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)

Q.1) Create the following database in 3NF using PostgresSQL. **[Total Marks: 10]**

Consider the following database of Bus-Transport System. Many buses run on one route. Drivers are allotted to the buses shift-wise.

Bus (Bus_no int , capacity int , depot_name varchar(20))

Route (Route_no int, source varchar(20), destination varchar (20), no_of_stations int)

Driver (Driver_no int , driver_name varchar(20), license_no int, address varchar (20),age int , salary float)

Relationship:

Bus and Route related with many to one relationship.

Bus and Driver related with many to many relationship with descriptive attributes, Shift – it can be 1 (Morning) or 2 (Evening) and Date_of_duty_allotted.

Constraints: Primary key, license_no must be unique, Bus capacity should not be null.

Create a View [**10]**

1. To display details of Bus_no 102 along with details of all drivers who have driven that bus.
2. To display the route details on which buses of capacity 30 runs.

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

1. Write a trigger before inserting the driver record in driver table, if the salary is less than or equal to zero, then return the error message 'Invalid Salary'. **[10]**

2. Write a function using cursor to display all the dates on which a driver has driven a bus (Accept the driver name as an input parameter) **[10]**

Q.3) External Viva **[05]**

Q.4) Internal Evaluation **[15]**

BUS-TRANSPORT SYSTEM

```
CREATE TABLE Bus (Bus_no INT PRIMARY KEY, capacity INT NOT NULL, depot_name VARCHAR(20));

CREATE TABLE Route (Route_no INT PRIMARY KEY, source VARCHAR(20), destination VARCHAR(20), no_of_stations INT);

CREATE TABLE Driver (Driver_no INT PRIMARY KEY, driver_name VARCHAR(20), license_no INT UNIQUE, address VARCHAR(20), age INT, salary DECIMAL);

CREATE TABLE Bus_Driver (Bus_no INT, Driver_no INT, Shift INT CHECK (Shift IN (1, 2)), Date_of_duty_allotted DATE, PRIMARY KEY (Bus_no, Driver_no, Shift, Date_of_duty_allotted), FOREIGN KEY (Bus_no) REFERENCES Bus(Bus_no), FOREIGN KEY (Driver_no) REFERENCES Driver(Driver_no));

CREATE TABLE Bus_Route (Bus_no INT PRIMARY KEY, Route_no INT, FOREIGN KEY (Bus_no) REFERENCES Bus(Bus_no), FOREIGN KEY (Route_no) REFERENCES Route(Route_no));

INSERT INTO Route VALUES (1, 'Mumbai', 'Pune', 5), (2, 'Nashik', 'Mumbai', 7), (3, 'Ahmednagar', 'Aurangabad', 6), (4, 'Pune', 'Nagpur', 10), (5, 'Mumbai', 'Goa', 8);

INSERT INTO Bus VALUES (101, 30, 'Depot A'), (102, 40, 'Depot B'), (103, 50, 'Depot C'), (104, 35, 'Depot D'), (105, 30, 'Depot E');

INSERT INTO Driver VALUES (1, 'Rajesh', 123456, 'Mumbai', 45, 25000), (2, 'Amit', 654321, 'Pune', 35, 18000), (3, 'Sunil', 789456, 'Nashik', 50, 22000), (4, 'Suresh', 111222, 'Aurangabad', 40, 24000), (5, 'Mahesh', 222333, 'Nagpur', 29, 26000), (6, 'Anil', 333444, 'Goa', 55, 27000);

INSERT INTO Bus_Route VALUES (101, 1), (102, 2), (103, 3), (104, 4), (105, 5);

INSERT INTO Bus_Driver VALUES (101, 1, 1, '2024-10-10'), (101, 1, 2, '2024-10-10'), (102, 2, 1, '2024-10-11'), (102, 3, 2, '2024-10-11'), (103, 4, 1, '2024-10-12'), (103, 5, 2, '2024-10-12'), (104, 5, 1, '2024-10-13'), (104, 6, 2, '2024-10-13'), (105, 1, 1, '2024-10-14'), (105, 6, 2, '2024-10-14');
```

Q.1) Create a View:

1.
```
CREATE VIEW Bus_102_Drivers AS SELECT b.Bus_no, b.capacity, b.depot_name, d.Driver_no, d.driver_name, d.license_no, bd.Shift, bd.Date_of_duty_allotted FROM Bus b JOIN Bus_Driver bd ON b.Bus_no = bd.Bus_no JOIN Driver d ON bd.Driver_no = d.Driver_no WHERE b.Bus_no = 102;
  SELECT * FROM Bus_102_Drivers;
```

2.
```
CREATE VIEW Route_Bus_Capacity_30 AS SELECT r.Route_no, r.source, r.destination, r.no_of_stations FROM Route r JOIN Bus_Route br ON r.Route_no = br.Route_no JOIN Bus b ON br.Bus_no = b.Bus_no WHERE b.capacity = 30;
  SELECT * FROM Route_Bus_Capacity_30;
```

Q.2) Using above database solve following questions:

1.
```
CREATE OR REPLACE FUNCTION check_driver_salary()
RETURNS TRIGGER AS $$
BEGIN
  IF NEW.salary <= 0 THEN
    RAISE EXCEPTION 'Invalid Salary: Salary must be greater than zero.';
  END IF;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_check_salary
BEFORE INSERT ON Driver
FOR EACH ROW
EXECUTE FUNCTION check_driver_salary();

INSERT INTO Driver VALUES (7, 'Karan', 445566, 'Chennai', 28, 0);

INSERT INTO Driver VALUES (7, 'Vijay', 223344, 'Goa', 32, 18000);
```

2.
```
CREATE OR REPLACE FUNCTION get_driver_dates(d_name VARCHAR)
RETURNS VOID AS $$
DECLARE
  rec RECORD;
BEGIN
  FOR rec IN
    SELECT bd.Date_of_duty_allotted
    FROM Bus_Driver bd
    JOIN Driver d ON bd.Driver_no = d.Driver_no
    WHERE d.driver_name = d_name
  LOOP
    RAISE NOTICE 'Date of duty: %', rec.Date_of_duty_allotted;
  END LOOP;

  IF NOT FOUND THEN
    RAISE NOTICE 'No records found for driver: %', d_name;
  END IF;
END;
$$ LANGUAGE plpgsql;

SELECT get_driver_dates('Rajesh');
```