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)

Q.1) Create the following database in 3NF using PostgresSQL. [Total Marks: 10]

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 movie names produced by ‘Mr. Subhash Ghai’.
2. To display actor names who do not live in Mumbai or Pune city.

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

1. Write a trigger before inserting record into movie table; check release_year should not be greater than current year. Display appropriate message. **[10]**

2. Write a cursor using function to list movie-wise charges of ‘Amitabh Bachchan’. **[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 Movies_Produced_By_Subhash_Ghai AS SELECT mp.m_name, mp.release_year FROM Movie_Producer mp JOIN Producer p ON mp.producer_id = p.producer_id WHERE p.pname = 'Mr. Subhash Ghai';

SELECT * FROM Movies_Produced_By_Subhash_Ghai;

CREATE VIEW Actors_Not_In_Mumbai_Or_Pune AS SELECT a_name FROM Actor WHERE city NOT IN ('Mumbai', 'Pune');

SELECT * FROM Actors_Not_In_Mumbai_Or_Pune;
```

Q.2) Using above database solve following questions:

```
CREATE OR REPLACE FUNCTION check_release_year()
RETURNS TRIGGER AS $$
BEGIN
  IF NEW.release_year > EXTRACT(YEAR FROM CURRENT_DATE) THEN
    RAISE EXCEPTION 'Release year cannot be greater than the current year.';
  END IF;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER before_insert_release_year
BEFORE INSERT ON Movie
FOR EACH ROW EXECUTE FUNCTION check_release_year();

INSERT INTO Movie VALUES ('Tere Naam', 2025, 6000000);
```

```
CREATE OR REPLACE FUNCTION charges_of_amitabh_bachchan()
RETURNS VOID AS $$
DECLARE
  rec RECORD;
  cur CURSOR FOR
    SELECT ma.m_name, ma.release_year, ma.charges
    FROM Movie_Actor ma
    WHERE ma.a_name = 'Amitabh Bachchan';

BEGIN
  OPEN cur;
  LOOP
    FETCH cur INTO rec;
    EXIT WHEN NOT FOUND;
    RAISE NOTICE 'Movie: %, Year: %, Charges: %', rec.m_name, rec.release_year, rec.charges;
  END LOOP;
  CLOSE cur;
END;
$$ LANGUAGE plpgsql;

SELECT charges_of_amitabh_bachchan();
```