# 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 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 student names who are taught by most experienced teacher.

2. To display subjects taught by each teacher.

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

**1.** Write a trigger before inserting the student record. If the sno is less than or equal to zero, then display the message 'Invalid student number'.                                         **[10]**

**2.** Write a stored function to count the number of students studying a subject named '_' (Accept the subject's name as an input parameter). Display error message for invalid subject name.

**[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 MostExperiencedTeacher AS SELECT s.sname FROM Student s JOIN Student_Teacher st ON s.Sno = st.Sno JOIN Teacher t ON st.Tno = t.Tno WHERE t.experience = (SELECT MAX(experience) FROM Teacher);

SELECT * FROM MostExperiencedTeacher;

CREATE VIEW SubjectsByTeacher AS SELECT t.tname, st.subject FROM Teacher t JOIN Student_Teacher st ON t.Tno = st.Tno;

SELECT * FROM SubjectsByTeacher;

## Q.2) Using above database solve following questions:

```
CREATE OR REPLACE FUNCTION validate_student_number()
RETURNS TRIGGER AS $$
BEGIN
   IF NEW.Sno <= 0 THEN
      RAISE EXCEPTION 'Invalid student number';
   END IF;
   RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER before_insert_student
BEFORE INSERT ON Student
FOR EACH ROW
EXECUTE FUNCTION validate_student_number();

INSERT INTO Student VALUES (0, 'John Doe', '10th', 'Mumbai');
```

```
CREATE OR REPLACE FUNCTION
count_students_by_subject(subject_name VARCHAR)
RETURNS INTEGER AS $$
DECLARE
   student_count INTEGER;
BEGIN
   IF subject_name IS NULL OR subject_name = '' THEN
      RAISE EXCEPTION 'Invalid subject name';
   END IF;

   SELECT COUNT(DISTINCT s.Sno) INTO student_count
   FROM Student s
   JOIN Student_Teacher st ON s.Sno = st.Sno
   WHERE st.subject = subject_name;

   RETURN student_count;
END;
$$ LANGUAGE plpgsql;

SELECT count_students_by_subject('Mathematics');
```