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 PostgresSQL. **[Total Marks: 10]**

Q1) Consider the following Student –Marks database
Student (Rollno integer, sname varchar (30), city varchar (50), class varchar (10))
Subject (Scode varchar (10), subject_name varchar (20))

Relationship:

Student-Subject related with many-to-many relationship with attributes marks_scored.

Constraints: Primary key, sname should not be null.

Create a View:

1. To display details of students whose name starts with the letter 'A'. **[10]**

2. To display details of students who has scored less than 40 marks.

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

1. Write a trigger to ensure that the marks entered for a student with respect to a subject is never < 0 and greater than 100. **[10]**

2. Write a stored function to accept city as an input parameter and display student details. **[10]**

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

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

Slip 17

STUDENT – MARKS DATABASE

```
CREATE TABLE Students (Rollno INTEGER PRIMARY KEY, sname VARCHAR(30) NOT NULL, city VARCHAR(50), class VARCHAR(10));

CREATE TABLE Subjects (Scode VARCHAR(10) PRIMARY KEY, subject_name VARCHAR(20));

CREATE TABLE Students_Subjects (Rollno INTEGER, Scode VARCHAR(10), marks_scored INTEGER, PRIMARY KEY (Rollno, Scode), FOREIGN KEY (Rollno) REFERENCES Students(Rollno), FOREIGN KEY (Scode) REFERENCES Subjects(Scode));

INSERT INTO Students (Rollno, sname, city, class) VALUES (1, 'Amit', 'Mumbai', 'FYBCA'), (2, 'Anjali', 'Pune', 'SYBCA'), (3, 'Rahul', 'Nagpur', 'TYBCA'), (4, 'Arjun', 'Nashik', 'FYBCA');

INSERT INTO Subjects (Scode, subject_name) VALUES ('S101', 'DBMS'), ('S102', 'Math'), ('S103', 'Networking');

INSERT INTO Students_Subjects (Rollno, Scode, marks_scored) VALUES (1, 'S101', 95), (1, 'S102', 85), (2, 'S101', 78), (2, 'S102', 88), (3, 'S101', 92), (3, 'S103', 65), (4, 'S101', 81), (4, 'S102', 38);
```

Q.1) Create a View:

```
CREATE VIEW Students_Name_Starts_A AS SELECT Rollno, sname, city, class FROM Students WHERE sname LIKE 'A%';

SELECT * FROM Students_Name_Starts_A;

CREATE VIEW Students_Scored_Below_40 AS SELECT s.Rollno, s.sname, s.city, s.class, sub.subject_name, ss.marks_scored FROM Students s JOIN Students_Subjects ss ON s.Rollno = ss.Rollno JOIN Subjects sub ON ss.Scode = sub.Scode WHERE ss.marks_scored < 40;

SELECT * FROM Students_Scored_Below_40;
```

Q.2) Using above database solve following questions:

```
CREATE OR REPLACE FUNCTION check_marks_range() RETURNS TRIGGER
AS $$
BEGIN
   IF NEW.marks_scored < 0 OR NEW.marks_scored > 100 THEN
      RAISE EXCEPTION 'Error: Marks must be between 0 and 100';
   END IF;
   RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER check_marks_before_insert
BEFORE INSERT OR UPDATE ON Students_Subjects
FOR EACH ROW
EXECUTE FUNCTION check_marks_range();

          INSERT INTO Students_Subjects (101, 'CS101', 85);

          INSERT INTO Students_Subjects VALUES (101, 'CS102', -5);

          INSERT INTO Students_Subjects VALUES (101, 'CS103', 105);
```

```
CREATE OR REPLACE FUNCTION
get_students_by_city(student_city VARCHAR) RETURNS VOID AS
$$
DECLARE
   student_rec RECORD;
BEGIN
   FOR student_rec IN
      SELECT Rollno, sname, city, class
      FROM Students
      WHERE city = student_city
   LOOP
      RAISE NOTICE 'Rollno: %, Name: %, City: %, Class: %',
student_rec.Rollno, student_rec.sname, student_rec.city,
student_rec.class;
   END LOOP;
END;
$$ LANGUAGE plpgsql;

SELECT get_students_by_city('Mumbai');
```