Experiment 6

Student Name: Bharat **UID: 23BCS13947**
Branch: CSE **Section/Group: KRG 3-A**
Semester: 5th **Date of Performance:25/09/2025**
Subject Name: ADBMS **Subject Code: 23CSP-333**

1. Aim:

Problem 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:

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

Problem 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:

- Verify stock availability for the requested product and quantity.
- 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!“ ○ If there is insufficient stock, the system must:
  - Reject the transaction and display: Insufficient Quantity Available!"

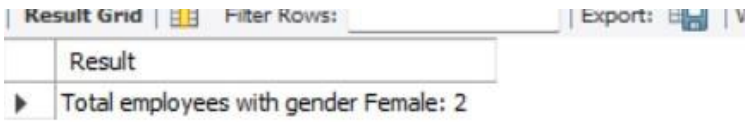
2. Objective:

- To learn how to design and implement stored procedures in PostgreSQL for business automation.
- To practice handling input/output parameters in stored procedures for dynamic reporting. ○ To understand database transaction handling for order management (commit/rollback). ○ To automate real-time updates of inventory and sales records in relational databases. ○ To gain skills in writing error-handling logic for business rules such as stock availability..

3. DBMS script and output:

----------Problem 1----------

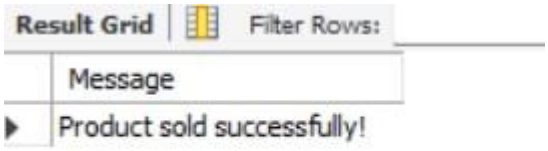
```
CREATE TABLE employees (
emp_id SERIAL PRIMARY KEY,
emp_name VARCHAR(100),
gender VARCHAR(10)
);

INSERT INTO employees (emp_name, gender) VALUES
('Amit Sharma', 'Male'),
('Priya Singh', 'Female'), ('Rohan
Verma', 'Male'),
('Neha Gupta', 'Female'),
('Kunal Mehta', 'Male');

CREATE OR REPLACE PROCEDURE get_employee_count_by_gender(
    IN p_gender VARCHAR,
    OUT p_count INT
)
LANGUAGE plpgsql
AS $$
BEGIN
    SELECT COUNT(*) INTO p_count
    FROM employees
    WHERE gender = p_gender;
    RAISE NOTICE 'Total employees with gender %: %', p_gender, p_count;
END;
$$;

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



----------Problem 2----------

```
CREATE TABLE products (
product_id SERIAL PRIMARY KEY,
product_name VARCHAR(100),     price
NUMERIC(10,2),     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(10,2),
   sale_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

INSERT INTO products (product_name, price, quantity_remaining) VALUES
('Smartphone', 15000, 10),
('Tablet', 25000, 5),
('Laptop', 60000, 3);

CREATE OR REPLACE PROCEDURE process_order(
   IN p_product_id INT,
   IN p_quantity INT
)
LANGUAGE plpgsql
AS $$
DECLARE
   v_price NUMERIC(10,2);
v_available INT;     v_total
NUMERIC(10,2);
BEGIN
   SELECT price, quantity_remaining INTO v_price, v_available
   FROM products
   WHERE product_id = p_product_id;

   IF v_available >= p_quantity THEN
v_total := v_price * p_quantity;
      INSERT INTO sales(product_id, quantity_ordered, total_price)

      VALUES (p_product_id, p_quantity, v_total);
      UPDATE products
```

```
    SET quantity_remaining = quantity_remaining - p_quantity,
quantity_sold = quantity_sold + p_quantity        WHERE
product_id = p_product_id;
    RAISE NOTICE 'Product sold successfully!';
  ELSE
    RAISE NOTICE 'Insufficient Quantity Available!';
  END IF;
END;
$$;

CALL process_order(1, 3);
CALL process_order(3, 5);
```



4. Learning Outcomes (What I have Learnt):

- Able to create stored procedures that return calculated values for business reporting (e.g., employee gender counts).
- The ability to process conditional logic (IF-ELSE) inside stored procedures to handle stock validation.
- Learned how to integrate multiple tables (employees, products, sales) in stored procedures for data consistency.
- Implemented real-time inventory updates in response to sales transactions.
- understood how automation reduces manual effort and ensures accuracy in HR and retail operations.