# ACCOLITE UNIVERSITY, JULY 2020

# SQL Concepts & Fundamentals

# Assignment Submission

By: Ayush Malik

The following problem description and assumptions have been taken into consideration while designing the database of the hospital system.

PROBLEM DESCRIPTION AND ASSUMPTIONS:

- The hospital management system handles 3 types of records: Patient, Doctor and Department.
- Every time a patient is admitted to the hospital, a new record is created. Even if the same patient gets admitted multiple times, a different record for him/her will be created each time. Once admitted, the patient can be visited by different doctors any number of times before he/she is discharged.
- A patient is admitted to only one department of the hospital.
- A doctor can belong to only one department of the hospital.
- A doctor can treat a patient from any department.
- A patient can be treated/visited by any number of doctors from any department.
- A patient's record consists of the following: patient id, name, identity proof number, contact person one, contact person two, date of admission, date of discharge, hospital charges, date of birth, department id, medical condition, phone number, blood group, the doctors he/she has been treated by along with the corresponding dates of visit/treatment, hospital fee.
  - Patient's id uniquely identifies each patient.
  - For both contact persons of the patient, the following information will be stored: name, phone number and address.
  - It is assumed that the visiting dates of the doctors will be between the date of admission and date of discharge of the patient (both inclusive).
  - Hospital Fee is further divided into the following types of fee: Doctor Fee, Room Charges, Medicine, Food and Miscellaneous.
  - It is assumed that the date of admission is always before the date of discharge or the same as the date of discharge.
  - A patient can be visited by the same doctor multiple times.
- A doctor's record consists of the following: doctor's id, name, identity proof number, department id, contact number and visiting charges.
  - A doctor's id uniquely identifies each doctor.
  - Visiting timings of the doctors have not been taken into consideration.
- A department's record consists of the following: department id, department name, phone and address.
  - A department's id uniquely identifies each department.
  - Visiting timings of the department/ hospital have not been taken into consideration.

    - It is assumed that a doctor’s visiting charges are the same for every visit.
- It is assumed that no photos are to be stored in the database.
- It is assumed that each address is a simple string value instead of being a composite attribute.

RELATIONAL MODEL OF THE HOSPITAL SYSTEM:

The normalized relational model of the system consists of 4 tables:

1. PATIENT

| | | |
|---|---|---|
| Patient_ID | Patient_Name | ID_Proof_Number |
| Phone | Date_of_Admission | Date_of_Discharge |
| Medical_Condition | Blood_Group | Department_ID |
| Contact_Person_1_Name | Contact_Person_1_Phone | Contact_Person_1_Address |
| Contact_Person_2_Name | Contact_Person_2_Phone | Contact_Person_1_Address |
| Medicine_Fee | Room_Charges | Food_Charges |
| Miscellaneous_Charges | Address | Date_of_Birth |

- Patient_ID is the primary key.
- Department_ID is a foreign key to the primary key (Department_ID) of the Department table.

2. DEPARTMENT

| | | | |
|---|---|---|---|
| Department_ID | Department_Name | Department_Address | Phone |

- Department_ID is the primary key.

3. DOCTOR

| | | |
|---|---|---|
| Doctor_ID | Doctor_Name | ID_Proof_Number |
| Visiting_Charge | Department_ID | Phone |

- Doctor_ID is the primary key.
- Department_ID is a foreign key to the primary key (Department_ID) of the Department table.

4. PATIENT_DOCTOR

| | | |
|---|---|---|
| Patient_ID | Doctor_ID | Date_of_Visit |

- Patient_ID, Doctor_ID and Date_of_Visit together form the primary key.
- Patient_ID is a foreign key to the primary key (Patient_ID) of the Patient table.
- Doctor_ID is a foreign key to the primary key (Doctor_ID) of the Doctor table.

**1. Create a database for the Hospital Management System based on your ER. Create appropriate tables & relationships.**

For easy demonstration, many attributes which should ideally be NOT NULL, have not been constrained.

The size of the datatype of attributes i.e. the number of bytes they occupy, have been adjusted for demonstration.

As already stated, it is assumed that each address is a simple string value instead of being a composite attribute.

**DDL QUERIES**

```
--------------------------CREATING department TABLE-------------------

CREATE TABLE `hospital`.`department` (
  `Department_ID` VARCHAR(15) NOT NULL,
  `Department_Name` VARCHAR(40) NULL,
  `Department_Address` VARCHAR(80) NULL,
  PRIMARY KEY (`Department_ID`));

--------------------------CREATING doctor TABLE----------------------

CREATE TABLE `hospital`.`doctor` (
  `Doctor_ID` VARCHAR(15) NOT NULL,
  `Doctor_Name` VARCHAR(40) NOT NULL,
  `Doctor_Phone` VARCHAR(20) NULL,
  `Department_ID` VARCHAR(15),
  `Visiting_Charges`FLOAT NULL,
  `ID_Proof_Number` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`Doctor_ID`),
  CONSTRAINT `Doctor_To_Department`
    FOREIGN KEY (`Department_ID`)
    REFERENCES `hospital`.`department` (`Department_ID`)
    ON DELETE SET NULL
    ON UPDATE CASCADE);

--------------------------CREATING patient TABLE---------------------

CREATE TABLE `hospital`.`patient` (
  `Patient_ID` VARCHAR(15) NOT NULL,
  `Patient_Name` VARCHAR(40) NOT NULL,
  `ID_Proof_Number` VARCHAR(20) NULL,
  `Phone` VARCHAR(20) NULL,
  `Contact_Person_1_Name` VARCHAR(40) NULL,
  `Contact_Person_1_Phone` VARCHAR(20) NULL,
  `Contact_Person_1_Address` VARCHAR(80)  NULL,
  `Address` VARCHAR(80) NULL,
  `Contact_Person_2_Name` VARCHAR(40) NULL,
  `Contact_Person_2_Phone` VARCHAR(20) NULL,
  `Department_ID` VARCHAR(15) NOT NULL,
  `Date_of_Admission` DATE NOT NULL,
  `Date_of_Discharge` DATE NULL,
  `Blood_Group` CHAR(3) NULL,
  `Contact_Person_2_Address` VARCHAR(80)NULL,
  `Miscellaneous_Charges` FLOAT NULL,
  `Food_Charges` FLOAT NULL,
  `Room_Charges` FLOAT NULL,
  `Medical Condition` VARCHAR(80) NULL,
  `Medicine_Fee` FLOAT NULL,
  PRIMARY KEY (`Patient_ID`),
  CONSTRAINT `Patient_to_Department`
    FOREIGN KEY (`Department_ID`)
    REFERENCES `hospital`.`department` (`Department_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
```

**---------------------CREATING patient_doctor TABLE------------------**

```
CREATE TABLE `hospital`.`patient_doctor` (
  `Patient_ID` VARCHAR(15) NOT NULL,
  `Doctor_ID` VARCHAR(15) NOT NULL,
  `Date_of_Visit` DATE NOT NULL,
  PRIMARY KEY (`Patient_ID`, `Doctor_ID`, `Date_of_Visit`),
  CONSTRAINT `Patient_Doctor_to_Patient`
    FOREIGN KEY (`Patient_ID`)
    REFERENCES `hospital`.`patient` (`Patient_ID`)
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT `Patient_Doctor_to_Doctor`
    FOREIGN KEY (`Doctor_ID`)
    REFERENCES `hospital`.`doctor` (`Doctor_ID`)
    ON DELETE NO ACTION
    ON UPDATE CASCADE);
```

# DML QUERIES

## -----------------------INSERTING DEPARTMENTS’ DATA------------------

```
INSERT INTO `hospital`.`department` (`Department_ID`, `Department_Name`,
`Department_Address`) VALUES ('D1', 'Cardiology', 'First Floor');

INSERT INTO `hospital`.`department` (`Department_ID`, `Department_Name`,
`Department_Address`) VALUES ('D2', 'ICU', 'Second Floor');

INSERT INTO `hospital`.`department` (`Department_ID`, `Department_Name`,
`Department_Address`) VALUES ('D3', 'Emergency', 'Third Floor');

INSERT INTO `hospital`.`department` (`Department_ID`, `Department_Name`,
`Department_Address`) VALUES ('D4', 'Gynaecology', 'Ground Floor');

INSERT INTO `hospital`.`department` (`Department_ID`, `Department_Name`,
`Department_Address`) VALUES ('D5', 'Neurology', 'East Wing');

INSERT INTO `hospital`.`doctor` (`Doctor_ID`, `Doctor_Name`,
`Department_ID`) VALUES ('Doc7', 'Ayush', 'D2');
```

## -----------------------INSERTING DOCTORS’ DATA---------------------

```
INSERT INTO `hospital`.`doctor` (`Doctor_ID`, `Doctor_Name`,
`Department_ID`) VALUES ('Doc1', 'Ramesh', 'D1');

INSERT INTO `hospital`.`doctor` (`Doctor_ID`, `Doctor_Name`,
`Department_ID`) VALUES ('Doc2', 'Suresh', 'D3');

INSERT INTO `hospital`.`doctor` (`Doctor_ID`, `Doctor_Name`,
`Department_ID`) VALUES ('Doc3', 'Rajesh', 'D1');

INSERT INTO `hospital`.`doctor` (`Doctor_ID`, `Doctor_Name`,
`Department_ID`) VALUES ('Doc4', 'Jignesh', 'D5');

INSERT INTO `hospital`.`doctor` (`Doctor_ID`, `Doctor_Name`,
`Department_ID`) VALUES ('Doc5', 'Aarti', 'D3');

INSERT INTO `hospital`.`doctor` (`Doctor_ID`, `Doctor_Name`,
`Department_ID`) VALUES ('Doc6', 'Aswani', 'D4');
```

**-----------------------INSERTING PATIENTS’ DATA---------------------**

```
INSERT INTO `hospital`.`patient` (`Patient_ID`, `Patient_Name`,
`Department_ID`, `Date_of_Admission`, `Date_of_Discharge`) VALUES ('P1',
'Arvind', 'D1', '2020-07-13', '2020-07-19');
INSERT INTO `hospital`.`patient` (`Patient_ID`, `Patient_Name`,
`Department_ID`, `Date_of_Admission`) VALUES ('P2', 'Mukesh', 'D2', '2020-
06-11');

INSERT INTO `hospital`.`patient` (`Patient_ID`, `Patient_Name`,
`Department_ID`, `Date_of_Admission`, `Date_of_Discharge`) VALUES ('P3',
'Vikas', 'D5', '2020-07-16', '2020-07-18');

INSERT INTO `hospital`.`patient` (`Patient_ID`, `Patient_Name`,
`Department_ID`, `Date_of_Admission`, `Date_of_Discharge`) VALUES ('P4',
'Swati', 'D1', '2020-06-11', '2020-07-15');

INSERT INTO `hospital`.`patient` (`Patient_ID`, `Patient_Name`,
`Department_ID`, `Date_of_Admission`, `Date_of_Discharge`) VALUES ('P5',
'Vineeta', 'D5', '2020-06-10', '2020-07-05');

INSERT INTO `hospital`.`patient` (`Patient_ID`, `Patient_Name`,
`Department_ID`, `Date_of_Admission`, `Date_of_Discharge`) VALUES ('P6',
'Aryan', 'D3', '2020-05-15', '2020-07-18');

INSERT INTO `hospital`.`patient` (`Patient_ID`, `Patient_Name`,
`Department_ID`, `Date_of_Admission`) VALUES ('P7', 'Vipin', 'D4', '2020-
07-08');

INSERT INTO `hospital`.`patient` (`Patient_ID`, `Patient_Name`,
`Department_ID`, `Date_of_Admission`, `Date_of_Discharge`) VALUES ('P8',
'Shweta', 'D1', '2020-07-09', '2020-07-19');
```

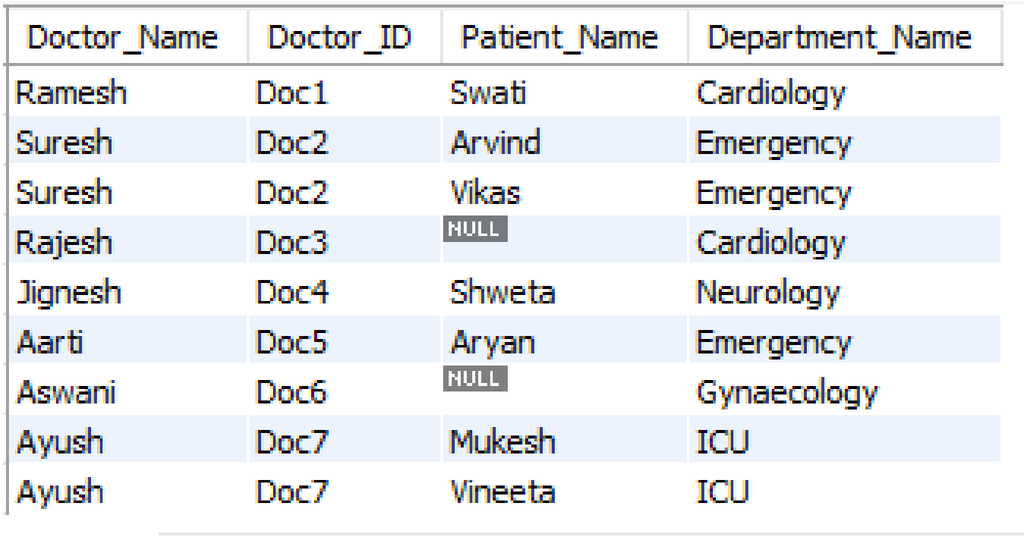
**--------------------INSERTING PATIENT-DOCTOR DATA-------------------**

```
INSERT INTO `hospital`.`patient_doctor` (`Patient_ID`, `Doctor_ID`,
`Date_of_Visit`) VALUES ('P1', 'Doc2', '2020-07-14');

INSERT INTO `hospital`.`patient_doctor` (`Patient_ID`, `Doctor_ID`,
`Date_of_Visit`) VALUES ('P2', 'Doc7', '2020-07-13');

INSERT INTO `hospital`.`patient_doctor` (`Patient_ID`, `Doctor_ID`,
`Date_of_Visit`) VALUES ('P3', 'Doc2', '2020-07-16');
```

```
INSERT INTO `hospital`.`patient_doctor` (`Patient_ID`, `Doctor_ID`,
`Date_of_Visit`) VALUES ('P4', 'Doc1', '2020-07-14');

INSERT INTO `hospital`.`patient_doctor` (`Patient_ID`, `Doctor_ID`,
`Date_of_Visit`) VALUES ('P5', 'Doc7', '2020-07-05');

INSERT INTO `hospital`.`patient_doctor` (`Patient_ID`, `Doctor_ID`,
`Date_of_Visit`) VALUES ('P6', 'Doc5', '2020-07-10');

INSERT INTO `hospital`.`patient_doctor` (`Patient_ID`, `Doctor_ID`,
`Date_of_Visit`) VALUES ('P8', 'Doc4', '2020-07-10');
```

**2. Design a query to provide a list of doctors, which department they belong to and patients treated by them (if any).**

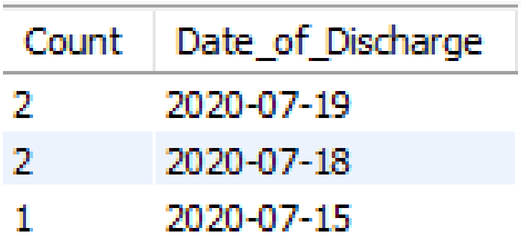
```
SELECT Doctor_Name, Doctor_ID, NULL AS Patient_Name, Department_Name
FROM hospital.doctor DOC, hospital.department DPT
WHERE DOC.Department_ID = DPT.Department_ID and Doctor_ID NOT IN
(SELECT Doctor_ID FROM hospital.patient_doctor)
UNION
SELECT Doctor_Name, PD.Doctor_ID, Patient_Name, Department_Name
FROM hospital.Department AS DPT, hospital.doctor AS
D,hospital.patient_doctor as PD ,hospital.patient as P
WHERE PD.patient_id = P.Patient_ID AND D.doctor_id = PD.Doctor_ID
AND D.Department_ID=DPT.Department_ID
ORDER BY Doctor_ID;
```



| Doctor_Name | Doctor_ID | Patient_Name | Department_Name |
|---|---|---|---|
| Ramesh | Doc1 | Swati | Cardiology |
| Suresh | Doc2 | Arvind | Emergency |
| Suresh | Doc2 | Vikas | Emergency |
| Rajesh | Doc3 | NULL | Cardiology |
| Jignesh | Doc4 | Shweta | Neurology |
| Aarti | Doc5 | Aryan | Emergency |
| Aswani | Doc6 | NULL | Gynaecology |
| Ayush | Doc7 | Mukesh | ICU |
| Ayush | Doc7 | Vineeta | ICU |

### 3. Query to provide the count of patients discharged per day in the last week.

```
SELECT Count(*) as Count, Date_of_Discharge
FROM hospital.patient
WHERE Date_of_Discharge BETWEEN '2020-07-13' and '2020-07-20'
GROUP BY Date_of_Discharge;
```



| Count | Date_of_Discharge |
|---|---|
| 2 | 2020-07-19 |
| 2 | 2020-07-18 |
| 1 | 2020-07-15 |

For missing Dates of the week Count=0;