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-Teacher database maintained by a college. It also gives information of the subject taught by the teachers.

Student (Sno integer, sname varchar (20), sclass varchar (10), saddr varchar (30))

Teacher (Tno integer, tname varchar (20), qualification char (15), experience integer)

Relationship:

Student-Teacher related with many to many relationship with descriptive attribute subject.

Constraints: Primary Key, student and teacher name should not be null.

Create a View: **[10]**

1. To display details of teachers having experience > 5 years.
2. To display details of teachers whose name start with the letter ‘S’.

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

1. Write a trigger before update a student’s class from student table. Display appropriate message. **[10]**

2. Write a function to count the number of teachers who are teaching to a student named‘ _________’. (Accept student name as an input parameter). **[10]**

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

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

STUDENT-TEACHER DATABASE

```
CREATE TABLE Student (Sno INTEGER PRIMARY KEY, sname VARCHAR(20) NOT NULL, sclass VARCHAR(10), saddr VARCHAR(30));

CREATE TABLE Teacher (Tno INTEGER PRIMARY KEY, tname VARCHAR(20) NOT NULL, qualification CHAR(15), experience INTEGER);

CREATE TABLE Student_Teacher (Sno INTEGER REFERENCES Student(Sno), Tno INTEGER REFERENCES Teacher(Tno), subject VARCHAR(30), PRIMARY KEY (Sno, Tno));

INSERT INTO Student (Sno, sname, sclass, saddr) VALUES (1, 'Rahul', '10th', 'Pune'), (2, 'Sneha', '12th', 'Mumbai'), (3, 'Amit', '11th', 'Pune'), (4, 'Vijay', '10th', 'Nashik');

INSERT INTO Teacher (Tno, tname, qualification, experience) VALUES (1, 'Sharma', 'Ph.D.', 10), (2, 'Joshi', 'M.Sc.', 4), (3, 'Singh', 'Ph.D.', 7), (4, 'Gupta', 'M.A.', 5);

INSERT INTO Student_Teacher (Sno, Tno, subject) VALUES (1, 1, 'Mathematics'), (1, 3, 'Physics'), (2, 2, 'Chemistry'), (3, 1, 'Mathematics'), (4, 3, 'Biology');
```

Q.1) Create a View:

```
CREATE VIEW ExperiencedTeachers AS SELECT * FROM Teacher WHERE experience > 5;

SELECT * FROM ExperiencedTeachers;

CREATE VIEW TeachersStartingWithS AS SELECT * FROM Teacher WHERE tname LIKE 'S%';

SELECT * FROM TeachersStartingWithS;
```

Q.2) Using above database solve following questions:

```
CREATE OR REPLACE FUNCTION before_update_student_class()
RETURNS TRIGGER AS $$
BEGIN
   IF OLD.sclass IS DISTINCT FROM NEW.sclass THEN
      RAISE NOTICE 'Updating class for student: %', OLD.sname;
   END IF;
   RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER before_update_student_class_trigger
BEFORE UPDATE ON Student
FOR EACH ROW
EXECUTE FUNCTION before_update_student_class();

UPDATE Student SET sclass = '12th' WHERE sname = 'Rahul';
```

```
CREATE OR REPLACE FUNCTION
count_teachers_for_student(student_name VARCHAR)
RETURNS INTEGER AS $$
DECLARE
   teacher_count INTEGER;
BEGIN
   SELECT COUNT(DISTINCT t.Tno) INTO teacher_count
   FROM Teacher t
   JOIN Student_Teacher st ON t.Tno = st.Tno
   JOIN Student s ON st.Sno = s.Sno
   WHERE s.sname = student_name;

   RETURN teacher_count;
END;
$$ LANGUAGE plpgsql;

SELECT count_teachers_for_student('Rahul');
```