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]

Q.1) 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 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 employee details and it should be sorted by employee's joining date.
2. To display employee and project details where employees worked less than 100 hours.

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

1. Write a trigger before inserting joining date into employee table, check joining date should be always less than current date. Display appropriate message. **[10]**

2. Write a stored function to accept project name as an input parameter and returns the number of employees working on that project. Raise an exception for an invalid project name. **[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 Employee_Details AS SELECT Eno, ename, qualification, joining_date FROM Employee ORDER BY joining_date;

SELECT * FROM Employee_Details;

CREATE OR REPLACE VIEW Employees_Worked_Less_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_Less_Than_100_Hours;
```

Q.2) Using above database solve following questions:

```
CREATE OR REPLACE FUNCTION check_joining_date()
RETURNS TRIGGER AS $$
BEGIN
  IF NEW.joining_date >= CURRENT_DATE THEN
    RAISE EXCEPTION 'Joining date must be before the current date.';
  END IF;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER validate_joining_date
BEFORE INSERT OR UPDATE ON Employee
FOR EACH ROW
EXECUTE FUNCTION check_joining_date();

INSERT INTO Employee VALUES (101, 'Vijay', 'Bachelors', '2022-09-01');

INSERT INTO Employee VALUES (102, 'Anil', 'Masters', CURRENT_DATE);
```

```
CREATE OR REPLACE FUNCTION get_employee_count_by_project(p_project_name VARCHAR)
RETURNS INTEGER AS $$
DECLARE
  employee_count INTEGER;
BEGIN
  SELECT COUNT(*)
  INTO employee_count
  FROM Project
  WHERE pname = p_project_name;

  IF employee_count = 0 THEN
    RAISE EXCEPTION 'Invalid project name: %', p_project_name;
  END IF;

  SELECT COUNT(*)
  INTO employee_count
  FROM Project_Employee
  WHERE project_name = p_project_name;

  RETURN employee_count;
END;
$$ LANGUAGE plpgsql;

SELECT get_employee_count_by_project('Robotics');
```