

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 the students name who scored more than 80 marks in 'DBMS' Subject.
2. To display student details of class 'TYBCA'.

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

1. Write a trigger after deleting a student record from the student table. Display the message "student record is being deleted". [10]
2. Write a stored function to accept student name as an input parameter and display their subject information. [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_DBMS_Above_80 AS SELECT s.sname FROM Students s JOIN Students_Subjects ss ON s.Rollno = ss.Rollno JOIN Subjects sub ON ss.Scode = sub.Scode WHERE sub.subject_name = 'DBMS' AND ss.marks_scored > 80;

SELECT * FROM Students_DBMS_Above_80;

CREATE VIEW Students_TYBCA_Details AS SELECT Rollno, sname, city, class FROM Students WHERE class = 'TYBCA';

SELECT * FROM Students_TYBCA_Details;
```

Q.2) Using above database solve following questions:

```
CREATE OR REPLACE FUNCTION notify_delete_student() RETURNS
TRIGGER AS $$
BEGIN
    RAISE NOTICE 'Student record is being deleted';
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER notify_student_delete
AFTER DELETE ON Students
FOR EACH ROW
EXECUTE FUNCTION notify_delete_student();

DELETE FROM Students WHERE Rollno = 101;
```

```
CREATE OR REPLACE FUNCTION get_subject_info(student_name
VARCHAR) RETURNS VOID AS $$
DECLARE
    subject_rec RECORD;
BEGIN
    FOR subject_rec IN
        SELECT sub.subject_name, ss.marks_scored
        FROM Students s
        JOIN Students_Subjects ss ON s.Rollno = ss.Rollno
        JOIN Subjects sub ON ss.Scode = sub.Scode
        WHERE s.sname = student_name
    LOOP
        RAISE NOTICE 'Subject: %, Marks: %',
subject_rec.subject_name, subject_rec.marks_scored;
    END LOOP;
END;
$$ LANGUAGE plpgsql;

SELECT get_subject_info('Amit');
```