

Semester - IV

Course Name - CSE250 Database Management System

Project Title - Hospital Database Management System

Group Members		
Roll Number	Name	Programme
AU1920082	Yuvraj Patel	BS (Hons) Computer Science
AU1940131	Vishwa Raval	BTech Computer Science & Engineering
AU1940150	Darsh Patel	BTech Computer Science & Engineering

## a. Description of Project :

The purpose of this project is to develop a system that handles different workflows related to a hospital. The main task is to develop a software that registers, stores, and meaningfully manipulates the information related to the patients, staff and doctors, appointments, visits, medicines, etc. in order to have some meaningful conclusions.

### Project Module flow

#### 1. Starts With Login Page

## Login

Please fill in your credentials to login.

Username

Password

Don't have an account? [Sign up now.](#)

The default admin username is "admin" and the password is "admin123"

- If user not **registered**,

## Sign Up

Please fill this form to create an account.

Username

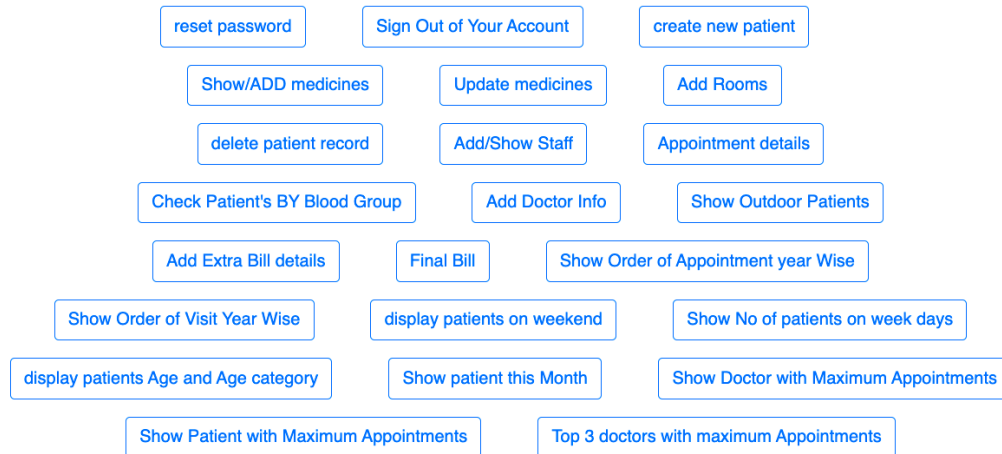
Password

Confirm Password

Already have an account? [Login here.](#)

## 2. Hospital Management Homepage contains list of all the Functionalities:

Hi, **admin**. Welcome to Hospital management site.



## 3. Creating new data or inserting values into database

- For example **Create New patient** leads to another page and gives **confirmation** if the record is inserted successfully or prompt error if otherwise.

Patient ID:

First Name:

Middle Name:

Last Name:

Gender (M or F) :

Date Of Birth :

Contact Number :

Email Address:

Address (city) :

Medical history:

Choose BloodGroup:

[Cancel](#)

**Patient Table**

Pat_ID	Pat_fname	Pat_mname	Pat_lname	Pat_gender	Pat_DOB	Pat_contactno	Pat_emailID	Pat_Address	Pat_MedHistory	Pat_BloodGroup
1	Dev		Mehta	M	2001-01-31	9239420134	dev@gmail.com	Ahmedabad		A+ve
2	Rahi		Shah	F	2001-11-02	9239445324	rahi@gmail.com	Ahmedabad		B+ve
3	Janvi		Joshi	F	2000-02-02	9239412342	janvi@gmail.com	Ahmedabad		O+ve
4	darsh	d	patel	M	0021-04-22	9879820606	d5@gmail.com	Mehsana	None	O+ve
5	Raina		Rai	F	1979-12-07	9898563427	raina@gmail.com	Mumbai		A-ve

- **Confirmation message :**

New record created successfully !

[Go back for inserting More patients](#)

id	action	created_date
31	patient inserted	2021-04-16 00:10:59
32	patient inserted	2021-04-16 03:07:57
33	patient inserted	2021-04-16 03:34:04
34	patient inserted	2021-04-16 04:44:57

- Users have the freedom to move to the **last visited page** with buttons like “**Go back For...**”.

#### 4. Updating and Deleting the records

- For Example user can update the corresponding details of medicines with “**Update**” Button.

Med_ID	Med_Name	Med_company	Med_price	Med_dose	Med_type	
Med101	DAN-P	UNISON	15	600mg	Anti inflammatory	<a href="#">Update</a>
Med102	DOLO	Torrent	20	650mg	Paracetamol	<a href="#">Update</a>
Med103			150	625mg		<a href="#">Update</a>

< [Cancel](#)

Medicine\_updates Table (Trigger 2)

change_date	field_name	before_value	after_value
2021-04-14 19:53:49	Med_Name	hg	qq
2021-04-14 19:53:49	Med_Name	ghj	qq
2021-04-14 19:53:49	Med_price	77	34
2021-04-14 19:53:49	Med_dose	76	22
2021-04-14 19:53:49	Med_type	ihq	qq

- Using a **HTML/CSS** based form, users are able to make the changes and get **notified** by “**Updated**” prompt for confirmation.

Med_ID	Med_Name	Med_company	Med_price	Med_dose	Med_type
Med101	DAN-P	UNISON	15	600mg	Anti inflammatory
Med102	DOLO	Torrent	20	650mg	Paracetamol
Med103			150	625mg	

Updated

[Medicine List](#)

Medicine ID:

Medicine name:

Medicine Company :

Medicine Price:

Medicine Dose:

Medicine type:

- The change is also **reflected** in the tables after pressing **back**, i.e. refer the change in ‘**Med103**’ ID medicine.

Med_ID	Med_Name	Med_company	Med_price	Med_dose	Med_type	
Med101	DAN-P	UNISON	15	600mg	Anti inflammatory	<a href="#">Update</a>
Med102	DOLO	Torrent	20	650mg	Paracetamol	<a href="#">Update</a>
Med103	calpol	Zydus	150	625	Anti Biotic	<a href="#">Update</a>

< [Cancel](#)

Medicine\_updates Table (Trigger 2)

change_date	field_name	before_value	after_value
2021-04-14 19:53:49	Med_Name	hg	qq
2021-04-14 19:53:49	Med_Name	ghj	qq
2021-04-14 19:53:49	Med_price	77	34

## 5. Viewing and Searching through the database

- Let's consider an example of finding **Patients** by their **BloodGroup**

Enter Blood Group from below :

[Cancel](#)

### Patient Blood Group Table

Pat_BloodGroup
A+ve
B+ve
O+ve
A-ve

- And getting the list of number of patients with Blood group “A+ve”

### Pat\_BloodGroup

1

[Go back and explore more bloodgroups](#)

6. Buttons other than **Insert, Update, Delete, Viewing** and **Searching** data we have implemented some **Stored Procedures , Functions** and **Triggers** for **User** to keep track of Some Important details like one of them is **Doctor with maximum Appointments**

#### **Doc With maximum appointments**

Doc_ID	Doc_fname	Doc_contactno	Doc_DOB	Doc_gender
Doc01	Mauli	1960-03-31	1960-03-31	F

[Cancel](#)

#### **Functionalities:**

##### 1) Managing **Patients' Information** :

- For **deriving the age of a patient, given the DOB**, we have created a function followed by a procedure. The function (**age(patientid int)**) with parameter patient id returns the current age of that patient. Then, if the current age is greater than 18 then the procedure (**patient\_age\_msg(patientid int)**) will **print a message** 'Patient is a Major', else it will print the message 'Minor Patient'.
- To **get an overview related to the blood groups of all the patients present in the database**, we have created a function (**patient\_bloodgroup(bloodgroup varchar(10))**) with parameter blood group. It returns the total number of patients of that particular blood group.
- To get the **details of the patient who has the maximum number of appointments** in this hospital, we have created a procedure **patient\_with\_maxApps()**.
- To **display the patients who took appointment on the weekends**, we have created a procedure called **display\_patients\_on\_weekend()**
- To **get the information about the patients who took an appointment during current month**, we have created a procedure **pat\_this\_month()**.
- In our database, we have a table for all the patients in the hospital, and then a separate table for indoor patients. However, we do not have **a table that shows the details of outdoor patients**. For which we have created a procedure called procedure **display\_out\_patient()**.
- Atlast, we have generated a **bill** for a visit of a patient, which includes a patient's visit charge, doctor charge, medicine charge etc. And generates the total amount to be paid.

##### 2) Managing **Doctors' Information** :

- To **get the information about the most famous doctor/s among the patients**, we have created a procedure, **maximum\_Doc\_appointments()** that displays information of the doctor who has the highest number of appointments. Further, we have also created a procedure, **Top3\_Doc\_appointments()** that returns the top three doctors who have the highest number of appointments.

3) Managing **Appointment Related Information**:

- To **get the trend of appointments for each weekday**, we have created a function with parameter i, **total\_patients\_on\_this\_day(dayno int)** that returns the number of patients who have appointments on day i. (1=Sunday, 2=Monday, 3=Tuesday, 4=Wednesday, 5=Thursday, 6=Friday, 7=Saturday)
- The above function is followed by a procedure (**patient\_count\_msg(dayno int)**) that **prints a message** 'Many patients on this day!!' if the number of patients is greater than 2 on that day, otherwise it prints the message 'Not many patients on this day!' (For bigger database we need to take a bigger number than 2)
- To **display patient-wise appointment details**, we have created a procedure called **display\_patient\_appointment\_info()**.
- To **display year-wise appointment details**, we have created a procedure called **display\_yearwise\_appointments()**. Also, all the patients who have taken the appointment might not end up visiting the hospital. So, to **display year-wise visits records**, we have created an additional procedure called **display\_yearwise\_visits(which\_year year)**.

4) Managing **Department wise HODs**:

- For **identifying an HOD of a department**, we have created a function **identify\_hod(doctorid varchar(6),departmentid varchar(6))**. If the doctor (doc\_id) is the HOD of the department (dept\_id), this function returns true, otherwise false.

5) Managing **Staff, Rooms, and Medicines related to Information**:

- If Staff\_dutystarttime is equal to Staff\_dutyendtime then a trigger will print **an error message, on inserting the data in the staff table**: "Duty start time cannot be same as Duty end time..." and won't allow the user to insert that data.
- We have created a trigger that **checks 'Room\_Charge' when a user inserts or updates Room\_Charge in Rooms table**. If Room\_Charge>10000, it won't allow you to insert/update the record and will display an appropriate error message 'Room charge cannot exceed 10000! Couldn't perform the operation...'
- We have created a trigger which triggers when any record is updated in the Medicines table. When the **value of any field is updated, we keep track of before and after values in the table "medicines\_updates"** for each field of the Medicines table. The "medicines\_updates" table contains the fields change\_date, field\_name, before\_value and after\_value.

6) Managing **Admin/Login Details**:

- We have created a **simple login and user registration** system using authentication from Admin User Table.
- There are some functionalities like creating a **new administrator** (if the person is new to the system), **logout** and **password change** (altering the password column of the user table).
- As a matter of Privacy Policy we don't save 'password' directly but instead store the HashCode corresponding to it using **Password\_Hash()**.

7) Managing other information:

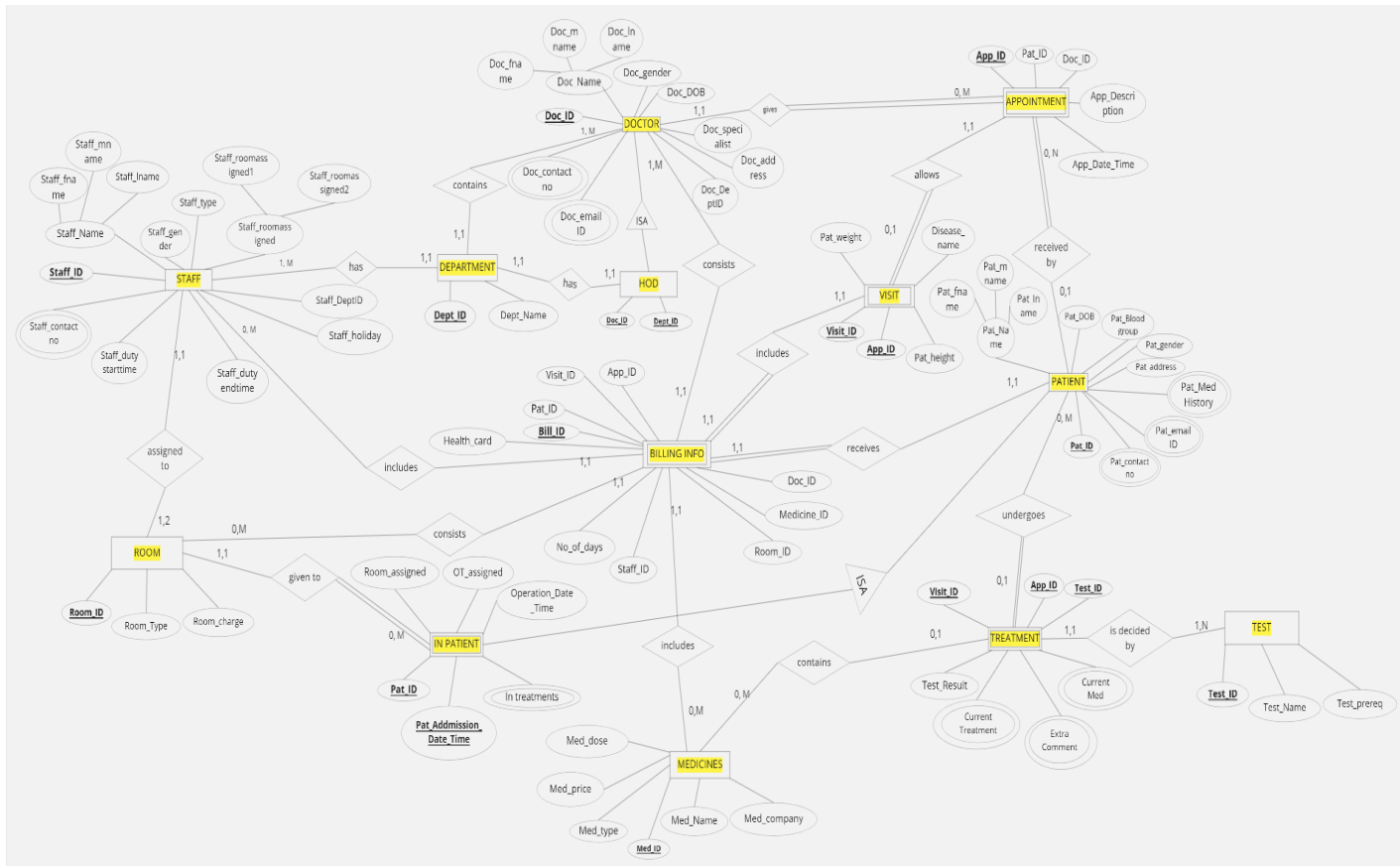
- If **Contact no. is not a 10 digit number** then a trigger will display **an error message**: "Invalid contact no" This is applied on In Patient\_personal\_info, Doctor\_info, and Staff tables.
- Whenever it is needed to permanently delete a record from patient\_personal\_info table, a trigger will automatically **delete all child records** from appointment, billing info, and in-patients tables. For this it is required to apply the same trigger on Appointment, and Visit table.
- For **managing patient log details**, we have created a trigger that keeps the track of new entries entered into patient\_personal\_info in a table called "logs table"

### **System Requirement Specification:**

We have used Netbean IDE 8.2 software to configure PHP/HTML/CSS scripts on our localhost web server. The operating system was Windows 10 / MacOS Catalina. The web browser is Chrome with PhpMyAdmin as Front-end and Back-end Integration. We deployed our web server on localhost using Apache(built-in server) using MAMP/WAMP (MacOS/Windows Apache MySQL PHP). At Least 8MB of disk space is required. We can implement it in modern browsers like IE7+, Firefox, Safari, Google Chrome and Opera that support CSS and HTML framework.



## b. Entity-Relationship Diagram:

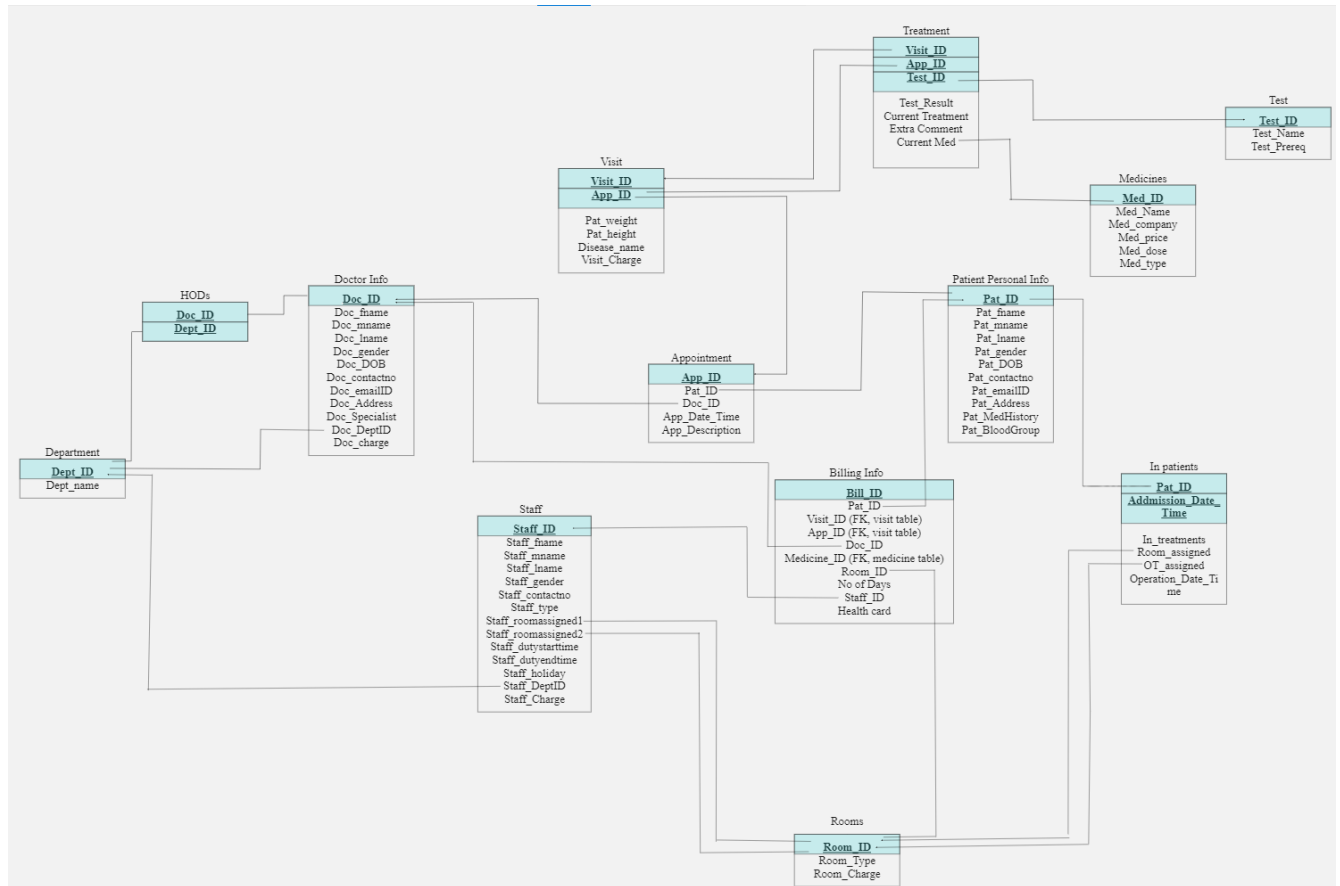


Link for the miro board: [https://miro.com/app/board/o9J\\_ILSPzV0=](https://miro.com/app/board/o9J_ILSPzV0=/)

(This link consists all the exercises we performed while preparing the conceptual model and schema including Normalisation Exercise, Defining Relationships and its Cardinality)

## c. Table Design (Data Dictionary):

### Schema:



### Syntax of create table:

```

CREATE TABLE Patient_personal_info(Pat_ID INT PRIMARY KEY,
    Pat_fname varchar(20) NOT NULL,
    Pat_mname varchar(20),
    Pat_lname varchar(20) NOT NULL,
    Pat_gender varchar(2) NOT NULL,
    Pat_DOB DATE NOT NULL,
    Pat_contactno BIGINT NOT NULL,
    Pat_emailID varchar(50) unique check(pat_emailID=lower(pat_emailID)),
    Pat_Address varchar(250),
    Pat_MedHistory varchar(500),
    Pat_BloodGroup varchar(10) NOT NULL );
    
```

```

insert into Patient_personal_info
values(1,'Dev',null,'Mehta','M','2001-01-31',9239420134,'dev@gmail.com','Ahmedabad',null,'A+ve');
    
```

```

insert into Patient_personal_info
values(2,'Rahi',null,'Shah','F','2001-11-02',9239445324,'rahi@gmail.com','Ahmedabad',null,'B+ve');
insert into Patient_personal_info
values(3,'Janvi',null,'Joshi','F','2000-02-02',9239412342,'janvi@gmail.com','Ahmedabad',null,'O+ve');
insert into Patient_personal_info
values(4,'Raj',null,'Majmudar','M','1989-06-22',9124563427,'raj@gmail.com','Surat',null,'O+ve');
insert into Patient_personal_info
values(5,'Raina',null,'Rai','F','1979-12-07',9898563427,'raina@gmail.com','Mumbai',null,'A-ve');

```

---

```

CREATE TABLE Department(Dept_ID varchar(6) PRIMARY KEY,
                        Dept_name varchar(50) NOT NULL
                        );

```

```

insert into Department values('Dept1','Cardiology');
insert into Department values('Dept2','Gastro');
insert into Department values('Dept3','GeneralSurgery');
insert into Department values('Dept4','Gynecology');
insert into Department values('Dept5','Neurology');
insert into Department values('Dept6','Physiotherapy');
insert into Department values('Dept7','Urology');

```

---

```

CREATE TABLE Doctor_info(Doc_ID varchar(6) PRIMARY KEY,
                          Doc_fname VARCHAR(20) NOT NULL,
                          Doc_mname VARCHAR(20),
                          Doc_lname VARCHAR(20) NOT NULL,
                          Doc_gender VARCHAR(2) NOT NULL,
                          Doc_DOB DATE NOT NULL,
                          Doc_contactno BIGINT NOT NULL,
                          Doc_emailID VARCHAR(50) unique
check(doc_emailID=lower(doc_emailID)),
                          Doc_Address VARCHAR(250),
                          Doc_Specialist VARCHAR(50),
                          Doc_DeptID varchar(6),
                          FOREIGN KEY (Doc_DeptID) references
Department(Dept_ID),
                          Doc_charge INT
                          );

```

```

insert into Doctor_info
values('Doc01','Mauli',null,'Patel','F','1960-03-31',9867420134,'mauli@gmail.com','Ahmedabad','Gynecol
ogist','Dept4',2000);
insert into Doctor_info
values('Doc02','Jay',null,'Patel','M','1950-12-21',6347420134,'jay@gmail.com','Ahmedabad','Cardiologist',
'Dept1',5000);

```

```

insert into Doctor_info
values('Doc03','Pranav',null,'Dava','M','1970-07-11',8947422334,'pranav@gmail.com','Vadodara','Surgeon
','Dept3',5000);
-- insert into Doctor_info
values('Doc09','Jay',null,'Patel','M','1950-12-21',6347420134,'jay1@gmail.com','Ahmedabad','Cardiologis
t','D1',5000);
select * from doctor_info;
select * from Doctor_info;

```

---

```

CREATE TABLE HODs (Doc_ID varchar(6),
                    Dept_ID varchar(6),
                    FOREIGN KEY (Doc_ID) REFERENCES Doctor_info(Doc_ID),
                    FOREIGN KEY (Dept_ID) REFERENCES Department(Dept_ID),
                    PRIMARY KEY (Doc_ID, Dept_ID)
);

```

```

INSERT INTO HODs VALUES ('Doc02','Dept1');
INSERT INTO HODs VALUES ('Doc01','Dept4');
INSERT INTO HODs VALUES ('Doc03','Dept3');

```

---

```

CREATE TABLE Appointment(App_ID varchar(10) PRIMARY KEY,
                          Pat_ID INT,
                          Doc_ID varchar(6),
                          App_Date_Time timestamp NOT NULL ,
                          App_Description VARCHAR(200),
                          FOREIGN KEY (Pat_ID) REFERENCES Patient_personal_info(Pat_ID),
                          FOREIGN KEY (Doc_ID) REFERENCES Doctor_info(Doc_ID)
);

```

```

insert into appointment values('App101',1,'Doc02',current_timestamp(),'NewCase');
insert into appointment values('App102',2,'Doc01',current_timestamp(),'NewCase');
insert into appointment values('App103',1,'Doc02',current_timestamp(),'Followup');
insert into appointment values ('App104',3,'Doc01', '2019-12-19 03:14:07','NewCase');
insert into appointment values ('App105',3,'Doc01', '2020-01-01 03:20:10','Dressing');
insert into appointment values ('App106',4,'Doc02', '2020-02-09 05:50:10','NewCase');
insert into appointment values ('App107',4,'Doc02', '2020-02-09 08:20:50',null);

```

---

```

CREATE TABLE Visit(Visit_ID varchar(10),
                   App_ID varchar(10),
                   Pat_weight INT NOT NULL,
                   Pat_height varchar(10) NOT NULL,
                   Disease_name VARCHAR(50),
                   Visit_charge INT,
                   FOREIGN KEY (App_ID) REFERENCES Appointment(App_ID),
                   PRIMARY KEY(Visit_ID, App_ID)
);

```

```

insert into visit values ('Vis101','App102',59,'5"6"', 'Cancer',500);
insert into visit values ('Vis102','App101',69,'5"9"', 'Dengue',800);
insert into visit values ('Vis103','App104',78,'6"0"', 'Malaria',700);
insert into visit values ('Vis104','App105',60,'5"4"', 'Broken Arm',800);
insert into visit values ('Vis105','App106',89,'5"9"', 'Cancer',800);

```

---

```

CREATE TABLE MEDICINES (Med_ID VARCHAR(10) PRIMARY KEY,
                        Med_Name VARCHAR(50) NOT NULL,
                        Med_company VARCHAR(30),
                        Med_price INT NOT NULL,
                        Med_dose varchar(10) NOT NULL,
                        Med_type VARCHAR(30)
                        );

```

```

insert into medicines values ('Med101','DAN-P','UNISON',15,'600mg','Anti inflammatory');
insert into medicines values ('Med102','DOLO','Torrent',20,'650mg','Paracetamol');
insert into medicines values ('Med103','Clavam','Zydus',120,'625mg','Anti biotic');
select * from medicines;

```

---

```

CREATE TABLE Test (Test_ID VARCHAR(5) PRIMARY KEY,
                    Test_Name VARCHAR(40) NOT NULL,
                    Test_Prereq VARCHAR(50)
                    );

```

```

insert into test values ('Test1','PPBS','Post meal');
insert into test values ('Test2','Lipid Profile','Fasting');
insert into test values ('Test3','Creatinine',null);
insert into test values ('Test4','TSH',null);
insert into test values ('Test5','Urine Culture',null);
insert into test values ('Test6','Blood Culture',null);

```

```

select * from test;

```

---

```

CREATE TABLE Treatment(Visit_ID varchar(10),
                        App_ID varchar(10),
                        Test_ID VARCHAR(5),
                                Test_Result boolean not null,
                        Current_Treatment varchar(200) default null,
                        Extra_Comment varchar(500) default null,
                        Current_Med varchar(150) default null,
                        PRIMARY KEY (Visit_ID, App_ID, Test_ID),
                        FOREIGN KEY (Visit_ID, App_ID) REFERENCES Visit(Visit_ID, App_ID),
                        FOREIGN KEY (Test_ID) REFERENCES Test(Test_ID),
                        FOREIGN KEY (Current_Med) REFERENCES MEDICINES(Med_ID)
                        );

```

```

insert into treatment values ('Vis101','App102', 'Test1',1,'IV fluid',null,'Med101');

```

```
select * from treatment;
```

---

```
CREATE TABLE Rooms(Room_ID VARCHAR(5) PRIMARY KEY,  
    Room_Type varchar(40),  
    Room_Charge INT NOT NULL  
);
```

```
INSERT INTO Rooms VALUES ('room1', 'Twin-sharing', 2500);  
INSERT INTO Rooms VALUES ('room2', 'Deluxe', 5000);  
INSERT INTO Rooms VALUES ('room3', 'OT2', 3500);  
INSERT INTO Rooms VALUES ('room4', 'General Ward', 1000);  
INSERT INTO Rooms VALUES ('room5', 'OT1', 3500);  
INSERT INTO Rooms VALUES ('room6', 'Consultation-1', 0);  
INSERT INTO Rooms VALUES ('room7', 'Consultation-2', 0);
```

---

```
CREATE TABLE Staff (Staff_ID VARCHAR(10) PRIMARY KEY,  
    Staff_fname VARCHAR(40) NOT NULL,  
    Staff_mname VARCHAR(40),  
    Staff_lname VARCHAR(40),  
    Staff_gender VARCHAR(2) NOT NULL,  
    Staff_contactno bigint NOT NULL,  
    Staff_type VARCHAR(30) ,  
    Staff_roomassigned1 VARCHAR(5) NOT NULL,  
    Staff_roomassigned2 VARCHAR(5),  
    Staff_dutystarttime time NOT NULL,  
    Staff_dutyendtime time NOT NULL,  
    Staff_holiday varchar(10),  
    Staff_DeptID varchar(6),  
    Staff_Charge int not null,  
    FOREIGN KEY (Staff_DeptID) REFERENCES Department(Dept_ID),  
    FOREIGN KEY (Staff_roomassigned1) REFERENCES Rooms(Room_ID),  
    FOREIGN KEY (Staff_roomassigned2) REFERENCES Rooms(Room_ID)  
);
```

```
insert into staff values ('S1','Deep',null,  
null,'M',9341568535,'wardboy','room4',null,'01:00:00','08:00:00','Wednesday','Dept1',500);  
insert into staff values ('S2','Prakash',null,  
null,'M',9823416792,'assistant','room5',null,'08:00:00','01:00:00','Wednesday','Dept3',1000);  
insert into staff values ('S3','Mamta',null,  
null,'F',9135968535,'nurse','room1','room3','09:00:00','05:00:00','Sunday','Dept4',500);
```

---

```
CREATE TABLE In_Patients (Pat_ID INT,  
    Admission_Date_Time timestamp not null,  
    In_treatments Varchar(100),  
    Room_assigned VARCHAR(5) Not Null,  
    OT_assigned VARCHAR(5),
```

```

    Operation_Date_Time timestamp,
    Primary Key(Pat_ID,Addmission_Date_Time),
    Foreign Key (Room_Assigned) REFERENCES Rooms(Room_ID),
    Foreign Key (OT_assigned) REFERENCES Rooms(Room_ID),
    Foreign Key (Pat_ID) references patient_personal_info(Pat_ID)
);

```

```

insert into in_patients values(1, '2021-04-07 09:35:48', 'hospitalisation', 'room1', 'room3', '2021-04-17
04:06:02');

```

```

insert into in_patients values(2, '2021-04-14 10:17:02', 'Hospitalisation', 'room2', 'room5', '2021-04-17
04:07:31');

```

---

```

CREATE TABLE Billing_info(Bill_ID varchar(10) primary key,
    Pat_ID INT NOT NULL,
    Visit_ID varchar(10) NOT NULL,
    App_ID varchar(10) NOT NULL,
    Doc_ID varchar(6) NOT NULL,
    Medicine_ID VARCHAR(10),
    Room_ID VARCHAR(5),
    No_of_Days INT NOT NULL,
    Staff_ID VARCHAR(10) NOT NULL,
    Health_Card boolean,
    FOREIGN KEY (Visit_ID) REFERENCES Visit(Visit_ID),
    FOREIGN KEY (App_ID) REFERENCES Appointment(App_ID),
    FOREIGN KEY (Room_ID) REFERENCES Rooms(Room_ID),
    FOREIGN KEY (Pat_ID) REFERENCES Patient_personal_info(Pat_ID),
    FOREIGN KEY (Doc_ID) REFERENCES Doctor_info(Doc_ID),
    FOREIGN KEY (Medicine_ID) REFERENCES Medicines(Med_ID)
);

```

```

insert into billing_info values ('Bill1',1,'Vis102','App101','Doc03','Med101','room1',3,'S3',1);

```

```

insert into billing_info values ('Bill2',3,'Vis103','App104','Doc01','Med103',null,0,'S1',0);

```

```

insert into billing_info values ('Bill3',2,'Vis101','App102','Doc02','Med102','room2',0,'S3',0);

```

---

## d. Stored Procedure, Functions and Triggers

### Triggers:

- 1) **Triggers when record is updated in the Medicines table. When the value of any field is updated, we keep track of before and after values in the table “medicines\_updates” for each field of the Medicines table. The “medicines\_updates” table contains the fields change\_date, field\_name, before\_value and after\_value:**

```
create table medicines_updates(change_date timestamp, field_name varchar(50), before_value
varchar(30), after_value varchar(30));
```

```
drop trigger if exists trigger1;
```

```
delimiter $$
```

```
create trigger trigger1 after update on MEDICINES
```

```
for each row
```

```
begin
```

```
    declare curr_date_time timestamp;
```

```
    select current_timestamp into curr_date_time;
```

```
    if (old.Med_ID != new.Med_ID) then
```

```
        insert into medicines_updates values (curr_date_time,
'Med_ID',old.Med_ID,new.Med_ID);
```

```
    end if;
```

```
    if (old.Med_Name !=new.Med_Name) then
```

```
        insert into medicines_updates values (curr_date_time,
'Med_Name',old.Med_Name,new.Med_Name);
```

```
    end if;
```

```
    if (old.Med_company !=new.Med_company) then
```

```
        insert into medicines_updates values (curr_date_time,
'Med_Name',old.Med_company,new.Med_company);
```

```
    end if;
```

```
    if (old.Med_price !=new.Med_price) then
```

```
        insert into medicines_updates values (curr_date_time,
'Med_price',old.Med_price,new.Med_price);
```

```
    end if;
```



```

    if (old.Med_dose !=new.Med_dose) then
        insert into medicines_updates values (curr_date_time,
'Med_dose',old.Med_dose,new.Med_dose);
    end if;

    if (old.Med_type !=new.Med_type) then
        insert into medicines_updates values (curr_date_time,
'Med_type',old.Med_type,new.Med_type);
    end if;
end
$$
delimiter ;

```

- 2) **This trigger checks 'Room\_Charge' when a user inserts or updates Room\_Charge in Rooms table. If Room\_Charge>10000, it won't allow you to insert/update the record and will display an appropriate error message 'Room charge cannot exceed 10000! Couldn't perform the operation...':**

```

drop trigger if exists trigger2_insert;
delimiter $$
create trigger trigger2_insert before insert on Rooms
for each row
begin
declare err_msg varchar(200);

if (new.Room_Charge>10000) then
set err_msg='Error: Room charge cannot exceed 10000! Couldn't perform the operation...';
signal sqlstate '45123' set message_text=err_msg;
end if;
end
$$
delimiter ;

```

```

drop trigger if exists trigger2_update;
delimiter $$
create trigger trigger2_update before update on Rooms
for each row
begin
declare err_msg varchar(200);

```

```

if (new.Room_Charge>10000) then
set err_msg='Error: Room charge cannot exceed 10000! Couldn't perform the operation...';
signal sqlstate '45123' set message_text=err_msg;
end if;
end
$$
delimiter ;

```

**3) If Contact no. is not a 10 digit number then this trigger will display message:"Invalid contact no" (In Patient\_personal\_info, Doctor\_info, and Staff tables):**

```

drop trigger if exists trigger3_doc;
delimiter $$
create trigger trigger3_doc before insert on Doctor_info
for each row
begin
declare err_msg varchar(200);
if (length(new.Doc_contactno)!=10) then
set err_msg='Error: Invalid Contact No...';
signal sqlstate '45123' set message_text=err_msg;
end if;
end
$$
delimiter ;

```

```

drop trigger if exists trigger3_pat;
delimiter $$
create trigger trigger3_pat before insert on patient_personal_info
for each row
begin
declare err_msg varchar(200);
if (length(new.Pat_contactno)!=10) then
set err_msg='Error: Invalid Contact No...';
signal sqlstate '45123' set message_text=err_msg;
end if;
end
$$
delimiter ;

```

```

drop trigger if exists trigger3_staff;

```

```

delimiter $$
create trigger trigger3_staff before insert on staff
for each row
begin
declare err_msg varchar(200);
if (length(new.Staff_contactno)!=10) then
set err_msg='Error: Invalid Contact No...';
signal sqlstate '45123' set message_text=err_msg;
end if;
end
$$
delimiter ;

```

- 4) This trigger will delete all child records from appointment, billing info, and in-patients tables when the patient record is deleted from the Patient\_personal\_info table:**

```

drop trigger if exists trigger4a;
delimiter $$
create trigger trigger4a before delete on Patient_personal_info
for each row
begin
    delete from Appointment where Appointment.Pat_ID=old.Pat_ID;
    delete from Billing_info where Billing_info.Pat_ID=old.Pat_ID;
    delete from In_Patients where In_Patients.Pat_ID=old.Pat_ID;
end
$$
delimiter ;

```

```

drop trigger if exists trigger4b;
delimiter $$
create trigger trigger4b before delete on Appointment
for each row
begin
    delete from Visit where Visit.App_ID=old.App_ID;
end
$$
delimiter ;

```

```

drop trigger if exists trigger4c;
delimiter $$

```

```

create trigger trigger4c before delete on Visit
for each row
begin
    delete from Treatment where Treatment.App_ID=old.App_ID and
Treatment.Visit_ID=old.Visit_ID;
end
$$
delimiter ;

```

- 5) Before insert on Staff table, if Staff\_dutystarttime is equal to Staff\_dutyendtime then this trigger will print an error message, "Duty start time cannot be same as Duty end time...":**

```

drop trigger if exists trigger5;
DELIMITER $$
CREATE TRIGGER trigger5 BEFORE INSERT ON Staff
FOR EACH ROW BEGIN
DECLARE err_msg varchar(200);
if(new.Staff_dutystarttime = new.Staff_dutyendtime) THEN
    set err_msg = 'Error: Duty start time cannot be same as Duty end time...';
    signal sqlstate '45123' set message_text = err_msg;
end if;
END
$$
DELIMITER ;

```

- 6) Keeps track of new entries entered into patient\_personal\_info in logs table:**

```

CREATE TRIGGER `insertlogs` AFTER INSERT ON `Patient_personal_info`
FOR EACH ROW INSERT INTO logs VALUES(null, "new patient admitted", NOW())

```

## Functions & Procedures:

### 1) Function age(patientid int) & Procedure patient\_age\_msg(patientid int):

```
drop function if exists age;
delimiter $$
CREATE FUNCTION age(patientid int) RETURNS int
deterministic

BEGIN
declare answer int;
declare DOB date;
select pat_dob from patient_personal_info where patient_personal_info.Pat_ID=patientid into
DOB;
        select DATE_FORMAT(FROM_DAYS(DATEDIFF(CURDATE(),DOB)), '%Y')+0 AS
age into answer;
        return answer;
END
$$
delimiter ;
select age(1);
```

```
-- Procedure (in continuation of the above function)
drop procedure if exists patient_age_msg;
delimiter $$
CREATE PROCEDURE patient_age_msg(patientid int)
BEGIN
        if (select age(patientid)>18) then
                select 'Patient is a Major';
        else
                select 'Minor Patient';
        end if;
END
$$
delimiter ;
```

```
call patient_age_msg(1);
```

**Call statement in PHP:**

```
$sql= "age('$patientid') ";
```

```
$sql= "patient_age_msg('$ref') ";
```

## 2) Function patient\_bloodgroup(bloodgroup varchar(10)):

```
drop function if exists patient_bloodgroup;
delimiter $$
CREATE FUNCTION patient_bloodgroup(bloodgroup varchar(10)) RETURNS int
deterministic

BEGIN
declare answer int;
      select count(Patient_personal_info.Pat_ID) from Patient_personal_info where
Patient_personal_info.Pat_BloodGroup=bloodgroup into answer;
      return answer;
END
$$
delimiter ;
```

```
select patient_bloodgroup('A+ve');
select patient_bloodgroup('B+ve');
select patient_bloodgroup('O-ve');
```

### Call statement in PHP:

```
$sql= "patient_bloodgroup('$bloodgroup')";
```

## 3) Function total\_patients\_on\_this\_day(dayno int):

```
drop function if exists total_patients_on_this_day;
delimiter $$
CREATE FUNCTION total_patients_on_this_day(dayno int) RETURNS int
deterministic

BEGIN
      declare answer int;
      select count(patient_personal_info.Pat_ID) from appointment natural join
patient_personal_info where dayofweek(appointment.app_date_time)=dayno into answer;
      return answer;
END
$$
delimiter ;
```

```
select total_patients_on_this_day(7);
```

### Call statement in PHP:

```
$sql= "total_patients_on_this_day($dayno)";
```

**4) Function total\_patients\_on\_this\_day(dayno int) & Procedure patient\_count\_msg(dayno int):**

```
drop function if exists total_patients_on_this_day;
delimiter $$
CREATE FUNCTION total_patients_on_this_day(dayno int) RETURNS int
deterministic

BEGIN
declare answer int;
    select count(patient_personal_info.Pat_ID) from appointment natural join
patient_personal_info where dayofweek(appointment.app_date_time)=dayno into answer;
    return answer;
END
$$
delimiter ;
select total_patients_on_this_day(7);

-- Procedure (in continuation with the above function)
drop procedure if exists patient_count_msg;
delimiter $$
CREATE PROCEDURE patient_count_msg(dayno int)
BEGIN
    if (select total_patients_on_this_day(dayno)>2) then
        select 'Many patients on this day!!';
    else
        select 'Not many patients on this day!';
    end if;
END
$$
delimiter ;

call patient_count_msg(7);
```

**Call in PHP**

```
$sql= "patient_count_msg($dayno)";
$sql= "total_patients_on_this_day($dayno)";
```

**5) Function identify\_hod(doctorid varchar(6),departmentid varchar(6)):**

```

drop function if exists identify_hod;
delimiter $$
create function identify_hod(doctorid varchar(6),departmentid varchar(6)) returns boolean
deterministic
begin
    declare answer boolean;
    declare doc varchar(50);
    declare dept varchar(50);
    -- select Doc_ID from HODs where (HODs.Dept_ID=departmentid and
HODs.Doc_ID=doctorid) into doc;
    select Doc_ID from HODs where HODs.Doc_ID=doctorid into doc;
    if doc = null then
        set answer=0;
    else
        select dept_id from HODs where HODs.doc_id=doc into dept;
        if (departmentid =dept) then
            set answer=1;
        else
            set answer=0;
        end if;
    end if;
    return answer;
end
$$
delimiter ;

```

**Call statement in PHP:**

**\$sql= "identify\_hod('\$doctorid','\$departmentid') ";**

**6)**

**A) Procedure maximum\_Doc\_appointments():**

```

DELIMITER $$
CREATE PROCEDURE maximum_Doc_appointments()
BEGIN
SELECT
    Appointment.Doc_ID, Doctor_info.Doc_fname,
    Doctor_info.Doc_contactno,Doctor_info.Doc_DOB,Doctor_info.Doc_gender,
    COUNT(Doc_ID) AS `value_occurrence`
FROM
    Appointment NATURAL JOIN Doctor_info
GROUP BY

```



```

    Doc_ID
ORDER BY
    `value_occurrence`
DESC
LIMIT 1;
END
$$
DELIMITER;
Call statement in PHP:
$sql= "maximum_Doc_appointments()";

```

**B) Procedure Top3\_Doc\_appointments():**

```

DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `Top3_Doc_appointments`()
    NO SQL
BEGIN
SELECT
    Appointment.Doc_ID, Doctor_info.Doc_fname,
    Doctor_info.Doc_contactno,Doctor_info.Doc_DOB,Doctor_info.Doc_gender,
    COUNT(Doc_ID) AS `value_occurrence`
FROM
    Appointment NATURAL JOIN Doctor_info
GROUP BY
    Doc_ID
ORDER BY
    `value_occurrence`
DESC
LIMIT 3;
END$$
DELIMITER ;
Call statement in PHP:
$sql= "Top3_Doc_appointments()";

```

**C) Procedure patient\_with\_maxApps():**

```

DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `patient_with_maxApps`()()
    NO SQL
BEGIN

```

```

SELECT
Patient_personal_info.Pat_ID,Patient_personal_info.Pat_fname,Patient_personal_info.Pat_conta
ctno,Patient_personal_info.Pat_DOB,Patient_personal_info.Pat_gender,
COUNT(App_ID) AS `value_occurrence`
FROM
Appointment
NATURAL JOIN Patient_personal_info
GROUP BY Pat_ID
ORDER BY
`value_occurrence`

DESC
LIMIT 1 ;
END$$
DELIMITER ;
Call statement in PHP:
$sql= "patient_with_maxApps() ";

```

- 7) The procedure `display_patient_appointment_info()` displays appointment details of patients in the following format:

**Patient Name:** <patient name1>  
**App\_ID | Doc\_ID | Doc\_DeptID | App\_Date\_Time**

---

**Patient Name:** <patient name2>  
**App\_ID | Doc\_ID | Doc\_DeptID | App\_Date\_Time**

---

```

drop procedure if exists display_patient_appointment_info;
delimiter $$
CREATE PROCEDURE display_patient_appointment_info()
BEGIN
declare r_patient_id varchar(50);
declare finished integer default 0;
declare c_patient_id cursor for select distinct Patient_personal_info.Pat_ID from
Patient_personal_info;
declare continue handler for not found set finished=1;

```

```

open c_patient_id;

getpatient_id:loop
    fetch c_patient_id into r_patient_id;
    if finished=1 then
        leave getpatient_id;
    end if;

    select r_patient_id as 'Patient ID';
    select distinct Appointment.App_ID, Doctor_info.Doc_ID, Doctor_info.Doc_DeptID,
Appointment.App_Date_Time from Patient_personal_info natural join Appointment natural join
Doctor_info where r_patient_id=Patient_personal_info.Pat_ID;

end loop getpatient_id;
close c_patient_id;

END
$$
delimiter ;

```

```

call display_patient_appointment_info();
Call statement in PHP:
$sql= "display_patient_appointment_info() ";

```

#### 8) Procedure display\_yearwise\_appointments(which\_year year) & Procedure display\_yearwise\_visits(which\_year year)

```

-- Procedure8a
-- Year wise appointments
drop procedure if exists display_yearwise_appointments;
delimiter $$
CREATE PROCEDURE display_yearwise_appointments(which_year year)
BEGIN
    SELECT * from appointment where year(appointment.app_date_time)=which_year;
END
$$
delimiter ;
Call statement in PHP:
$sql= "display_yearwise_appointments('$Appyear')";

```

```

-- Procedure8b
-- Year wise patient visits
drop procedure if exists display_yearwise_visits;
delimiter $$
CREATE PROCEDURE display_yearwise_visits(which_year year)
BEGIN
    SELECT
visit.Visit_ID,appointment.Pat_ID,patient_personal_info.Pat_fname,patient_personal_info.Pat_D
OB from visit natural join appointment natural join patient_personal_info where
year(appointment.app_date_time)=which_year;
END
$$
delimiter ;
Call statement in PHP:
$sql= "display_yearwise_visits('$Visit_Year')";

```

#### 9) Procedure display\_patients\_on\_weekend():

```

drop procedure if exists display_patients_on_weekend;
delimiter $$
CREATE PROCEDURE display_patients_on_weekend()
BEGIN
    -- select patient_personal_info.Pat_ID,patient_personal_info.Pat_fname,
patient_personal_info.Pat_contactno,appointment.App_ID,
appointment.App_Date_Time,dayofweek(appointment.App_Date_Time) from appointment
natural join patient_personal_info where dayofweek(appointment.app_date_time)=1 or
dayofweek(appointment.app_date_time)=7;
    declare finished int default 0;

    declare r_Pat_ID int;
    declare r_Pat_fname varchar(20);
    declare r_Pat_contactno bigint;
    declare r_App_ID varchar(10);
    declare r_App_Date_Time timestamp;

    declare c_display_patients_on_weekend cursor for select p.Pat_ID, p.Pat_fname,
p.Pat_contactno, a.App_ID, a.App_Date_Time from patient_personal_info p natural join
appointment a where dayofweek(a.app_date_time)=1 or dayofweek(a.app_date_time)=7;
    declare continue handler for not found set finished = 1;

```

```

        open c_display_patients_on_weekend;
        c_loop: loop
            fetch c_display_patients_on_weekend into r_Pat_ID, r_Pat_fname, r_Pat_contactno,
r_App_ID, r_App_Date_Time;
            if finished = 1 then
                leave c_loop;
            end if;
            select r_Pat_ID as "PAT_ID", r_Pat_fname as "PAT FNAME", r_Pat_contactno as
"PAT CONTACTNO", r_App_ID as "APP ID", r_App_Date_Time as "APP DATE TIME";
        end loop;

```

END

\$\$

delimiter ;

**Call statement in PHP:**

**\$sql= “display\_patients\_on\_weekend()“;**

#### **10) display\_out\_patient():**

```

drop procedure if exists display_out_patient;
delimiter $$
CREATE PROCEDURE display_out_patient()
BEGIN
    SELECT * from patient_personal_info where Pat_ID not in (SELECT Pat_ID from
in_patients);
END
$$
delimiter ;

```

call display\_out\_patient();

**Call statement in PHP:**

**\$sql= “ display\_out\_patient()“;**

#### **11) Procedure pat\_this\_month():**

```

drop procedure if exists pat_this_month;
delimiter $$
CREATE PROCEDURE pat_this_month()
BEGIN
    declare r_pat_id varchar(50);

```

```

declare finished integer default 0;
declare c_pat_id cursor for select distinct appointment.pat_id from appointment;
declare continue handler for not found set finished=1;

open c_pat_id;

getpat_id:loop
    fetch c_pat_id into r_pat_id;
    if finished=1 then
        leave getpat_id;
    end if;

    -- select * from emp where month(joining_date)=month(current_date());
    select distinct patient_personal_info.Pat_ID, patient_personal_info.Pat_contactno,
patient_personal_info.Pat_fname,patient_personal_info.Pat_gender from appointment natural
join patient_personal_info where month(appointment.app_date_time)=month(current_date()) and
year(appointment.app_date_time)=year(current_date());

end loop getpat_id;
close c_pat_id;

END
$$
delimiter ;

```

```

call pat_this_month();
Call statement in PHP:
$sql= "pat_this_month() ";

```

## 12) Generating Bill

```

Call statement in PHP:
<?php
include_once 'database.php';
if (isset($_POST['save'])) {
    $Pat_ID1 = $_POST['Pat_ID1'];
    $Doc_ID1 = $_POST['Doc_ID1'];
    $visit_charge1 = $_POST['visit_charge1'];
    $Doc_charge1 = $_POST['Doc_charge1'];
    $Med_price1 = $_POST['Med_price1'];

```

```

$no_of_days1 = $_POST['no_of_days1'];

echo 'PATIENT ID IS <br>';
print($Pat_ID1);
echo '<br>DOCTOR ID IS <br>';
print($Doc_ID1);
echo '<br>VISIT CHARGE IS <br>';
print($visit_charge1);
echo '<br>DOCTOR CHARGE IS <br>';
print($Doc_charge1);
echo '<br>MEDICINE PRICE IS <br>';
print($Med_price1);
echo '<br>NUMBER OF DAYS THE PATIENT VISITED/STAYED <br>';
print($no_of_days1);
echo '<br>Total Charge For this patient is <br>';
$total_charge= ($visit_charge1+$Doc_charge1)*$no_of_days1+$Med_price1;
print($total_charge);
echo '<br>( Total_charge = (visit_charge+Doc_charge)*no_of_days+Med_price )';
echo '<br>';
    mysqli_close($conn);
}
?>

```

**e. Screenshots of results generated after procedure and function are called on Front-end**

**Functions & Procedures:**

**1) Function age(patientid int) & Procedure patient\_age\_msg:**

Enter Patient ID :  
  
 [Cancel](#)

age(1)  
20  
Patient is a Major  
[Go back and explore more patient Age](#)

**2) Function patient\_bloodgroup(bloodgroup varchar(10)):**

Enter Blood Group from below :  
  
 [Cancel](#)

**Patient Blood Group Table**  

Pat_BloodGroup
A+ve
B+ve
O+ve
A-ve

Pat\_BloodGroup  
2  
[Go back and explore more bloodgroups](#)

Pat_ID	Pat_fname	Pat_mname	Pat_lname	Pat_gender	Pat_DOB	Pat_contactno	Pat_emailID	Pat_Address	Pat_MedHistory	Pat_BloodGroup
1	Dev	NULL	Mehta	M	2001-01-31	9239420134	dev@gmail.com	Ahmedabad	NULL	A+ve
2	Rahi	NULL	Shah	F	2001-11-02	9239445324	rahi@gmail.com	Ahmedabad	NULL	B+ve
3	Janvi	NULL	Joshi	F	2000-02-02	9239412342	janvi@gmail.com	Ahmedabad	NULL	O+ve
4	Raj	NULL	Majmudar	M	1989-06-22	9124563427	raj@gmail.com	Surat	NULL	O+ve
5	Raina	NULL	Rai	F	1979-12-07	9898563427	raina@gmail.com	Mumbai	NULL	A-ve



3) & 4) Function total\_patients\_on\_this\_day(dayno int) & Procedure patient\_count\_msg(dayno int):

Enter day of the week (considering sunday to saturday as 1-7) :

[Submit](#)

count(patient\_personal\_info.Pat\_ID)

6

Many patients on this day!!

[Go back and explore more](#)

App_ID	Pat_ID	Doc_ID	App_Date_Time	App_Description
App101	1	Doc02	2021-04-18 00:06:35	NewCase
App102	2	Doc01	2021-04-18 00:06:35	NewCase
App103	1	Doc02	2021-04-18 00:06:35	Followup
App104	3	Doc01	2019-12-19 03:14:07	NewCase
App105	3	Doc01	2020-01-01 03:20:10	Dressing
App106	4	Doc02	2020-02-09 05:50:10	NewCase
App107	4	Doc02	2021-04-18 06:19:24	FOLLOWUP
App108	3	Doc01	2021-04-18 06:19:32	NEW CASE

5) Function identify\_hod(doctorid varchar(6),departmentid varchar(6)):

Enter DOC ID

Enter Department ID

[submit](#) [Cancel](#)

HOD Table

Doc_ID	Dept_ID
Doc02	Dept1
Doc01	Dept4

identify\_hod

1

[Go back and explore more HODs](#)

6A) Procedure maximum\_Doc\_appointments():

**Doc With maximum appointments**

Doc_ID	Doc_fname	Doc_contactno	Doc_DOB	Doc_gender
Doc01	Mauli	1960-03-31	1960-03-31	F

[Cancel](#)

## 6B) Procedure Top3\_Doc\_appointments()

**Top 3 Doctors with maximum Appointment (in Decreasing order)**

Doc_ID	Doc_fname	Doc_contactno	Doc_DOB	Doc_gender
Doc01	Mauli	1960-03-31	1960-03-31	F
Doc03	Pranav	1970-07-11	1970-07-11	M
Doc02	Jay	1950-12-21	1950-12-21	M

[Cancel](#)

## 6C) Procedure patient\_with\_maxApps()

**Patients With maximum appointments**

Pat_ID	Pat_fname	Pat_contactno	Pat_DOB	Pat_gender
3	Janvi	9239412342	2000-02-02	F

[Cancel](#)

## 7) Procedure display\_patient\_appointment\_info():

Patient_ID
6

Patient_ID
1

App_ID	Doc_ID	Doc_DeptID	App_Date_Time
App101	Doc02	Dept1	2021-04-18 00:06:35
App103	Doc02	Dept1	2021-04-18 00:06:35

Patient_ID
3

App_ID	Doc_ID	Doc_DeptID	App_Date_Time
App104	Doc01	Dept4	2019-12-19 03:14:07
App105	Doc01	Dept4	2020-01-01 03:20:10
App108	Doc01	Dept4	2021-04-18 06:19:32

Patient_ID
2

App_ID	Doc_ID	Doc_DeptID	App_Date_Time
App102	Doc01	Dept4	2021-04-18 00:06:35

Patient_ID
5

Patient_ID
4

App_ID	Doc_ID	Doc_DeptID	App_Date_Time
App106	Doc02	Dept1	2020-02-09 05:50:10
App107	Doc02	Dept1	2021-04-18 06:19:24

8A) Procedure display\_yearwise\_appointments(which\_year year):

Appointment Year to be displayed :

2020  
submit Cancel

Appointment date and time
2021-04-18 00:06:35
2021-04-18 00:06:35
2021-04-18 00:06:35
2019-12-19 03:14:07
2020-01-01 03:20:10
2020-02-09 05:50:10
2021-04-18 06:19:24
2021-04-18 06:19:32

**App\_ID Pat\_ID Doc\_ID App\_Date\_Time App\_Description**  
 App105 3 Doc01 2020-01-01 03:20:10 Dressing  
**App\_ID Pat\_ID Doc\_ID App\_Date\_Time App\_Description**  
 App106 4 Doc02 2020-02-09 05:50:10 NewCase

[Go back and explore more App years](#)

8B) Procedure display\_yearwise\_visits(which\_year year):

Appointment Year to be displayed :

2021  
submit Cancel

Appointment date and time
2021-04-18 00:06:35
2021-04-18 00:06:35
2021-04-18 00:06:35
2019-12-19 03:14:07
2020-01-01 03:20:10
2020-02-09 05:50:10
2021-04-18 06:19:24
2021-04-18 06:19:32

---

**App\_ID Pat\_ID Doc\_ID App\_Date\_Time App\_Description**  
 App101 1 Doc02 2021-04-18 00:06:35 NewCase  
**App\_ID Pat\_ID Doc\_ID App\_Date\_Time App\_Description**  
 App102 2 Doc01 2021-04-18 00:06:35 NewCase  
**App\_ID Pat\_ID Doc\_ID App\_Date\_Time App\_Description**  
 App103 1 Doc02 2021-04-18 00:06:35 Followup  
**App\_ID Pat\_ID Doc\_ID App\_Date\_Time App\_Description**  
 App107 4 Doc02 2021-04-18 06:19:24 FOLLOWUP  
**App\_ID Pat\_ID Doc\_ID App\_Date\_Time App\_Description**  
 App108 3 Doc01 2021-04-18 06:19:32 NEW CASE

[Go back and explore more App years](#)

## 9) Procedure display\_patients\_on\_weekend():

Pat_ID	Pat_fname	Pat_contactno	App_ID	App_Date_Time
1	Dev	9239420134	App101	2021-04-18 00:06:35
2	Rahi	9239445324	App102	2021-04-18 00:06:35
1	Dev	9239420134	App103	2021-04-18 00:06:35
4	Raj	9124563427	App106	2020-02-09 05:50:10
4	Raj	9124563427	App107	2021-04-18 06:19:24
3	Janvi	9239412342	App108	2021-04-18 06:19:32

[Go back and explore more](#)

## 10) Procedure display\_out\_patient():

Outdoor patients info (patients who are not in in\_patients)

Pat_ID	Pat_fname	Pat_mname	Pat_lname	Pat_gender	Pat_DOB	Pat_contactno	Pat_emailID	Pat_Address	Pat_MedHistory	Pat_BloodGroup
1	Dev		Mehta	M	2001-01-31	9239420134	dev@gmail.com	Ahmedabad		A+ve
2	Rahi		Shah	F	2001-11-02	9239445324	rahi@gmail.com	Ahmedabad		B+ve
3	Janvi		Joshi	F	2000-02-02	9239412342	janvi@gmail.com	Ahmedabad		O+ve
4	Raj		Majmudar	M	1989-06-22	9124563427	raj@gmail.com	Surat		O+ve
5	Raina		Rai	F	1979-12-07	9898563427	raina@gmail.com	Mumbai		A-ve
6	darsh	d	patel	M	2001-04-22	9090909001	d2@gmail.com	ahmedabad	none	O+ve

[Cancel](#)

## 11) Procedure pat\_this\_month():

Pat_ID	Pat_contactno	Pat_fname	Pat_gender	app_date_time
1	9239420134	Dev	Dev	2021-04-18 00:06:35
2	9239445324	Rahi	Rahi	2021-04-18 00:06:35
4	9124563427	Raj	Raj	2021-04-18 06:19:24
3	9239412342	Janvi	Janvi	2021-04-18 06:19:32

[Cancel](#)

## 12) Generate Individual Patient's Bill for a particular Doctor:

- Enter given data :

### Generate individual Patient's Bill for a particular Doctor

(\*referring the Below Data)

Pat\_ID  
2  
Doc\_ID  
Doc01  
visit\_charge  
500  
Doc\_charge  
2000  
Med\_price  
15  
no\_of\_days  
5

[Cancel](#)

Generate Bill with the help below Data

Pat_ID	Doc_ID	Doc_charge	Visit_ID
2	Doc01	2000	Vis101
3	Doc01	2000	Vis103
3	Doc01	2000	Vis104
1	Doc02	5000	Vis102
Visit_ID	Current_Med	Visit_charge	
Vis101	Med101	500	
Med_ID	Med_price		
Med101	15		
Med102	20		
Med103	150		

#### - Bill with Total Charge generated :

PATIENT ID IS  
2  
DOCTOR ID IS  
Doc01  
VISIT CHARGE IS  
500  
DOCTOR CHARGE IS  
2000  
MEDICINE PRICE IS  
15  
NUMBER OF DAYS THE PATIENT VISITED/STAYED  
5  
Total Charge For this patient is  
12515  
( Total\_charge = (visit\_charge+Doc\_charge)\*no\_of\_days+Med\_price )

[Go back for Generating More Bills](#)

## f. Screenshots of errors generated on front-end when trigger is fired

### Triggers:

#### 1) Trigger1 (On Medicines Table: Updates stored in Medicine\_Updates table)

Med_ID	Med_Name	Med_company	Med_price	Med_dose	Med_type
Med101	DAN-P	UNISON	15	600mg	Anti inflammatory
Med102	DOLO	Torrent	20	650mg	Paracetamol
Med103	Clavam	Zydus	125	625mg	Anti biotic

#### Medicine List

Medicine ID:

Med103

Medicine name:

Medicine Company :

Medicine Price:

150

Medicine Dose:

625mg

Medicine type:

Update

[Go back and update other medicines](#)

#### “Medicine Updates” Table:

2021-04-14 20:16:27	Med_price	66	
2021-04-14 20:16:27	Med_dose	66	55
2021-04-14 20:16:27	Med_type	mm	na
2021-04-14 20:17:15	Med_Name	ll	updatenname 2
2021-04-14 20:17:15	Med_Name	ll	update cmpany 2
2021-04-14 20:17:15	Med_type	ll	na 2
2021-04-14 20:17:15	Med_Name	ll	updatenname 2
2021-04-14 20:17:15	Med_Name	ll	update cmpany 2
2021-04-14 20:17:15	Med_type	ll	na 2
2021-04-14 21:40:49	Med_Name	company	GlaxoSmithKline
2021-04-15 15:20:31	Med_Name	et	name
2021-04-15 15:20:31	Med_Name	et	company
2021-04-15 15:20:31	Med_price	44	15
2021-04-15 15:20:31	Med_dose	et	15mg
2021-04-15 15:20:31	Med_type	et	type
2021-04-18 10:23:17	Med_price	120	125
2021-04-18 10:32:43	Med_Name	Clavam	
2021-04-18 10:32:43	Med_Name	Zydus	
2021-04-18 10:32:43	Med_price	125	150
2021-04-18 10:32:43	Med_type	Anti biotic	

2) Trigger 2 (On Room Table: 'Room charge cannot exceed 10000! Couldn't perform the operation...')

---

ROOM ID :  
  
Room TYPE:  
  
Room Charge:

[Cancel](#)

Error:  
INSERT INTO `Rooms` (`Room\_ID`, `Room\_Type`, `Room\_Charge`) VALUES ('room180','super deluxe','20000') [Error: Room charge cannot exceed 10000! Couldn't perform the operation...](#)  
[Go back and add other rooms](#)

3) Trigger 3 (On Patient\_personal\_info, Doctor\_info, and Staff tables: contact no length not equal to 10 then "Invalid contact no")

Doctor ID:  
  
Doctor First Name:  
  
Doctor Middle Name  
  
Doctor Last name  
  
gender:  
  
Date OF Birth  
   
Doctor Contact Number:  
  
Doctor Email ID :  
  
Address :  
  
Doctor-Specialist :  
  
Doctor Department ID  
  
Doctor Charge

[Cancel](#)

## Error:

Error: INSERT INTO `Doctor\_info` (`Doc\_ID`, `Doc\_fname`, `Doc\_mname`, `Doc\_Lname`, `Doc\_gender`, `Doc\_DOB`, `Doc\_contactno`, `Doc\_emailID`, `Doc\_Address`, `Doc\_Specialist`, `Doc\_DeptID`, `Doc\_charge`) VALUES ('doc07','darsh','d','patel','m','2001-04-22','98789212341','d2@gmail.com','11d@gmail.com','surgeon','Dept3','1000') **Error: Invalid Contact No..**

[Go back for inserting More patients](#)

## 4) Trigger 4 (On Appointment, billing info, in-patients, Patient\_personal\_info, Visit, and treatments table: Deleting record deletes records from child tables)

### Verification (step by step) in Appointment table :

[Go back and Delete more patient Records](#) Check appointment table for Trigger 4 Confirmation

App_ID	Pat_ID	Doc_ID	App_Date_Time	App_Description
App101	1	Doc02	2021-04-18 00:06:35	NewCase
App102	2	Doc01	2021-04-18 00:06:35	NewCase
App103	1	Doc02	2021-04-18 00:06:35	Followup
App104	3	Doc01	2019-12-19 03:14:07	NewCase
App105	3	Doc01	2020-01-01 03:20:10	Dressing
App106	4	Doc02	2020-02-09 05:50:10	NewCase
App107	4	Doc02	2021-04-18 06:19:24	FOLLOWUP
App108	3	Doc01	2021-04-18 06:19:32	NEW CASE

Pat_ID	Pat_fname	Pat_mname	Pat_Lname	Pat_gender	Pat_DOB	Pat_contactno	Pat_emailID	Pat_Address	Pat_MedHistory	Pat_BloodGroup	
1	Dev		Mehta	M	2001-01-31	9239420134	dev@gmail.com	Ahmedabad		A+ve	<a href="#">Delete</a>
2	Rahi		Shah	F	2001-11-02	9239445324	rahi@gmail.com	Ahmedabad		B+ve	<a href="#">Delete</a>
3	Janvi		Joshi	F	2000-02-02	9239412342	janvi@gmail.com	Ahmedabad		O+ve	<a href="#">Delete</a>
4	Raj		Majmudar	M	1989-06-22	9124563427	raj@gmail.com	Surat		O+ve	<a href="#">Delete</a>
5	Raina		Rai	F	1979-12-07	9898563427	raina@gmail.com	Mumbai		A-ve	<a href="#">Delete</a>

[Cancel](#)

Record deleted successfully

Corresponding Child Records from Appointment, Billing\_info and In\_Patients deleted successfully

[Go back and Delete more patient Records](#) Check appointment table for Trigger 4 Confirmation

App_ID	Pat_ID	Doc_ID	App_Date_Time	App_Description
App101	1	Doc02	2021-04-18 00:06:35	NewCase
App102	2	Doc01	2021-04-18 00:06:35	NewCase
App103	1	Doc02	2021-04-18 00:06:35	Followup
App104	3	Doc01	2019-12-19 03:14:07	NewCase
App105	3	Doc01	2020-01-01 03:20:10	Dressing
App108	3	Doc01	2021-04-18 06:19:32	NEW CASE



## 5) Trigger5 (On Staff Table: Duty start time cannot be same as Duty end time...)

Staff ID:  
S4

First Name:  
darsh

Middle Name:  
d

Last Name:  
patel

Gender (M or F) :  
M

Contact number :  
9879820606

Staff Type :  
WardBoy

Staff Assigned Room 1:  
room2

Staff Assigned Room 2:  
room6

Staff duty start time:  
10:48 AM

Staff duty end time:  
10:48 AM

Staff holiday:  
monday

Staff Department ID:  
Dept1

Staff Charge:  
600

[Cancel](#)

Staff_ID	Staff_fname	Staff_mname	Staff_lname	Staff_gender	Staff_contactno	Staff_type	Staff_roomassigned1	Staff_roomassigned2	Staff_dutystarttime	Staff_dutyendtime	Staff_holiday	Sta
S1	Deep			M	9341568535	wardboy	room4		01:00:00	08:00:00	Wednesday	Dej
S2	Prakash			M	9823416792	assistant	room5		08:00:00	01:00:00	Wednesday	Dej
S3	Mamta			F	9135968535	nurse	room1	room3	09:00:00	05:00:00	Sunday	Dej

## Error Prompt

Error: INSERT INTO `Staff` (`Staff\_ID`, `Staff\_fname`, `Staff\_mname`, `Staff\_lname`, `Staff\_gender`, `Staff\_contactno`, `Staff\_type`, `Staff\_roomassigned1`, `Staff\_roomassigned2`, `Staff\_dutystarttime`, `Staff\_dutyendtime`, `Staff\_holiday`, `Staff\_DeptID`, `Staff\_Charge`) VALUES ('S4','darsh','d','patel','M','9879820606','WardBoy','room2','room6','10:48','10:48','monday','Dept1','600') **Error: Duty start time cannot be same as Duty end time...**

[Go back for inserting More patients](#)

## 6) Logs Table -

Patient ID:  
4

First Name:  
darsh

Middle Name:  
d

Last Name:  
patel

Gender (M or F) :  
M

Date Of Birth :  
22 / 04 / 0021

Contact Number :  
9879820606

Email Address:  
d5@gmail.com

Address (city) :  
Mehsana

Medical history:  
None

Choose BloodGroup:  
O+ve

[Cancel](#)

Patient Table

Pat_ID	Pat_fname	Pat_mname	Pat_lname	Pat_gender	Pat_DOB	Pat_contactno	Pat_emailID	Pat_Address	Pat_MedHistory	Pat_BloodGroup
1	Dev		Mehta	M	2001-01-31	9239420134	dev@gmail.com	Ahmedabad		A+ve
2	Rahi		Shah	F	2001-11-02	9239445324	rahi@gmail.com	Ahmedabad		B+ve
3	Janvi		Joshi	F	2000-02-02	9239412342	janvi@gmail.com	Ahmedabad		O+ve
5	Raina		Rai	F	1979-12-07	9898563427	raina@gmail.com	Mumbai		A-ve

## - Log is inserted after adding a new patient

New record created successfully !

[Go back for inserting More patients](#)

id	action	created_date
31	patient inserted	2021-04-16 00:10:59
32	patient inserted	2021-04-16 03:07:57
33	patient inserted	2021-04-16 03:34:04
34	patient inserted	2021-04-16 04:44:57
35	patient inserted	2021-04-16 11:30:39
36	patient inserted	2021-04-16 11:36:28
37	patient inserted	2021-04-16 11:36:28
38	patient inserted	2021-04-16 11:36:28
39	patient inserted	2021-04-16 11:36:28
40	patient inserted	2021-04-16 11:36:28
41	patient inserted	2021-04-16 11:36:28
42	patient inserted	2021-04-16 11:43:05
43	patient inserted	2021-04-16 11:45:44
44	patient inserted	2021-04-17 03:43:35
49	patient inserted	2021-04-17 06:00:13
50	patient inserted	2021-04-17 07:19:51
51	patient inserted	2021-04-17 07:19:51
52	patient inserted	2021-04-17 07:29:59
53	patient inserted	2021-04-17 07:29:59
54	patient inserted	2021-04-17 07:29:59
55	new patient admitted	2021-04-17 23:44:57
56	new patient admitted	2021-04-17 23:44:57
57	new patient admitted	2021-04-17 23:44:57
58	new patient admitted	2021-04-17 23:44:57
59	new patient admitted	2021-04-17 23:44:57
60	new patient admitted	2021-04-18 09:09:56
61	new patient admitted	2021-04-18 10:53:01

---