_test DATE NOT NULL,
    -> Time_of_test TIME NOT NULL,
    -> Prescription_no INT NOT NULL,
    -> FOREIGN KEY (Prescription_no) REFERENCES PRESCRIPTION (Prescription_ID)
    -> ON DELETE CASCADE ON UPDATE CASCADE,
    -> Bill_no INT NOT NULL,
    -> FOREIGN KEY (Bill_no) REFERENCES BILL (Bill_no)
    -> ON DELETE CASCADE ON UPDATE CASCADE);
Query OK, 0 rows affected (0.04 sec)
```

Altering tables

```
mysql> ALTER TABLE BILL
    -> ADD Total_charge FLOAT;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Here, we have added a new column for Total_charge to the BILL relation.

Creating insert and update triggers

```
mysql> DELIMITER //
mysql> CREATE TRIGGER `Calculate_Total_Charge`
    -> BEFORE INSERT ON `BILL`
    -> FOR EACH ROW
    -> BEGIN
    -> SET NEW.Total_charge = NEW.Room_charge + NEW.Test_charge + NEW.Consultation_charge;
    -> END;//
Query OK, 0 rows affected (0.02 sec)

mysql> DELIMITER ;
mysql>
```

Here, we have used the insert trigger to insert the total charge on the bill.

```
mysql> DELIMITER //
mysql> CREATE TRIGGER Update_Total_Charge
    -> BEFORE UPDATE ON BILL
    -> FOR EACH ROW
    -> BEGIN
    -> IF !(NEW.Room_charge <=> OLD.Room_charge) OR !(NEW.Test_charge <=> OLD.Test_charge) OR !(NEW.Consultation_charge
<=> OLD.Consultation_charge) THEN
    -> SET NEW.Total_charge = NEW.Room_charge + NEW.Test_charge + NEW.Consultation_charge;
    -> END IF;
    -> END;//
Query OK, 0 rows affected, 3 warnings (0.01 sec)
```

We have used the update trigger to update the total charge on the bill if any of the component charges i.e., the room charge, test charge or consultation charge changes.

Creating views

```
mysql> CREATE VIEW PATIENT_DOCTOR_CONSULTATIONS AS
    -> SELECT P.Name AS 'Patient_Name', C.Date_of_consultation, C.Time_of_consultation, D.Name AS 'Doctor_Name'
    -> FROM PATIENT P
    -> INNER JOIN
    -> CONSULTATION C
    -> ON P.Patient_ID = C.Patient_no
    -> INNER JOIN
    -> DOCTOR D
    -> ON C.Doctor_no = D.Doctor_ID
    -> WHERE P.Gender = 'Male';
Query OK, 0 rows affected (0.01 sec)
```

A view has been created to show the patient and doctor consultations by using inner join for displaying the date and time of each consultation that a patient had with a doctor.

Populating values in tables

The relevant data has been inserted in the tables.

PATIENT

```
mysql> INSERT INTO PATIENT (Patient_ID, Name, Age, Gender, Address)
    -> VALUES (1, 'John Smith', 25, 'Male', 'B-15, North Avenue, Vincent Park, New York');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO PATIENT (Patient_ID, Name, Age, Gender, Address)
    -> VALUES (2, 'Oliver Patrick', 40, 'Male', 'C-17, North Avenue, Vincent Park, New York');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO PATIENT (Patient_ID, Name, Age, Gender, Address)
    -> VALUES (3, 'Jane Conlan', 30, 'Female', 'C-17, South Avenue, Central Park, Dallas');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PATIENT (Patient_ID, Name, Age, Gender, Address)
    -> VALUES (4, 'Charlie Powderly', 35, 'Male', 'A-11, South Avenue, Central Park, Dallas');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PATIENT (Patient_ID, Name, Age, Gender, Address)
    -> VALUES (5, 'Malini Singh', 60, 'Female', 'C-17, East Avenue, Central Park, Sydney');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PATIENT (Patient_ID, Name, Age, Gender, Address)
    -> VALUES (6, 'Albert Johnson', 23, 'Others', 'D-25, West Street, Vincent Park, Sydney');
Query OK, 1 row affected (0.00 sec)
```

DOCTOR

```
mysql> INSERT INTO DOCTOR (Doctor_ID, Name, Age, Gender, Address, Specialization)
    -> VALUES (1, 'Shahrukh Khan', 42, 'Male', 'E-17, West Street, Vincent Park, Sydney', 'Pediatrics');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO DOCTOR (Doctor_ID, Name, Age, Gender, Address, Specialization)
    -> VALUES (2, 'Neena Smith', 36, 'Female', 'D-16, East Street, Vincent Park, Berlin', 'Neurology');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO DOCTOR (Doctor_ID, Name, Age, Gender, Address, Specialization)
    -> VALUES (3, 'Lucy Johnson', 38, 'Female', 'D-16, North Avenue, Central Park, Berlin', 'Dentistry');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO DOCTOR (Doctor_ID, Name, Age, Gender, Address, Specialization)
    -> VALUES (4, 'William Wade', 51, 'Male', 'D-16, East Street, Vincent Park, London', 'Dentistry');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO DOCTOR (Doctor_ID, Name, Age, Gender, Address, Specialization)
    -> VALUES (5, 'Harry Hill', 56, 'Others', 'A-22, South Avenue, Central Park, London', 'Orthopedics');
Query OK, 1 row affected (0.01 sec)
```

DOCTOR QUALIFICATION

```
mysql> INSERT INTO DOCTOR_QUALIFICATION (Doctor_ID, Qualification)
    -> VALUES (1, 'MBBS');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO DOCTOR_QUALIFICATION (Doctor_ID, Qualification)
    -> VALUES (2, 'MBBS');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO DOCTOR_QUALIFICATION (Doctor_ID, Qualification)
    -> VALUES (2, 'MD');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO DOCTOR_QUALIFICATION (Doctor_ID, Qualification)
    -> VALUES (3, 'BDS');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO DOCTOR_QUALIFICATION (Doctor_ID, Qualification)
    -> VALUES (4, 'BDS');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO DOCTOR_QUALIFICATION (Doctor_ID, Qualification)
    -> VALUES (5, 'MBBS');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO DOCTOR_QUALIFICATION (Doctor_ID, Qualification)
    -> VALUES (5, 'MD');
Query OK, 1 row affected (0.01 sec)
```

CONSULTATION

```
mysql> INSERT INTO CONSULTATION (Consultation_ID, Patient_no, Date_of_consultation, Time_of_consultation, Doctor_no)
    -> VALUES (1, 1, '2022-07-13', '09:25', 3);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CONSULTATION (Consultation_ID, Patient_no, Date_of_consultation, Time_of_consultation, Doctor_no)
    -> VALUES (2, 1, '2021-07-18', '15:25', 2);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CONSULTATION (Consultation_ID, Patient_no, Date_of_consultation, Time_of_consultation, Doctor_no)
    -> VALUES (3, 2, '2022-09-10', '12:30', 1);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CONSULTATION (Consultation_ID, Patient_no, Date_of_consultation, Time_of_consultation, Doctor_no)
    -> VALUES (4, 2, '2022-03-19', '14:45', 2);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CONSULTATION (Consultation_ID, Patient_no, Date_of_consultation, Time_of_consultation, Doctor_no)
    -> VALUES (5, 2, '2021-08-12', '12:50', 4);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CONSULTATION (Consultation_ID, Patient_no, Date_of_consultation, Time_of_consultation, Doctor_no)
    -> VALUES (6, 3, '2022-05-25', '11:10', 4);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CONSULTATION (Consultation_ID, Patient_no, Date_of_consultation, Time_of_consultation, Doctor_no)
    -> VALUES (7, 4, '2022-06-23', '13:40', 5);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CONSULTATION (Consultation_ID, Patient_no, Date_of_consultation, Time_of_consultation, Doctor_no)
    -> VALUES (8, 4, '2022-06-24', '11:10', 5);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CONSULTATION (Consultation_ID, Patient_no, Date_of_consultation, Time_of_consultation, Doctor_no)
    -> VALUES (9, 4, '2022-06-25', '16:40', 5);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CONSULTATION (Consultation_ID, Patient_no, Date_of_consultation, Time_of_consultation, Doctor_no)
    -> VALUES (10, 5, '2022-10-28', '11:30', 1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CONSULTATION (Consultation_ID, Patient_no, Date_of_consultation, Time_of_consultation, Doctor_no)
    -> VALUES (11, 5, '2021-02-14', '16:40', 2);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CONSULTATION (Consultation_ID, Patient_no, Date_of_consultation, Time_of_consultation, Doctor_no)
    -> VALUES (12, 6, '2022-10-20', '18:30', 3);
Query OK, 1 row affected (0.01 sec)
```

ROOM

```
mysql> INSERT INTO ROOM (Room_ID, Room_type, Status, Floor, Room_no)
    -> VALUES (1, 'Normal', 'Occupied', 1, 14)
    -> ^C
mysql> INSERT INTO ROOM (Room_ID, Room_type, Status, Floor, Room_no)
    -> VALUES (1, 'Normal', 'Occupied', 1, 14);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO ROOM (Room_ID, Room_type, Status, Floor, Room_no)
    -> VALUES (2, 'Normal', 'Not occupied', 1, 20);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO ROOM (Room_ID, Room_type, Status, Floor, Room_no)
    -> VALUES (3, 'Premium', 'Occupied', 2, 12);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO ROOM (Room_ID, Room_type, Status, Floor, Room_no)
    -> VALUES (4, 'Premium', 'Occupied', 2, 17);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO ROOM (Room_ID, Room_type, Status, Floor, Room_no)
    -> VALUES (5, 'Intensive Care', 'Unoccupied', 3, 5);
ERROR 3819 (HY000): Check constraint 'room_chk_2' is violated.
mysql> INSERT INTO ROOM (Room_ID, Room_type, Status, Floor, Room_no)
    -> VALUES (5, 'Intensive Care', 'Not occupied', 3, 5);
Query OK, 1 row affected (0.00 sec)
```

ROOM_BOOKING

```
mysql> INSERT INTO ROOM_BOOKING (Patient_ID, Room_ID)
    -> VALUES (4, 5);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO ROOM_BOOKING (Patient_ID, Room_ID)
    -> VALUES (4, 1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO ROOM_BOOKING (Patient_ID, Room_ID)
    -> VALUES (5, 3);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO ROOM_BOOKING (Patient_ID, Room_ID)
    -> VALUES (3, 3);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO ROOM_BOOKING (Patient_ID, Room_ID)
    -> VALUES (1, 2);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO ROOM_BOOKING (Patient_ID, Room_ID)
    -> VALUES (2, 4);
Query OK, 1 row affected (0.01 sec)
```

PRESCRIPTION

```
mysql> INSERT INTO PRESCRIPTION (Prescription_ID, Disease, Consultation_ID)
    -> VALUES (1, 'Gingivitis', 1);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PRESCRIPTION (Prescription_ID, Disease, Consultation_ID)
    -> VALUES (2, 'Stroke', 2);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PRESCRIPTION (Prescription_ID, Disease, Consultation_ID)
    -> VALUES (3, 'Malaria', 3);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PRESCRIPTION (Prescription_ID, Disease, Consultation_ID)
    -> VALUES (4, 'Stroke', 4);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PRESCRIPTION (Prescription_ID, Disease, Consultation_ID)
    -> VALUES (5, 'Gingivitis', 5);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PRESCRIPTION (Prescription_ID, Disease, Consultation_ID)
    -> VALUES (6, 'Gingivitis', 6);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PRESCRIPTION (Prescription_ID, Disease, Consultation_ID)
    -> VALUES (7, 'Arthritis', 7);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PRESCRIPTION (Prescription_ID, Disease, Consultation_ID)
    -> VALUES (8, 'Back-pain', 8);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PRESCRIPTION (Prescription_ID, Disease, Consultation_ID)
    -> VALUES (9, 'Back-pain', 9);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PRESCRIPTION (Prescription_ID, Disease, Consultation_ID)
    -> VALUES (10, 'Common-cold', 10);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PRESCRIPTION (Prescription_ID, Disease, Consultation_ID)
    -> VALUES (11, 'Brain-tumour', 11);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PRESCRIPTION (Prescription_ID, Disease, Consultation_ID)
    -> VALUES (12, 'Cavities', 12);
Query OK, 1 row affected (0.01 sec)
```

REMEDY

```
mysql> INSERT INTO REMEDY (Medicines_Treatments, Prescription_ID)
    -> VALUES ('Sensitive toothpaste', 1);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO REMEDY (Medicines_Treatments, Prescription_ID)
    -> VALUES ('Paracetamol', 2);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO REMEDY (Medicines_Treatments, Prescription_ID)
    -> VALUES ('Paracetamol', 3);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO REMEDY (Medicines_Treatments, Prescription_ID)
    -> VALUES ('Regular exercise', 4);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO REMEDY (Medicines_Treatments, Prescription_ID)
    -> VALUES ('Sensitive toothpaste', 5);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO REMEDY (Medicines_Treatments, Prescription_ID)
    -> VALUES ('Mouthwash', 6);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO REMEDY (Medicines_Treatments, Prescription_ID)
    -> VALUES ('Bedrest', 7);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO REMEDY (Medicines_Treatments, Prescription_ID)
    -> VALUES ('Regular exercise', 8);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO REMEDY (Medicines_Treatments, Prescription_ID)
    -> VALUES ('Bedrest', 9);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO REMEDY (Medicines_Treatments, Prescription_ID)
    -> VALUES ('Benadril', 10);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO REMEDY (Medicines_Treatments, Prescription_ID)
    -> VALUES ('Bedrest', 11);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO REMEDY (Medicines_Treatments, Prescription_ID)
    -> VALUES ('Sensitive toothpaste', 12);
Query OK, 1 row affected (0.01 sec)
```

BILL

```
mysql> INSERT INTO LAB_TEST (Test_ID, Test_name, Result, Date_of_test, Time_of_test, Prescription_no, Bill_no)
    -> VALUES (1, 'Blood-test', 'Anaemia', '2022-07-13', '11:00', 1, 1);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`hospital_management_system`.`lab_test`, CONSTRAINT `lab_test_ibfk_2` FOREIGN KEY (`Bill_no`) REFERENCES `bill` (`Bill_no`) O
N DELETE CASCADE ON UPDATE CASCADE)
mysql> INSERT INTO BILL (Bill_no, Test_charge, Room_charge, Consultation_charge, No_of_days, Prescription_ID)
    -> VALUES (1, 11.5, 50.0, 20.3, 1, 1);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO BILL (Bill_no, Test_charge, Room_charge, Consultation_charge, No_of_days, Prescription_ID)
    -> VALUES (2, 0.0, 0.0, 25.3, 0, 2);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO BILL (Bill_no, Test_charge, Room_charge, Consultation_charge, No_of_days, Prescription_ID)
    -> VALUES (3, 0.0, 50.0, 12.6, 2, 3);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO BILL (Bill_no, Test_charge, Room_charge, Consultation_charge, No_of_days, Prescription_ID)
    -> VALUES (4, 50.0, 150.0, 48.9, 3, 4);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO BILL (Bill_no, Test_charge, Room_charge, Consultation_charge, No_of_days, Prescription_ID)
    -> VALUES (5, 0.0, 0.0, 25.4, 0, 5);
Query OK, 1 row affected (0.01 sec)

mysql> DELETE FROM BILL WHERE Bill_no=4;
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO BILL (Bill_no, Test_charge, Room_charge, Consultation_charge, No_of_days, Prescription_ID)
    -> VALUES (4, 30.0, 0.0, 28.5, 0, 4);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO BILL (Bill_no, Test_charge, Room_charge, Consultation_charge, No_of_days, Prescription_ID)
    -> VALUES (6, 15.5, 100.0, 25.4, 2, 6);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO BILL (Bill_no, Test_charge, Room_charge, Consultation_charge, No_of_days, Prescription_ID)
    -> VALUES (7, 20.1, 100.0, 46.8, 2, 7);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO BILL (Bill_no, Test_charge, Room_charge, Consultation_charge, No_of_days, Prescription_ID)
    -> VALUES (8, 0.0, 200.0, 50.3, 2, 8);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO BILL (Bill_no, Test_charge, Room_charge, Consultation_charge, No_of_days, Prescription_ID)
    -> VALUES (9, 0.0, 100.0, 30.7, 1, 9);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO BILL (Bill_no, Test_charge, Room_charge, Consultation_charge, No_of_days, Prescription_ID)
    -> VALUES (10, 0.0, 50.0, 28.5, 1, 10);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO BILL (Bill_no, Test_charge, Room_charge, Consultation_charge, No_of_days, Prescription_ID)
    -> VALUES (11, 0.0, 0.0, 36.6, 0, 11);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO BILL (Bill_no, Test_charge, Room_charge, Consultation_charge, No_of_days, Prescription_ID)
    -> VALUES (12, 12.2, 0.0, 36.6, 0, 12);
Query OK, 1 row affected (0.01 sec)
```

Total_charge is not added to the bill as it will be calculated by the trigger Calculate_Total_Charge.

LAB_TEST

```
mysql> INSERT INTO LAB_TEST (Test_ID, Test_name, Result, Date_of_test, Time_of_test, Prescription_no, Bill_no)
    -> VALUES (1, 'Blood test', 'Anaemia', '2022-07-13', '11:00', 1, 1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO LAB_TEST (Test_ID, Test_name, Result, Date_of_test, Time_of_test, Prescription_no, Bill_no)
    -> VALUES (2, 'Liver function test', 'Cirrhosis', '2022-03-19', '15:23', 4, 4);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO LAB_TEST (Test_ID, Test_name, Result, Date_of_test, Time_of_test, Prescription_no, Bill_no)
    -> VALUES (3, 'Tooth X-Ray', 'Cavity', '2022-05-25', '12:15', 6, 6);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO LAB_TEST (Test_ID, Test_name, Result, Date_of_test, Time_of_test, Prescription_no, Bill_no)
    -> VALUES (4, 'Back X-Ray', 'Osteoporosis', '2022-06-23', '14:23', 7, 7);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO LAB_TEST (Test_ID, Test_name, Result, Date_of_test, Time_of_test, Prescription_no, Bill_no)
    -> VALUES (5, 'Tooth X-Ray', 'Cavity', '2022-10-20', '20:10', 12, 12);
Query OK, 1 row affected (0.00 sec)
```

ROOM_BILLING

```
mysql> INSERT INTO ROOM_BILLING (Bill_no, Room_ID)
    -> VALUES (1, 1);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO ROOM_BILLING (Bill_no, Room_ID)
    -> VALUES (3, 1);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO ROOM_BILLING (Bill_no, Room_ID)
    -> VALUES (6, 2);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO ROOM_BILLING (Bill_no, Room_ID)
    -> VALUES (7, 2);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO ROOM_BILLING (Bill_no, Room_ID)
    -> VALUES (8, 3);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO ROOM_BILLING (Bill_no, Room_ID)
    -> VALUES (9, 5);
Query OK, 1 row affected (0.01 sec)

mysql> DELETE FROM ROOM_BILLING WHERE Bill_no=8;
Query OK, 1 row affected (0.01 sec)

mysql> DELETE FROM ROOM_BILLING WHERE Bill_no=9;
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO ROOM_BILLING (Bill_no, Room_ID)
    -> VALUES (8, 5);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO ROOM_BILLING (Bill_no, Room_ID)
    -> VALUES (9, 3);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO ROOM_BILLING (Bill_no, Room_ID)
    -> VALUES (10, 2);
Query OK, 1 row affected (0.01 sec)
```

Retrieving information from the database

```
mysql> SELECT P.Name AS 'Patient_Name', C.Date_of_consultation, PRES.Disease, R.Medicines_Treatments FROM
    -> PATIENT P
    -> INNER JOIN
    -> CONSULTATION C
    -> ON P.Patient_ID = C.Patient_no
    -> INNER JOIN
    -> PRESCRIPTION PRES
    -> ON PRES.Consultation_ID = C.Consultation_ID
    -> INNER JOIN
    -> REMEDY R
    -> ON PRES.Prescription_ID = R.Prescription_ID;
+----------------+---------------------+-------------+---------------------+
| Patient_Name   | Date_of_consultation | Disease     | Medicines_Treatments |
+----------------+---------------------+-------------+---------------------+
| John Smith     | 2022-07-13          | Gingivitis  | Sensitive toothpaste |
| John Smith     | 2021-07-18          | Stroke      | Paracetamol          |
| Oliver Patrick | 2022-09-10          | Malaria     | Paracetamol          |
| Oliver Patrick | 2022-03-19          | Stroke      | Regular exercise     |
| Oliver Patrick | 2021-08-12          | Gingivitis  | Sensitive toothpaste |
| Jane Conlan    | 2022-05-25          | Gingivitis  | Mouthwash            |
| Charlie Powderly | 2022-06-23        | Arthritis   | Bedrest              |
| Charlie Powderly | 2022-06-24        | Back-pain   | Regular exercise     |
| Charlie Powderly | 2022-06-25        | Back-pain   | Bedrest              |
| Malini Singh   | 2022-10-28          | Common-cold | Benadril             |
| Malini Singh   | 2021-02-14          | Brain-tumour | Bedrest             |
| Albert Johnson | 2022-10-20          | Cavities    | Sensitive toothpaste |
+----------------+---------------------+-------------+---------------------+
12 rows in set (0.00 sec)
```

The above SQL command joins the PATIENT, CONSULTATION, PRESCRIPTION and REMEDY tables to show when a patient took an appointment, what was the diagnostic and treatments given.

```
mysql> SELECT D.Name AS 'Doctor_Name', COUNT(C.Consultation_ID) AS 'No_of_consultations' FROM
    -> DOCTOR D
    -> LEFT JOIN
    -> CONSULTATION C
    -> ON D.Doctor_ID = C.Doctor_no
    -> GROUP BY D.Doctor_ID;
+---------------+---------------------+
| Doctor_Name   | No_of_consultations |
+---------------+---------------------+
| Shahrukh Khan |                   2 |
| Neena Smith   |                   3 |
| Lucy Johnson  |                   2 |
| William Wade  |                   2 |
| Harry Hill    |                   3 |
+---------------+---------------------+
5 rows in set (0.01 sec)
```

The above SQL command shows which doctor is the most popular by counting how many consultations each doctor has completed.

Creating new users and managing privileges for security

```
mysql> GRANT INSERT, DELETE, UPDATE, SELECT ON PATIENT TO admin;
ERROR 1410 (42000): You are not allowed to create a user with GRANT
mysql> CREATE USER 'admin'@'localhost' IDENTIFIED BY 'Vanshika@03';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT INSERT, DELETE, UPDATE, SELECT ON PATIENT TO admin;
ERROR 1410 (42000): You are not allowed to create a user with GRANT
mysql> GRANT INSERT, DELETE, UPDATE, SELECT ON PATIENT TO 'admin'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT INSERT, DELETE, UPDATE, SELECT ON DOCTOR TO 'admin'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE USER 'finance'@'localhost' IDENTIFIED BY 'Vanshika@03';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT INSERT, DELETE, UPDATE, SELECT ON BILL TO 'finance'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT UPDATE, SELECT ON ROOM_BILLING TO 'finance'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT UPDATE, SELECT ON LAB_TEST TO 'finance'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> REVOKE UPDATE ON LAB_TEST FROM 'finance'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

Two new users, admin and finance are created and provided access to the relevant tables for their use case. The admin manages the patient and doctor data, while finance has access to room, test and billing information.

References

- https://www.w3resource.com/mysql/mysql-tutorials.php
- https://dev.mysql.com/doc/
- https://opentextbc.ca/dbdesign01/chapter/chapter-11-functional-dependencies/