show databases;

create database lab\_practical;

show tables;

use lab\_practical;

-- Question: A

-- 1. Create the necessary tables for the identified database objects from the ER Diagram of your business domain with necessary constraints (Primary Key, Unique, Not Null, Check, Default and Foreign Key).

create table Doctor(d\_id int primary key, d\_name varchar(50) not null, speciality varchar(30), charges int default(300));

desc Doctor;

select \* from Doctor;

-- drop table Patient;

-- drop table Appointment;

-- drop table Doctor;

create table Patient(p\_id int primary key, p\_name varchar(50) not null, age int, gender varchar(10), check(gender='Male' or gender = "Female"));

desc Patient;

create table Appointment(a\_id int primary key, d\_id int references Doctor.d\_id, p\_id int references Patient.p\_id);

desc Appointment;

-- 2. Demonstrate the enforcement of constraint with necessary inputs.

insert into Doctor values (101, "Dr. Shubam B.", "Neurologist", "500"),

(102, "Dr. Prati S.", "Cardiologist", "400"),

(103, "Dr. S. Swami", "Neutritionist", "200"),

(104, "Dr. S. Kumar", "Homiopathy", "200");

select \* from Doctor;

insert into Patient values (1, "S Katkam", 27, "Male"),

(2, "Shivani P", 47, "Female"),

(3, "Sharad Bhai k.", 40, "Male"),

(4, "K P Hudda", 19, "Male"),

(5, "M S Raj", 42, "Male"),

(6, "V V Varma", 22, "Male"),

(7, "Riya Jadhav", 16, "Female"),

(8, "Goutami H", 59, "Female");

select \* from Patient;

insert into Appointment values(1001, 101, 1),

(1002, 101, 2),

(1003, 102, 3),

(1004, 102, 4),

(1005, 103, 5),

(1006, 103, 6),

(1007, 104, 7),

(1008, 104, 8);

-- 3. Demonstrate the enforcement of constraint using alter table command.

-- alter table Patient set

-- 4. List the constraints enforced on each table.

desc Doctor;

desc Patient;

desc Appointment;

-- 5. Demonstrate dropping of any one constraint of an object.

alter table Patient drop column gender;

desc Patient;

-- 6. Demonstrate Drop and Truncate Commands onto the objects and display the records.

-- drop query:

drop table patient;

desc patient;

-- truncate query:

truncate table patient;

desc patient;

select \* from patient;

-- Question B

select \* from Doctor;

select \* from Patient;

select \* from Appointment;

-- The Operations should be performed on tables with logical sense and mention valid justification.

-- making some basic changes into appointment table

alter table Appointment add column is\_OPD\_appointment boolean;

desc Appointment;

update Appointment a set a.is\_OPD\_appointment=1 where a.d\_id=101;

update Appointment a set a.is\_OPD\_appointment=1 where a.d\_id=102;

update Appointment a set a.is\_OPD\_appointment=0 where a.is\_OPD\_appointment is null;

select \* from Appointment;

-- Query 1: write a query to find name, age and gender of the patients who has taken OPD.

select p.p\_name as Patient\_Name, p.age as Patient\_Age, p.gender as Patient\_Gender from Appointment a

inner join Patient p on p.p\_id=a.p\_id where a.is\_OPD\_appointment=1;

-- Query 2: Write a query to find patient name and age who has appointment with 'Neurologist' with doctors name and appointment id.

select a.a\_id as Appointment\_ID , p.p\_name as Patient\_Name, p.age as Patient\_Age, d.d\_name from Appointment a

inner join Patient p on p.p\_id=a.p\_id inner join Doctor d on d.d\_id=a.d\_id where d.speciality = "Neurologist";

-- Queries for performing other join except above used join

-- Cross Join

select \* from Appointment a cross join Doctor d;

-- Left Outer Join

select a.a\_id, p.p\_name from Appointment a left outer join Patient p on a.p\_id=p.p\_id;

-- Right outer Join

select a.a\_id, d.d\_name from Appointment a left outer join doctor d on a.d\_id=d.d\_id;

-- Self Join

select p.p\_id, p.p\_name, p1.age, p1.gender from Patient p join Patient p1 where p.p\_id = p1.p\_id;

-- C. Write at least 5 compound SQL statements to demonstrate at least any 5 of the following Aggregate functions on the tables created for your application

-- domain (AVG, SUM, COUNT, MIN, MAX, MEDIAN, CORR and STDDEV) and appropriately use GROUPBY and HAVING clause

-- Query 1: Write a query to find total no of the patients visited in a day in our Hospital

select count(p.p\_id) from Patient p;

-- Query 2: Write a query to find doctors charing minimun charges and maximum charges in ascending order from the patient

-- 1. min()

select d\_name from Doctor group by d\_id having min(charges);

-- 2. max()

select d\_name, charges from Doctor group by d\_id having max(charges);

-- Query 4: Write an sql query for finding average cost of visiting hospital

select avg(charges) as The\_average\_cost from Doctor;

-- Query 5: Write an query to find a name and speciality of the doctor who is charging maximun amount of money from a patient of a visit.

select d\_name, speciality, charges from doctor where charges in (select max(charges) as Charge from Doctor);