

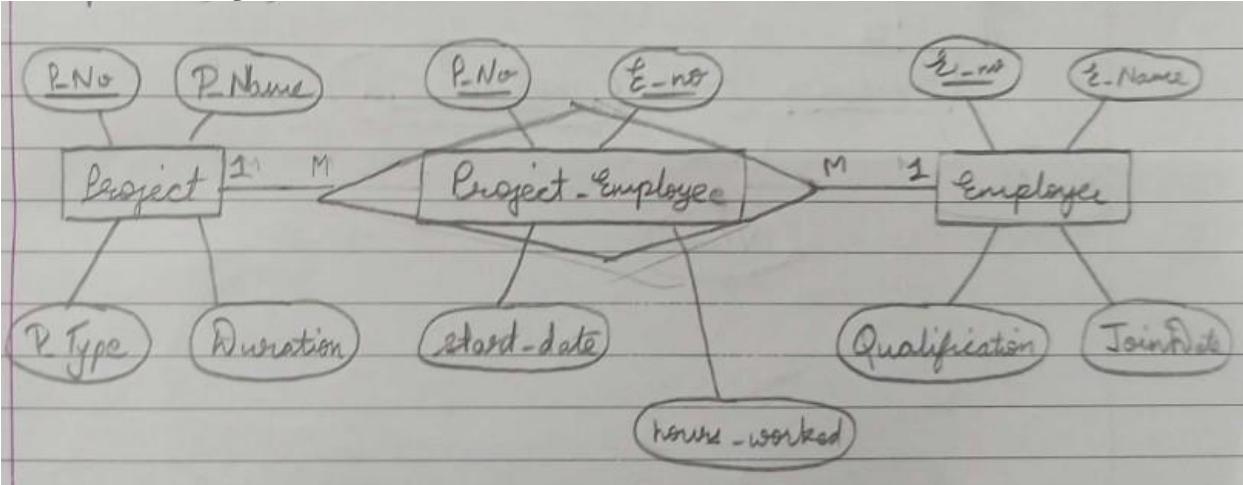
Q1) Practical Questions on PostgresSQL

Project(P_No, P_Name, P_Type, Duration) Employee(E_no, E_Name, Qualification, JoinDate)

Project and Employee: M-M relationship, with descriptive attributes as start_date(date), no_of_hours_worked(integer).

Assume appropriate data types for all the attributes.

a) Draw the ER diagram for above relational schema and normalize it in 3NF.



b) Create the above database in 3NF form in PostgresSQL using constraints.

```
CREATE TABLE Project (P_No SERIAL PRIMARY KEY, P_Name VARCHAR(100) NOT NULL, P_Type VARCHAR(50) NOT NULL, Duration INTEGER NOT NULL);
```

```
CREATE TABLE Employee (E_No SERIAL PRIMARY KEY, E_Name VARCHAR(100) NOT NULL, Qualification VARCHAR(100) NOT NULL, JoinDate DATE NOT NULL);
```

```
CREATE TABLE Project_Employee (P_No INTEGER REFERENCES Project(P_No), E_No INTEGER REFERENCES Employee(E_No), start_date DATE NOT NULL, no_of_hours_worked INTEGER NOT NULL, PRIMARY KEY (P_No, E_No));
```

```
INSERT INTO Project (P_Name, P_Type, Duration) VALUES ('Robotics', 'Research', 2), ('AI Chatbot', 'Development', 4), ('Data Analysis', 'Analysis', 3);
```

```
INSERT INTO Employee (E_Name, Qualification, JoinDate) VALUES ('John Doe', 'Computer Science', '2020-01-01'), ('Jane Smith', 'Electrical Engineering', '2019-05-15'), ('David Johnson', 'Data Science', '2021-03-10');
```

```
INSERT INTO Project_Employee (P_No, E_No, start_date, no_of_hours_worked) VALUES (1, 1, '2020-01-01', 8), (2, 2, '2019-05-15', 12), (3, 3, '2021-03-10', 10), (2, 3, '2021-03-10', 6);
```

Q2) Using above database, solve the following queries:

a) Find the employee numbers of the employees, who do not work on project “Robotics”.

```
SELECT DISTINCT E_No FROM Employee WHERE E_No NOT IN (SELECT E_No FROM Project_Employee WHERE P_No = (SELECT P_No FROM Project WHERE P_Name = 'Robotics'));
```

b) Find the names of the employees whose duration is more than three years.

```
SELECT E_Name FROM Employee WHERE EXTRACT(YEAR FROM NOW()) - EXTRACT(YEAR FROM JoinDate) > 3;
```

c) List the names of employees who is worked for more than 10 hrs on at least one project

```
SELECT DISTINCT E.E_Name FROM Employee E JOIN Project_Employee PE ON E.E_No = PE.E_No WHERE PE.no_of_hours_worked > 10;
```

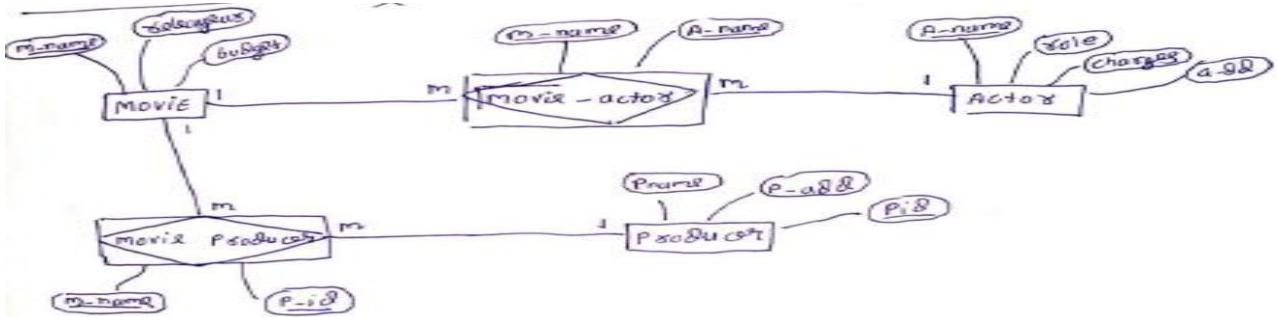
d) Delete the details of the employees starting with ‘S’.

```
DELETE FROM Employee WHERE E_Name LIKE 'S%';
```

Q1) Practical Questions on PostgreSQL

Movie (M_Name, release_year, budget) Actor (A_name, role, charges, a_address) Producer (producer_id, Name, P_address)

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Movie (M_Name VARCHAR(100) PRIMARY KEY, release_year INTEGER NOT NULL, budget NUMERIC NOT NULL);
```

```
CREATE TABLE Actor (A_name VARCHAR(100) PRIMARY KEY, role VARCHAR(100) NOT NULL, charges NUMERIC NOT NULL, a_address VARCHAR(100) NOT NULL);
```

```
CREATE TABLE Producer (producer_id SERIAL PRIMARY KEY, Name VARCHAR(100) NOT NULL, P_address VARCHAR(100) NOT NULL);
```

```
CREATE TABLE Movie_Actor (M_Name VARCHAR(100) REFERENCES Movie, A_name VARCHAR(100) REFERENCES Actor, role VARCHAR(100), PRIMARY KEY (M_Name, A_name, role));
```

```
CREATE TABLE Movie_Producer (M_Name VARCHAR(100) REFERENCES Movie, producer_id INTEGER REFERENCES Producer, PRIMARY KEY (M_Name, producer_id));
```

```
INSERT INTO Movie (M_Name, release_year, budget) VALUES ('Bahubali', 2015, 100000000), ('Dangal', 2016, 80000000), ('Tanhaji', 2020, 120000000);
```

```
INSERT INTO Actor (A_name, role, charges, a_address) VALUES ('Rajkumar Rao', 'Hero', 5000000, 'Mumbai'), ('Deepika Padukone', 'Heroine', 8000000, 'Bengaluru'), ('Ranveer Singh', 'Hero', 6000000, 'Mumbai');
```

```
INSERT INTO Producer (Name, P_address) VALUES ('Karan Johar', 'Mumbai'), ('Sajid Nadiadwala', 'Mumbai'), ('Ajay Devgn', 'Pune');
```

```
INSERT INTO Movie_Actor (M_Name, A_name, role) VALUES ('Bahubali', 'Rajkumar Rao', 'Hero'), ('Dangal', 'Deepika Padukone', 'Heroine'), ('Tanhaji', 'Ranveer Singh', 'Hero');
```

```
INSERT INTO Movie_Producer (M_Name, producer_id) VALUES ('Bahubali', 1), ('Dangal', 2), ('Tanhaji', 3);
```

Q2) Using above database, solve the following queries:

- a) List the names of movies with the highest budget.

```
SELECT M_Name FROM Movie WHERE budget = (SELECT MAX(budget) FROM Movie);
```

- b) List the names of producers who produce the same movie as “Tanhaji”.

```
SELECT p.Name FROM Producer p JOIN Movie_Producer mp ON p.producer_id = mp.producer_id WHERE mp.M_Name = 'Tanhaji';
```

- c) List the names of actors who do not live in Pune or Mumbai city.

```
SELECT DISTINCT A_name FROM Actor WHERE a_address NOT IN ('Pune', 'Mumbai');
```

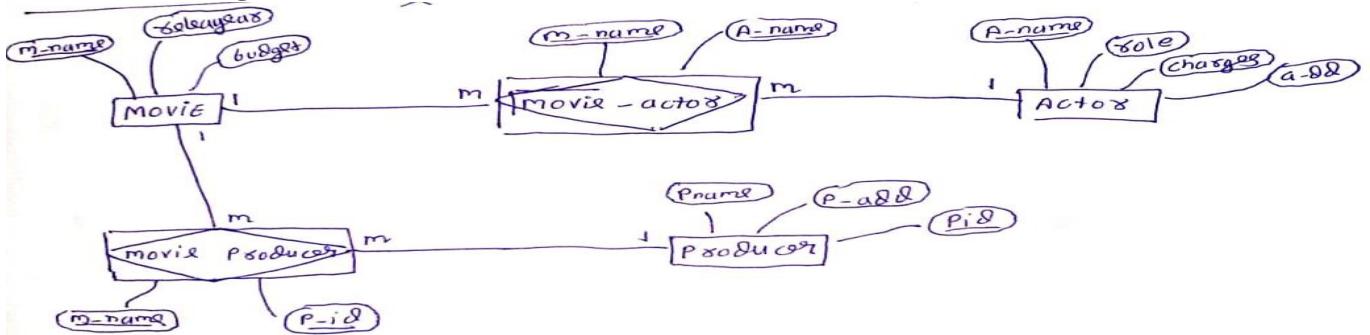
- d) Delete the information of Producers who are living in Pune city.

```
DELETE FROM Movie_Producer WHERE producer_id = 3;
```

```
DELETE FROM Producer WHERE P_address = 'Pune';
```

Q1) Practical Questions on PostgresSQL Movie (M_Name,release_year,budget) Actor (A_name, role, charges,a_address) Producer (producer_id, Name,P_address)

a) Draw the ER diagram for above relational schema and normalize it in 3NF.



b) Create the above database in 3NF form in PostgresSQL using constraints.

CREATE TABLE Movie (M_Name VARCHAR(100) PRIMARY KEY, release_year INTEGER NOT NULL, budget NUMERIC(10, 2) NOT NULL);

CREATE TABLE Actor (A_Name VARCHAR(100) NOT NULL, Role VARCHAR(100) NOT NULL, Charges NUMERIC(10, 2) NOT NULL, A_Address VARCHAR(255) NOT NULL, PRIMARY KEY (A_Name, Role));

CREATE TABLE Producer (Producer_ID SERIAL PRIMARY KEY, Name VARCHAR(100) NOT NULL, P_Address VARCHAR(255) NOT NULL);

CREATE TABLE Movie_Actor (M_Name VARCHAR(100) REFERENCES Movie(M_Name), A_Name VARCHAR(100) REFERENCES Actor(A_Name), Role VARCHAR(100), PRIMARY KEY (M_Name, A_Name, Role));

CREATE TABLE Movie_Producer (M_Name VARCHAR(100) REFERENCES Movie(M_Name), Producer_ID INTEGER REFERENCES Producer(Producer_ID), PRIMARY KEY (M_Name, Producer_ID));

INSERT INTO Movie (M_Name, release_year, budget) VALUES ('Tanhaji', 2020, 300000000), ('Avatar', 2009, 237000000), ('Inception', 2010, 160000000), ('Interstellar', 2014, 165000000);

INSERT INTO Actor (A_Name, Role, Charges, A_Address) VALUES ('Ajay Devgn', 'Tanhaji', 15000000, 'Mumbai'), ('Leonardo DiCaprio', 'Cobb', 25000000, 'Los Angeles'), ('Tom Hardy', 'Eames', 12000000, 'London'), ('Sigourney Weaver', 'Dr. Grace Augustine', 20000000, 'New York');

INSERT INTO Producer (Name, P_Address) VALUES ('Rohit Shetty', 'Mumbai'), ('Christopher Nolan', 'Los Angeles'), ('James Cameron', 'Los Angeles');

INSERT INTO Movie_Actor (M_Name, A_Name, Role) VALUES ('Tanhaji', 'Ajay Devgn', 'Tanhaji'), ('Avatar', 'Sigourney Weaver', 'Dr. Grace Augustine'), ('Inception', 'Leonardo DiCaprio', 'Cobb'), ('Interstellar', 'Matthew McConaughey', 'Cooper');

INSERT INTO Movie_Producer (M_Name, Producer_ID) VALUES ('Tanhaji', 1), ('Avatar', 3), ('Inception', 2), ('Interstellar', 2), ('Interstellar', 3);

Q2) Using above database, solve the following queries:

a) Gives count of movies whose budget is greater than 3 crores.

SELECT COUNT(*) FROM Movie WHERE budget > 30000000;

b) List details of actors who have acted in movie “Tanhaji”.

SELECT * FROM Actor WHERE A_Name IN (SELECT A_Name FROM Movie_Actor WHERE M_Name = 'Tanhaji');

c) Delete the details movie starting with ‘a’.

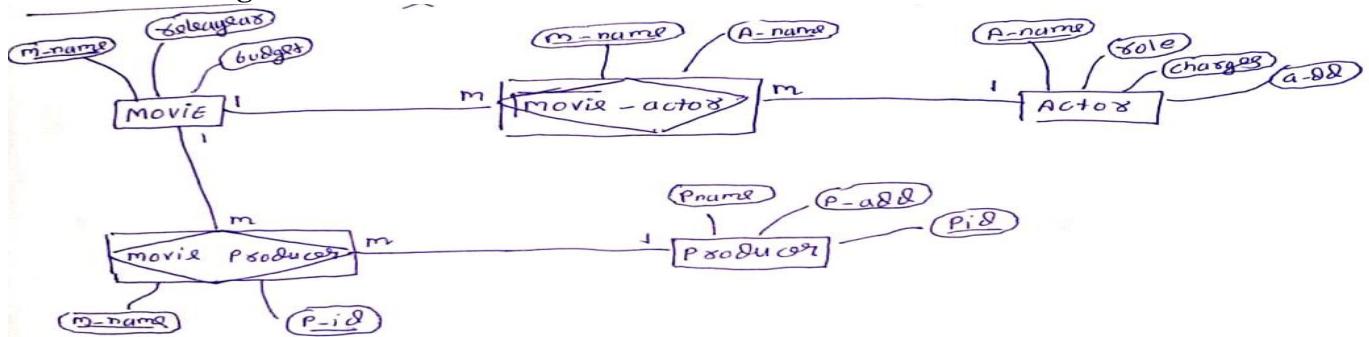
DELETE FROM Movie WHERE M_Name LIKE 'a%';

d) List the names of movies, produced by more than one producer.

SELECT M_Name FROM Movie GROUP BY M_Name HAVING COUNT(*) > 1;

Q1) Practical Questions on PostgresSQL Movie (M_Name, release_year, budget) Actor (A_name ,role, charges,a_address) Producer (producer_id, Name, P_address)

a) Draw the ER diagram for above relational schema and normalize it in 3NF.



b) Create the above database in 3NF form in PostgresSQL using constraints.

CREATE TABLE Movie (M_Name VARCHAR(100) PRIMARY KEY, release_year INTEGER NOT NULL, budget DECIMAL(10, 2) NOT NULL);

CREATE TABLE Actor (A_name VARCHAR(100) NOT NULL, role VARCHAR(100) NOT NULL, charges DECIMAL(10, 2) NOT NULL, a_address VARCHAR(255) NOT NULL, PRIMARY KEY (A_name, role));

CREATE TABLE Producer (producer_id SERIAL PRIMARY KEY, Name VARCHAR(100) NOT NULL, P_address VARCHAR(255) NOT NULL);

CREATE TABLE Movie_Actor (M_Name VARCHAR(100) REFERENCES Movie(M_Name), A_name VARCHAR(100) REFERENCES Actor(A_name), role VARCHAR(100) REFERENCES Actor(role), PRIMARY KEY (M_Name, A_name, role));

CREATE TABLE Movie_Producer (M_Name VARCHAR(100) REFERENCES Movie(M_Name), producer_id INTEGER REFERENCES Producer(producer_id), PRIMARY KEY (M_Name, producer_id));

INSERT INTO Movie (M_Name, release_year, budget) VALUES ('Spiderman', 2000, 5000000), ('Star Wars', 2000, 10000000), ('Avatar', 2009, 250000000);

INSERT INTO Actor (A_name, role, charges, a_address) VALUES ('Tom Holland', 'Spiderman', 1000000, 'Los Angeles'), ('Mark Hamill', 'Luke Skywalker', 500000, 'New York'), ('Zoe Saldana', 'Neytiri', 2000000, 'Los Angeles');

INSERT INTO Producer (Name, P_address) VALUES ('Kevin Feige', 'Los Angeles'), ('George Lucas', 'San Francisco'), ('James Cameron', 'Los Angeles');

INSERT INTO Movie_Actor (M_Name, A_name, role) VALUES ('Spiderman', 'Tom Holland', 'Spiderman'), ('Star Wars', 'Mark Hamill', 'Luke Skywalker'), ('Avatar', 'Zoe Saldana', 'Neytiri');

INSERT INTO Movie_Producer (M_Name, producer_id) VALUES ('Spiderman', 1), ('Star Wars', 2), ('Avatar', 3);

Q2) Using above database, solve the following queries:

a) List the movie name starting with 'S'.

SELECT M_Name FROM Movie WHERE M_Name LIKE 'S%';

b) List the names of actors who have acted in the maximum number of movies.

SELECT A_name FROM Actor WHERE (A_name, role) IN (SELECT A_name, role FROM Movie_Actor GROUP BY A_name, role HAVING COUNT(*) = (SELECT MAX(count) FROM (SELECT COUNT(*) as count FROM Movie_Actor GROUP BY A_name, role) AS subquery));

c) Count all the movies names released in the year 2000.

SELECT COUNT(*) FROM Movie WHERE release_year = 2000;

d) Display all movies of "Ranveer Singh"

SELECT M_Name FROM Movie WHERE M_Name IN (SELECT M_Name FROM Movie_Actor WHERE A_name = 'Ranveer Singh');

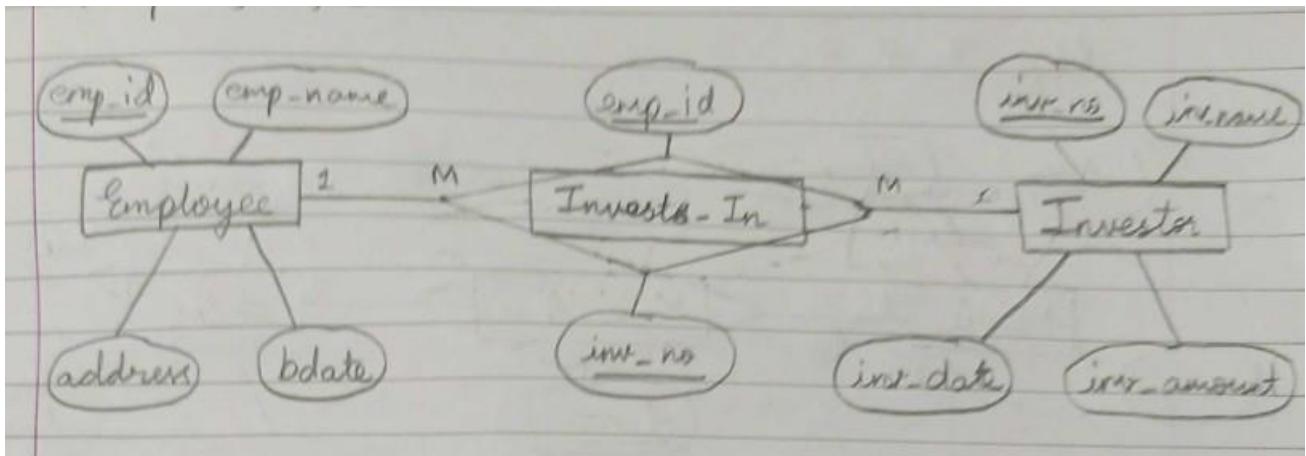
Q1) Practical Questions on PostgresSQL

Employee (emp_id, emp_name, address, bdate)
Investor (inv_no, inv_name, inv_date, inv_amt)

An employee may invest in one or more investments, hence he can be an investor. But an investor need not be an employee of the firm.

Assume appropriate data types for all the attributes.

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgresSQL using constraints.

```
CREATE TABLE Employee (emp_id SERIAL PRIMARY KEY, emp_name VARCHAR(100) NOT NULL, address VARCHAR(255), bdate DATE);
```

```
CREATE TABLE Investor (inv_no SERIAL PRIMARY KEY, inv_name VARCHAR(100) NOT NULL, inv_date DATE, inv_amt NUMERIC(10, 2));
```

```
CREATE TABLE Employee_Investor (emp_id INTEGER REFERENCES Employee(emp_id), inv_no INTEGER REFERENCES Investor(inv_no), PRIMARY KEY (emp_id, inv_no));
```

```
INSERT INTO Employee (emp_name, address, bdate) VALUES ('John Doe', '123 Main St, New York', '1990-01-15'), ('Sarah Smith', '456 Oak St, Los Angeles', '1985-05-20'), ('Michael Johnson', '789 Pine St, Chicago', '1982-08-10');
```

```
INSERT INTO Investor (inv_name, inv_date, inv_amt) VALUES ('Investor 1', '2023-01-01', 8000), ('Investor 2', '2023-02-15', 12000), ('Investor 3', '2023-03-20', 5000), ('Investor 4', '2023-04-10', 15000);
```

```
INSERT INTO Employee_Investor (emp_id, inv_no) VALUES (1, 1), (1, 2), (2, 3), (3, 4);
```

Q2) Using above database, solve the following queries:

- a) List the names of customers who are either employees, or investors or both.

```
SELECT emp_name FROM Employee UNION SELECT inv_name FROM Investor UNION SELECT emp_name FROM Employee WHERE emp_id IN (SELECT emp_id FROM Employee_Investor);
```

- b) Delete the employee whose names starts with 'S'

```
DELETE FROM Employee WHERE emp_name LIKE 'S%';
```

- c) List the employees who are living in Mumbai city.

```
SELECT emp_name FROM Employee WHERE address LIKE '%Mumbai%';
```

- d) Update amount to Rs. 50000 of investors having amount less than Rs 10,000.

```
UPDATE Investor SET inv_amt = 50000 WHERE inv_amt < 10000;
```

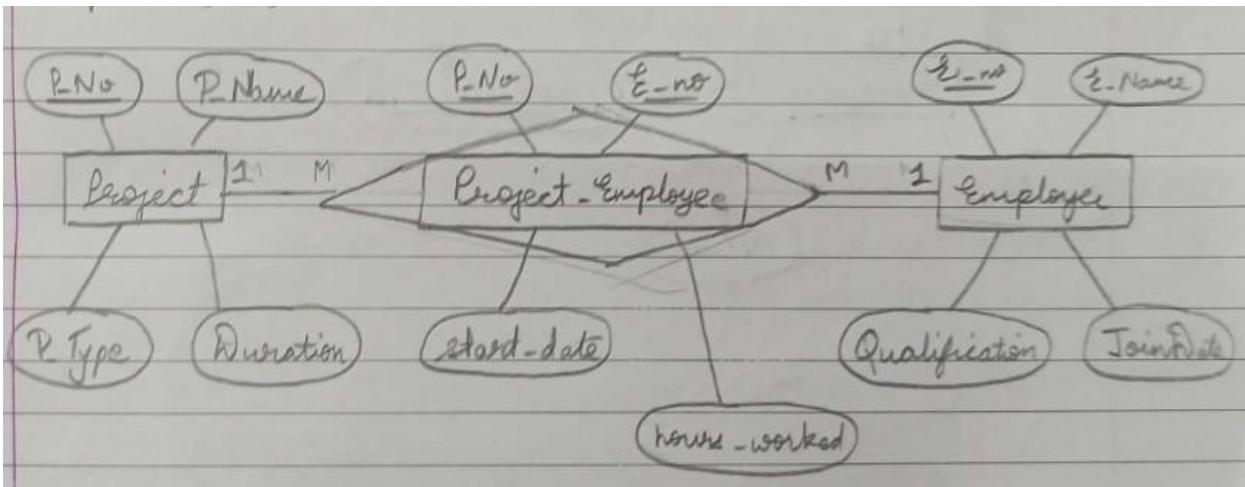
Q1) Practical Questions on PostgresSQL

Project (P_No, P_Name, P_Type, Duration) Employee
(E_no, E_Name, Qualification, JoinDate)

Project and Employee: M-M relationship, with descriptive attributes as start_date(date), no_of_hours_worked(integer).

Assume appropriate data types for all the attributes.

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgresSQL using constraints.

```
CREATE TABLE Project (P_No SERIAL PRIMARY KEY, P_Name VARCHAR(100) NOT NULL, P_Type VARCHAR(50), Duration INTEGER);
```

```
CREATE TABLE Employee (E_No SERIAL PRIMARY KEY, E_Name VARCHAR(100) NOT NULL, Qualification VARCHAR(100), JoinDate DATE);
```

```
CREATE TABLE Project_Employee (Project_No INTEGER REFERENCES Project(P_No), Employee_No INTEGER REFERENCES Employee(E_No), Start_Date DATE, No_of_Hours_Worked INTEGER, PRIMARY KEY (Project_No, Employee_No));
```

```
INSERT INTO Project (P_Name, P_Type, Duration) VALUES ('System', 'Type A', 3), ('Database', 'Type B', 6), ('Web App', 'Type C', 4);
```

```
INSERT INTO Employee (E_Name, Qualification, JoinDate) VALUES ('Alice', 'Bachelor', '2019-01-01'), ('Bob', 'Master', '2020-03-15'), ('Charlie', 'PhD', '2018-05-20');
```

```
INSERT INTO Project_Employee (Project_No, Employee_No, Start_Date, No_of_Hours_Worked) VALUES (1, 1, '2019-01-01', 40), (1, 2, '2020-03-15', 35), (2, 3, '2018-05-20', 45), (3, 1, '2019-01-01', 30);
```

Q2) Using above database, solve the following queries:

- a) Find the employees whose name starts with 'A'.

```
SELECT * FROM Employee WHERE E_Name LIKE 'A%';
```

- b) Find the details of employees working on the project "System".

```
SELECT E.* FROM Employee E JOIN Project_Employee PE ON E.E_No = PE.Employee_No JOIN Project P ON P.P_No = PE.Project_No WHERE P.P_Name = 'System';
```

- c) List the names of the first three employees in alphabetical order.

```
SELECT E_Name FROM Employee ORDER BY E_Name LIMIT 3;
```

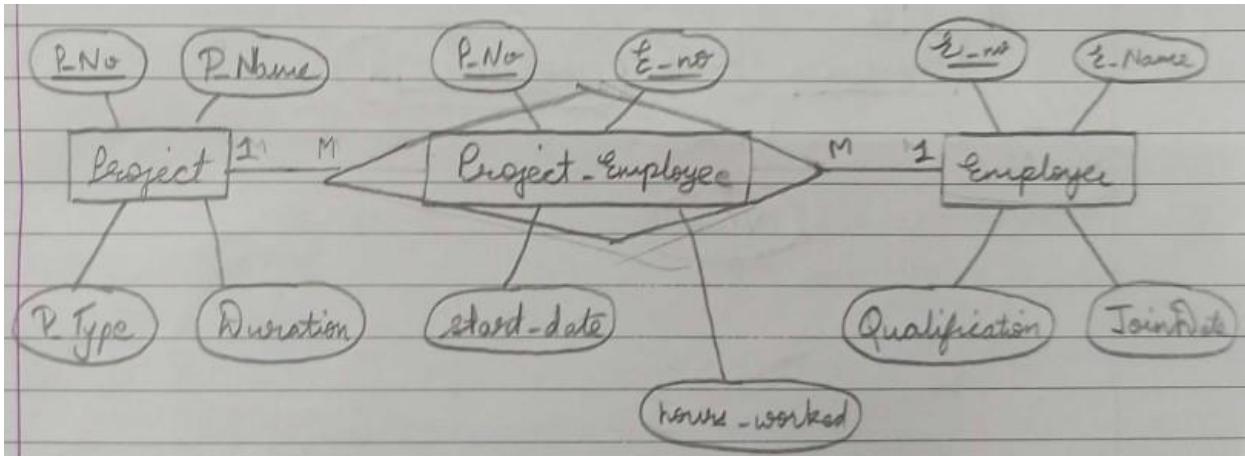
- d) Find the names of the employees whose duration is more than five years.

```
SELECT E.* FROM Employee E JOIN Project_Employee PE ON E.E_No = PE.Employee_No JOIN Project P ON P.P_No = PE.Project_No WHERE P.Duration > 5;
```

Q1) Practical Questions on PostgreSQL

Project (P_No, P_Name, P_Type, Duration) Employee (E_no, E_Name, Qualification, JoinDate) Project and Employee: M-M relationship, with descriptive attributes as start_date (date), no_of_hours_worked (integer). Assume appropriate data types for all the attributes.

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Project (P_No SERIAL PRIMARY KEY, P_Name VARCHAR(100), P_Type VARCHAR(50), Duration INTEGER);
```

```
CREATE TABLE Employee (E_No SERIAL PRIMARY KEY, E_Name VARCHAR(100), Qualification VARCHAR(50), JoinDate DATE);
```

```
CREATE TABLE Project_Employee (P_No INTEGER REFERENCES Project(P_No), E_No INTEGER REFERENCES Employee(E_No), Start_Date DATE, No_of_Hours_Worked INTEGER, PRIMARY KEY (P_No, E_No));
```

```
INSERT INTO Project (P_Name, P_Type, Duration) VALUES ('Robotics', 'Research', 5), ('AI Chatbot', 'Development', 3), ('Smart Home Automation', 'Integration', 7);
```

```
INSERT INTO Employee (E_Name, Qualification, JoinDate) VALUES ('John Doe', 'M.Sc', '2010-05-15'), ('Jane Smith', 'B.E', '2015-08-20'), ('Alice Johnson', 'Ph.D', '2012-03-10');
```

```
INSERT INTO Project_Employee (P_No, E_No, Start_Date, No_of_Hours_Worked) VALUES (1, 1, '2020-01-01', 50), (1, 2, '2021-02-01', 45), (2, 3, '2019-03-01', 60), (3, 1, '2018-06-01', 30), (3, 3, '2020-10-01', 55);
```

Q2) Using above database, solve the following queries:

- Find the names of the employees whose duration is more than 10 years
`SELECT E_Name FROM Employee WHERE EXTRACT(YEAR FROM AGE(JoinDate)) > 10;`
- Find the details of employees working on the project “Robotics”.
`SELECT Employee.* FROM Employee INNER JOIN Project_Employee ON Employee.E_No = Project_Employee.E_No INNER JOIN Project ON Project_Employee.P_No = Project.P_No WHERE Project.P_Name = 'Robotics';`
- List the names of the employees in alphabetical order.
`SELECT E_Name FROM Employee ORDER BY E_Name ASC;`
- Delete the information of employee whose qualification is B.E.
`DELETE FROM project_employee WHERE e_no = 2;`
`DELETE FROM Employee WHERE Qualification = 'B.E';`

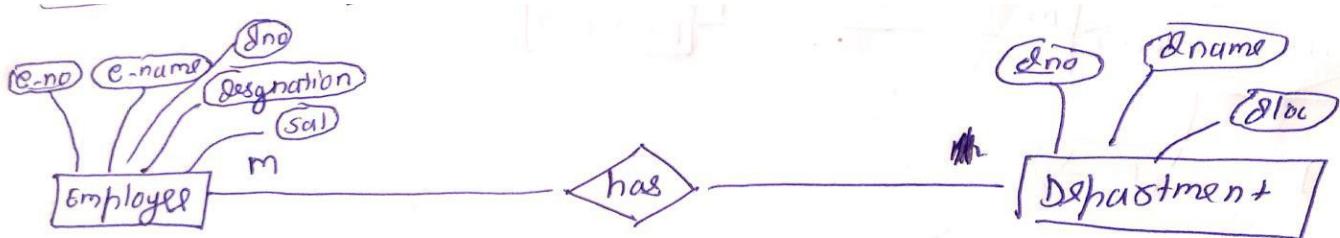
Q.1) Practical Questions on PostgreSQL Consider the following database

Employee (eno, ename, designation, sal)

Department
(dno, dname, dloc)

There exists a one-to-many relationship between department and employee. Create the Relations accordingly, so that the relationship is handled properly and the relations are in normalized form(3NF).

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Department (dno INT PRIMARY KEY, dname VARCHAR(50) NOT NULL, dloc VARCHAR(50) NOT NULL);
```

```
CREATE TABLE Employee (eno INT PRIMARY KEY, ename VARCHAR(50) NOT NULL, designation VARCHAR(50) NOT NULL, sal DECIMAL(10,2), dno INT, FOREIGN KEY (dno) REFERENCES Department(dno));
```

```
INSERT INTO Department (dno, dname, dloc) VALUES (101, 'Marketing', 'Mumbai'), (102, 'Sales', 'Delhi'), (103, 'Finance', 'Chennai'), (104, 'IT', 'Pune');
```

```
INSERT INTO Employee (eno, ename, designation, sal, dno) VALUES (2001, 'Alice Smith', 'Manager', 35000.00, 101), (2002, 'Bob Brown', 'Sales Representative', 22000.00, 102), (2003, 'Charlie Chen', 'Accountant', 28000.00, 103), (2004, 'David Davis', 'H.O.D.', 50000.00, 101), (2005, 'Emily Evans', 'Marketing Associate', 18000.00, 101), (2006, 'Rita Sharma', 'Clerk', 15000.00, 104);
```

Q2.) Using above database, solve the following queries:

- a) Find the employee details whose name starts with R.

```
SELECT * FROM Employee WHERE ename LIKE 'R%';
```

- b) Find sum of salary of employees department wise

```
SELECT D.dname, SUM(E.sal) AS total_salary FROM Department D INNER JOIN Employee E ON E.dno = D.dno GROUP BY D.dname;
```

- c) Delete employee details who are working as designation 'clerk' located in 'Pune'

```
DELETE FROM Employee E USING Department D WHERE E.designation = 'clerk' AND E.dno = D.dno AND D.dloc = 'Pune';
```

- d) List employees with department details whose salary is in between 20000 to 30000.

```
SELECT E.eno, E.ename, E.designation, E.sal, D.dname, D.dloc FROM Employee E INNER JOIN Department D ON E.dno = D.dno WHERE E.sal BETWEEN 20000 AND 30000;
```

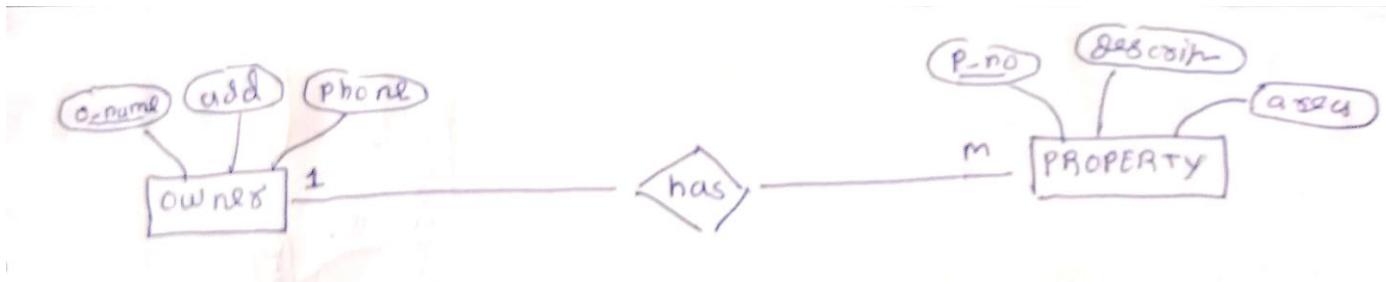
Q.1) Practical Questions on PostgreSQL Consider the following database

Property (pno, description, area)

Owner (oname, address, phone)

An owner can have one or more properties, but a property belongs to exactly one owner. Create the relations accordingly, so that the relationship is handled properly and the relations are in normalized form (3NF).

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Owner (oname VARCHAR(50) PRIMARY KEY, address VARCHAR(255) NOT NULL, phone VARCHAR(20));
```

```
CREATE TABLE Property (pno INT PRIMARY KEY, description VARCHAR(255), area VARCHAR(50) NOT NULL, oname VARCHAR(50) NOT NULL, FOREIGN KEY (oname) REFERENCES Owner(oname));
```

```
INSERT INTO Owner (oname, address, phone) VALUES ('Mr. Kadam', '123 Main St, Pune', '1234567890'), ('Ms. Patel', '456 Hill Rd, Mumbai', '9876543210'), ('Mr. Patil', '789 Park Ave, Delhi', '0123456789');
```

```
INSERT INTO Property (pno, description, area, oname) VALUES (101, 'Spacious apartment', 'Moshi', 'Mr. Kadam'), (102, 'Cozy bungalow', 'Hinjewadi', 'Mr. Kadam'), (103, 'Modern studio', 'Bandra', 'Ms. Patel'), (104, 'Luxurious villa', 'Juhu', 'Ms. Patel');
```

Q2.) Using above database, solve the following queries:

- a) List details of property where area is 'Moshi'

```
SELECT * FROM Property WHERE area = 'Moshi';
```

- b) List area wise owner property details.

```
SELECT P.area, O.oname, O.address, O.phone, P.description, P.pno FROM Property P INNER JOIN Owner O ON P.oname = O.oname ORDER BY P.area;
```

- c) List property details owned by 'Mr. Kadam'

```
SELECT * FROM Property P INNER JOIN Owner O ON P.oname = O.oname WHERE O.oname = 'Mr. Kadam';
```

- d) Update phone no of 'Mr. Patil' to 987842621

```
SELECT * FROM Property P INNER JOIN Owner O ON P.oname = O.oname WHERE O.oname = 'Mr. Kadam';
UPDATE Owner SET phone = '987842621' WHERE oname = 'Mr. Patil';
```

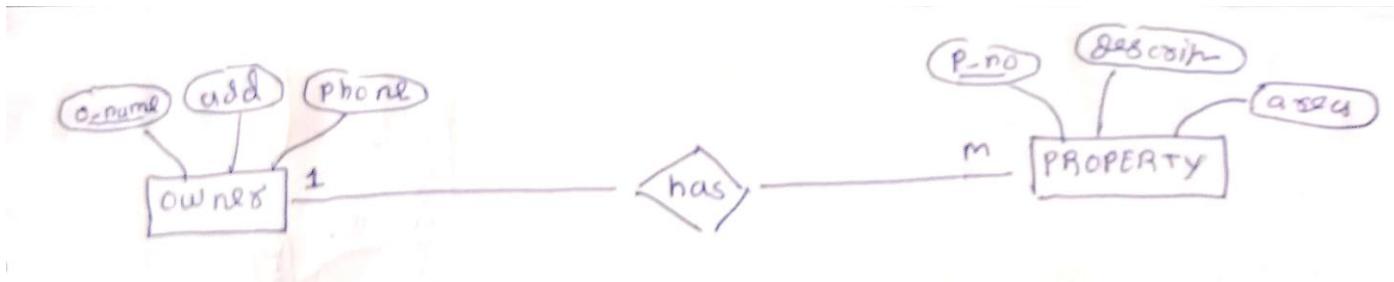
Q.1) Practical Questions on PostgreSQL Consider the following database

Property (pno, description, area)

Owner (oname, address, phone)

An owner can have one or more properties, but a property belongs to exactly one owner. Create the relations accordingly, so that the relationship is handled properly and the relations are in normalized form (3NF).

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Owner (oname VARCHAR(50) PRIMARY KEY, address VARCHAR(255) NOT NULL, phone VARCHAR(20), age INT);
```

```
CREATE TABLE Property (pno INT PRIMARY KEY, description VARCHAR(255), area VARCHAR(50) NOT NULL, oname VARCHAR(50) NOT NULL, FOREIGN KEY (oname) REFERENCES Owner(oname));
```

```
INSERT INTO Owner (oname, address, phone, age) VALUES ('Mr. Kadam', '123 Main St, Pune', '1234567890', 45), ('Ms. Patel', '456 Hill Rd, Mumbai', '9876543210', 32), ('Mr. Patil', '789 Park Ave, Delhi', '0123456789', 50);
```

```
INSERT INTO Property (pno, description, area, oname) VALUES (101, 'Spacious apartment', 'Moshi', 'Mr. Kadam'), (102, 'Cozy bungalow', 'Hinjewadi', 'Mr. Kadam'), (103, 'Modern studio', 'Bandra', 'Ms. Patel'), (104, 'Luxurious villa', 'Juhu', 'Ms. Patel');
```

Q2.) Using above database, solve the following queries:

- a) List details of property whose description 'Banglow'

```
SELECT * FROM Property WHERE description = 'Banglow';
```

- b) List properties details with owner name in 'Chinchwad' area.

```
SELECT P.pno, P.description, P.area, O.oname, O.address, O.phone FROM Property P INNER JOIN Owner O ON P.oname = O.oname WHERE P.area = 'Chinchwad';
```

- c) Alter table owner to add owner Age attribute.

```
ALTER TABLE Owner ADD COLUMN age INT;
```

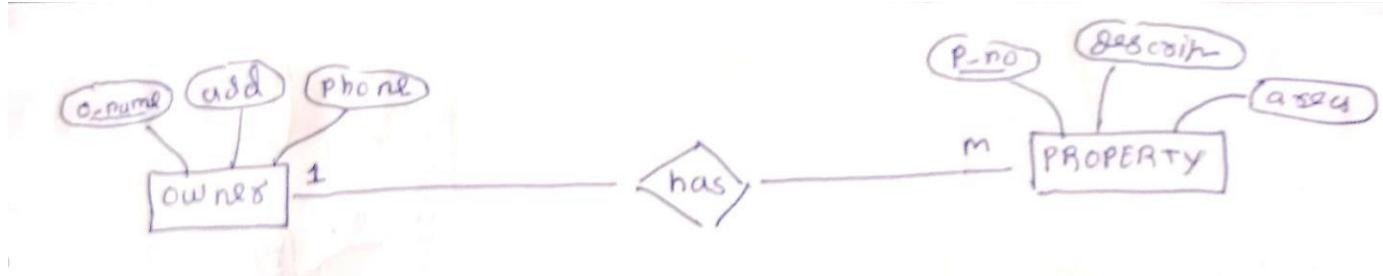
- d) Delete properties of 'Mr.Patil' located in 'Shivajinagar' area.

```
DELETE FROM Property WHERE oname = 'Mr. Patil' AND area = 'Shivajinagar';
```

Q.1) Practical Questions on PostgreSQL Consider the following database

Property (pno, description, area) Owner (oname, address, phone) An owner can have one or more properties, but a property belongs to exactly one owner. Create the relations accordingly, so that the relationship is handled properly and the relations are in normalized form (3NF).

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Owner (oname VARCHAR(50) PRIMARY KEY, address VARCHAR(255) NOT NULL, phone VARCHAR(20));
```

```
CREATE TABLE Property (pno INT PRIMARY KEY, description VARCHAR(255), area VARCHAR(50) NOT NULL, oname VARCHAR(50) NOT NULL, FOREIGN KEY (oname) REFERENCES Owner(oname));
```

```
INSERT INTO Owner (oname, address, phone) VALUES ('Mr. Khan', '123 Main Street', '123-456-7890'), ('Mr. Patil', '456 Elm Street', '987-654-3210'), ('Ms. Singh', '789 Oak Street', '456-789-0123');
```

```
INSERT INTO Property (pno, description, area, oname) VALUES (1, '2BHK Apartment', '1000 sqft', 'Mr. Khan'), (2, 'Villa', '2000 sqft', 'Mr. Patil'), (3, 'Townhouse', '1500 sqft', 'Ms. Singh'), (4, 'Penthouse', '1800 sqft', 'Mr. Khan');
```

Q2.) Using above database, solve the following queries:

- a) List details of property whose owner name is 'Mr. Khan'.

```
SELECT * FROM Property WHERE oname = 'Mr. Khan';
```

- b) Find areawise property details with owner name.

```
SELECT oname, description, area FROM Property ORDER BY area;
```

- c) Count owner wise details of property.

```
SELECT oname, COUNT(*) AS num_properties FROM Property GROUP BY oname;
```

- d) Update owner 'Mr. Patil' property description to '3BHK'.

```
UPDATE Property SET description = '3BHK' WHERE oname = 'Mr. Patil';
```

Q.1) Practical Questions on PostgreSQL

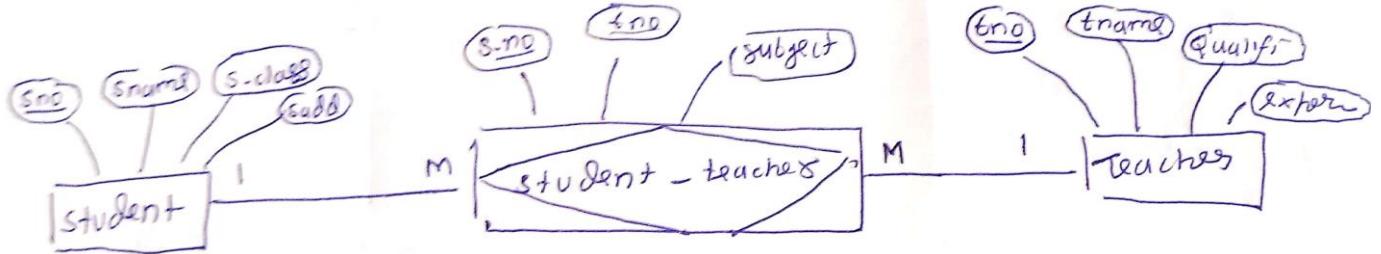
Consider the following database

Student (sno, s_name, s_class, s_addr)

Teacher (tno, t_name, qualification, experience)

Student-Teacher: M-M with descriptive attributes Subject.

a) Draw the ER diagram for above relational schema and normalize it in 3NF.



b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Student (sno SERIAL PRIMARY KEY, s_name VARCHAR(50), s_class VARCHAR(20), s_addr VARCHAR(100));
```

```
CREATE TABLE Teacher (tno SERIAL PRIMARY KEY, t_name VARCHAR(50), qualification VARCHAR(50), experience INT);
```

```
CREATE TABLE Student_Teacher (sno INT REFERENCES Student(sno), tno INT REFERENCES Teacher(tno), Subject VARCHAR(50), PRIMARY KEY (sno, tno));
```

```
INSERT INTO Student (s_name, s_class, s_addr) VALUES ('Rahul', 'F.Y.B.C.A', 'Address 1, Pune'), ('Priya', 'S.Y.B.Sc', 'Address 2, Mumbai'), ('Amit', 'T.Y.B.Com', 'Address 3, Delhi'), ('Riya', 'F.Y.B.A', 'Address 4, Bangalore'), ('Suresh', 'S.Y.B.Tech', 'Address 5, Chennai');
```

```
INSERT INTO Teacher (t_name, qualification, experience) VALUES ('Mr. Kumar', 'M.Tech', 5), ('Mrs. Patel', 'M.Sc', 8), ('Dr. Sharma', 'Ph.D', 12), ('Ms. Gupta', 'M.A', 6), ('Mr. Singh', 'B.Ed', 3);
```

```
INSERT INTO Student_Teacher (sno, tno, Subject) VALUES (1, 1, 'DBMS'), (2, 2, 'Physics'), (3, 3, 'Maths'), (4, 4, 'English'), (5, 5, 'History');
```

Q2.) Using above database, solve the following queries

a) List teacher details who taught subject 'DBMS'

```
SELECT * FROM Teacher WHERE tno IN (SELECT tno FROM Student_Teacher WHERE Subject = 'DBMS');
```

b) Find student details with teacher name who taught class 'F.Y.B.C.A'

```
SELECT s.*, t.t_name FROM Student s JOIN Student_Teacher st ON s.sno = st.sno JOIN Teacher t ON st.tno = t.tno WHERE s.s_class = 'F.Y.B.C.A';
```

c) Find the maximum experience details of teacher.

```
SELECT * FROM Teacher ORDER BY experience DESC LIMIT 1;
```

d) Update teacher qualification to 'Ph.D' of 'Mr.Kumar'.

```
UPDATE Teacher SET qualification = 'Ph.D' WHERE t_name = 'Mr. Kumar';
```

Q.1) Practical Questions on PostgreSQL

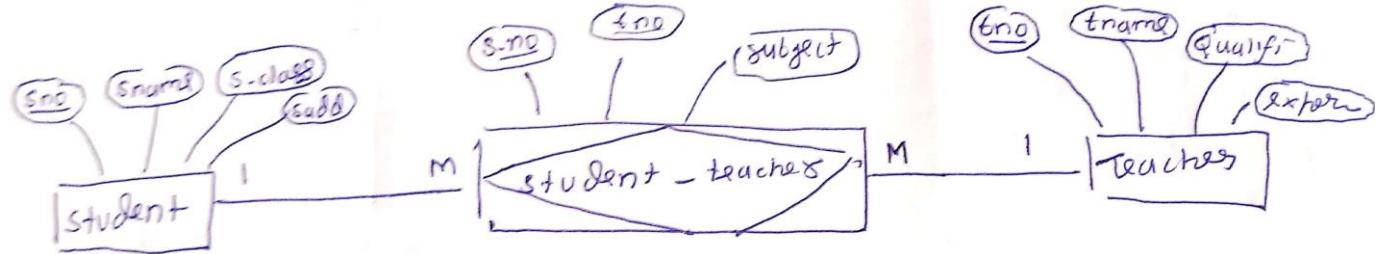
Consider the following database

Student (sno, s_name, s_class, s_addr)

Teacher (tno, t_name, qualification, experience)

Student-Teacher: M-M with descriptive attributes Subject.

a) Draw the ER diagram for above relational schema and normalize it in 3NF.



b) Create the above database in 3NF form in PostgreSQL using constraints.

```
create table stud(sno int primary key,sname varchar(20),sclass text,addr text);
```

```
create table teacher(tno int primary key,tname varchar(20),qualification text, experience int);
```

```
create table stte(sno int,tno int,subject text,primary key(sno,tno),foreign key(sno)references stud(sno),foreign key(tno)references teacher(tno));
```

```
insert into stud values(1,'Kunal','FYBCA','Beed'),(2,'Hrushi','SYBCA','Alandi'),(3,'Tushar','TYBCA','Pune'),(4,'Aditya','FYBCA','Pimpri'),(5,'Sujal','SYBCA','Nashik');
```

```
insert into teacher values(11,'Mr.Rajesh','MPhil',5),(12,'Mr.Kumar','BED',4),(13,'Mrs.Tejaswee','MPhil',3),(14,'Mrs.Aarya','PhD',7),(15,'Mr.Kunal','Diploma',4);
```

```
insert into stte values(1,11,'DBMS'),(2,12,'Computer'),(3,13,'C'),(4,14,'OS'),(5,15,'CO'),(4,12,'DBMS'),(2,11,'OS');
```

Q2.) Using above database, solve the following queries:

a) List teacher details who taught subject 'DBMS'.

```
select * from teacher where tno in(select tno from stte where subject='DBMS');
```

b) Find student details with teacher name who taught class 'F.Y.B.C.A'.

```
select stud.sname,stud.sclass,teacher.tname from stud join stte on stud.sno=stte.sno join teacher on stte.tno=teacher.tno where stud.sclass='FYBCA';
```

c) Find the maximum experience details of teacher.

```
select * from teacher where experience=(select max(experience) from teacher);
```

d) Update teacher qualification to 'Ph.D' of 'Mr.Kumar'.

```
update teacher set qualification='Ph.D' where tname='Mr.Kumar';
```

Q.1) Practical Questions on PostgreSQL

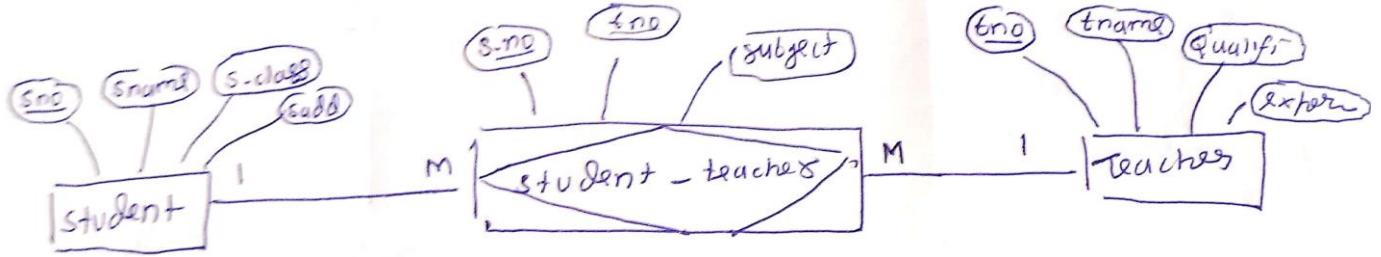
Consider the following database

Student (sno, s_name, s_class, s_addr)

Teacher (tno, t_name, qualification, experience)

Student-Teacher: M-M with descriptive attributes Subject.

a) Draw the ER diagram for above relational schema and normalize it in 3NF.



b) Create the above database in 3NF form in PostgreSQL using constraints.

```
create table stud(sno int primary key,sname varchar(20),sclass text,addr text);
```

```
create table teacher(tno int primary key,tname varchar(20),qualification text,experience int);
```

```
create table stte(sno int,tno int,subject text,primary key(sno,tno),foreign key(sno)references stud(sno),foreign key(tno)references teacher(tno));
```

```
insert into stud values(1,'Kunal','FYBCA','Beed'),(2,'Hrushi','SYBCA','Alandi'),(3,'Tushar','TYBCA','Pune'),(4,'Aditya','FYBCA','Pimpri'),(5,'Sujal','SYBCA','Nashik');
```

```
insert into teacher values(11,'Mr.Rajesh','MPhil',5),(12,'Mr.Kumar','BED',4),(13,'Mrs.Tejaswee','MPhil',3),(14,'Mrs.Aarya','PhD',7),(15,'Mr.Kunal','Diploma',4);
```

```
insert into stte values(1,11,'DBMS'),(2,12,'Computer'),(3,13,'C'),(4,14,'OS'),(5,15,'CO'),(4,12,'DBMS'),(2,11,'OS');
```

Q2.) Using above database, solve the following queries:

a) List the students details who study in class ‘S.Y.B.C.A’.

```
select * from stud where sclass='SYBCA';
```

b) Find subject wise teacher details.

```
select stte.subject,teacher.* from stte join teacher on stte.tno=teacher.tno;
```

c) Count the total no. of students whose class is ‘T.Y.B.C.A’

```
select count(*)from stud where sclass='TYBCA';
```

d) Delete teacher details whose name is ‘Mr. Rajesh’.

```
delete from teacher where tname='Mr.Rajesh';
```

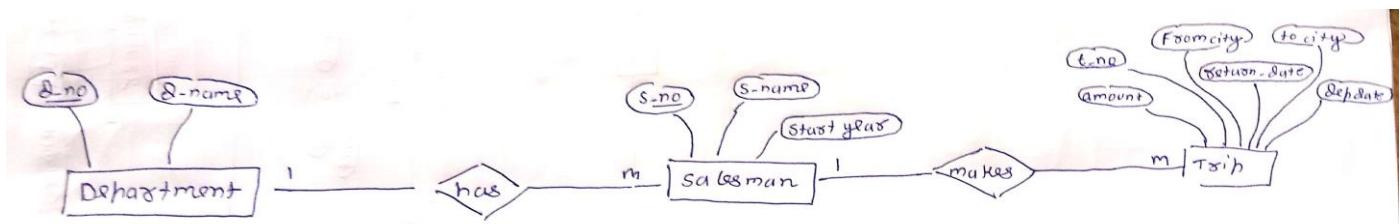
Q.1) Practical Questions on PostgreSQL Consider the following database

Salesman (sno, s_name, start_year)

Trip (tno, from_city, to_city, departure_date, return_date, amount) Dept (deptno, dept_name)

The relationship is as follows. Dept-salesman 1 To M Salesman-Trip 1 ToM

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Dept (deptno INT PRIMARY KEY, dept_name VARCHAR(50) NOT NULL);
```

```
CREATE TABLE Salesman (sno INT PRIMARY KEY, s_name VARCHAR(50) NOT NULL, start_year INT, deptno INT REFERENCES Dept(deptno));
```

```
CREATE TABLE Trip (tno INT PRIMARY KEY, from_city VARCHAR(50) NOT NULL, to_city VARCHAR(50) NOT NULL, departure_date DATE, return_date DATE, amount DECIMAL(10,2));
```

```
INSERT INTO Dept (deptno, dept_name) VALUES (10, 'Computer'), (20, 'Marketing'), (30, 'Accounts');
```

```
INSERT INTO Salesman (sno, s_name, start_year, deptno) VALUES (1, 'John', 2020, 10), (2, 'Alice', 2021, 20), (3, 'Bob', 2019, 30), (4, 'Mary', 2022, 10);
```

```
INSERT INTO Trip (tno, from_city, to_city, departure_date, return_date, amount) VALUES (101, 'Mumbai', 'Delhi', '2024-04-01', '2024-04-07', 25000.00), (102, 'Pune', 'Bangalore', '2024-04-10', '2024-04-15', 32000.00), (103, 'Chennai', 'Hyderabad', '2024-04-18', '2024-04-22', 18000.00), (104, 'Kolkata', 'Jaipur', '2024-04-25', '2024-04-30', 40000.00);
```

Q2.) Using above database, solve the following queries:

- a) List salesman details whose department is 'Computer'

```
SELECT S.sno, S.s_name, S.start_year, D.dept_name FROM Salesman S INNER JOIN Dept D ON S.deptno = D.deptno WHERE D.dept_name = 'Computer';
```

- b) List department wise salesman details

```
SELECT D.dept_name, S.sno, S.s_name, S.start_year FROM Dept D INNER JOIN Salesman S ON S.deptno = D.deptno ORDER BY D.dept_name, S.s_name;
```

- c) Find salesman details whose trip budget is less than 30000.

```
SELECT S.sno, S.s_name, T.amount FROM Salesman S INNER JOIN Trip T ON S.sno = T.tno WHERE T.amount < 30000;
```

- d) Alter table dept to add attribute 'dept_location'.

```
ALTER TABLE Dept ADD COLUMN dept_location VARCHAR(50);
```

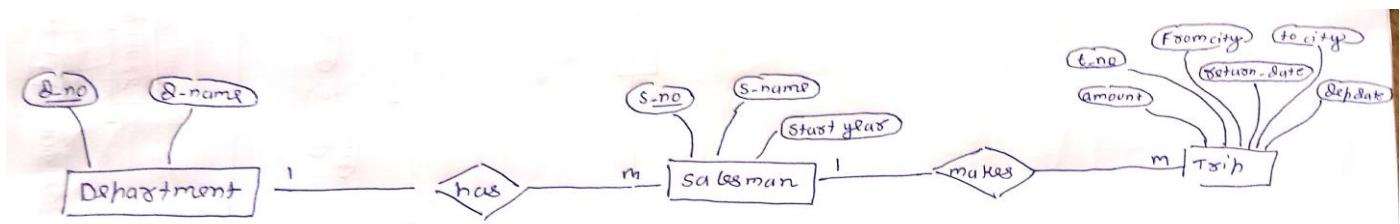
Q.1) Practical Questions on PostgreSQL Consider the following database

Salesman (sno, s_name, start_year)

Trip (tno, from_city, to_city, departure_date, return_date, amount) Dept (deptno, dept_name)

The relationship is as follows. Dept-salesman 1 To M Salesman-Trip 1 ToM

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Dept (deptno SERIAL PRIMARY KEY, dept_name VARCHAR(50) NOT NULL UNIQUE);
```

```
CREATE TABLE Salesman (sno SERIAL PRIMARY KEY, s_name VARCHAR(50) NOT NULL, start_year INTEGER NOT NULL, deptno INTEGER REFERENCES Dept(deptno) NOT NULL);
```

```
CREATE TABLE Trip (tno SERIAL PRIMARY KEY, from_city VARCHAR(50) NOT NULL, to_city VARCHAR(50) NOT NULL, departure_date DATE NOT NULL, return_date DATE NOT NULL, amount DECIMAL(10,2) NOT NULL, sno INTEGER REFERENCES Salesman(sno) NOT NULL);
```

```
INSERT INTO Dept (dept_name) VALUES ('Marketing'), ('Sales'), ('Engineering');
```

```
INSERT INTO Salesman (s_name, start_year, deptno) VALUES ('Mr. Patil', 2015, 1), ('Ms. Sharma', 2018, 2), ('Mr. Khan', 2020, 3);
```

```
INSERT INTO Trip (from_city, to_city, departure_date, return_date, amount, sno) VALUES ('Mumbai', 'Delhi', '2024-04-01', '2024-04-05', 75000.00, 1), ('Chennai', 'Kolkata', '2024-03-15', '2024-03-20', 50000.00, 2), ('Bangalore', 'Hyderabad', '2024-02-10', '2024-02-14', 80000.00, 3);
```

Q2.) Using above database, solve the following queries:

- a) Find trip details whose salesman 'Mr.Patil'.

```
SELECT t.tno, t.from_city, t.to_city, t.departure_date, t.return_date, t.amount FROM Trip t INNER JOIN Salesman s ON t.sno = s.sno WHERE s.s_name = 'Mr.Patil';
```

- b) List department wise trip details

```
SELECT d.dept_name, t.tno, t.from_city, t.to_city, t.departure_date, t.return_date, t.amount FROM Trip t INNER JOIN Salesman s ON t.sno = s.sno INNER JOIN Dept d ON s.deptno = d.deptno;
```

- c) Find salesman with trip details whose expense is between 50000 to 100000 expenses.

```
SELECT s.sno, s.s_name, s.start_year, t.tno, t.from_city, t.to_city, t.departure_date, t.return_date, t.amount FROM Trip t INNER JOIN Salesman s ON t.sno = s.sno WHERE t.amount BETWEEN 50000 AND 100000;
```

- d) Update trip from city 'Pune' to 'Delhi' of Mr.Patil

```
UPDATE Trip SET from_city = 'Delhi' WHERE sno = (SELECT sno FROM Salesman WHERE s_name = 'Mr.Patil');
```

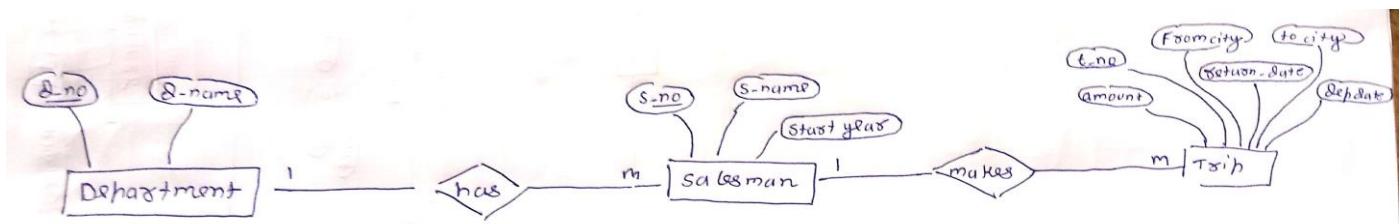
Q.1) Practical Questions on PostgreSQL Consider the following database

Salesman (sno, s_name, start_year)

Trip (tno, from_city, to_city, departure_date, return_date, amount) Dept (deptno, dept_name)

The relationship is as follows. Dept-salesman 1 To M Salesman-Trip 1 ToM

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Dept (deptno SERIAL PRIMARY KEY, dept_name VARCHAR(50) NOT NULL UNIQUE);
```

```
CREATE TABLE Salesman (sno SERIAL PRIMARY KEY, s_name VARCHAR(50) NOT NULL, start_year INTEGER NOT NULL, deptno INTEGER REFERENCES Dept(deptno) NOT NULL, experience INTEGER AS (EXTRACT(YEAR FROM CURRENT_DATE) - start_year) STORED);
```

```
CREATE TABLE Trip (tno SERIAL PRIMARY KEY, from_city VARCHAR(50) NOT NULL, to_city VARCHAR(50) NOT NULL, departure_date DATE NOT NULL, return_date DATE NOT NULL, amount DECIMAL(10,2) NOT NULL, sno INTEGER REFERENCES Salesman(sno) NOT NULL);
```

```
INSERT INTO Dept (dept_name) VALUES ('Marketing'), ('Sales'), ('Engineering');
```

```
INSERT INTO Salesman (s_name, start_year, deptno) VALUES ('Mr. Patil', 2015, 1), ('Ms. Sharma', 2018, 2), ('Mr. Khan', 2020, 3);
```

```
INSERT INTO Trip (from_city, to_city, departure_date, return_date, amount, sno) VALUES ('Mumbai', 'Delhi', '2024-04-01', '2024-04-05', 75000.00, 1), ('Chennai', 'Kolkata', '2024-03-15', '2024-03-20', 50000.00, 2), ('Bangalore', 'Hyderabad', '2024-02-10', '2024-02-14', 80000.00, 3), ('Pune', 'Jaipur', '2024-01-20', '2024-01-25', 60000.00, 1);
```

Q2.) Using above database, solve the following queries:

- a) List salesman details whose department name is 'Marketing'

```
SELECT s.sno, s.s_name, s.start_year, s.experience FROM Salesman s INNER JOIN Dept d ON s.deptno = d.deptno WHERE d.dept_name = 'Marketing';
```

- b) Find trip details along with salesman whose experience is maximum.

```
SELECT t.*, s.* FROM Trip t INNER JOIN Salesman s ON t.sno = s.sno WHERE s.experience = (SELECT MAX(experience) FROM Salesman);
```

- c) Alter table salesman to add attribute 'experience'.

- d) Find the departments from which the salesman has done number of trips more than two.

```
SELECT d.dept_name FROM Dept d INNER JOIN Salesman s ON s.deptno = d.deptno GROUP BY d.deptno, d.dept_name HAVING COUNT(DISTINCT tno) > 2;
```

Q.1) Practical Questions on PostgreSQL

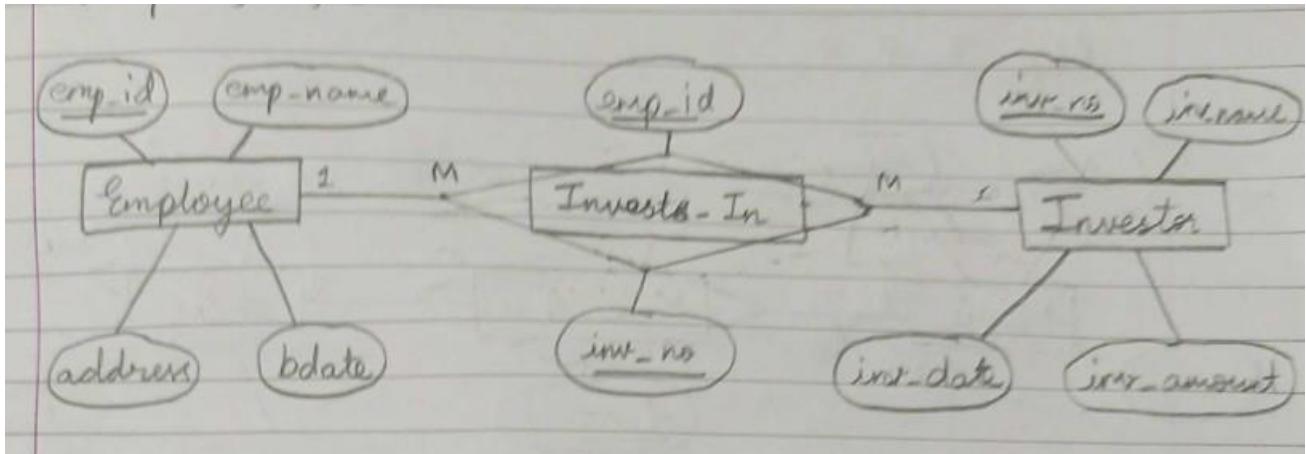
Consider the following database

Employee (emp_id, emp_name, address, bdate)
Investor (inv_no, inv_name, inv_date, inv_amt)

An employee may invest in one or more investments; hence he can be an investor. But an investor need not be an employee of the firm.

Assume appropriate data types for all the attributes.

a) Draw the ER diagram for above relational schema and normalize it in 3NF.



b) Create the above database in 3NF form in PostgreSQL using constraints.

```
create table Employee(emp_id int primary key, emp_name varchar(20), address text, bdate date);
```

```
create table Investor(inv_no int primary key, inv_name varchar(20), inv_date date, inv_amount float);
```

```
create table Employeeinvestor(emp_id int references employee(emp_id), inv_no int references investor(inv_no), primary key(emp_id, inv_no));
```

```
insert into Employee values(1,'nikita','warora','2005-06-14'),(2,'tejaswee','dighi','2005-01-23'),(3,'mrudula','pune','2005-08-16'), (4,'prachi','nagpur','2005-08-10'),(5,'mansi','chandrapur','2005-04-15');
```

```
insert into Investor values(101,'manasvi','2024-04-23',10000),(102,'pranjali','2024-02-12',20000),(103,'ranveer','2024-04-11',40000),(104,'ragav','2024-09-23',45000),(105,'payal','2024-11-21',33000);
```

```
insert into Employeeinvestor values(1,101),(2,102),(3,103),(4,104),(5,105);
```

Q2.) Using above database, solve the following queries:

a) List the distinct names of customers who are either employees, or investors or both.

```
SELECT DISTINCT emp_name FROM employee UNION SELECT DISTINCT inv_name FROM investor;
```

b) List the employees who are living in pune.

```
SELECT emp_name FROM Employee WHERE address = 'Pune';
```

c) Find the count of employee area wise.

```
SELECT address, COUNT(*) AS employee_count FROM Employee GROUP BY address;
```

d) Delete information of Investors whose amount is less than 10,000.

```
DELETE FROM Investor WHERE inv_amt < 10000;
```

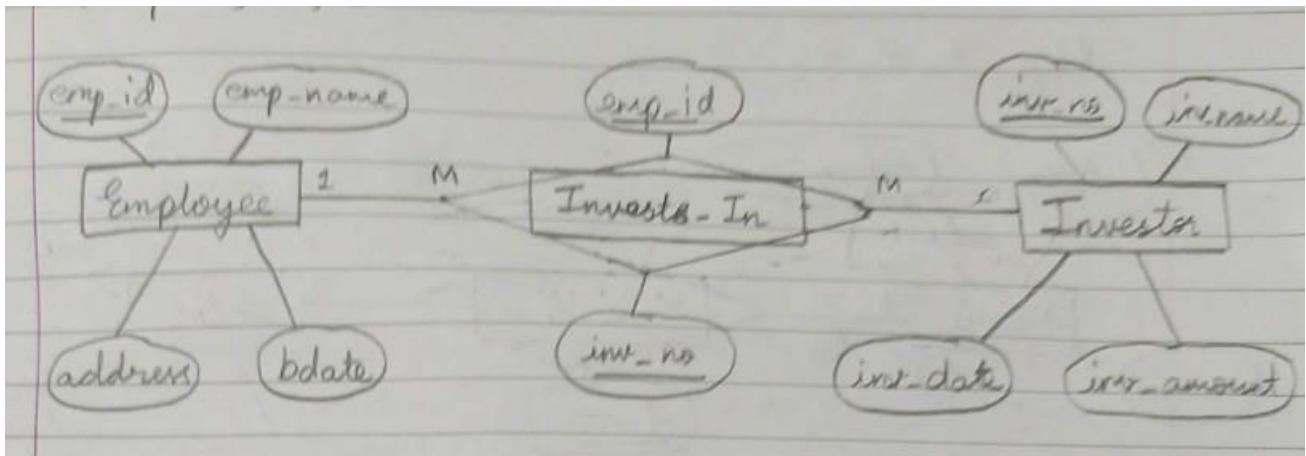
Q1) Practical Questions on PostgresSQL

Employee (emp_id, emp_name, address, bdate)
Investor (inv_no, inv_name, inv_date, inv_amt)

An employee may invest in one or more investments, hence he can be an investor. But an investor need not be an employee of the firm.

Assume appropriate data types for all the attributes.

a) Draw the ER diagram for above relational schema and normalize it in 3NF.



b) Create the above database in 3NF form in PostgresSQL using constraints.

```
CREATE TABLE Employee (emp_id SERIAL PRIMARY KEY, emp_name VARCHAR(100) NOT NULL, address TEXT, bdate DATE);
```

```
CREATE TABLE Investor (inv_no SERIAL PRIMARY KEY, inv_name VARCHAR(100) NOT NULL, inv_date DATE, inv_amt DECIMAL(12, 2) NOT NULL);
```

```
CREATE TABLE EmployeeInvestor (emp_id INT REFERENCES Employee(emp_id), inv_no INT REFERENCES Investor(inv_no), PRIMARY KEY (emp_id, inv_no));
```

```
INSERT INTO Employee (emp_name, address, bdate) VALUES ('John Doe', '123 Main St, Anytown, USA', '1990-05-15'), ('Mary Smith', '456 Elm St, Somewhere, USA', '1985-09-20'), ('Alice Johnson', '789 Oak St, Nowhere, USA', '1992-03-10');
```

```
INSERT INTO Investor (inv_name, inv_date, inv_amt) VALUES ('Mary Smith', '2023-01-05', 15000.00), ('John Doe', '2023-02-10', 20000.00), ('Michael Brown', '2023-03-15', 25000.00);
```

```
INSERT INTO EmployeeInvestor (emp_id, inv_no) VALUES (1, 1), (2, 2), (3, 3);
```

Q2) Using above database, solve the following queries:

a) List the names of employees who are also investors.

```
SELECT e.emp_name FROM Employee e JOIN EmployeeInvestor ei ON e.emp_id = ei.emp_id;
```

b) List the information of Investors whose name start with 'M'

```
SELECT * FROM Investor WHERE inv_name LIKE 'M%';
```

c) Increase the amount of investors by 10%.

```
UPDATE Investor SET inv_amt = inv_amt * 1.10;
```

d) Delete the investor information whose amount is 20,000.

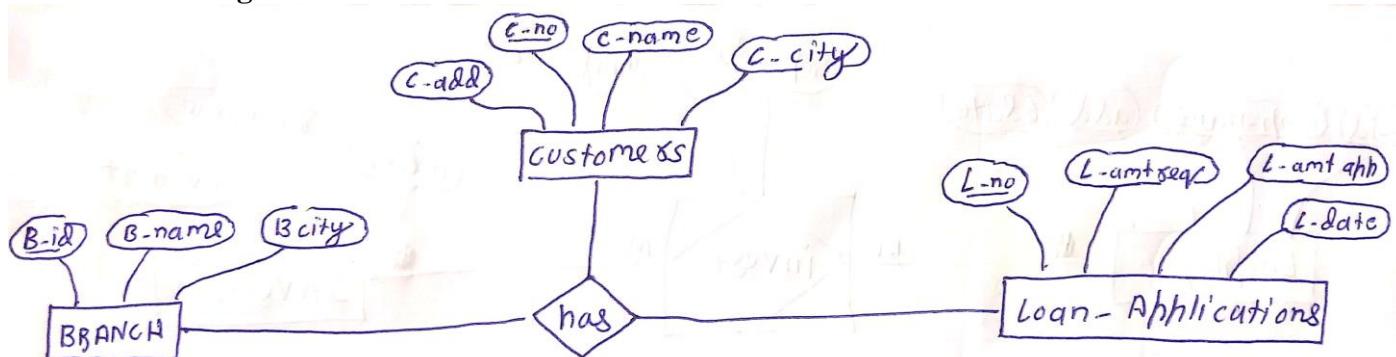
```
DELETE FROM Investor WHERE inv_amt = 20000.00;
```

Q1) Practical Questions on PostgresSQL

Branch (B_id, Brname, Brcity) Customer (C_no, Cname, Caddress, City) Loan_Application (L_no, L_amt_required, L_amt_approved,L_date)

Branch, Customer, Loan_Application are related with ternary relationship. Ternary (B_id, C_no, L_no) Assume appropriate data types for all the attributes.

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgresSQL using constraints.

```
CREATE TABLE Branch (B_id INT PRIMARY KEY,Brname VARCHAR(50) NOT NULL);
```

```
CREATE TABLE Customer (C_no INT PRIMARY KEY,Cname VARCHAR(50) NOT NULL,Caddress VARCHAR(255) NOT NULL,City VARCHAR(50) NOT NULL);
```

```
CREATE TABLE Loan_Application (L_no INT PRIMARY KEY,L_amt_required DECIMAL(10,2) NOT NULL, L_amt_approved DECIMAL(10,2) NOT NULL,L_date DATE NOT NULL);
```

```
CREATE TABLE Ternary (B_id INT,C_no INT,L_no INT,PRIMARY KEY (B_id, C_no, L_no),FOREIGN KEY (B_id) REFERENCES Branch(B_id),FOREIGN KEY (C_no) REFERENCES Customer(C_no),FOREIGN KEY (L_no) REFERENCES Loan_Application(L_no));
```

```
INSERT INTO Branch VALUES (1, 'Aundh'), (2, 'M.G. Road');
```

```
INSERT INTO Customer VALUES (101, 'John Doe', '123 Main St', 'Pune'),(102, 'Jane Smith', '456 Elm St', 'Mumbai'), (103, 'Alice Miller', '789 Oak Ave', 'Aundh');
```

```
INSERT INTO Loan_Application VALUES (2001, 100000.00, 80000.00, '2024-03-20'), (2002, 50000.00, 45000.00, '2024-03-25'),(2003, 150000.00, 120000.00, '2024-03-28');
```

```
INSERT INTO Ternary VALUES(1, 101, 2001),(2, 102, 2002),(1, 103, 2003);
```

Q2) Using above database, solve the following queries:

- a) List the names of the customers for the “Aundh” branch.

```
SELECT C.Cname FROM Customer C INNER JOIN Ternary T ON C.C_no = T.C_no INNER JOIN Branch B ON T.B_id = B.B_id WHERE B.Brname = 'Aundh';
```

- b) Find the maximum loan amount approved.

```
SELECT MAX(L_amt_approved) AS Max_Approved_Loan FROM Loan_Application;
```

- c) Count the number of loan application received by “M.G. Road” branch.

```
SELECT COUNT(*) AS Loan_Applications FROM Loan_Application LA INNER JOIN Ternary T ON LA.L_no = T.L_no INNER JOIN Branch B ON T.B_id = B.B_id WHERE B.Brname = 'M.G. Road';
```

- d) Increase the approved loan amount by 2%.

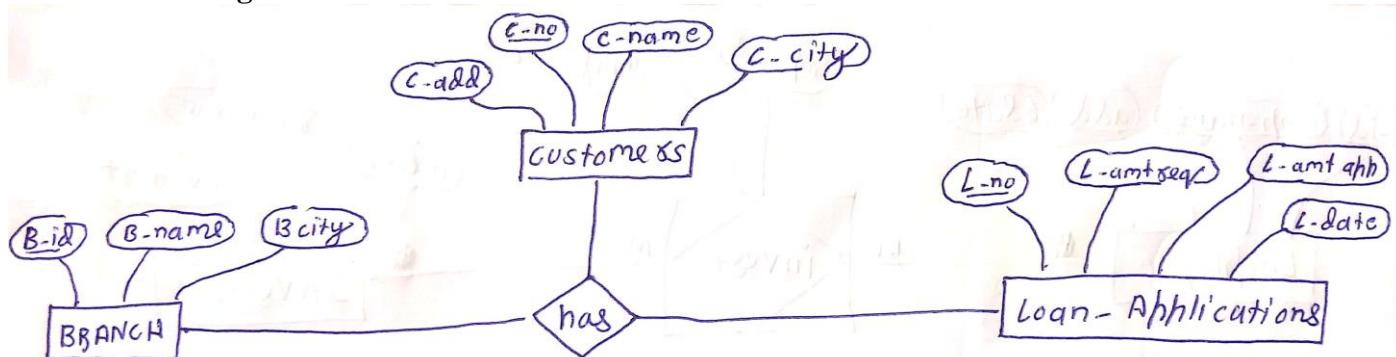
```
SET L_amt_approved = L_amt_approved * 1.02;
```

Q1) Practical Questions on PostgreSQL

Branch (B_id, Brname, Brcity) Customer (C_no, Cname, Caddress, City) Loan_Application (L_no, L_amt_required, L_amt_approved, L_date)

Branch, Customer, Loan_Application are related with ternary relationship. Ternary (B_id, C_no, L_no) Assume appropriate data types for all the attributes.

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Branch ( B_id INT PRIMARY KEY, Brname VARCHAR(50) NOT NULL UNIQUE, -- Unique constraint for branch name, Brcity VARCHAR(50) NOT NULL);
```

```
CREATE TABLE Customer ( C_no INT PRIMARY KEY, Cname VARCHAR(50) NOT NULL, Caddress VARCHAR(255) NOT NULL, City VARCHAR(50) NOT NULL);
```

```
CREATE TABLE Loan_Application ( L_no INT PRIMARY KEY, L_amt_required DECIMAL(10,2) NOT NULL, L_amt_approved DECIMAL(10,2) NOT NULL, L_date DATE NOT NULL);
```

```
CREATE TABLE Ternary ( B_id INT, C_no INT, L_no INT, PRIMARY KEY (B_id, C_no, L_no), FOREIGN KEY (B_id) REFERENCES Branch(B_id), FOREIGN KEY (C_no) REFERENCES Customer(C_no), FOREIGN KEY (L_no) REFERENCES Loan_Application(L_no));
```

```
INSERT INTO Branch VALUES (1, 'Aundh', 'Pune'), (2, 'Deccan', 'Pune'), (3, 'Shivajinagar', 'Pune');
```

```
INSERT INTO Customer VALUES (101, 'John Doe', '123 Main St', 'Pune'), (102, 'Jane Smith', '456 Elm St', 'Mumbai'), (103, 'Alice Miller', '789 Oak Ave', 'Pune');
```

```
INSERT INTO Loan_Application VALUES (2001, 100000.00, 80000.00, '2024-03-20'), (2002, 50000.00, 45000.00, '2024-03-25'), (2003, 150000.00, 120000.00, '2024-03-28');
```

```
INSERT INTO Ternary VALUES (1, 101, 2001), (2, 102, 2002), (1, 103, 2003);
```

Q2) Using above database, solve the following queries:

- a) List the names of the customers for the “Shivajinagar” branch.

```
SELECT C.Cname FROM Customer C INNER JOIN Ternary T ON C.C_no = T.C_no INNER JOIN Branch B ON T.B_id = B.B_id WHERE B.Brname = 'Shivajinagar';
```

- b) Find out the total loan amount sanctioned by “Deccan” branch.

```
SELECT SUM(LA.L_amt_approved) AS Total_Sanctioned_Amount FROM Loan_Application LA INNER JOIN Ternary T ON LA.L_no = T.L_no INNER JOIN Branch B ON T.B_id = B.B_id WHERE B.Brname = 'Deccan';
```

- c) List the names of customers who live in Pune city.

```
SELECT C.Cname FROM Customer C WHERE C.City = 'Pune';
```

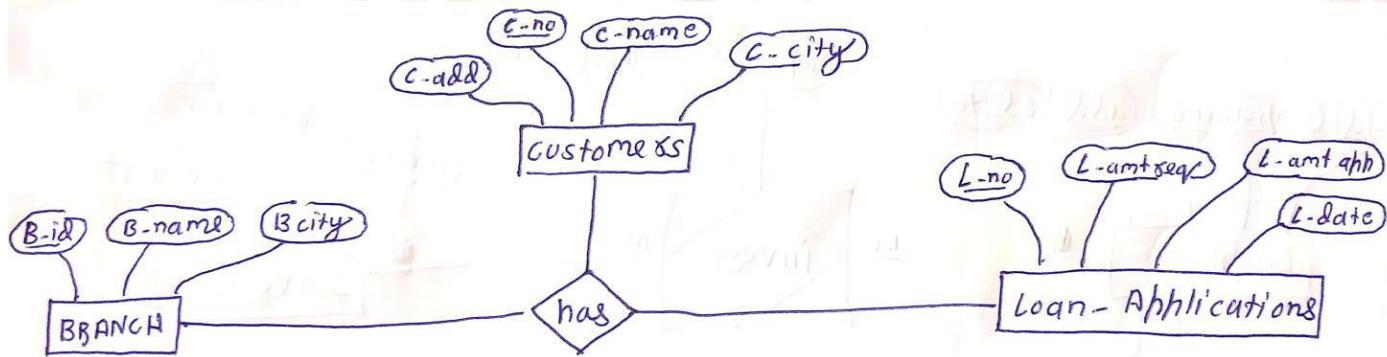
- d) Count the number of loan applications received by “Aundh” branch

```
SELECT COUNT(*) AS Loan_Applications FROM Loan_Application LA INNER JOIN Ternary T ON LA.L_no = T.L_no INNER JOIN Branch B ON T.B_id = B.B_id WHERE B.Brname = 'Aundh';
```

Q1) Practical Questions on PostgreSQL

Branch (B_id, Brname, Brcity) Customer (C_no, Cname, Caddress, City) Loan_Application (L_no, L_amt_required, L_amt_approved, L_date) Branch, Customer, Loan_Application are related with ternary relationship. Ternary (B_id, C_no, L_no) Assume appropriate data types for all the attributes.

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```

CREATE TABLE Branch (B_id INT PRIMARY KEY, Brname VARCHAR(50) NOT NULL UNIQUE, Brcity VARCHAR(50) NOT NULL);

CREATE TABLE Customer (C_no INT PRIMARY KEY, Cname VARCHAR(50) NOT NULL, Caddress VARCHAR(255) NOT NULL, City VARCHAR(50) NOT NULL);

CREATE TABLE Loan_Application (L_no INT PRIMARY KEY, L_amt_required DECIMAL(10,2) NOT NULL, L_amt_approved DECIMAL(10,2) NOT NULL, L_date DATE NOT NULL);

CREATE TABLE Ternary (B_id INT, C_no INT, L_no INT, PRIMARY KEY (B_id, C_no, L_no), FOREIGN KEY (B_id) REFERENCES Branch(B_id), FOREIGN KEY (C_no) REFERENCES Customer(C_no), FOREIGN KEY (L_no) REFERENCES Loan_Application(L_no));

INSERT INTO Branch (B_id, Brname, Brcity) VALUES (1, 'Aundh', 'Pune'), (2, 'Deccan', 'Pune'), (3, 'Borivali', 'Mumbai');

INSERT INTO Customer (C_no, Cname, Caddress, City) VALUES (101, 'John Doe', '123 Main St', 'Pune'), (102, 'Jane Smith', '456 Elm St', 'Mumbai'), (103, 'Alice Miller', '789 Oak Ave', 'Pune'), (104, 'Sunil Sharma', '400 Park Lane', 'Mumbai');

INSERT INTO Loan_Application (L_no, L_amt_required, L_amt_approved, L_date) VALUES (2001, 100000.00, 80000.00, '2024-03-20'), (2002, 50000.00, 45000.00, '2024-03-25'), (2003, 150000.00, 120000.00, '2024-09-10'), (2004, 75000.00, 60000.00, '2024-09-15');

INSERT INTO Ternary (B_id, C_no, L_no) VALUES (1, 101, 2001), (2, 102, 2002), (1, 103, 2003), (3, 104, 2004);
  
```

Q2) Using above database, solve the following queries:

- List the details of customers whose name starts with „S“.
`SELECT * FROM Customer WHERE Cname LIKE 'S%';`
- List the names of the customer along with the branch names who have applied for loan in the month of September.
`SELECT C.Cname, B.Brname FROM Customer C INNER JOIN Ternary T ON C.C_no = T.C_no INNER JOIN Branch B ON T.B_id = B.B_id INNER JOIN Loan_Application LA ON LA.L_no = T.L_no WHERE EXTRACT(MONTH FROM LA.L_date) = 9;`
- Find the maximum loan amount approved.
`SELECT MAX(L_amt_approved) AS Max_Approved_Loan FROM Loan_Application;`
- List the branch details of “Mumbai” city
`SELECT * FROM Branch WHERE Brcity = 'Mumbai';`

Q.1) Practical Questions on PostgreSQL the following database

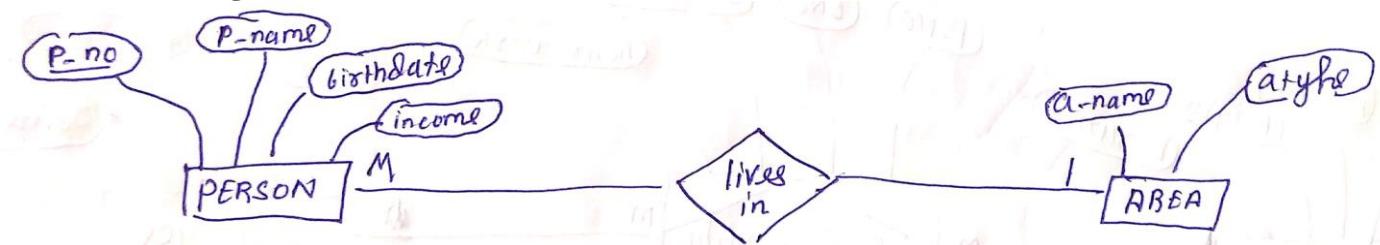
Person (pnumber, pname, birthdate, income)
Area

(aname, area_type)

An area can have one or more person living in it , but a person belongs to exactly one area.The attribute ‘area_type’ can have values as either urban or rural.

Assume appropriate data types for all the attributes.

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Area (aname VARCHAR(50) PRIMARY KEY, area_type VARCHAR(10));
```

```
CREATE TABLE Person (pnumber INT PRIMARY KEY, pname VARCHAR(50), birthdate DATE, income  
FLOAT, aname VARCHAR(50) REFERENCES Area);
```

```
INSERT INTO Area VALUES ('Pune', 'urban'), ('Mumbai', 'urban'), ('Alandi', 'urban'), ('Khed', 'rural'), ('Chakan',  
'rural');
```

```
INSERT INTO Person VALUES (1,'Tushar', '2003-03-15', 15000.00, 'Pune'),(2,'Sujal', '2005-04-20', 12000.00,  
'Pune'), (3,'Hrushi', '2004-02-19', 8000.00, 'Alandi'), (4,'Aditya', '18-06-2005', 11000, 'Alandi'),(5,'Sumeet', '07-06-  
2005', 9000, 'Mumbai'),(6,'Suyog', '01-05-2005', 20000, 'Khed'),(7,'Omkar', '23-07-2006', 9500,  
'Chakan'),(8,'Shubham', '08-09-2005', 7500, 'Khed'),(9,'Raj', '01-02-2006', 13000, 'Mumbai'),(10,'Sarthak', '28-01-  
2004', 18000, 'Chakan');
```

Q2.) Using above database, solve the following queries : [Marks 20]

- a) List the names of all people living in 'Pune' area.

```
SELECT pname FROM Person WHERE aname = 'Pune';
```

- b) Give the count of people whose income is below Rs.10000 living in urban area.

```
SELECT COUNT(*) FROM Person WHERE income < 10000 AND aname IN (SELECT aname  
FROM Area WHERE area_type = 'urban');
```

- c) List all people in rural area having maximum income.

```
SELECT pname, income FROM Person WHERE aname IN (SELECT aname FROM Area WHERE  
area_type = 'rural')AND income =(SELECT MAX(income) FROM Person WHERE aname IN  
(SELECT aname FROM Area WHERE area_type = 'rural'));
```

- d) Delete information of all people staying in 'urban' area.

```
DELETE FROM Person WHERE aname IN (SELECT aname FROM Area WHERE area_type =  
'urban');
```

Q.1) Practical Questions on PostgreSQL the following database

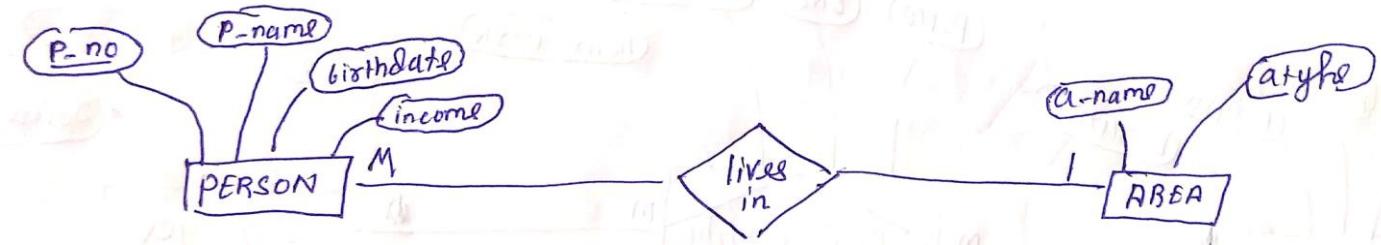
Person (pnumber, pname, birthdate, income)Area

(aname,area_type)

An area can have one or more person living in it , but a person belongs to exactly one area.The attribute ‘area_type’ can have values as either urban or rural.

Assume appropriate data types for all the attributes.

a) Draw the ER diagram for above relational schema and normalize it in 3NF.



b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Area (aname VARCHAR(50) PRIMARY KEY, area_type VARCHAR(10));
```

```
CREATE TABLE Person (pnumber INT PRIMARY KEY, pname VARCHAR(50), birthdate DATE, income  
FLOAT, aname VARCHAR(50) REFERENCES Area);
```

```
INSERT INTO Area VALUES ('Pune', 'urban'), ('Mumbai', 'urban'), ('Alandi', 'urban'), ('Khed', 'rural'), ('Chakan',  
'rural');
```

```
INSERT INTO Person VALUES (1,'Tushar', '2003-03-15', 15000.00, 'Pune'),(2,'Sujal', '2005-04-20', 12000.00,  
'Pune'), (3,'Hrushi', '2004-02-19', 8000.00, 'Alandi'), (4,'Aditya', '18-06-2005', 11000, 'Alandi'),(5,'Sumeet', '07-06-  
2005', 9000, 'Mumbai'),(6,'Suyog', '01-05-2005', 20000, 'Khed'),(7,'Omkar', '23-07-2006', 9500,  
'Chakan'),(8,'Shubham', '08-09-2005', 7500, 'Khed'),(9,'Raj', '01-02-2006', 13000, 'Mumbai'),(10,'Sarthak', '28-01-  
2004', 18000, 'Chakan');
```

Q2) Using above database, solve the following queries:

a) List all people area wise living in rural area type.

```
SELECT p.pname, a.aname FROM Person p JOIN Area a ON p.aname = a.aname  
WHERE a.area_type = 'rural';
```

b) Find the count of people area wise.

```
SELECT a.aname, COUNT(p.pnumber) AS people_count FROM Person p JOIN Area a ON  
p.aname = a.aname GROUP BY a.aname;
```

c) Find average income of people living in urban area.

```
SELECT AVG(p.income) AS average_income FROM Person p JOIN Area a ON p.aname =  
a.aname WHERE a.area_type = 'urban';
```

d) Delete people staying in ‘rural’ area and having income less than Rs 50000.

```
DELETE FROM Person WHERE aname IN (SELECT aname FROM Area WHERE  
area_type = 'rural') AND income < 50000;
```

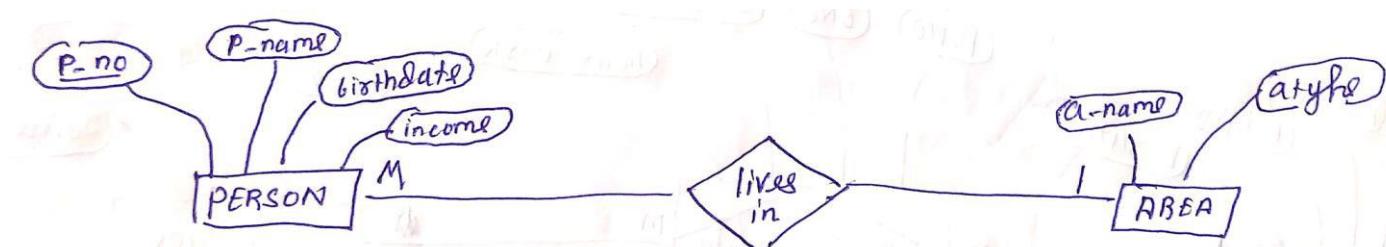
Q.1) Practical Questions on PostgreSQL the following database

Person (pnumber, pname, birthdate, income) Area (aname,area_type)

An area can have one or more person living in it , but a person belongs to exactly one area. The attribute ‘area_type’ can have values as either urban or rural.

Assume appropriate data types for all the attributes.

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Area (aname VARCHAR(50) PRIMARY KEY, area_type VARCHAR(10));
```

```
CREATE TABLE Person (pnumber INT PRIMARY KEY, pname VARCHAR(50), birthdate DATE, income FLOAT, aname VARCHAR(50) REFERENCES Area);
```

```
INSERT INTO Area VALUES ('Pune', 'urban'), ('Mumbai', 'urban'), ('Alandi', 'urban'), ('Khed', 'rural'), ('Chakan', 'rural');
```

```
INSERT INTO Person VALUES (1,'Tushar', '2003-03-15', 15000.00, 'Pune'),(2,'Sujal', '2005-04-20', 12000.00, 'Pune'), (3,'Hrushi', '2004-02-19', 8000.00, 'Alandi'), (4,'Aditya', '18-06-2005', 11000, 'Alandi'),(5,'Sumeet', '07-06-2005', 9000, 'Mumbai'),(6,'Suyog', '01-05-2005', 20000, 'Khed'),(7,'Omkar', '23-07-2006', 9500, 'Chakan'),(8,'Shubham', '08-09-2005', 7500, 'Khed'),(9,'Raj', '01-02-2006', 13000, 'Mumbai'),(10,'Sarthak', '28-01-2004', 18000, 'Chakan');
```

Q2) Using above database, solve the following queries:

- a) List the names of all people whose income is between is Rs10,000 and Rs.20000.

```
SELECT pname FROM Person WHERE income BETWEEN 10000 AND 20000;
```

- b) Find the count of people living in Pune area.

```
SELECT COUNT(*) FROM Person p JOIN Area a ON p.aname = a.aname WHERE a.aname = 'Pune';
```

- c) List names of people area wise having minimum income.

```
SELECT pname, area_type FROM Person p JOIN Area a ON p.aname = a.aname WHERE (p.income, p.aname) IN (SELECT MIN(income), aname FROM Person GROUP BY aname);
```

- d) Update income to Rs. 10000 of people staying in ‘urban’ area and having income less than Rs5000 .

```
UPDATE Person SET income = 10000 WHERE aname IN ( SELECT aname FROM Area WHERE area_type = 'urban') AND income < 5000;
```

Q1) Practical Questions on PostgreSQL

STUDENT (sreg_no, name , class)

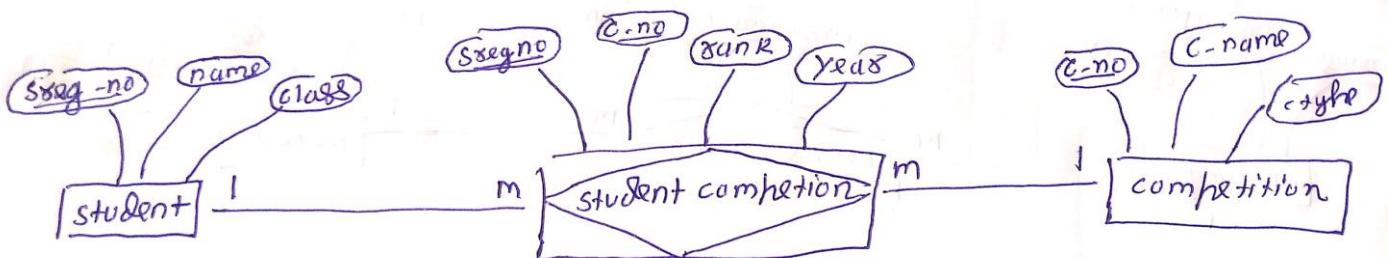
COMPETITION (c_no , name , C_type)

The relationship is as follows:

STUDENT-COMPETITION: M-M with described attributes rank and year. Class should be 'FYBCA' , 'SYBCA' and 'TYBCA'

Assume appropriate data types for all the attributes.

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Student (sreg_no INT PRIMARY KEY, name VARCHAR(50) NOT NULL, class VARCHAR(10)
CHECK (class IN ('FYBCA', 'SYBCA', 'TYBCA')));
```

```
CREATE TABLE Competition (c_no INT PRIMARY KEY, name VARCHAR(50) NOT NULL, C_type
VARCHAR(50) NOT NULL);
```

```
CREATE TABLE Participates_In (sreg_no INT NOT NULL, c_no INT NOT NULL, rank INT, year INT, PRIMARY
KEY (sreg_no, c_no), FOREIGN KEY (sreg_no) REFERENCES Student(sreg_no), FOREIGN KEY (c_no)
REFERENCES Competition(c_no));
```

```
INSERT INTO Student (sreg_no, name, class) VALUES (1001, 'Alice Smith', 'FYBCA'), (1002, 'Bob Brown',
'SYBCA'), (1003, 'Charlie Chen', 'TYBCA'), (1004, 'David Davis', 'FYBCA'), (1005, 'Emily Evans', 'SYBCA');
```

```
INSERT INTO Competition (c_no, name, C_type) VALUES (2001, 'Essay Competition', 'Writing'), (2002, 'Debate
Competition', 'Public Speaking'), (2003, 'Poster Competition', 'Visual Arts'), (2004, 'Quiz Competition', 'Knowledge');
```

```
INSERT INTO Participates_In (sreg_no, c_no, rank, year) VALUES (1001, 2001, 1, 2023), (1001, 2003, 2, 2023),
(1002, 2002, 1, 2022), (1003, 2004, 3, 2022), (1004, 2002, 2, 2023), (1004, 2004, 1, 2023), (1005, 2001, 3, 2023);
```

Q2) Using above database, solve the following queries:

- a) List the names of all students studying in FYBCA.

```
SELECT name FROM Student WHERE class = 'FYBCA';
```

- b) Find the count of students participated in Poster competition class wise.

```
SELECT class, COUNT(DISTINCT s.sreg_no) AS Participated_In FROM Student s INNER JOIN
Participates_In p ON s.sreg_no = p.sreg_no INNER JOIN Competition c ON p.c_no = c.c_no WHERE c.C_type
= 'Poster' GROUP BY class;
```

- c) List the names of students scoring 1st rank in all different competition.

```
SELECT S.name FROM Student S INNER JOIN Participates_In PI ON S.sreg_no = PI.sreg_no WHERE PI.rank
= 1 GROUP BY S.name HAVING COUNT(DISTINCT PI.c_no) = (SELECT COUNT(DISTINCT c_no) FROM
Competition);
```

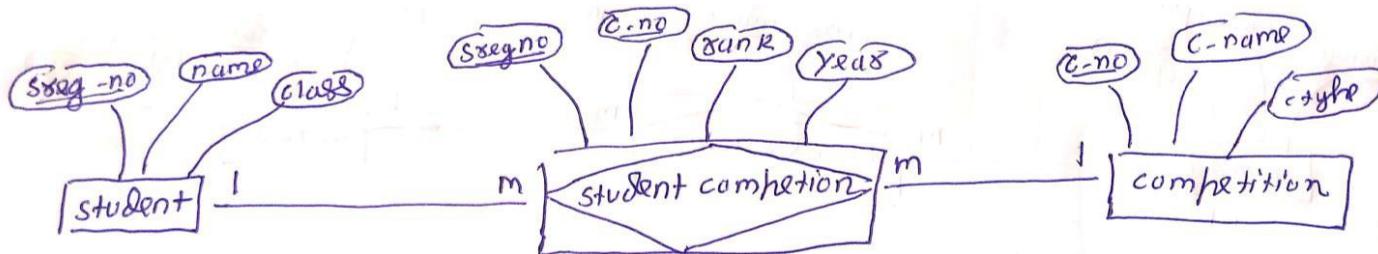
- d) Delete all students of class FYBCA participated in Quiz of competition in year 2018.

```
DELETE FROM Student S USING Participates_In PI, Competition C WHERE S.sreg_no = PI.sreg_no AND
PI.c_no = C.c_no AND S.class = 'FYBCA' AND C.C_type = 'Quiz' AND PI.year = 2018;
```

Q1) Practical Questions on PostgresSQL

STUDENT(sreg_no, name , class) COMPETITION(c_no , name , C_type)

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgresSQL using constraints.

```
CREATE TABLE STUDENT (sreg_no SERIAL PRIMARY KEY, name VARCHAR(100), class VARCHAR(10));
```

```
CREATE TABLE COMPETITION (c_no SERIAL PRIMARY KEY, name VARCHAR(100), c_type VARCHAR(20));
```

```
CREATE TABLE STUDENT_COMPETITION (sreg_no INT REFERENCES STUDENT, c_no INT REFERENCES COMPETITION, rank INT, year INT);
```

```
INSERT INTO STUDENT (name, class) VALUES ('Amit Kumar', 'FYBCA'), ('Priya Patel', 'SYBCA'), ('Rahul Sharma', 'FYBCA'), ('Neha Gupta', 'TYBCA'), ('Kiran Joshi', 'SYBCA');
```

```
INSERT INTO COMPETITION (name, c_type) VALUES ('Debate Competition', 'Debate'), ('Poster Making Competition', 'Poster'), ('Quiz Competition', 'Quiz');
```

```
INSERT INTO STUDENT_COMPETITION (sreg_no, c_no, rank, year) VALUES (1, 2, 1, 2018), (2, 2, 1, 2019), (3, 2, 2, 2019), (4, 2, 1, 2019), (5, 2, 2, 2019), (1, 3, 2, 2019), (2, 3, 1, 2019), (3, 3, 2, 2019), (4, 3, 1, 2019), (5, 3, 1, 2019);
```

Q2) Using above database, solve the following queries:

- a) List the names of all students participated in Quiz Competition.

```
SELECT S.name FROM Student S INNER JOIN Participates_In PI ON S.sreg_no = PI.sreg_no INNER JOIN Competition C ON PI.c_no = C.c_no WHERE C.C_type = 'Quiz';
```

- b) List the names of students participated in more than two competitions.

```
SELECT s.name FROM STUDENT s JOIN STUDENT_COMPETITION sc ON s.sreg_no = sc.sreg_no GROUP BY s.name HAVING COUNT(DISTINCT sc.c_no) > 2;
```

- c) List the names of students of TYBCA scoring 1st rank in all different competitions in year 2019.

```
SELECT S.name FROM Student S INNER JOIN Participates_In PI ON S.sreg_no = PI.sreg_no WHERE S.class = 'TYBCA' AND PI.rank = 1 AND PI.year = 2019 GROUP BY S.name HAVING COUNT(DISTINCT PI.c_no) = (SELECT COUNT(DISTINCT c_no) FROM Competition);
```

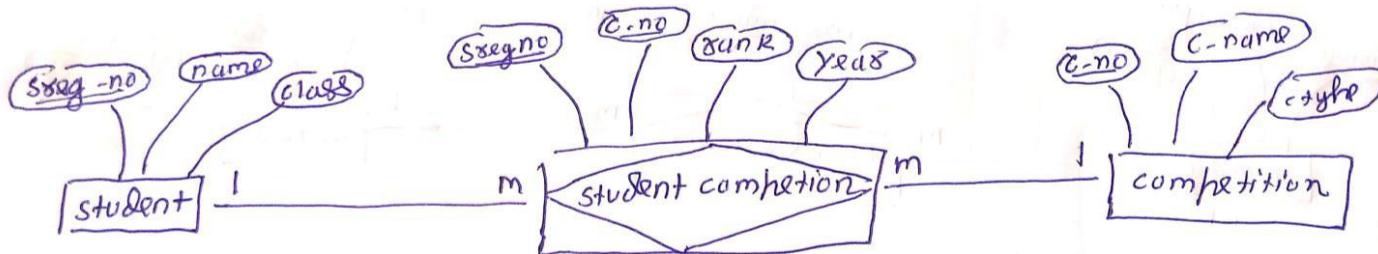
- d) Change rank of student KIRAN JOSHI of class SYBCA to 2nd rank participated in Poster competition

```
UPDATE Participates_In PI SET rank = 2 FROM Student S WHERE S.name = 'Kiran Joshi' AND S.class = 'SYBCA' AND PI.sreg_no = S.sreg_no AND (SELECT C.C_type FROM Competition C WHERE C.c_no = PI.c_no) = 'Poster';
```

Q1) Practical Questions on PostgreSQL

STUDENT(sreg_no, name , class) COMPETITION(c_no , name , C_type)

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```

CREATE TABLE Student (sreg_no INT PRIMARY KEY, name VARCHAR(50) NOT NULL, class VARCHAR(10)
CHECK (class IN ('FYBCA', 'SYBCA', 'TYBCA')));

CREATE TABLE Competition (c_no INT PRIMARY KEY, name VARCHAR(50) NOT NULL, C_type
VARCHAR(50) NOT NULL);

```

```

CREATE TABLE Participates_In (sreg_no INT NOT NULL, c_no INT NOT NULL, rank INT, year INT, PRIMARY
KEY (sreg_no, c_no), FOREIGN KEY (sreg_no) REFERENCES Student(sreg_no), FOREIGN KEY (c_no)
REFERENCES Competition(c_no));

```

```

INSERT INTO Student (sreg_no, name, class) VALUES(1, 'Amit Kumar', 'FYBCA'),(2, 'Priya Patel', 'SYBCA'),(3,
'Rahul Sharma', 'FYBCA'),(4, 'Neha Gupta', 'TYBCA'),(5, 'Kiran Joshi', 'SYBCA');

INSERT INTO Competition (c_no, name, C_type) VALUES (1, 'Debate Competition', 'Debate'), (2, 'Poster Making
Competition', 'Poster'),(3, 'Quiz Competition', 'Quiz');

INSERT INTO Participates_In (sreg_no, c_no, rank, year) VALUES (1, 2, 1, 2019),(1, 3, 1, 2019), (4, 1, 1, 2019),(1,
1, 1, 2020),(2, 2, 1, 2020),(4, 3, 1, 2020),(5, 2, 1, 2020),(3, 3, 2, 2019);

```

Q2) Using above database, solve the following queries:

- a) List the name of all competitions held in year 2019.

```

SELECT name FROM Competition WHERE c_no IN (SELECT DISTINCT c_no FROM Participates_In
WHERE year = 2019);

```

- b) List the name of students of FYBCA who participated in Quiz as well as in Poster

```

SELECT s.name FROM Student s JOIN Participates_In p ON s.sreg_no = p.sreg_no JOIN Competition c ON
p.c_no = c.c_no WHERE s.class = 'FYBCA' AND c.C_type IN ('Quiz', 'Poster');

```

- c) List the name of students class wise scoring 1st rank in all different competitions held in year 2020.

```

SELECT s.name, s.class FROM Student s WHERE s.sreg_no NOT IN (SELECT p.sreg_no FROM
Participates_In p WHERE p.year = 2020 AND p.rank != 1)GROUP BY s.name, s.class;

```

- d) Change class of student KIRAN JOSHI to FYBCA who participated in Poster competition.

```

UPDATE Student SET class = 'FYBCA' WHERE name = 'Kiran Joshi' AND sreg_no IN (SELECT
sreg_no FROM Participates_In WHERE c_no = 2);

```

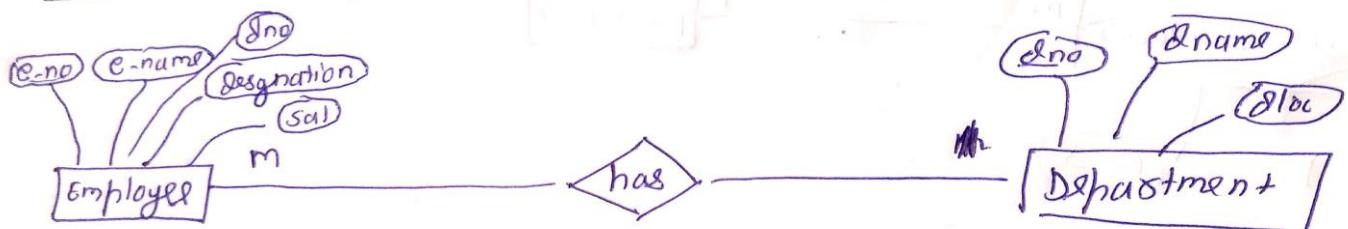
Q.1) Practical Questions on PostgreSQL Consider the following database

Employee (eno, ename, designation, sal)

Department
(dno, dname, dloc)

There exists a one-to-many relationship between department and employee. Create the Relations accordingly, so that the relationship is handled properly and the relations are in normalized form(3NF).

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Department (dno INT PRIMARY KEY, dname VARCHAR(50) NOT NULL, dloc  
VARCHAR(50) NOT NULL);
```

```
CREATE TABLE Employee (eno INT PRIMARY KEY, ename VARCHAR(50) NOT NULL, designation  
VARCHAR(50) NOT NULL, sal INT, dno INT, FOREIGN KEY (dno) REFERENCES Department(dno));
```

```
INSERT INTO Department (dno, dname, dloc) VALUES (101, 'Marketing', 'Mumbai'), (102, 'Sales', 'Delhi'), (103,  
'Finance', 'Chennai');
```

```
INSERT INTO Employee (eno, ename, designation, sal, dno) VALUES (2001, 'Alice Smith', 'Manager', 35000, 101),  
(2002, 'Bob Brown', 'Sales Representative', 22000, 102), (2003, 'Charlie Chen', 'Accountant', 28000, 103), (2004,  
'David Davis', 'H.O.D.', 50000, 101), (2005, 'Emily Evans', 'Marketing Associate', 18000, 101);
```

Q2.) Using above database, solve the following queries:

- a) List the details of employee whose designation is 'H.O.D'

```
SELECT * FROM Employee WHERE designation = 'H.O.D';
```

- b) List the name of employees whose salary is above 20000

```
SELECT ename FROM Employee WHERE sal > 20000;
```

- c) Count the number of employees in each department.

```
SELECT D.dname, COUNT(*) AS num_employees FROM Department D INNER JOIN Employee E ON D.dno  
= E.dno GROUP BY D.dname;
```

- d) Display department wise employee list.

```
SELECT D.dname, E.eno, E.ename, E.designation, E.sal FROM Department D INNER JOIN Employee E ON  
D.dno = E.dno ORDER BY D.dname;
```

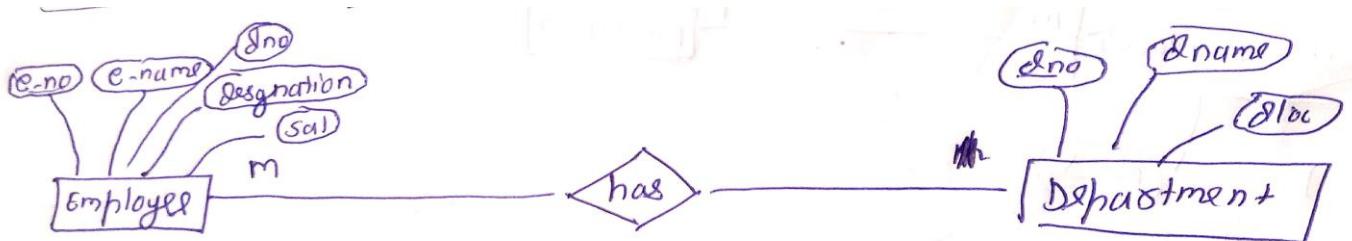
Q.1) Practical Questions on PostgreSQL Consider the following database

Employee (eno, ename, designation, sal)

Department
(dno, dname, dloc)

There exists a one-to-many relationship between department and employee. Create the Relations accordingly, so that the relationship is handled properly and the relations are in normalized form(3NF).

- a) Draw the ER diagram for above relational schema and normalize it in 3NF.



- b) Create the above database in 3NF form in PostgreSQL using constraints.

```
CREATE TABLE Department (dno INT PRIMARY KEY, dname VARCHAR(50) NOT NULL, dloc VARCHAR(50) NOT NULL);
```

```
CREATE TABLE Employee (eno INT PRIMARY KEY, ename VARCHAR(50) NOT NULL, designation VARCHAR(50) NOT NULL, sal DECIMAL(10,2), dno INT, FOREIGN KEY (dno) REFERENCES Department(dno));
```

```
INSERT INTO Department (dno, dname, dloc) VALUES (101, 'Marketing', 'Mumbai'), (102, 'Sales', 'Delhi'), (103, 'Finance', 'Chennai'), (104, 'IT', 'Pune');
```

```
INSERT INTO Employee (eno, ename, designation, sal, dno) VALUES (2001, 'Alice Smith', 'Manager', 35000.00, 101), (2002, 'Bob Brown', 'Sales Representative', 22000.00, 102), (2003, 'Charlie Chen', 'Accountant', 28000.00, 103), (2004, 'David Davis', 'H.O.D.', 50000.00, 101), (2005, 'Emily Evans', 'Marketing Associate', 18000.00, 101);
```

Q2.) Using above database, solve the following queries:

- a) List the employees and the department details location in 'Mumbai'

```
SELECT E.eno, E.ename, E.designation, E.sal, D.dname, D.dloc FROM Employee E INNER JOIN Department D ON E.dno = D.dno WHERE D.dloc = 'Mumbai';
```

- b) List employee details whose designation is 'HR' and location in 'Pune'

```
SELECT E.eno, E.ename, E.designation, E.sal, D.dname, D.dloc FROM Employee E INNER JOIN Department D ON E.dno = D.dno WHERE E.designation = 'HR' AND D.dloc = 'Pune';
```

- c) Update all employee's salary increase by 15%

```
UPDATE Employee SET sal = sal * 1.15;
```

- d) Find the maximum salary of employee from each department located at 'Pune'

```
SELECT D.dname, MAX(E.sal) AS max_salary FROM Department D INNER JOIN Employee E ON E.dno = D.dno WHERE D.dloc = 'Pune' GROUP BY D.dname;
```