

# Savitribai Phule Pune University

S.Y. B.C.A. (Science) (Semester-III) Practical Examination

**BCA 235: s(Database Management Systems II Laboratory)**

Duration: 3Hrs.

Max Marks: 35+15=50

- Note: -**
1. Read the questions carefully and insert data in the database accordingly.
  2. Insert sufficient number of records in the database.
  3. No query should generate empty output.
  4. For count queries output should be more than 2 records. (If asked)

**Create the following database in 3NF using PostgreSQL. [Total Marks: 10]**

**Q1)** Consider the following database of Movie\_Actor\_Producer.

**Movie** (m\_name varchar (25), release\_year integer, budget money)

**Actor** (a\_name char (30), city varchar(30))

**Producer** (producer\_id integer, pname char (30), p\_address varchar (30))

**Relationship:**

Movie and Actor related with many-to-many relationship with descriptive attributes role and charges. Producer and Movie related with many-to-many relationship.

**Constraints:** Primary key, release\_year should not be null.

**Create a View [10]**

1. To display actor names who lives in 'Mumbai'.
2. To display actors information in each movie.

**Q.2) Using above database solve following questions: [Total Marks: 20]**

1. Write a trigger before inserting budget into a movie table. Budget should be minimum 60 lakh. Display appropriate message. [10]
2. Write a stored function to accept producer name as an input parameter and display count of movies that producer has produced. [10]

**Q.3) External Viva [05]**

**Q.4) Internal Evaluation [15]**

MOVIE ACTOR PRODUCER

```
CREATE TABLE Movie (m_name VARCHAR(25), release_year INTEGER NOT NULL, budget DECIMAL, PRIMARY KEY (m_name, release_year));

CREATE TABLE Actor (a_name CHAR(30), city VARCHAR(30), PRIMARY KEY (a_name));

CREATE TABLE Producer (producer_id INTEGER, pname CHAR(30), p_address VARCHAR(30), PRIMARY KEY (producer_id));

CREATE TABLE Movie_Actor (m_name VARCHAR(25), release_year INTEGER, a_name CHAR(30), role VARCHAR(50), charges DECIMAL, PRIMARY KEY (m_name, release_year, a_name), FOREIGN KEY (m_name, release_year) REFERENCES Movie(m_name, release_year), FOREIGN KEY (a_name) REFERENCES Actor(a_name));

CREATE TABLE Movie_Producer (m_name VARCHAR(25), release_year INTEGER, producer_id INTEGER, PRIMARY KEY (m_name, release_year, producer_id), FOREIGN KEY (m_name, release_year) REFERENCES Movie(m_name, release_year), FOREIGN KEY (producer_id) REFERENCES Producer(producer_id));

INSERT INTO Movie VALUES ('Sholey', 1975, 5000000), ('Lagaan', 2001, 3000000), ('Taal', 1999, 2000000);

INSERT INTO Actor VALUES ('Amitabh Bachchan', 'Mumbai'), ('Aamir Khan', 'Mumbai'), ('Dharmendra', 'Pune'), ('Hema Malini', 'Delhi');

INSERT INTO Producer VALUES (1, 'Mr. Subhash Ghai', 'Mumbai'), (2, 'Yash Chopra', 'Pune');

INSERT INTO Movie_Actor VALUES ('Sholey', 1975, 'Amitabh Bachchan', 'Jai', 1000000), ('Sholey', 1975, 'Dharmendra', 'Veeru', 800000), ('Lagaan', 2001, 'Aamir Khan', 'Bhuvan', 1200000);

INSERT INTO Movie_Producer VALUES ('Sholey', 1975, 1), ('Lagaan', 2001, 2), ('Lagaan', 2001, 1) ('Taal', 1999, 1);
```

Q.1) Create a View:

```
CREATE VIEW Actors_In_Mumbai AS SELECT a_name FROM Actor WHERE city = 'Mumbai';

SELECT * FROM Actors_In_Mumbai;

CREATE VIEW Actors_In_Movies AS SELECT ma.m_name, ma.release_year, a.a_name, ma.role, ma.charges FROM Movie_Actor ma JOIN Actor a ON ma.a_name = a.a_name;

SELECT * FROM Actors_In_Movies;
```

Q.2) Using above database solve following questions:

```
CREATE OR REPLACE FUNCTION check_budget()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.budget < 6000000 THEN
        RAISE EXCEPTION 'Budget must be at least 60 lakhs.';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER before_insert_budget
BEFORE INSERT ON Movie
FOR EACH ROW EXECUTE FUNCTION check_budget();

INSERT INTO Movie VALUES ('New Movie', 2023, 5000000);
```

```
CREATE OR REPLACE FUNCTION
count_movies_by_producer(producer_name CHAR(30))
RETURNS INTEGER AS $$
DECLARE
    movie_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO movie_count
    FROM Movie_Producer mp
    JOIN Producer p ON mp.producer_id = p.producer_id
    WHERE p.pname = producer_name;

    RETURN movie_count;
END;
$$ LANGUAGE plpgsql;

SELECT count_movies_by_producer('Mr. Subhash Ghai');
```