

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 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 [10]

1. To display names of students class 'FYBCA'.
2. To display students name, subject and marks who has scored more than 90 marks.

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

1. Write a trigger before inserting Rollno into Student table. Display error message if entered Rollno less than equal to zero. [10]
2. Write a function using cursor, to calculate total marks of each student and display it. [10]

Q.3) External Viva [05]

Q.4) Internal Evaluation [15]

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_FYBCA AS SELECT sname FROM Students WHERE class = 'FYBCA';

SELECT * FROM Students_FYBCA;

CREATE VIEW Students_Scored_Above_90 AS SELECT s.sname, 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 > 90;

SELECT * FROM Students_Scored_Above_90;

Q.2) Using above database solve following questions:

```
CREATE OR REPLACE FUNCTION check_rollno() RETURNS TRIGGER
AS $$
BEGIN
    IF NEW.Rollno <= 0 THEN
        RAISE EXCEPTION 'Error: Roll number must be greater than
zero';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER check_rollno_before_insert
BEFORE INSERT ON Students
FOR EACH ROW
EXECUTE FUNCTION check_rollno();

INSERT INTO Students VALUES (101, 'Rahul', 'Pune', 'FYBCA');

INSERT INTO Students VALUES (0, 'Priya', 'Mumbai', 'SYBCA');
```

```
CREATE OR REPLACE FUNCTION calculate_total_marks() RETURNS VOID
AS $$
DECLARE
    student_rec RECORD;
    total INTEGER;
BEGIN
    FOR student_rec IN SELECT DISTINCT Rollno FROM Students_Subjects
    LOOP
        SELECT SUM(marks_scored) INTO total
        FROM Students_Subjects
        WHERE Rollno = student_rec.Rollno;

        RAISE NOTICE 'Rollno: %, Total Marks: %', student_rec.Rollno, total;
    END LOOP;
END;
$$ LANGUAGE plpgsql;

SELECT calculate_total_marks();
```