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)

Q1) 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 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 driver details working in Morning shift.
2. To display driver details having salary > 20,000.

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 age is not between 18 and 35, then display error message 'Invalid input'. **[10]**

2. Write a stored function to display details of buses running on route_no = '_'. (Accept route_no 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 Morning_Shift_Drivers AS SELECT d.Driver_no, d.driver_name, d.license_no, d.address, d.age, d.salary FROM Driver d JOIN Bus_Driver bd ON d.Driver_no = bd.Driver_no WHERE bd.Shift = 1;
   SELECT * FROM Morning_Shift_Drivers;

2. CREATE VIEW High_Salary_Drivers AS SELECT * FROM Driver WHERE salary > 20000;
   SELECT * FROM High_Salary_Drivers;

Q.2) Using above database solve following questions:

1.
```
CREATE OR REPLACE FUNCTION check_driver_age()
RETURNS TRIGGER AS $$
BEGIN
IF NEW.age < 18 OR NEW.age > 35 THEN
RAISE EXCEPTION 'Invalid input: Driver age must be between 18 and 35.';
END IF;
RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_check_age
BEFORE INSERT ON Driver
FOR EACH ROW
EXECUTE FUNCTION check_driver_age();

INSERT INTO Driver VALUES (6, 'Manoj', 998877, 'Nagpur', 40, 20000);

INSERT INTO Driver VALUES (5, 'Suresh', 112233, 'Delhi', 30, 25000);
```

2.
```
CREATE OR REPLACE FUNCTION get_buses_by_route(route_id INT)
RETURNS VOID AS $$
DECLARE
  bus_record RECORD;
BEGIN
  RAISE NOTICE 'Function get_buses_by_route called with route_id: %', route_id;

  IF NOT EXISTS (SELECT 1 FROM Route WHERE Route_no = route_id) THEN
    RAISE EXCEPTION 'Route ID % does not exist', route_id;
  END IF;

  FOR bus_record IN
    SELECT b.Bus_no, b.capacity, b.depot_name
    FROM Bus b
    JOIN Bus_Route br ON b.Bus_no = br.Bus_no
    WHERE br.Route_no = route_id
  LOOP
    RAISE NOTICE 'Bus_no: %, Capacity: %, Depot_name: %',
          bus_record.Bus_no, bus_record.capacity, bus_record.depot_name;
  END LOOP;
END;
$$ LANGUAGE plpgsql;

SELECT get_buses_by_route(1);
```