

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 Project-Employee database, which is managed by a company and stores the details of projects assigned to employees.

Project (Pno int, pname varchar (30), ptype varchar (20), duration integer)

Employee (Eno integer, ename varchar (20), qualification char (15), joining_date date)

Relationship:

Project-Employee related with many-to-many relationship, with descriptive attributes as start_date_of_Project, no_of_hours_worked.

Constraints: Primary key, pname should not be null.

Create a View: [10]

1. To display the project name, project type, and project start date, sorted by project start date.
2. To display details of employees working on 'Robotics' project.

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

1. Write a trigger before inserting the duration into the project table and make sure that the duration is always greater than zero. Display appropriate message. [10]
2. Write function using cursor to accept project name as an input parameter and display names of employees working on that project. [10]

Q.3) External Viva [05]

Q.4) Internal Evaluation [15]

PROJECT-EMPLOYEE DATABASE

```
CREATE TABLE Project (Pno INTEGER PRIMARY KEY, pname VARCHAR(30) NOT NULL, ptype VARCHAR(20), duration INTEGER);

CREATE TABLE Employee (Eno INTEGER PRIMARY KEY, ename VARCHAR(20), qualification CHAR(15), joining_date DATE);

CREATE TABLE Project_Employee (Pno INTEGER, Eno INTEGER, start_date_of_project DATE, no_of_hours_worked INTEGER, PRIMARY KEY (Pno, Eno), FOREIGN KEY (Pno) REFERENCES Project(Pno), FOREIGN KEY (Eno) REFERENCES Employee(Eno));

INSERT INTO Project VALUES (1, 'Robotics', 'Research', 24), (2, 'ERP', 'Development', 18), (3, 'AI Model', 'Research', 12), (4, 'Web Application', 'Development', 9);

INSERT INTO Employee VALUES (101, 'Amit', 'B.Tech', '2020-01-10'), (102, 'Priya', 'MCA', '2021-03-15'), (103, 'Rahul', 'B.Sc', '2019-07-22'), (104, 'Sneha', 'M.Tech', '2022-06-10');

INSERT INTO Project_Employee VALUES (1, 101, '2022-05-01', 120), (2, 102, '2022-04-15', 90), (1, 103, '2021-08-10', 50), (3, 104, '2023-01-12', 60), (2, 101, '2022-08-22', 130), (4, 102, '2022-09-05', 70);
```

Q.1) Create a View:

```
CREATE OR REPLACE VIEW Project_Details AS SELECT p.pname, p.ptype, pe.start_date_of_project FROM Project p JOIN Project_Employee pe ON p.Pno = pe.Pno ORDER BY pe.start_date_of_project;

SELECT * FROM Project_Details;

CREATE OR REPLACE VIEW Employees_On_Robotics AS SELECT e.Eno, e.ename, e.qualification, e.joining_date FROM Employee e JOIN Project_Employee pe ON e.Eno = pe.Eno JOIN Project p ON p.Pno = pe.Pno WHERE p.pname = 'Robotics';

SELECT * FROM Employees_On_Robotics;
```

Q.2) Using above database solve following questions:

<pre>CREATE OR REPLACE FUNCTION check_duration() RETURNS TRIGGER AS \$\$ BEGIN IF NEW.duration <= 0 THEN RAISE EXCEPTION 'Duration must be greater than zero'; END IF; RETURN NEW; END; \$\$ LANGUAGE plpgsql; CREATE TRIGGER before_insert_duration BEFORE INSERT ON Project FOR EACH ROW EXECUTE FUNCTION check_duration(); INSERT INTO Project VALUES (101, 'AI Research', 'Research', 0);</pre>	<pre>CREATE OR REPLACE FUNCTION get_employees_by_project(p_name VARCHAR) RETURNS VOID AS \$\$ DECLARE emp_record RECORD; emp_cursor CURSOR FOR SELECT e.ename FROM Employee e JOIN Project_Employee pe ON e.Eno = pe.Eno JOIN Project p ON pe.Pno = p.Pno WHERE p.pname = p_name; BEGIN OPEN emp_cursor; LOOP FETCH emp_cursor INTO emp_record; EXIT WHEN NOT FOUND; RAISE NOTICE 'Employee Name: %', emp_record.ename; END LOOP; CLOSE emp_cursor; END; \$\$ LANGUAGE plpgsql; SELECT get_employees_by_project('Robotics');</pre>
--	---