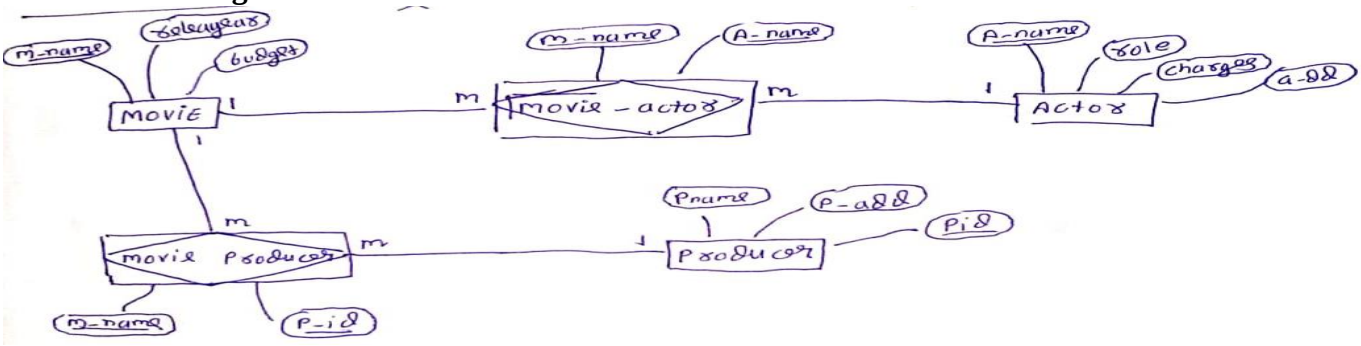
Q1) Practical Questions on PostgresSQL Movie (M_Name, release_year, budget) Actor (A_name ,role, charges,a_address) Producer (producer_id, Name, P_address)

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');
```