

# University Institute of Engineering Department of Computer Science & Engineering

# **EXPERIMENT:6**

NAME : VANSH KUMAR

BRANCH : BE-CSE

SEMESTER : 5<sup>TH</sup>

UID : 23BCS10117

SECTION : KRG\_1A

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
```

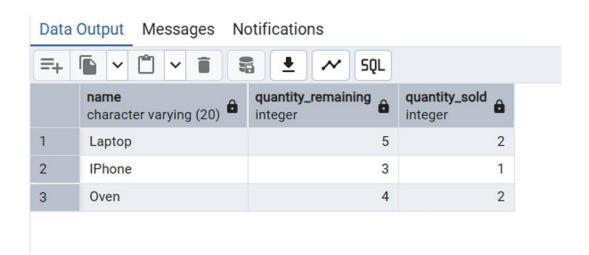
```
FROM products
  WHERE product name = p product name
  AND quantity remaining >= p quantity;
  -- Step 2: If sufficient stock
  IF v count > 0 THEN
    -- Fetch product code and price
    SELECT product code, price
    INTO v product code, v price
    FROM products
    WHERE product name = p product 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:-
    Data Output Messages Notifications
    NOTICE: Total number of Male employees: 3
    NOTICE: Total number of Female employees: 2
    NOTICE: Captured result: 2 Female employees
```

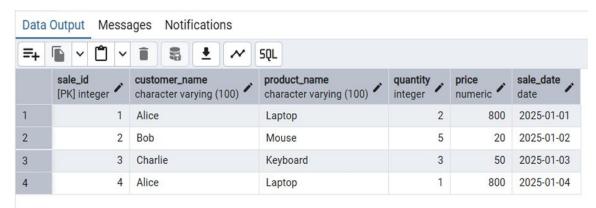
DO

Query returned successfully in 70 msec.

# [HARD]

Data Output	Messages	Notifications
NOTICE: Pro	oduct sold	successfully!
Query retur	ned success	fully in 62 msec.





# **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.