University Institute of Engineering

Department of Computer Science & Engineering

EXPERIMENT:6

NAME : SATYA PRAKASH SWAIN **UID : 23BCS11072**
BRANCH : BE-CSE **SECTION : KRG_1A**
SEMESTER : 5TH **SUBJECT : 23CSP-339**
SUBJECT NAME : ADBMS

1. AIM:-

[MEDIUM]

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.

[HARD]

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!"

2. TOOLS USED :-

PgAdmin4

3. CODE:-

[MEDIUM]

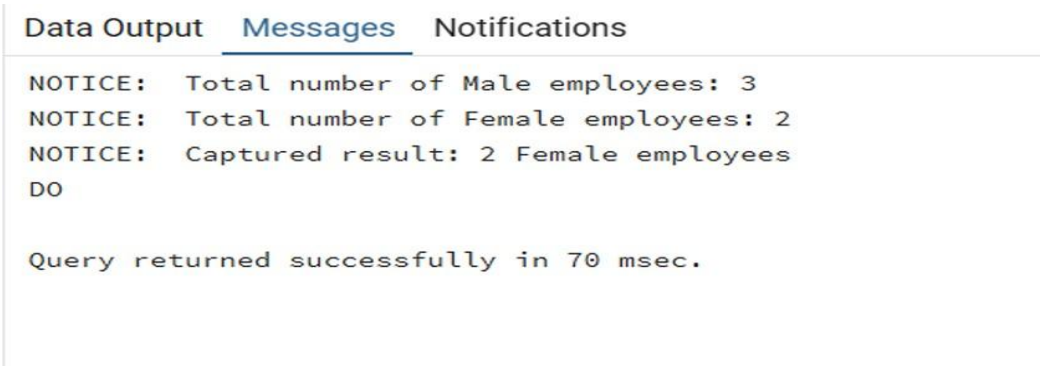
```
CREATE OR REPLACE PROCEDURE sp_get_employees_by_gender(
   IN p_gender VARCHAR(50),
   OUT p_employee_count INT
)
LANGUAGE plpgsql
AS $$
BEGIN
   -- Count total employees by gender
   SELECT COUNT(id)
   INTO p_employee_count
   FROM employee_info
   WHERE gender = p_gender;

   -- Display the result
   RAISE NOTICE 'Total employees with gender %: %', p_gender, p_employee_count;
END;
$$;

CALL sp_get_employees_by_gender('Male', NULL);
```

[HARD]

```
CREATE OR REPLACE PROCEDURE
   pr_buy_products( IN p_product_name VARCHAR,
   IN p_quantity INT
)
LANGUAGE plpgsql
AS $$
DECLARE
   v_product_code VARCHAR(20);
   v_price FLOAT;
   v_count INT;
BEGIN
   -- Step 1: Check if product exists and has enough quantity
   SELECT COUNT(*)
   INTO v_count
```

```
    FROM products
    WHERE product_name = p_product_name
    AND quantity_remaining >= p_quantity;

    -- Step 2: If sufficient stock
    IF v_count > 0 THEN
        -- Fetch product code and price
        SELECT product_code, price
        INTO v_product_code, v_price
        FROM products
        WHERE product_name = p_product_name;

        -- Insert a new record into the sales table
        INSERT INTO sales (order_date, product_code, quantity_ordered, sale_price)
        VALUES (CURRENT_DATE, v_product_code, p_quantity, (v_price * p_quantity));

        -- Update stock details
        UPDATE products
        SET quantity_remaining = quantity_remaining - p_quantity,
            quantity_sold = quantity_sold + p_quantity
        WHERE product_code = v_product_code;
        -- Confirmation message
        RAISE NOTICE 'PRODUCT SOLD..! Order placed successfully for % unit(s) of %.',
p_quantity, p_product_name;

    ELSE
        -- Step 3: If stock is insufficient
        RAISE NOTICE 'INSUFFICIENT QUANTITY..! Order cannot be processed for % unit(s)
of %.', p_quantity, p_product_name;
    END IF;
END;
$$;
CALL pr_buy_products ('MacBook Pro 14"', 1);
```

4. OUTPUT:-



[HARD]

Data Output | Messages | Notifications

```
NOTICE:  Product sold successfully!
CALL

Query returned successfully in 62 msec.
```

Data Output | Messages | Notifications

	name character varying (20)	quantity_remaining integer	quantity_sold integer
1	Laptop	5	2
2	IPhone	3	1
3	Oven	4	2

Data Output | Messages | Notifications

	sale_id [PK] integer	customer_name character varying (100)	product_name character varying (100)	quantity integer	price numeric	sale_date date
1	1	Alice	Laptop	2	800	2025-01-01
2	2	Bob	Mouse	5	20	2025-01-02
3	3	Charlie	Keyboard	3	50	2025-01-03
4	4	Alice	Laptop	1	800	2025-01-04

5. LEARNING OUTCOMES:-

1. Ability to write PostgreSQL stored procedures with input and output parameters.
2. Understanding how to aggregate and filter data using COUNT() and WHERE.
3. Learn to automate HR reporting and other repetitive tasks.
4. Skills in transaction management for multiple related database operations.
5. Implementing conditional logic in SQL to handle different scenarios.
6. Understanding inventory and sales tracking in real-time.
7. Ability to update multiple tables consistently to maintain data integrity.

8. Learn to provide clear, user-friendly output for reporting or confirmations.
9. Understanding how to automate business processes to reduce manual effort.
10. Awareness of performance and efficiency considerations in database operations.