Experiment 6

Student Name: Umang Kaushik **UID: 23BCS10712**
Branch: BE CSE **Section/Group: KRG-1B**
Semester: 4th **Date of Performance: 16th Sep**
Subject Name: DBMS **Subject Code: 23CSP-333**

1. Questions and Code:

Question 1 -

TechSphere Solutions, a growing IT services company with offices across India, wants to track and monitor gender diversity within its workforce. The HR department frequently needs to know the total number of employees by gender (Male or Female) . To solve this problem, the company needs an automated database-driven solution that can instantly return the count of employees by gender through a stored procedure that:

1. Create a PostgreSQL stored procedure that:
2. Takes a gender (e.g., 'Male' or 'Female') as input.
3. Calculates the total count of employees for that gender.
4. Returns the result as an output parameter.
5. Displays the result clearly for HR reporting purposes.

Code -

```
CREATE TABLE employees (
    employee_id SERIAL PRIMARY KEY,
    employee_name VARCHAR(100),
```

```
    gender VARCHAR(10)
);

INSERT INTO employees (employee_name, gender) VALUES
('Aarav Sharma', 'Male'),
('Priya Patel', 'Female'),
('Rohan Mehta', 'Male'),
('Anika Gupta', 'Female'),
('Vikram Singh', 'Male'),
('Saanvi Reddy', 'Female'),
('Aditya Kumar', 'Male');

CREATE OR REPLACE PROCEDURE
get_employee_count_by_gender(
    IN p_gender VARCHAR(10),
    OUT p_count INT
)
LANGUAGE plpgsql
AS $$
BEGIN
    SELECT COUNT(*)
    INTO p_count
    FROM employees
    WHERE gender = p_gender;
END;
$$;

CALL get_employee_count_by_gender('Male', 0);
CALL get_employee_count_by_gender('Female', 0);
```

Question 2 –

SmartShop is a modern retail company that sells electronic gadgets like smartphones, tablets, and laptops.
The company wants to automate its ordering and inventory management process. Whenever a customer places an order, the system must:
1. Verify stock availability for the requested product and quantity.
2. If sufficient stock is available:
- Log the order in the sales table with the ordered quantity and total price. - Update the inventory in the products table by reducing quantity_remaining and increasing quantity_sold. - Display a real-time confirmation message: “Product sold successfully!”
3. If there is insufficient stock, the system must: - Reject the transaction and display: “Insufficient Quantity Available!”

Code –

```
CREATE TABLE products (
    product_id SERIAL PRIMARY KEY,
    product_name VARCHAR(100),
    price NUMERIC,
    quantity_remaining INT,
    quantity_sold INT DEFAULT 0
);


CREATE TABLE sales (
    sale_id SERIAL PRIMARY KEY,
    product_id INT REFERENCES products(product_id),
    quantity_ordered INT,
```

```
    total_price NUMERIC,
    sale_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);


INSERT INTO products (product_name, price, quantity_remaining)
VALUES
('Smartphone X', 60000.00, 50),
('Tablet Pro', 45000.00, 30),
('Laptop Ultra', 120000.00, 15);


CREATE OR REPLACE PROCEDURE place_order(
    p_product_id INT,
    p_ordered_quantity INT
)
LANGUAGE plpgsql
AS $$
DECLARE
    v_available_quantity INT;
    v_product_price NUMERIC;
BEGIN
    SELECT quantity_remaining, price
    INTO v_available_quantity, v_product_price
    FROM products
    WHERE product_id = p_product_id
```

```
    FOR UPDATE;

    IF v_available_quantity >= p_ordered_quantity THEN

        UPDATE products
        SET
            quantity_remaining = quantity_remaining - p_ordered_quantity,
            quantity_sold = quantity_sold + p_ordered_quantity
        WHERE product_id = p_product_id;

        INSERT INTO sales (product_id, quantity_ordered, total_price)
        VALUES (p_product_id, p_ordered_quantity, v_product_price *
p_ordered_quantity);

        RAISE NOTICE 'Product sold successfully!';
    ELSE

        RAISE EXCEPTION 'Insufficient Quantity Available!';
    END IF;
END;
$$;
CALL place_order(1, 5);

CALL place_order(3, 20);
```