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 Project-Employee database, which is managed by a company and storesthe 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 relationships, 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 employee details working on 'ERP' Project.
2. To display employee and project details where employees worked more than 100 hours.

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

1. Write a trigger after deleting Project record from Project table. Display the message "Project record is being deleted". **[10]**

2. Write a function to find the number of employees whose date of joining is before 03-10-2022. **[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 Employees_On_ERP 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 = 'ERP';

SELECT * FROM Employees_On_ERP;

CREATE OR REPLACE VIEW Employees_Worked_More_Than_100_Hours AS SELECT e.Eno, e.ename, p.pname, pe.no_of_hours_worked FROM Employee e JOIN Project_Employee pe ON e.Eno = pe.Eno JOIN Project p ON p.Pno = pe.Pno WHERE pe.no_of_hours_worked > 100;

SELECT * FROM Employees_Worked_More_Than_100_Hours;
```

Q.2) Using above database solve following questions:

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

CREATE TRIGGER project_delete_trigger
AFTER DELETE ON Project
FOR EACH ROW EXECUTE FUNCTION after_project_delete();


DELETE FROM Project WHERE Pno = 1;
```

```
        CREATE OR REPLACE FUNCTION
count_employees_before_joining_date(target_date DATE)
        RETURNS INTEGER AS $$
        DECLARE
          employee_count INTEGER;
        BEGIN
          SELECT COUNT(*)
          INTO employee_count
          FROM Employee
          WHERE joining_date < target_date;

          IF employee_count = 0 THEN
            RAISE EXCEPTION 'No employees found who joined before %',
target_date;
          END IF;

          RETURN employee_count;
        END;
        $$ LANGUAGE plpgsql;

        SELECT count_employees_before_joining_date('2022-10-03');
```