DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

PROJECT REPORT

PROJECT TITLE: HOSPITAL MANAGEMENT SYSTEM

Course Name:	CC2141L - Database Systems
Course Instructor:	Muhammad Ehsan
Section:	V5

EXECUTIVE SUMMARY:

The project is a detailed overview of **HOSPITAL MANAGEMENT SYSTEM**

PROBLEM STATEMENT:

Medical care is one of the most essential and in demand service for all. It requires a lot of attention and high-quality service that also causes health care workers to do a lot of effort. These issues also add the situations where there's a need for physical attendant for every patient wherein it could be automated and handled with technology.

SOLUTION:

To address the issues faced by hospitals, a system named Hospital Management System is proposed. This system will handle information such as **patents with mild diagnosis and prescriptions**. The idea of having the automated patient management is a big help for our health care workers and physicians to monitor and take good care of the patients.

SCOPE:

The hospital management system could handle specific task such as securing various information of the patients. This will help them secure the data to keep patient-doctor confidentiality as well as assure them of their healing factors. It can also improve the productivity of the, health care workers and could let them accommodate more patients

Entities Selection

- Patient

S NO	**COLUMN NAME**	**DATA TYPE**	**CONSTRAINTS**	**DESCRIPTION**
1.	Patient_ID	INT	Primary-Key	Contains unique ID of patient
2.	Name	VARCHAR		Contains name
3.	DOB	VARCHAR		Contains DOB
4.	Age	FLOAT		Contains Age
5.	Phone_no	INT	Multi-Valued Attribute	Contains Phone Number
6.	Address	VARCHAR		Contains Address
7.	Sex	VARCHAR		Contains Sex
9.	Admit_date	DATE		Contains Admit Date
9.	Blood Group	Varchar		Contains Blood Group of patients

- Doctor

S No	Column Name	Data Type	Constraints	Description
1.	Name F_Name L_Name	VARCHAR	Composite Attribute	Contains Name
2.	D_Password	VARCHAR		Contains Password
3.	D-Id	VARCHAR	Primary-Key	Contains Unique ID
4.	Phone-No	INT	Multi-Valued Attribute	Contains phone number
5.	Salary	FLOAT		Contains salary
6.	Email	VARCHAR		Contains email id
7.	Sex	VARCHAR		Contains sex
8.	Cabin No	INT		Contains Cabin number
9.	Specializations	VARCHAR	Multi-Valued Attribute	Contains specialization
10.	Address	VARCHAR		Contains Address of Doctor

- Pharmacy

S No	Column Name	Data Type	Constraints	Description
1.	Medbill_no	Int	Primary key	Contains Bill no of medicines
2.	Amount	float		Contains Amount

- Receptionist

S No	Column Name	Data Type	Constraints	Description
1.	Rec_ID	VARCHAR	Primary Key	Contains unique receptionist id
2.	Rec_Password	VARCHAR		Contains Password
3.	R-Name	Varchar		Contains Name
4.	Phone-No	Int	Multi-Valued Attribute	Contains phone number
5.	Gender	Varchar		Contains gender
6.	Email	Varchar		Contains Email id
7.	Salary	Float		Contains Salary
8.	Address	VARCHAR		Contains address of receptionist

- Department

S No	Column Name	Data Type	Constraints	Description
1.	Dept_ID	Int	Primary Key	Contains unique dept_id
2.	Dept_Name	varchar		Contains name for departments
3.	Dept_location	Varchar		Contains location of each department

- Admin

S No	Column name	Data Type	Constraints	Description
1.	Admin_ID	VARCHAR	Primary Key	Contains unique ID of Admin
2.	Ad_Password	VARCHAR		Contains Password
3.	Email	Varchar		Contains email
4.	Admin name	Varchar		Contains admin name
5.	Phone_no	Int	Multi-Valued Attribute	Contains phone number
6.	Gender	Varchar		Contains gender
7.	Salary	float		Contains salary

- Nurse

S No	Column Name	Data Type	Constraints	Description
1.	Nurse_ID	Varchar	Primary Key	Contains unique nurse ID (This also serves as username for nurse)
2.	N_Password	Varchar		Contains password
3.	N_Name	Varchar		Contains nurse name
4.	Salary	Float		Contains salary
5.	Phone_No	Int	Mulit-Valued Attribute	Contains nurse phone number
6.	Gender	Varchar		Contains gender
7.	Address	Varchar		Contains address

- Laboratory

S No	Column Name	Data Type	Constraints	Description
1.	Attendant_ID	varchar	Primary Key	Contains unique Attendant ID
2.	L_Password	Int		Contains password
3.	Attendant_Name	Varchar		Contains Attendant name
4.	Phone_No	Int	Multi-Valued Attribute	Contains phone number
5.	Gender	Varchar		Contains Gender
6.	Email	Varchar		Contains email
7.	Address	Varchar		Contains address
8.	Salary	Float		Contains salary

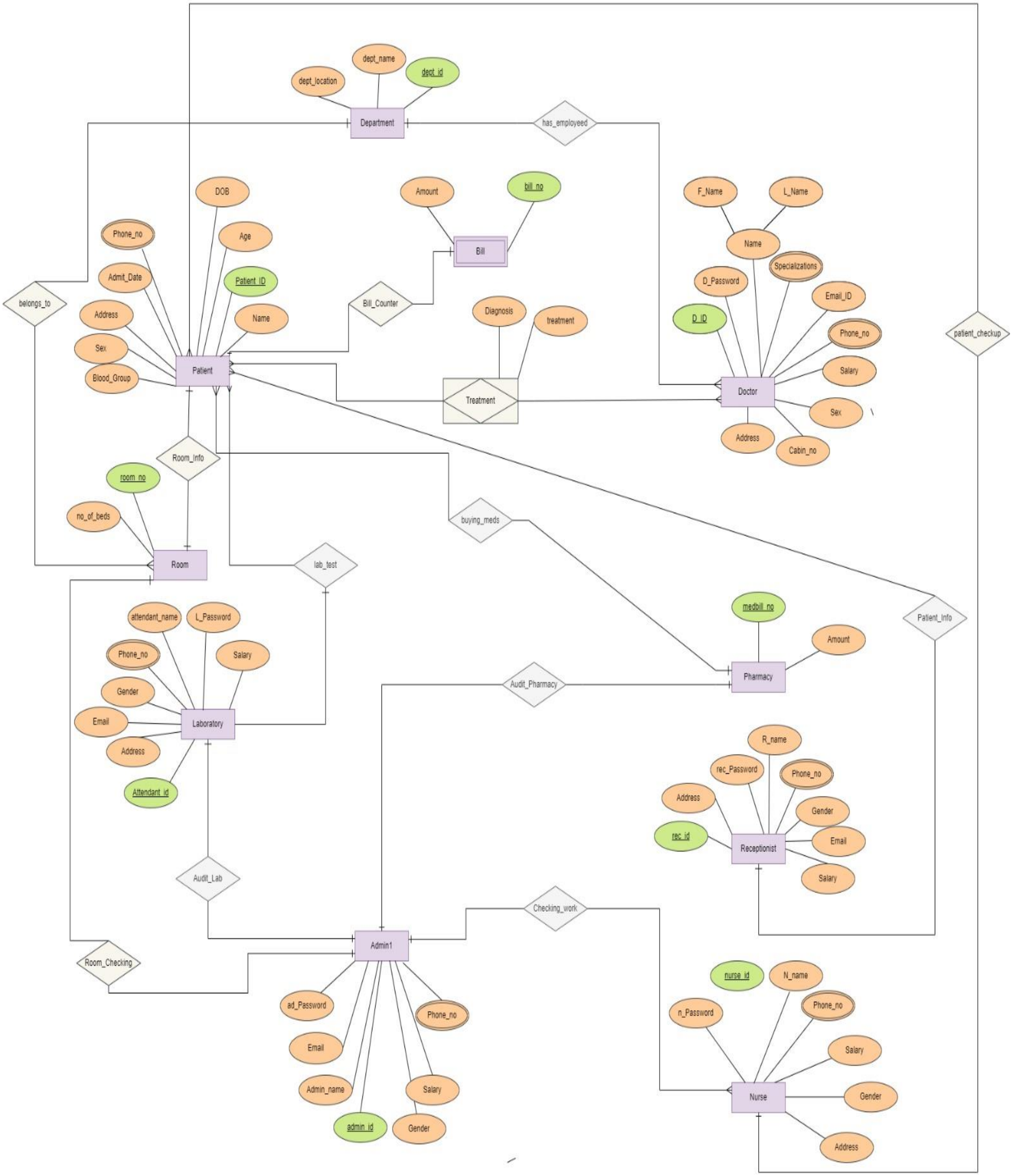
- Room

S No	Column Name	Data Type	Constraints	Description
1.	Room_no	Varchar	Primary Key	Contains room number
2.	No_of_beds	Int		Contains no of beds

- Bill (Weak Entity):

S No	Column Name	Data Type	Constraints	Description
1.	Amount	Int		Contains Bill Amount
2.	Bill_No	Int	Primary key	Contains Unique bill number that contains Doctor’s fee

Entity Relationship Diagram:



Relations:

There are following relationships in the above HOSPITAL MANAGEMENT SYSTEM

Treatment: ASSOCIATIVE ENTRY

- A patient is treated by many Doctors (1:M)
- Many patients are treated by multiple doctors (M:M)

Bill Counter:

- Only one bill with certain amount is charged for one patient with certain ailment (1:1)

Buying_Meds:

- A patient goes to pharmacy to buy medicines and at pharmacy many patients come to buy medicines(1:M)

Patient Info:

- Receptionist stores many patients' information but a patient information is stored by particular receptionist (1:M)

Audit Pharmacy:

- Admin stores audit of multiple pharmacies but the audit of any particular pharmacy is stored by one admin (1:M)

Audit Lab:

- Admin stores audit of one lab and audit of any particular laboratory is stored by one admin (1:1)

Room Checking:

- Admin can check many rooms but rooms are checked by one admin (1:M)

Checking Work:

- Admin can check hours for many nurses but the hours of nurse is checked by only one admin (1:M)

LAB_Test:

- A laboratory can store the records of test of many patients but the patient record is stored at given laboratory (1:M)

Room Info:

- Room Attendant can store the records of multiple patients but patients' records are stored for only one room (1:M)

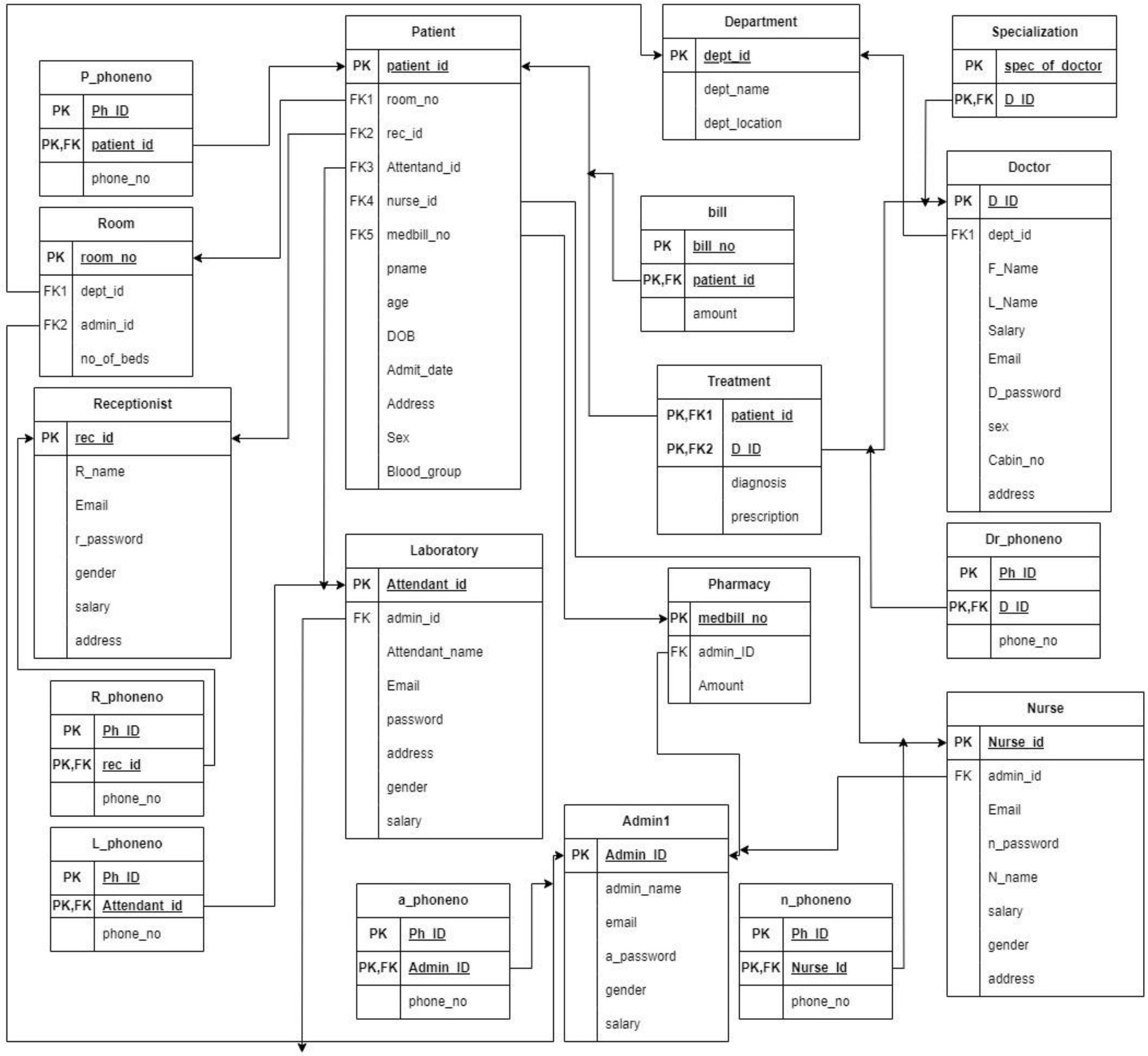
Patient Checkup:

- A nurse can check many patients but a patient is checked by one nurse (1:M)

Belongs_To:

- A room belongs to a single department but a department has many rooms (1:M)

Relational Schema:



MYSQL Queries:

```
create database hospital;
use hospital;
-- DDL Commands
create table patient(
Patient_Id varchar(10) primary key,
P_Name varchar(50) not null,
P_Age int,
P_DOB date not null,
Admit_Date date not null,
Address varchar(100) not null,
Sex varchar(50) not null,
Blood_Group varchar(50) ,
Room_no varchar(10), foreign key(Room_no)references Room(Room_no) on update cascade on delete cascade,
Rec_ID varchar(10),  foreign key(Rec_ID )references Receptionist(Rec_ID) on update cascade on delete cascade,
Attendant_ID varchar(10), foreign key(Attendant_ID) references Laboratory(Attendant_ID) on update cascade on delete cascade,
Nurse_ID varchar(10),  foreign key(Nurse_ID)references Nurse(Nurse_ID)on update cascade on delete cascade,
Medbill_No varchar(10), foreign key(Medbill_No)references pharmacy(Medbill_No) on update cascade on delete cascade
);
select * from patient;
create table P_phoneno
(
 Ph_ID varchar(10),
 Patient_Id varchar(10) , foreign key(Patient_Id)references patient(Patient_Id),
 phone_no int,
 primary key( Ph_ID,Patient_Id)
);
-- weak entity
create table bill(
Bill_No varchar(10),
 Patient_Id varchar(10), foreign key(Patient_Id)references patient(Patient_Id),
Amount int,
primary key(Bill_No,Patient_Id)
);
create table department(
Dept_ID int primary key,
Dept_name varchar(100),
Dept_Location varchar(100)
);
create table doctor(
D_ID varchar(10) primary key,
F_name varchar(20),
L_name varchar(20),
D_Password varchar(20),
Sex varchar(10),
Salary float,
Email varchar(50),
Cabin_no int,
Dept_ID int , foreign key(Dept_ID)references Department(Dept_ID) on update cascade on delete cascade
);
create table Dr_phoneno
(
 Ph_ID varchar(10),
 D_ID varchar(10), foreign key(D_ID)references doctor(D_ID) on update cascade on delete cascade,
 phone_no int,
 primary key(Ph_ID,D_ID)
);
create table specialization
(
 spec_of_doctor varchar(100),
 D_ID varchar(10), foreign key(D_ID)references doctor(D_ID) on update cascade on delete cascade,
  primary key(spec_of_doctor,D_ID)
);
create table Treatment
(
Patient_Id varchar(10), foreign key(Patient_Id)references patient(Patient_Id) on update cascade on delete cascade,
 D_ID varchar(10), foreign key(D_ID)references doctor(D_ID) on update cascade on delete cascade,
 Diagnosis varchar(100),
 prescription varchar(100),
 primary key(Patient_Id , D_ID )
);
create table pharmacy
(
 Medbill_No varchar(10) primary key,
 Admin_ID varchar(10), foreign key(Admin_ID)references admin1(Admin_ID) on update cascade on delete
cascade,
 Amount int
);
create table Receptionist(
Rec_ID varchar(10) primary key,
R_Name varchar(15) ,
Email varchar(50),
R_password varchar(15) ,
Gender varchar(10),
Salary int,
address varchar(50)
);
create table R_phoneno(
Ph_ID varchar(10),                                        /*Multivalued attribute*/
Rec_ID varchar(10),  foreign key(Rec_ID )references Receptionist(Rec_ID) on update cascade on delete cascade,
phone_no int,
primary key(Ph_ID ,Rec_ID)
);
```

```
create table Nurse(
Nurse_ID varchar(10) primary key,
N_Name varchar(15),
email varchar(50),
N_password varchar(20),
Salary int,
Gender varchar(10),
Address varchar(50),
Admin_ID varchar(10), foreign key(Admin_ID)references admin1(Admin_ID) on update cascade on delete cascade
);
create table N_phoneno(
Ph_ID varchar(10) ,
Nurse_ID varchar(10),  foreign key(Nurse_ID)references Nurse(Nurse_ID) on update cascade on delete cascade,
phone_no int,
primary key(Ph_ID,Nurse_ID)
);
create table Admin1(
Admin_ID varchar(10) primary key,
Admin_Name varchar(20),
Email varchar(20),
A_password varchar(20),
Gender varchar(10),
Salary int
);
create table A_phoneno(
Ph_ID varchar(10) ,
Admin_ID varchar(10), foreign key(Admin_ID) references Admin1(Admin_ID) on update cascade on delete cascade,
phone_no int,
primary key(Ph_ID ,Admin_ID )
);
create table Laboratory(
Attendant_ID varchar(10) primary key,
Attendant_name varchar(15),
Email varchar(50),
L_password varchar(20),
Gender varchar(10),
Address varchar(30),
Salary int,
Admin_ID varchar(10), foreign key(Admin_ID) references Admin1(Admin_ID) on update cascade on delete cascade
);
create table L_phoneno(
Ph_ID varchar(10),
Attendant_ID varchar(10), foreign key(Attendant_ID) references Laboratory(Attendant_ID),
phone_no int,
primary key(Ph_ID, Attendant_ID)
);
create table Room(
room_no varchar(10) primary key,
Dept_ID int, foreign key(Dept_ID)references Department(Dept_ID),
Admin_ID varchar(10), foreign key(Admin_ID)references admin1(Admin_ID),
no_of_beds int
);
-- RENAME COLUMN
alter table patient change Patient_Condition patient_condition varchar(20) ;
-- DELETION
delete from Pharmacy where Amount = 0;
-- drop
alter table admin1 drop column attendance;

-- DML commands
insert into department
values (1,"Radiology","1st floor right side"),
(2,"OPD","1st floor left side"),
(3,"pediatrics","2nd floor left side"),
(4,"Orthopedics","2nd floor right side"),
(5,"ENT","ground floor center");
insert into doctor
values("D1",'Jaweria','Rizwan','@3974','Female',1500000.0,'javeria@gmail.com',01,1),
("D2",'Zara','Abrar','@5544','Female',14000000.0,'zzaranoor8@gmail.com',02,2),
("D3",'Reshmail','Fatima','@5804','Female',1300000.0,'resh04@gmail.com',05,2),
("D4",'Alizey','Nadeem','@8624','Female',1200000.0,'alizeygull@gmail.com',15,4),
("D5",'Haroon','Javaid','@6700','Male',1700000.0,'haroon.458@gmail.com',13,3),
("D6",'Asad','Jaffar','@2948','Male',1500000.0,'jaffar@gmail.com',07,5);
insert into dr_phoneno
values("D1PH1","D1",929999),
("D1PH2","D1",929429),
("D2PH1","D2",927523),
("D3PH1","D3",920000),
("D4PH1","D4",924360),
("D5PH1","D5",927841),
("D5PH2","D5",928475),
("D6PH1","D6",927625);
insert into specialization
values("radiology specialist","D1"),
("Physical examination","D2"),
("Physical examination","D3"),
("ENT specialist","D3"),
("orthopedics specialist","D4"),
("pediatrics specialist","D5"),
("ENT specialist","D6");
insert into receptionist
values("R1",'Zunaira Ali','zunaira02@gmail.com',"@456",'Female',20000,"abcLahore"),
("R2",'Furqan Ahmad','Furqanahmad@gmail.com',"@601",'Male',22000,"123Lahore"),
("R3",'Muhammad Jameel','Jameel07@gmail.com',"@994",'Male',17000,"xyzLahore"),
("R4",'Rubab Khurram','rubab__@gmail.com',"@230",'Female',20000,"ghiLahore");
insert into r_phoneno values("R1PH1","R1",921001),
("R2PH1","R2",921111),
 ("R3PH1","R3",924512),
("R4PH1","R4",921199),
("R4PH2","R4",923499);
insert into admin1
values("A1",'Adeel Hayat','ade.el@gmail.com',"@356",'Male',200000),
( "A2",'Nabeel Bukhari','nabe.el@gmail.com',"@854",'Male',150000),
("A3", 'Malik Ahsan','Malik@gmail.com',"@444",'Male',3700000),
("A4",'Asmat Kazmi','Kazmi@gmail.com',"@450",'Male',300000);
insert into a_phoneno values("A1PH1","A1",921481);
insert into a_phoneno values("A2PH1","A2",921777);
insert into a_phoneno values("A3PH1","A3",9214912);
insert into a_phoneno values("A4PH1","A4",921759);
/*Pharmacy*/
insert into Pharmacy values('ph1','A1',1500);
insert into Pharmacy values('ph2','A2',600);
insert into Pharmacy values('ph3','A3',500);
insert into Pharmacy values('ph4','A4',1200);
/*Nurse*/
insert into Nurse values('N1','Fakhar Jawad','Abid',579,70000,'Female','Lahore','A1');
insert into Nurse values('N2','Abdul Jabbar','Khurram',975,50000,'Male','Lahore','A2');
insert into Nurse values('N3','Abdul Rehman','Liaqat',452,55000,'Male','Karachi','A3');
insert into Nurse values('N4','Anoushe Bajwa','Shafqat',502,45000,'Female','Islamabad','A4');
/*Nurse phone number*/
insert into N_phoneno values("N1PH1","N1",921444);
insert into N_phoneno values("N2PH1","N2",921098);
insert into N_phoneno values("N3PH1","N3",921987);
insert into N_phoneno values("N4PH1","N4",921993);
/*Laboratory*/
insert into Laboratory values('L1','Shujah Haider','naeem09@gmail.com',987,'Male','defLahore',15000,'A1');
insert into Laboratory values('L2','Danish Ali','danishali09@gmail.com',640,'Male','jklLahore',17000,'A2');

insert into Laboratory values('L3','Abu-Bakar','abu.bakar@gmail.com',742,'Male','mnoLahore',12000,'A3');

insert into Laboratory values('L4','Bushra Bibi','Bushra00@gmail.com',540,'Female','pqrLahore',15000,'A4');

insert into Laboratory values('L5','Amna Bibi','amna12@gmail.com',542,'Female','stuLahore',15000,'A4');

/*Laboratory PHONE NUMBER*/

insert into L_phoneno values("L1PH1","L1",921444);

insert into L_phoneno values("L2PH1","L2",921098);

insert into L_phoneno values("L3PH1","L3",921987);

insert into L_phoneno values("L4PH1","L4",921993);

/*Room*/

insert into Room values('RO1',1,'A1',3);

insert into Room values('RO2',1,'A1',5);

insert into Room values('RO3',2,'A2',8);

insert into Room values('RO4',2,'A2',7);

insert into Room values('RO5',3,'A3',9);

insert into Room values('RO6',4,'A4',11);


/*patient*/

insert into patient values("P1",'Asma Amir',21,'2002-12-14','2023-05-10','Shahdara Lahore','MALE','A+','RO1','R1','L1','N1','ph1');

insert into patient values("P2",'Usman Khalid',22,'2001-09-04','2022-11-10','Lahore','MALE','B+','RO2','R2','L2','N2','ph2');

insert into patient values("P3",'Talha Waqar',22,'2001-07-14','2023-05-28','Queta','MALE','O+','RO3','R3','L3','N3','ph3');

insert into patient values("P4",'Wassam Shah',21,'2001-09-01','2020-05-10','DHA Lahore','MALE','A-','RO4','R4','L4','N4','ph4');

/*Patient PHONE NUMBER*/

insert into P_phoneno values("P1PH1","P1",921444);

insert into P_phoneno values("P2PH1","P2",921098);

insert into P_phoneno values("P3PH1","P3",921987);

insert into P_phoneno values("P4PH1","P4",921993);

/*BILL*/

insert into bill values("B1","P1",1200);
insert into bill values("B2","P2",800);
insert into bill values("B3","P3",900);
insert into bill values("B4","P4",1000);
/*Treatment*/
insert into treatment values("P1","D1",'common cold','Paracetamol, 500 mg,3 times a day for 3 days');
insert into treatment values("P2","D2",'seasonal allergies','Steam inhalation, As required, Until symptoms subside');
insert into treatment values("P3","D3",'migraine','Antihistamine, 10 mg, Once daily for 1 week');
insert into treatment values("P4","D4",'acid reflux','Antacids, As directed, Until symptoms improve');

-- ALTER COMMAND
alter table Patient add Patient_Condition varchar(30);
-- UPDATION
update Patient set Patient_Condition = 'Mild cold' where Patient_ID = 1;
update Patient set Patient_Condition = 'serious allergy' where Patient_ID = 2;
update Patient set Patient_Condition = 'Under-Observation' where Patient_ID = 3;
update Patient set Patient_Condition = 'Better' where Patient_ID = 4;
-- ALTER COMMAND
alter table bill add Bill_Status varchar (15);
-- UPDATION
update Bill set Bill_Status = 'Paid' where Patient_ID = 1;
update Bill set Bill_Status = 'Unpaid' where Patient_ID = 2;
update Bill set Bill_Status = 'Paid' where Patient_ID = 3;
update Bill set Bill_Status = 'Paid' where Patient_ID = 4;

-- selection with different operators
-- display doctors with salary between 1000000 to 1500000
select *from doctor where salary between 1000000 and 1500000;

-- display all patients with names starting with A
select *from patient where P_Name like ("A%");

-- display total bill of patients from highest to lowest
```

```
select P_Name, sum(bill.Amount + pharmacy.Amount ) as Totalbill from patient join bill on
patient.Patient_Id = bill.Patient_Id
join pharmacy on patient.Medbill_No= pharmacy.Medbill_No group by patient.Patient_Id
order by Totalbill desc;

-- Add new column
alter table admin1 add column attendance varchar(20);
update admin1 set attendance="Present" where Admin_ID ="A1";

-- subquery
-- find the receptionist with salary greater than bill amount
select R_Name , salary from receptionist where Salary > any (select Amount from bill);

--  Aggregate functions (AVG, COUNT, MIN, MAX, SUM)
-- sample queries
-- 1 display the total amount each patient spends in hospital
-- sum of salaries of all doctors
select sum(doctor.salary) as totalSalaries from doctor;

-- display total amount each patient spent in the hospital
select P_Name, sum(bill.Amount + pharmacy.Amount ) as Totalbill from patient join bill on
patient.Patient_Id = bill.Patient_Id
join pharmacy on patient.Medbill_No= pharmacy.Medbill_No group by patient.Patient_Id ;

-- Find the pharmacy bill with the most amount
select Medbill_No ,max(pharmacy.Amount) as maxAmount from pharmacy;

-- Find the pharmacy bill with the least amount
select Medbill_No ,min(pharmacy.Amount) as minAmount from pharmacy;

-- find average salary a laboratory attendant takes
select  avg(Salary) as avgSalary from laboratory;
```

```
-- No.of patients visited hospital
select count(Patient_Id) as no_ofPatients from patient;

-- Joins

-- display patientname with the doctor he has been appointed to and his treatment details
select pat.P_Name , Dr.F_name,  Dr.L_name , tr.Diagnosis, tr.prescription from patient as pat
join treatment as tr on pat.Patient_Id = tr.Patient_Id join doctor as Dr on Dr.D_ID = tr.D_ID;

-- display the entities names that are being checked by admin
select lab.Attendant_name, n.N_Name, r.room_no, ph.medbill_no, ad.Admin_Name from admin1 as ad
join laboratory as lab on ad.Admin_ID = lab.Admin_ID
join room as r on ad.Admin_ID = r.Admin_ID
join nurse as n on ad.Admin_ID = n.Admin_ID
join pharmacy as ph on ad.Admin_ID = ph.Admin_ID ;

-- display the rooms of every department
select r.room_no, r.no_of_beds, dep.dept_name from department as dep
left join room as r on dep.Dept_ID = r.Dept_ID;

-- Views
-- Display the doctors with their specializations
-- Display the doctors with their specializations
create view doctor_spec as
SELECT doctor.F_name, doctor.L_name, specialization.spec_of_doctor
FROM doctor
JOIN specialization ON doctor.D_ID = specialization.D_ID;
select * from doctor_spec;

-- Procedures
-- search for details of a doctor by Id
```

```
Delimiter !!
create  procedure DoctorDetails (IN p_DID varchar(10))
begin
select *from doctor dr join dr_phoneno d_ph on dr.D_ID = d_ph.D_ID join specialization sp on
 dr.D_ID = sp.D_ID
 where dr.D_ID = p_DID;
 end!!
 Delimiter ;
 drop procedure DoctorDetails;
 call DoctorDetails ("D3");

 -- update condition of a patient
 Delimiter !!
 drop procedure if exists updateP_Condition;     /*It will replace the old procedure with the newly created one*/
 create procedure updateP_Condition(IN p_P_Condition varchar(20), p_PatientID varchar(10))
 begin
 update patient
 set patient_condition =  p_P_Condition
 where Patient_ID = p_PatientID;
 end !!
 delimiter ;
 call updateP_Condition("mild","P1");

 -- Count number of nurses serving in the hospital
 delimiter !!
create procedure CountNurses(out No_ofNurses int )
begin
select count(Nurse_ID) into No_ofNurses from nurse;
end !!
delimiter ;
call CountNurses(@No_ofNurses) ;
select @No_ofNurses as No_ofNurses;
```

```
-- search for how many phone numbers a patient has stored
delimiter //
create procedure patientphone(in id varchar(50))
begin
SELECT COUNT(p.phone_no) AS Number_of_Phone_Numbers
FROM P_phoneno as p
WHERE Patient_ID = id;
end //
delimiter ;
call  patientphone("P4");
drop procedure patientphone;
-- Add bonus to the salary of entities and add to new updated column entry
DELIMITER $$
CREATE PROCEDURE AddBonus(IN bonus_amount FLOAT)
BEGIN
    -- Add bonus to Doctor's salary
    UPDATE doctor
    SET Salary = Salary + bonus_amount;
    select salary as doctor_salary from doctor;

    -- Add bonus to Receptionist's salary
    UPDATE receptionist
    SET Salary = Salary + bonus_amount;
    select salary as Receptionist_salary from receptionist;

    -- Add bonus to Laboratory Attendant's salary
    UPDATE laboratory
    SET Salary = Salary + bonus_amount;
    select salary as Lab_Attendent_salary from laboratory;

    -- Add bonus to Admin's salary
    UPDATE admin1
    SET Salary = Salary + bonus_amount;
    select salary as admin_salary from admin1;

    -- Add bonus to Nurse's salary
    UPDATE nurse
    SET Salary = Salary + bonus_amount;
    select salary as nurse_salary from nurse;
END$$
DELIMITER ;

CALL AddBonus(10000);
```