REVIEW – 1

1. Project Title : INVENTORY MANAGEMENT SYSTEM

Course : Database Management Systems (DBMS) Mini Project

Team Members :

Noshita Reddy	PES1UG23CS409
P R Mythri	PES1UG23CS413
Narasappagari Himavarshini	PES1UG23CS382

2.Description of the Problem Statement (Abstract) :

The **Inventory Management System (IMS)** is designed to efficiently manage stock, suppliers, customers, purchase orders, and sales orders in a business environment. Traditional manual methods often lead to problems like stock shortages, delays in procurement, and difficulty in tracking sales and purchases.

This system provides a structured **database-driven approach** to manage:

- Product details, suppliers, and customers.
- Recording of purchase and sales transactions.
- Tracking of available stock.
- Employee responsibilities in handling orders.

By implementing this system, businesses can ensure better stock control, streamlined purchase/sales operations, and accurate reporting for decision-making.

3.User Requirement Specification (URS) :

Functional Requirements --

Supplier Management

- Add, update, and delete supplier details.
- View list of suppliers and contact details.

Product Management

- Add, update, and delete product details.
- Track available stock, unit price .
- Link each product with a supplier.

Customer Management

- Maintain customer details.

Purchase Order Management

- Create purchase orders for suppliers.
- Track ordered products, quantities, and costs.

Sales Order Management

- Create sales orders for customers.
- Add multiple products to each sales order.
- Calculate total amount for each order.

Employee Management

- Assign employees to handle purchase and sales orders.
- Maintain employee roles (Manager, Admin and Staff).

Inventory/Stock Management

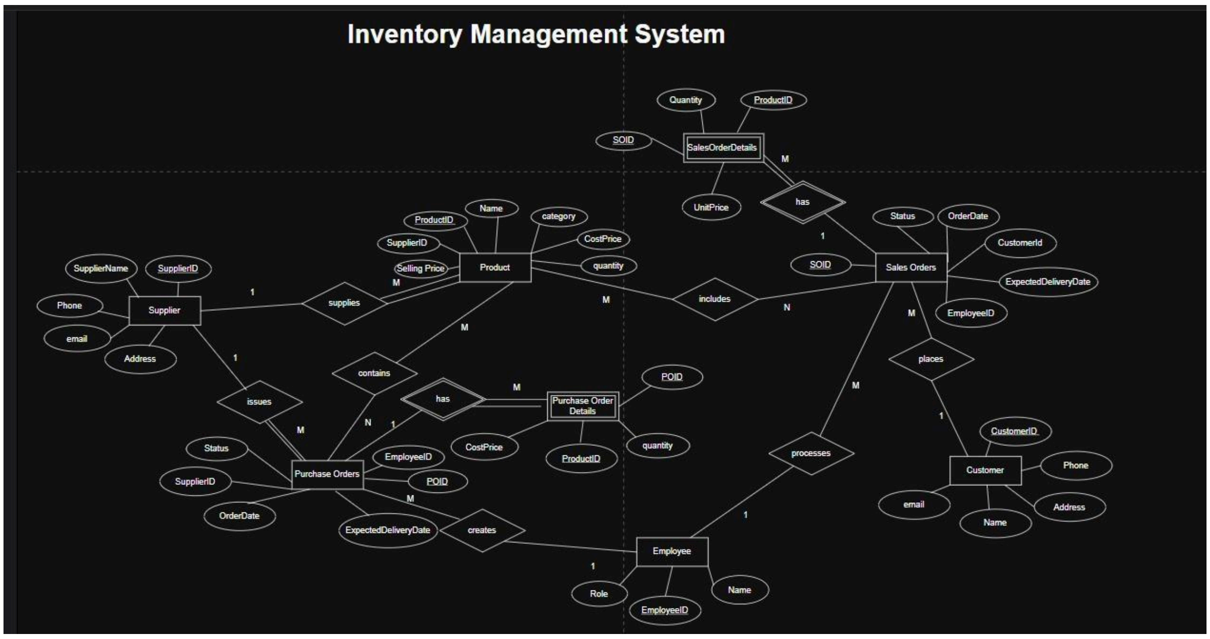
- Track stock availability and update quantity after purchase/sales transactions.
- Link stock information with product and supplier details.

Non – Functional Requirements --

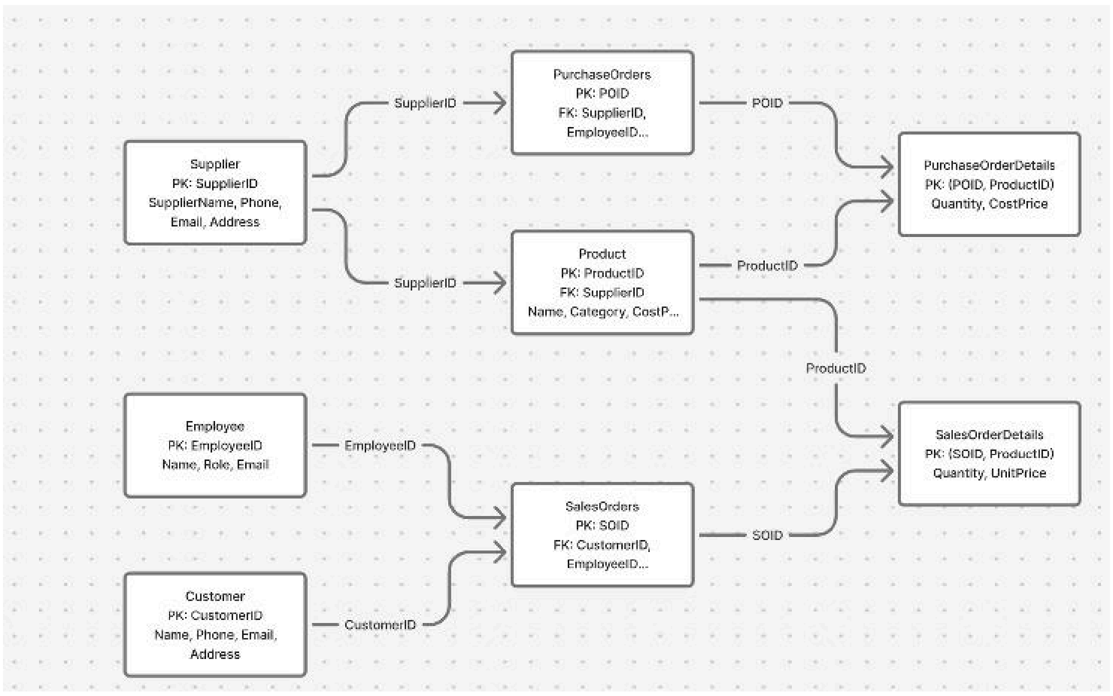
- **Usability**: Easy-to-use interface (GUI) for employees.
- **Reliability**: Data consistency through primary and foreign keys.
- **Scalability**: Can support large number of products and orders.
- **Security**: Ensure referential integrity and constraints (CHECK, UNIQUE, NOT NULL)

4. List of Software/Tools/Programming Languages Used

Draw.io



Relational Schema :



DDL Commands:

```
-- Create the database for the Inventory Management System
CREATE DATABASE IF NOT EXISTS
InventoryManagementSystem;

-- Select the newly created database to work with USE
InventoryManagementSystem;

--
=====================================================
================ -- TABLE CREATION --
=====================================================
================

-- Create the Suppliers table -- This table stores information
about product suppliers. CREATE TABLE Suppliers ( SupplierID
INT PRIMARY KEY AUTO_INCREMENT, SupplierName
VARCHAR(255) NOT NULL, Phone VARCHAR(20), Email
VARCHAR(255) UNIQUE, Address TEXT );

-- Create the Employees table -- This table stores information
about employees who manage orders. CREATE TABLE
Employees ( EmployeeID INT PRIMARY KEY
AUTO_INCREMENT, Name VARCHAR(255) NOT NULL, Role
VARCHAR(100) );

-- Create the Customers table -- This table stores information
about customers who place sales orders. CREATE TABLE
Customers ( CustomerID INT PRIMARY KEY
AUTO_INCREMENT, Name VARCHAR(255) NOT NULL, Phone
VARCHAR(20), Email VARCHAR(255) UNIQUE, Address TEXT );
```

-- Create the Products table -- This table holds all product details. It references the Suppliers table. CREATE TABLE Products (ProductID INT PRIMARY KEY AUTO_INCREMENT, Name VARCHAR(255) NOT NULL, Category VARCHAR(100), CostPrice DECIMAL(10, 2) NOT NULL, SellingPrice DECIMAL(10, 2) NOT NULL, Quantity INT NOT NULL DEFAULT 0, SupplierID INT, FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID) ON DELETE SET NULL);

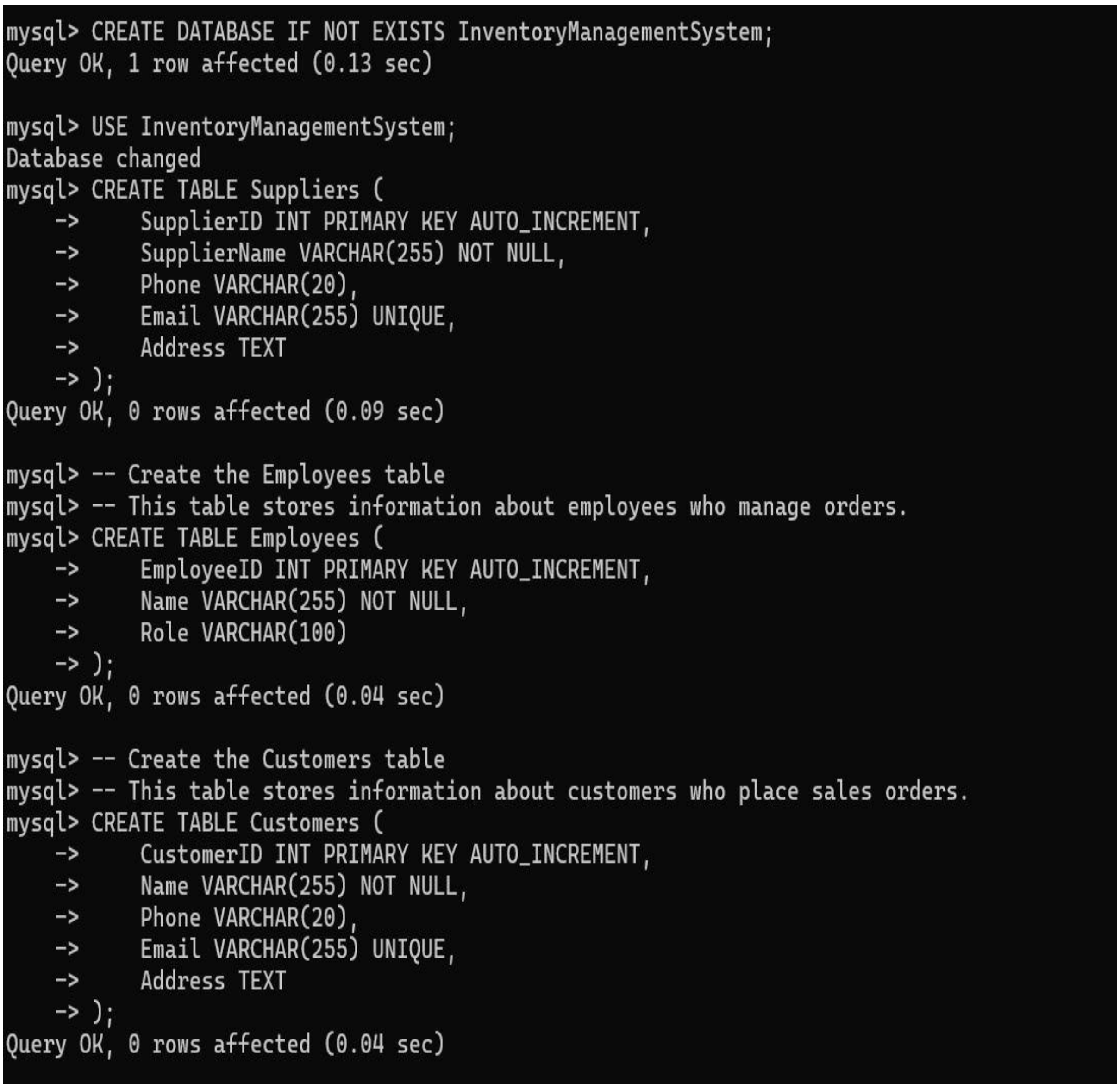
-- Create the PurchaseOrders table -- This table stores header information for orders placed with suppliers. CREATE TABLE PurchaseOrders (POID INT PRIMARY KEY AUTO_INCREMENT, OrderDate DATE NOT NULL, ExpectedDeliveryDate DATE, Status VARCHAR(50) DEFAULT 'Pending', SupplierID INT, EmployeeID INT, FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID) ON DELETE RESTRICT, FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID) ON DELETE SET NULL);

-- Create the SalesOrders table -- This table stores header information for sales orders from customers. CREATE TABLE SalesOrders (SOID INT PRIMARY KEY AUTO_INCREMENT, OrderDate DATE NOT NULL, ExpectedDeliveryDate DATE, Status VARCHAR(50) DEFAULT 'Pending', CustomerID INT, EmployeeID INT, FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) ON DELETE RESTRICT, FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID) ON DELETE SET NULL);

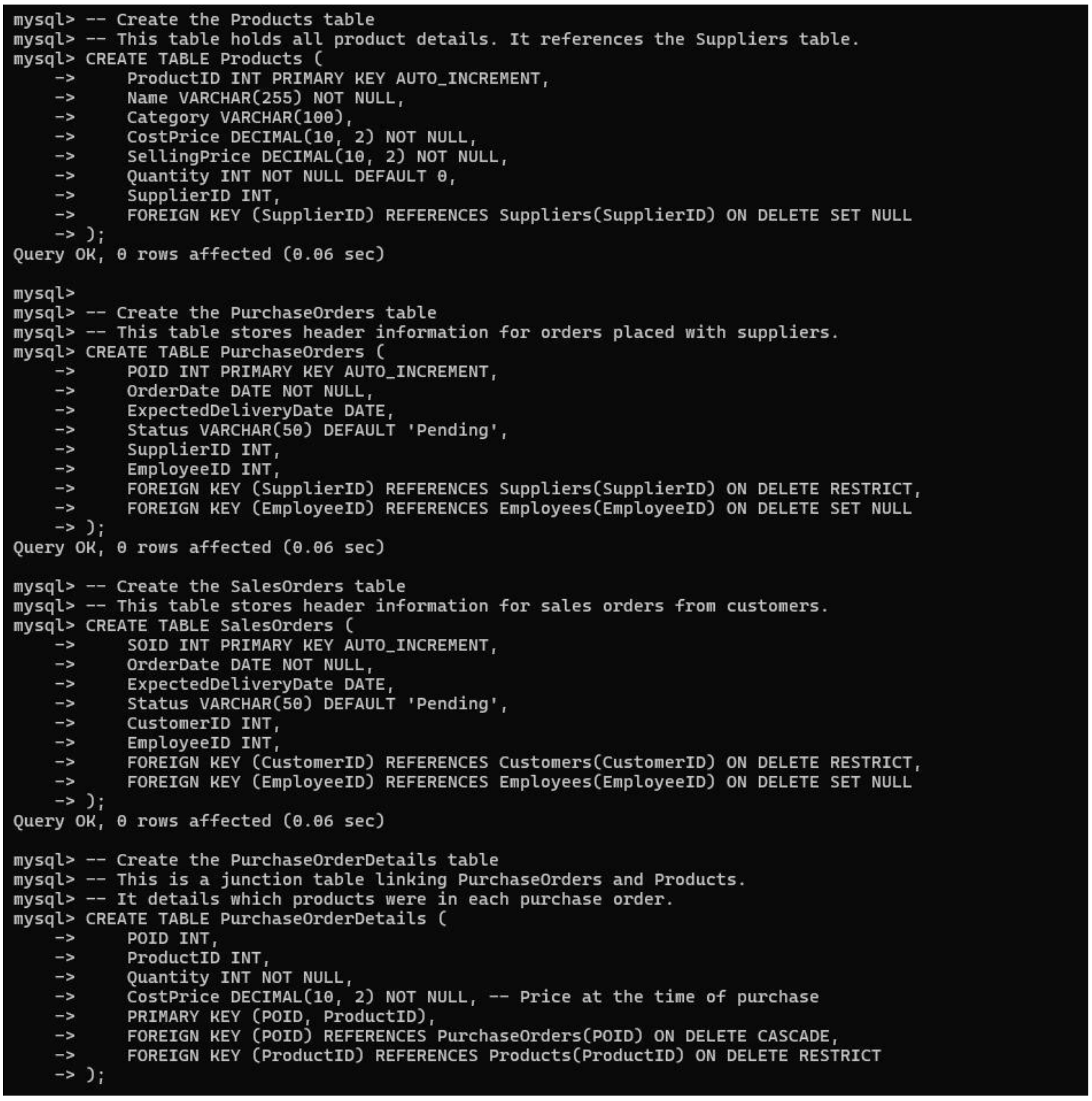
-- Create the PurchaseOrderDetails table -- This is a junction table linking PurchaseOrders and Products. -- It details which products were in each purchase order. CREATE TABLE PurchaseOrderDetails (POID INT, ProductID INT, Quantity INT NOT NULL, CostPrice DECIMAL(10, 2) NOT NULL, -- Price at the time of purchase PRIMARY KEY (POID, ProductID), FOREIGN KEY (POID) REFERENCES PurchaseOrders(POID) ON DELETE CASCADE, FOREIGN KEY (ProductID) REFERENCES Products(ProductID) ON DELETE RESTRICT);

-- Create the SalesOrderDetails table -- This is a junction table linking SalesOrders and Products. -- It details which products were in each sales order. CREATE TABLE SalesOrderDetails (SOID INT, ProductID INT, Quantity INT NOT NULL, UnitPrice DECIMAL(10, 2) NOT NULL, -- Price at the time of sale PRIMARY KEY (SOID, ProductID), FOREIGN KEY (SOID) REFERENCES SalesOrders(SOID) ON DELETE CASCADE, FOREIGN KEY (ProductID) REFERENCES Products(ProductID) ON DELETE RESTRICT);

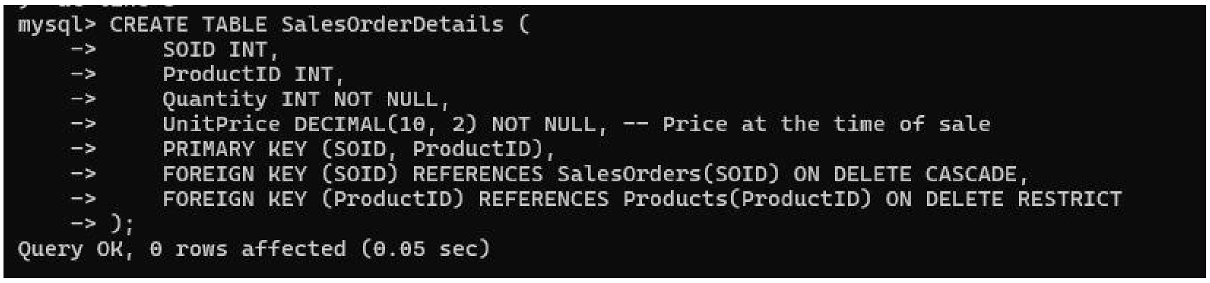
DDL AND DML SCREENSHOTS :



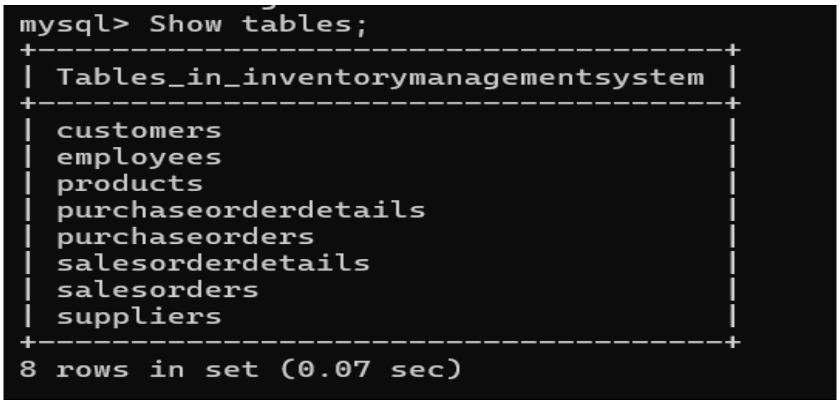
```
mysql> CREATE DATABASE IF NOT EXISTS InventoryManagementSystem;
Query OK, 1 row affected (0.13 sec)

mysql> USE InventoryManagementSystem;
Database changed
mysql> CREATE TABLE Suppliers (
    ->     SupplierID INT PRIMARY KEY AUTO_INCREMENT,
    ->     SupplierName VARCHAR(255) NOT NULL,
    ->     Phone VARCHAR(20),
    ->     Email VARCHAR(255) UNIQUE,
    ->     Address TEXT
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql> -- Create the Employees table
mysql> -- This table stores information about employees who manage orders.
mysql> CREATE TABLE Employees (
    ->     EmployeeID INT PRIMARY KEY AUTO_INCREMENT,
    ->     Name VARCHAR(255) NOT NULL,
    ->     Role VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> -- Create the Customers table
mysql> -- This table stores information about customers who place sales orders.
mysql> CREATE TABLE Customers (
    ->     CustomerID INT PRIMARY KEY AUTO_INCREMENT,
    ->     Name VARCHAR(255) NOT NULL,
    ->     Phone VARCHAR(20),
    ->     Email VARCHAR(255) UNIQUE,
    ->     Address TEXT
    -> );
Query OK, 0 rows affected (0.04 sec)
```



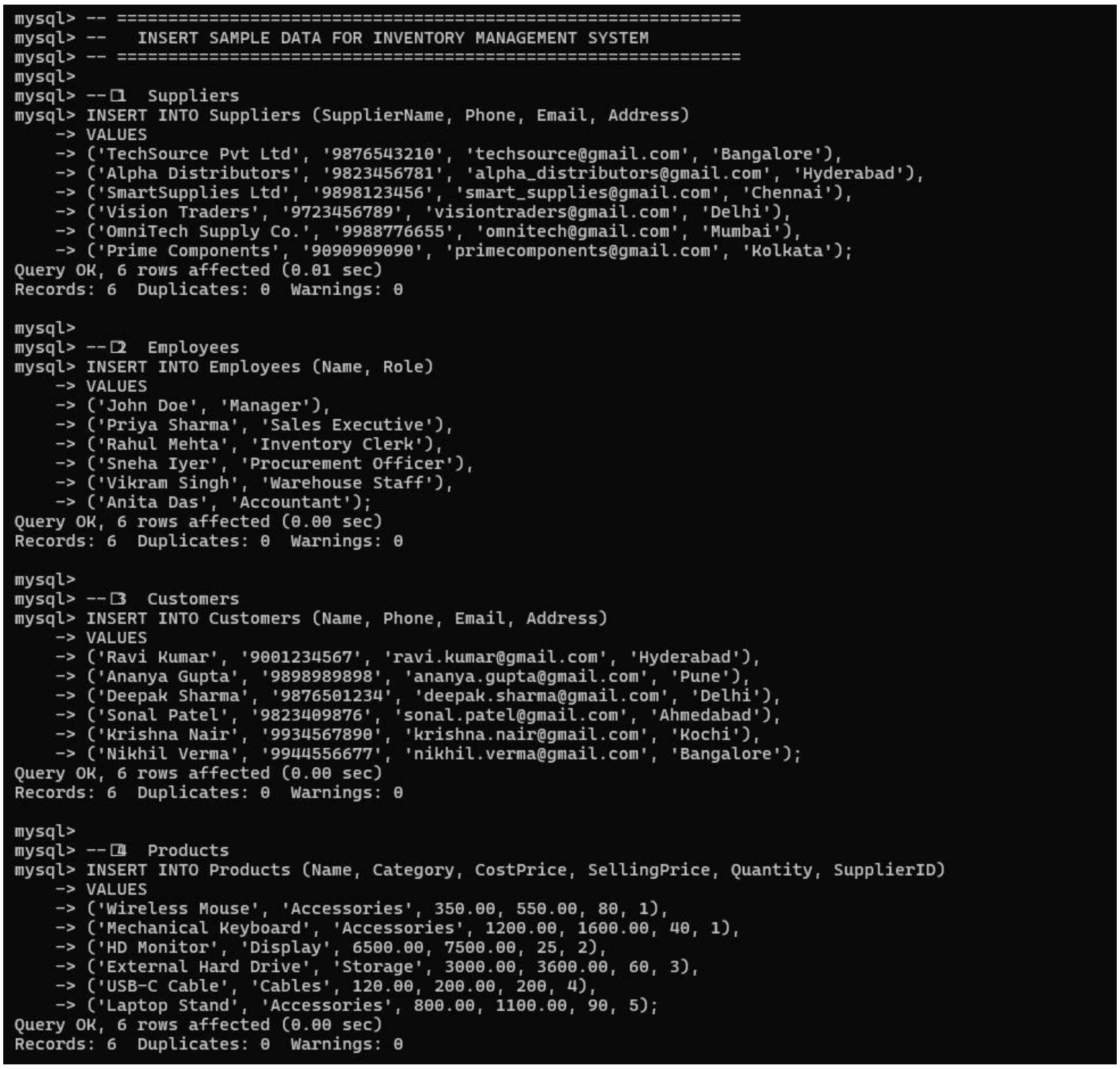
```
mysql> -- Create the Products table
mysql> -- This table holds all product details. It references the Suppliers table.
mysql> CREATE TABLE Products (
    ->     ProductID INT PRIMARY KEY AUTO_INCREMENT,
    ->     Name VARCHAR(255) NOT NULL,
    ->     Category VARCHAR(100),
    ->     CostPrice DECIMAL(10, 2) NOT NULL,
    ->     SellingPrice DECIMAL(10, 2) NOT NULL,
    ->     Quantity INT NOT NULL DEFAULT 0,
    ->     SupplierID INT,
    ->     FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID) ON DELETE SET NULL
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql>
mysql> -- Create the PurchaseOrders table
mysql> -- This table stores header information for orders placed with suppliers.
mysql> CREATE TABLE PurchaseOrders (
    ->     POID INT PRIMARY KEY AUTO_INCREMENT,
    ->     OrderDate DATE NOT NULL,
    ->     ExpectedDeliveryDate DATE,
    ->     Status VARCHAR(50) DEFAULT 'Pending',
    ->     SupplierID INT,
    ->     EmployeeID INT,
    ->     FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID) ON DELETE RESTRICT,
    ->     FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID) ON DELETE SET NULL
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> -- Create the SalesOrders table
mysql> -- This table stores header information for sales orders from customers.
mysql> CREATE TABLE SalesOrders (
    ->     SOID INT PRIMARY KEY AUTO_INCREMENT,
    ->     OrderDate DATE NOT NULL,
    ->     ExpectedDeliveryDate DATE,
    ->     Status VARCHAR(50) DEFAULT 'Pending',
    ->     CustomerID INT,
    ->     EmployeeID INT,
    ->     FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) ON DELETE RESTRICT,
    ->     FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID) ON DELETE SET NULL
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> -- Create the PurchaseOrderDetails table
mysql> -- This is a junction table linking PurchaseOrders and Products.
mysql> -- It details which products were in each purchase order.
mysql> CREATE TABLE PurchaseOrderDetails (
    ->     POID INT,
    ->     ProductID INT,
    ->     Quantity INT NOT NULL,
    ->     CostPrice DECIMAL(10, 2) NOT NULL, -- Price at the time of purchase
    ->     PRIMARY KEY (POID, ProductID),
    ->     FOREIGN KEY (POID) REFERENCES PurchaseOrders(POID) ON DELETE CASCADE,
    ->     FOREIGN KEY (ProductID) REFERENCES Products(ProductID) ON DELETE RESTRICT
    -> );
```



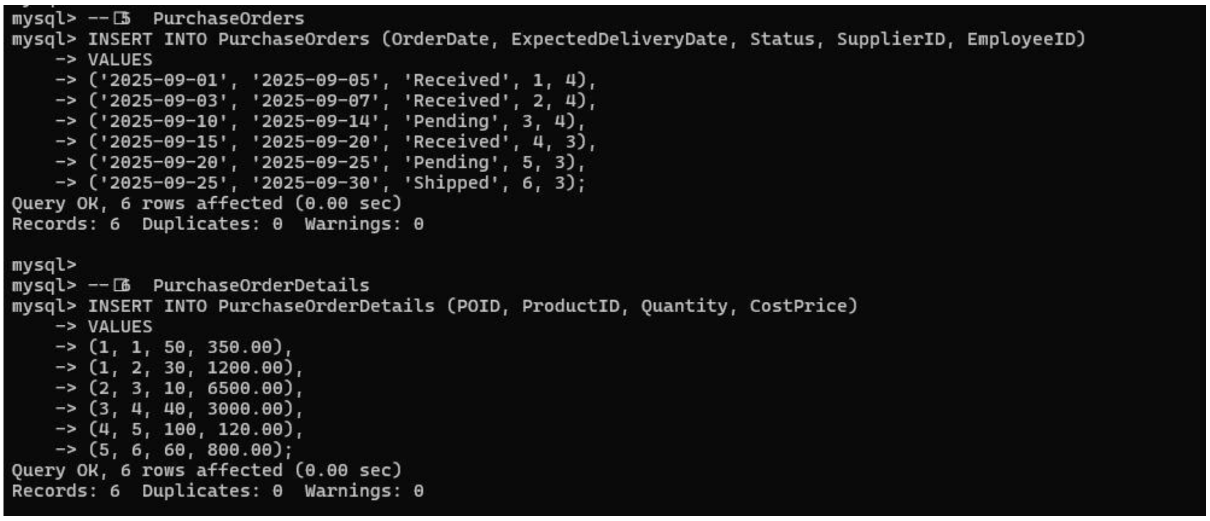
```
mysql> CREATE TABLE SalesOrderDetails (
    ->     SOID INT,
    ->     ProductID INT,
    ->     Quantity INT NOT NULL,
    ->     UnitPrice DECIMAL(10, 2) NOT NULL, -- Price at the time of sale
    ->     PRIMARY KEY (SOID, ProductID),
    ->     FOREIGN KEY (SOID) REFERENCES SalesOrders(SOID) ON DELETE CASCADE,
    ->     FOREIGN KEY (ProductID) REFERENCES Products(ProductID) ON DELETE RESTRICT
    -> );
Query OK, 0 rows affected (0.05 sec)
```



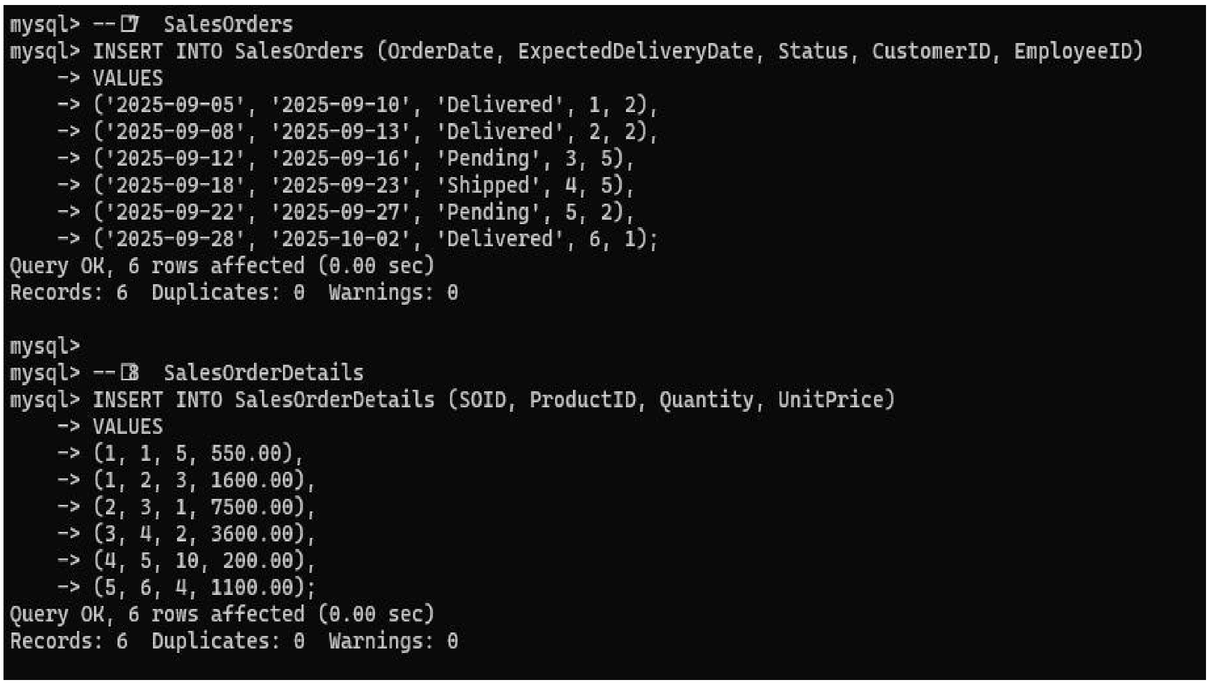
```
mysql> Show tables;
+------------------------------------+
| Tables_in_inventorymanagementsystem |
+------------------------------------+
| customers                          |
| employees                          |
| products                           |
| purchaseorderdetails               |
| purchaseorders                     |
| salesorderdetails                  |
| salesorders                        |
| suppliers                          |
+------------------------------------+
8 rows in set (0.07 sec)
```



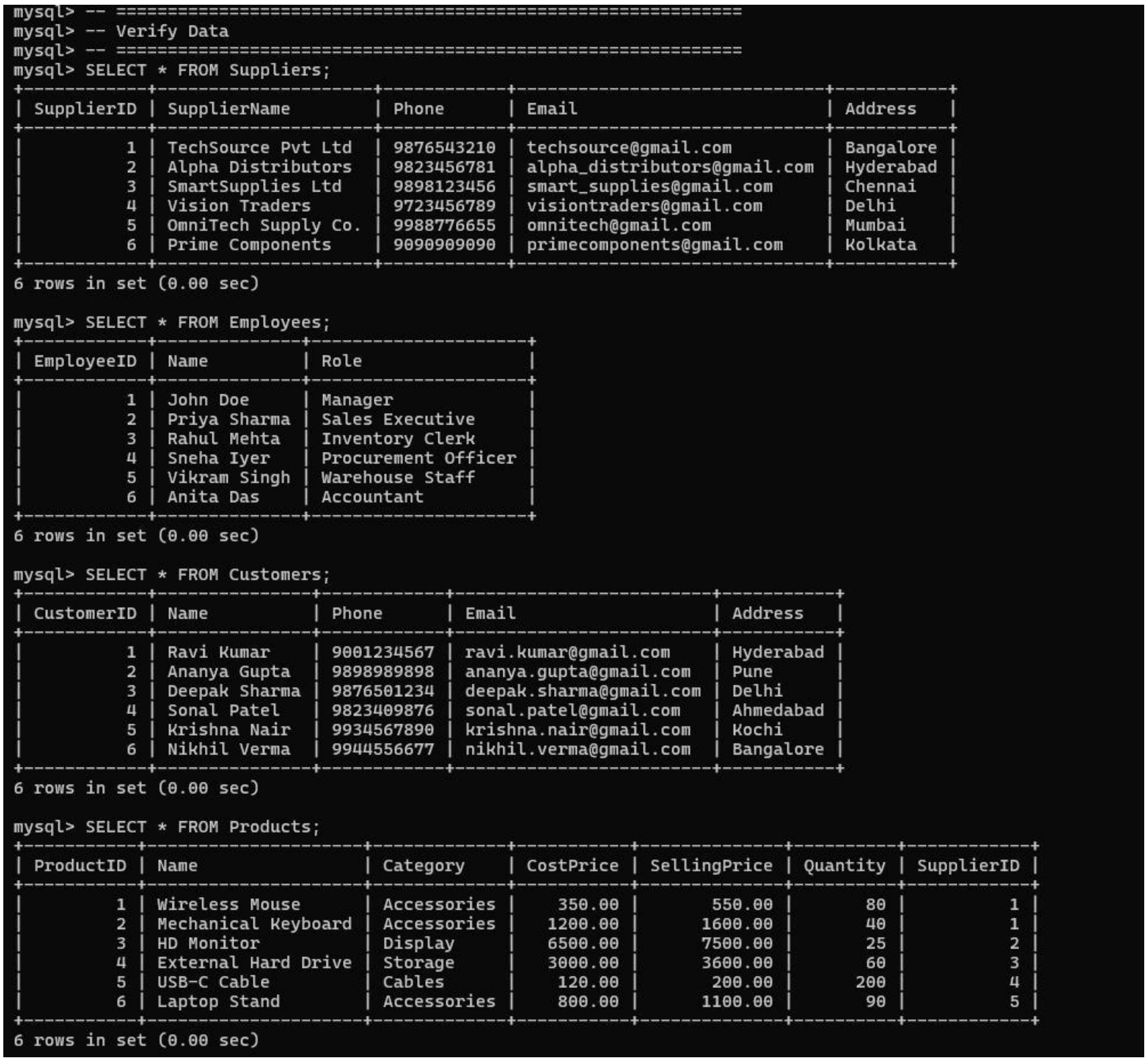
```
mysql> -- ============================================================
mysql> --   INSERT SAMPLE DATA FOR INVENTORY MANAGEMENT SYSTEM
mysql> -- ============================================================
mysql>
mysql> --□1  Suppliers
mysql> INSERT INTO Suppliers (SupplierName, Phone, Email, Address)
    -> VALUES
    -> ('TechSource Pvt Ltd', '9876543210', 'techsource@gmail.com', 'Bangalore'),
    -> ('Alpha Distributors', '9823456781', 'alpha_distributors@gmail.com', 'Hyderabad'),
    -> ('SmartSupplies Ltd', '9898123456', 'smart_supplies@gmail.com', 'Chennai'),
    -> ('Vision Traders', '9723456789', 'visiontraders@gmail.com', 'Delhi'),
    -> ('OmniTech Supply Co.', '9988776655', 'omnitech@gmail.com', 'Mumbai'),
    -> ('Prime Components', '9090909090', 'primecomponents@gmail.com', 'Kolkata');
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql>
mysql> --□2  Employees
mysql> INSERT INTO Employees (Name, Role)
    -> VALUES
    -> ('John Doe', 'Manager'),
    -> ('Priya Sharma', 'Sales Executive'),
    -> ('Rahul Mehta', 'Inventory Clerk'),
    -> ('Sneha Iyer', 'Procurement Officer'),
    -> ('Vikram Singh', 'Warehouse Staff'),
    -> ('Anita Das', 'Accountant');
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql>
mysql> --□3  Customers
mysql> INSERT INTO Customers (Name, Phone, Email, Address)
    -> VALUES
    -> ('Ravi Kumar', '9001234567', 'ravi.kumar@gmail.com', 'Hyderabad'),
    -> ('Ananya Gupta', '9898989898', 'ananya.gupta@gmail.com', 'Pune'),
    -> ('Deepak Sharma', '9876501234', 'deepak.sharma@gmail.com', 'Delhi'),
    -> ('Sonal Patel', '9823409876', 'sonal.patel@gmail.com', 'Ahmedabad'),
    -> ('Krishna Nair', '9934567890', 'krishna.nair@gmail.com', 'Kochi'),
    -> ('Nikhil Verma', '9944556677', 'nikhil.verma@gmail.com', 'Bangalore');
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql>
mysql> --□4  Products
mysql> INSERT INTO Products (Name, Category, CostPrice, SellingPrice, Quantity, SupplierID)
    -> VALUES
    -> ('Wireless Mouse', 'Accessories', 350.00, 550.00, 80, 1),
    -> ('Mechanical Keyboard', 'Accessories', 1200.00, 1600.00, 40, 1),
    -> ('HD Monitor', 'Display', 6500.00, 7500.00, 25, 2),
    -> ('External Hard Drive', 'Storage', 3000.00, 3600.00, 60, 3),
    -> ('USB-C Cable', 'Cables', 120.00, 200.00, 200, 4),
    -> ('Laptop Stand', 'Accessories', 800.00, 1100.00, 90, 5);
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0
```



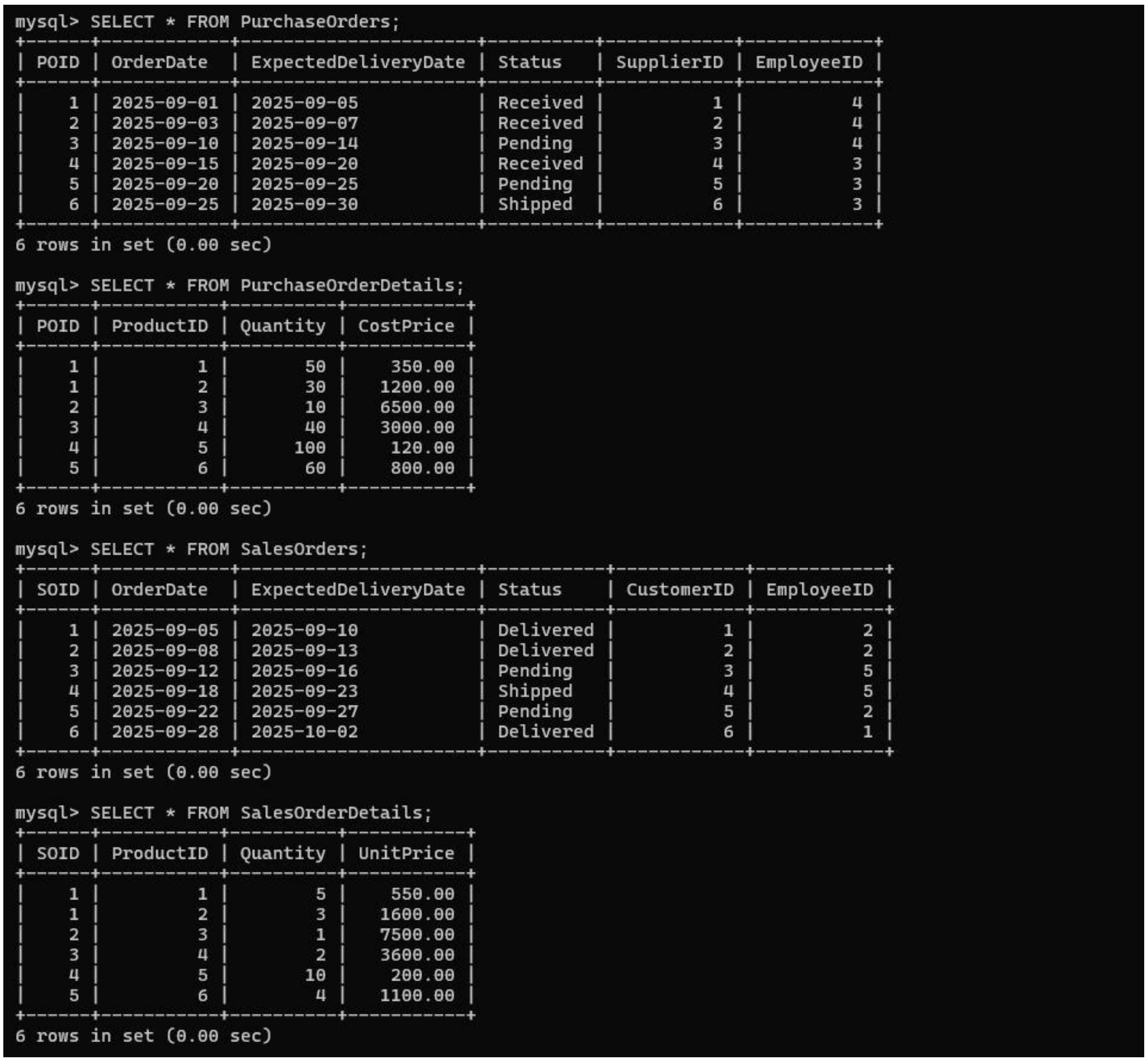
```
mysql> --□5  PurchaseOrders
mysql> INSERT INTO PurchaseOrders (OrderDate, ExpectedDeliveryDate, Status, SupplierID, EmployeeID)
    -> VALUES
    -> ('2025-09-01', '2025-09-05', 'Received', 1, 4),
    -> ('2025-09-03', '2025-09-07', 'Received', 2, 4),
    -> ('2025-09-10', '2025-09-14', 'Pending', 3, 4),
    -> ('2025-09-15', '2025-09-20', 'Received', 4, 3),
    -> ('2025-09-20', '2025-09-25', 'Pending', 5, 3),
    -> ('2025-09-25', '2025-09-30', 'Shipped', 6, 3);
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql>
mysql> --□6  PurchaseOrderDetails
mysql> INSERT INTO PurchaseOrderDetails (POID, ProductID, Quantity, CostPrice)
    -> VALUES
    -> (1, 1, 50, 350.00),
    -> (1, 2, 30, 1200.00),
    -> (2, 3, 10, 6500.00),
    -> (3, 4, 40, 3000.00),
    -> (4, 5, 100, 120.00),
    -> (5, 6, 60, 800.00);
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0
```



```
mysql> --□7  SalesOrders
mysql> INSERT INTO SalesOrders (OrderDate, ExpectedDeliveryDate, Status, CustomerID, EmployeeID)
    -> VALUES
    -> ('2025-09-05', '2025-09-10', 'Delivered', 1, 2),
    -> ('2025-09-08', '2025-09-13', 'Delivered', 2, 2),
    -> ('2025-09-12', '2025-09-16', 'Pending', 3, 5),
    -> ('2025-09-18', '2025-09-23', 'Shipped', 4, 5),
    -> ('2025-09-22', '2025-09-27', 'Pending', 5, 2),
    -> ('2025-09-28', '2025-10-02', 'Delivered', 6, 1);
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql>
mysql> --□8  SalesOrderDetails
mysql> INSERT INTO SalesOrderDetails (SOID, ProductID, Quantity, UnitPrice)
    -> VALUES
    -> (1, 1, 5, 550.00),
    -> (1, 2, 3, 1600.00),
    -> (2, 3, 1, 7500.00),
    -> (3, 4, 2, 3600.00),
    -> (4, 5, 10, 200.00),
    -> (5, 6, 4, 1100.00);
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0
```



```
mysql> -- ============================================================
mysql> -- Verify Data
mysql> -- ============================================================
mysql> SELECT * FROM Suppliers;
+------------+----------------------+------------+-------------------------------+-----------+
| SupplierID | SupplierName         | Phone      | Email                         | Address   |
+------------+----------------------+------------+-------------------------------+-----------+
|          1 | TechSource Pvt Ltd   | 9876543210 | techsource@gmail.com          | Bangalore |
|          2 | Alpha Distributors   | 9823456781 | alpha_distributors@gmail.com  | Hyderabad |
|          3 | SmartSupplies Ltd    | 9898123456 | smart_supplies@gmail.com      | Chennai   |
|          4 | Vision Traders       | 9723456789 | visiontraders@gmail.com       | Delhi     |
|          5 | OmniTech Supply Co.  | 9988776655 | omnitech@gmail.com            | Mumbai    |
|          6 | Prime Components     | 9090909090 | primecomponents@gmail.com     | Kolkata   |
+------------+----------------------+------------+-------------------------------+-----------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM Employees;
+------------+--------------+----------------------+
| EmployeeID | Name         | Role                 |
+------------+--------------+----------------------+
|          1 | John Doe     | Manager              |
|          2 | Priya Sharma | Sales Executive      |
|          3 | Rahul Mehta  | Inventory Clerk      |
|          4 | Sneha Iyer   | Procurement Officer  |
|          5 | Vikram Singh | Warehouse Staff      |
|          6 | Anita Das    | Accountant           |
+------------+--------------+----------------------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM Customers;
+------------+---------------+------------+--------------------------+-----------+
| CustomerID | Name          | Phone      | Email                    | Address   |
+------------+---------------+------------+--------------------------+-----------+
|          1 | Ravi Kumar    | 9001234567 | ravi.kumar@gmail.com     | Hyderabad |
|          2 | Ananya Gupta  | 9898989898 | ananya.gupta@gmail.com   | Pune      |
|          3 | Deepak Sharma | 9876501234 | deepak.sharma@gmail.com  | Delhi     |
|          4 | Sonal Patel   | 9823409876 | sonal.patel@gmail.com    | Ahmedabad |
|          5 | Krishna Nair  | 9934567890 | krishna.nair@gmail.com   | Kochi     |
|          6 | Nikhil Verma  | 9944556677 | nikhil.verma@gmail.com   | Bangalore |
+------------+---------------+------------+--------------------------+-----------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM Products;
+-----------+---------------------+-------------+-----------+--------------+----------+------------+
| ProductID | Name                | Category    | CostPrice | SellingPrice | Quantity | SupplierID |
+-----------+---------------------+-------------+-----------+--------------+----------+------------+
|         1 | Wireless Mouse      | Accessories |    350.00 |       550.00 |       80 |          1 |
|         2 | Mechanical Keyboard | Accessories |   1200.00 |      1600.00 |       40 |          1 |
|         3 | HD Monitor          | Display     |   6500.00 |      7500.00 |       25 |          2 |
|         4 | External Hard Drive | Storage     |   3000.00 |      3600.00 |       60 |          3 |
|         5 | USB-C Cable         | Cables      |    120.00 |       200.00 |      200 |          4 |
|         6 | Laptop Stand        | Accessories |    800.00 |      1100.00 |       90 |          5 |
+-----------+---------------------+-------------+-----------+--------------+----------+------------+
6 rows in set (0.00 sec)
```



```
mysql> SELECT * FROM PurchaseOrders;
+------+------------+----------------------+----------+------------+------------+
| POID | OrderDate  | ExpectedDeliveryDate | Status   | SupplierID | EmployeeID |
+------+------------+----------------------+----------+------------+------------+
|    1 | 2025-09-01 | 2025-09-05           | Received |          1 |          4 |
|    2 | 2025-09-03 | 2025-09-07           | Received |          2 |          4 |
|    3 | 2025-09-10 | 2025-09-14           | Pending  |          3 |          4 |
|    4 | 2025-09-15 | 2025-09-20           | Received |          4 |          3 |
|    5 | 2025-09-20 | 2025-09-25           | Pending  |          5 |          3 |
|    6 | 2025-09-25 | 2025-09-30           | Shipped  |          6 |          3 |
+------+------------+----------------------+----------+------------+------------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM PurchaseOrderDetails;
+------+-----------+----------+-----------+
| POID | ProductID | Quantity | CostPrice |
+------+-----------+----------+-----------+
|    1 |         1 |       50 |    350.00 |
|    1 |         2 |       30 |   1200.00 |
|    2 |         3 |       10 |   6500.00 |
|    3 |         4 |       40 |   3000.00 |
|    4 |         5 |      100 |    120.00 |
|    5 |         6 |       60 |    800.00 |
+------+-----------+----------+-----------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM SalesOrders;
+------+------------+----------------------+-----------+------------+------------+
| SOID | OrderDate  | ExpectedDeliveryDate | Status    | CustomerID | EmployeeID |
+------+------------+----------------------+-----------+------------+------------+
|    1 | 2025-09-05 | 2025-09-10           | Delivered |          1 |          2 |
|    2 | 2025-09-08 | 2025-09-13           | Delivered |          2 |          2 |
|    3 | 2025-09-12 | 2025-09-16           | Pending   |          3 |          5 |
|    4 | 2025-09-18 | 2025-09-23           | Shipped   |          4 |          5 |
|    5 | 2025-09-22 | 2025-09-27           | Pending   |          5 |          2 |
|    6 | 2025-09-28 | 2025-10-02           | Delivered |          6 |          1 |
+------+------------+----------------------+-----------+------------+------------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM SalesOrderDetails;
+------+-----------+----------+-----------+
| SOID | ProductID | Quantity | UnitPrice |
+------+-----------+----------+-----------+
|    1 |         1 |        5 |    550.00 |
|    1 |         2 |        3 |   1600.00 |
|    2 |         3 |        1 |   7500.00 |
|    3 |         4 |        2 |   3600.00 |
|    4 |         5 |       10 |    200.00 |
|    5 |         6 |        4 |   1100.00 |
+------+-----------+----------+-----------+
6 rows in set (0.00 sec)
```

Trigger/procedure/Function Commands:

Stored Procedure

1. Add Sales Order and auto-update stock

DELIMITER //

CREATE PROCEDURE AddSalesOrder(IN p_customerID INT, IN p_employeeID INT, IN p_productID INT, IN p_quantity INT)

BEGIN DECLARE p_price DECIMAL(10,2);

SELECT SellingPrice INTO p_price FROM Products WHERE ProductID = p_productID;

```
INSERT INTO SalesOrders (OrderDate, ExpectedDeliveryDate, Status,
CustomerID, EmployeeID)
VALUES (CURDATE(), DATE_ADD(CURDATE(), INTERVAL 5 DAY), 'Pending',
p_customerID, p_employeeID);

SET @soid = LAST_INSERT_ID();

INSERT INTO SalesOrderDetails (SOID, ProductID, Quantity, UnitPrice)
VALUES (@soid, p_productID, p_quantity, p_price);

UPDATE Products SET Quantity = Quantity - p_quantity WHERE ProductID =
p_productID;
```

END // DELIMITER ;

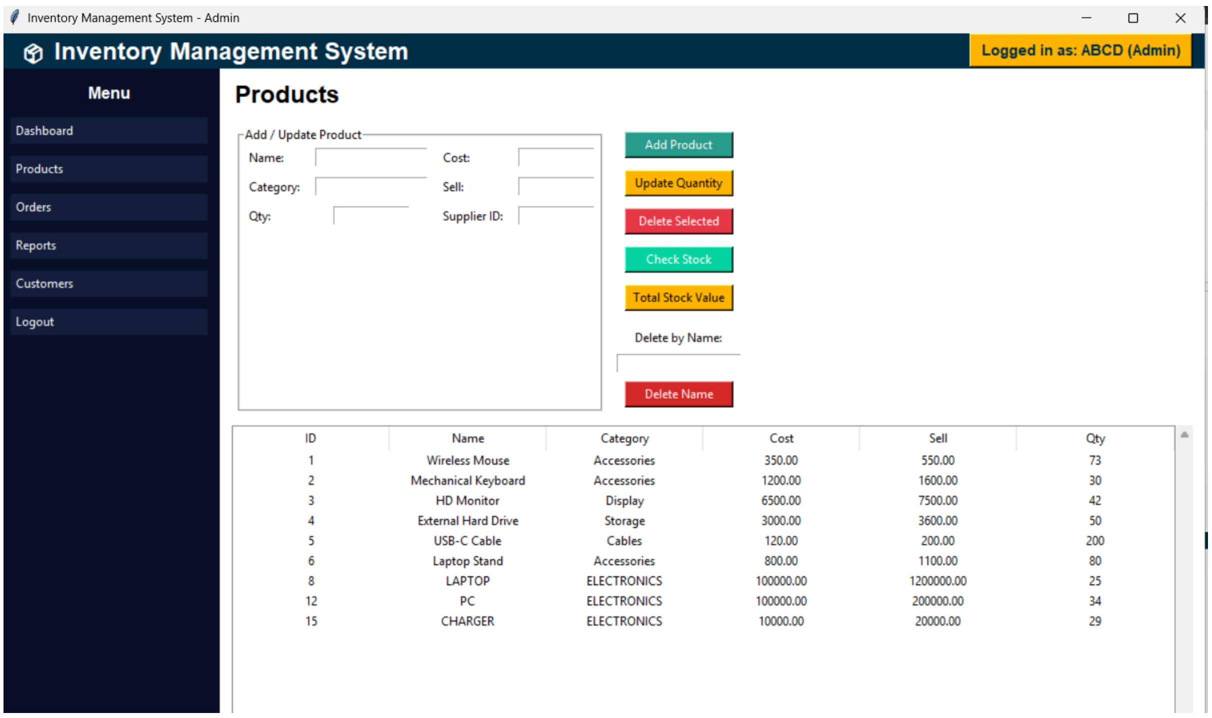
2. Add Purchase Order and auto-increase stock

DELIMITER // CREATE PROCEDURE AddPurchaseOrder(IN p_supplierID INT, IN p_employeeID INT, IN p_productID INT, IN p_quantity INT) BEGIN DECLARE p_cost DECIMAL(10,2); SELECT CostPrice INTO p_cost FROM Products WHERE ProductID = p_productID;

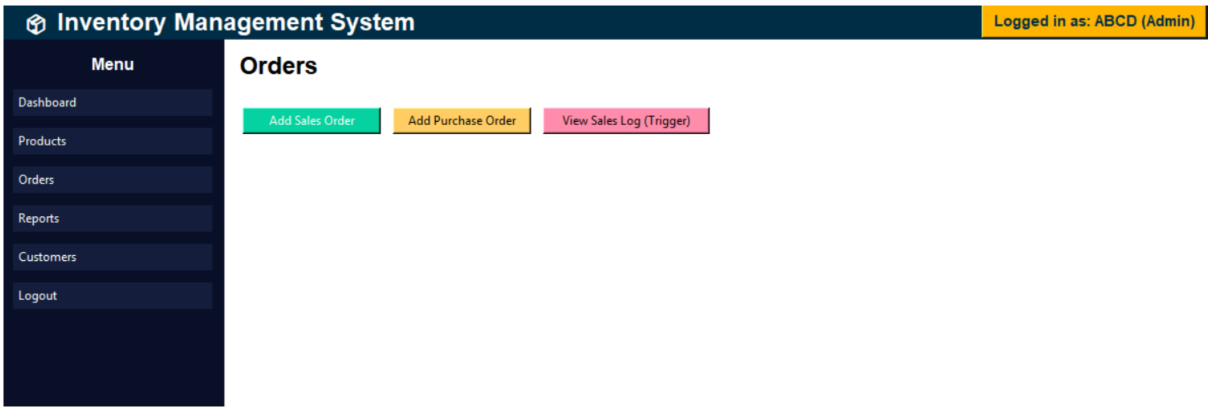
```
INSERT INTO PurchaseOrders (OrderDate, ExpectedDeliveryDate, Status,
SupplierID, EmployeeID)
VALUES (CURDATE(), DATE_ADD(CURDATE(), INTERVAL 7 DAY), 'Pending',
p_supplierID, p_employeeID);

SET @poid = LAST_INSERT_ID();

INSERT INTO PurchaseOrderDetails (POID, ProductID, Quantity, CostPrice)
VALUES (@poid, p_productID, p_quantity, p_cost);

UPDATE Products SET Quantity = Quantity + p_quantity WHERE ProductID =
p_productID;
```

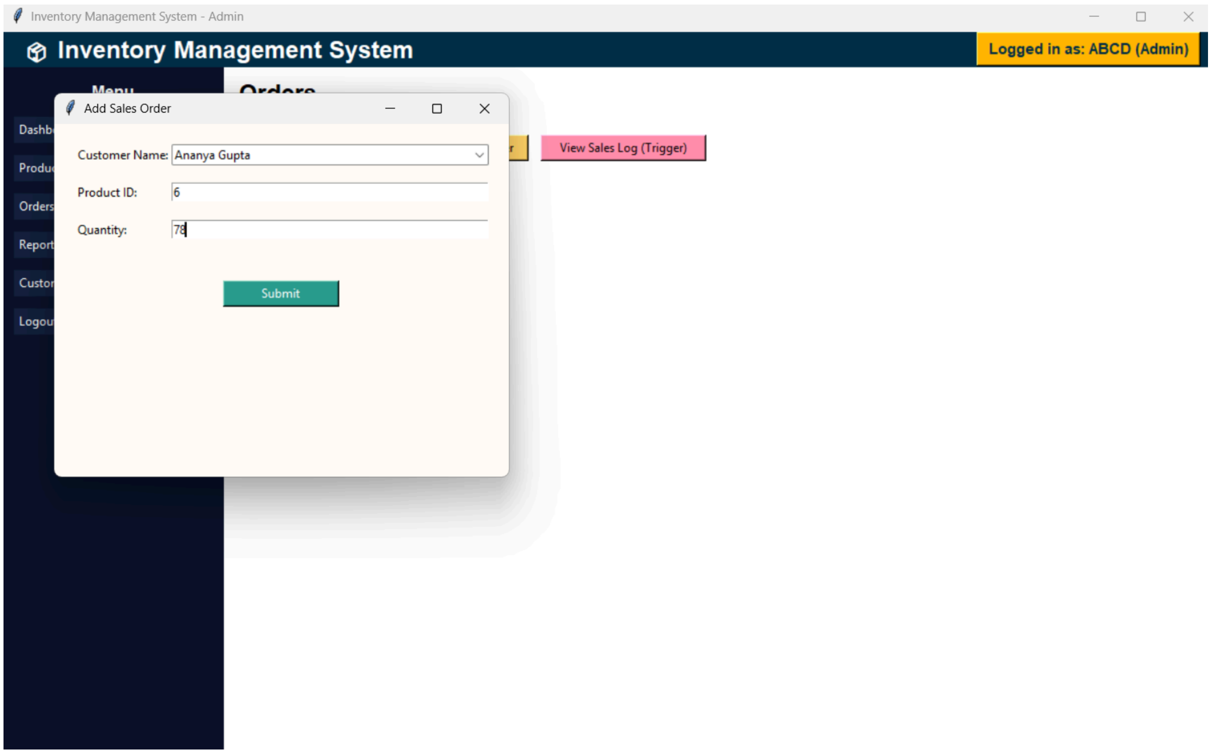
END // DELIMITER ;

ADD SALES ORDER :



ID	Name	Category	Cost	Sell	Qty
1	Wireless Mouse	Accessories	350.00	550.00	73
2	Mechanical Keyboard	Accessories	1200.00	1600.00	30
3	HD Monitor	Display	6500.00	7500.00	42
4	External Hard Drive	Storage	3000.00	3600.00	50
5	USB-C Cable	Cables	120.00	200.00	200
6	Laptop Stand	Accessories	800.00	1100.00	80
8	LAPTOP	ELECTRONICS	100000.00	1200000.00	25
12	PC	ELECTRONICS	100000.00	200000.00	34
15	CHARGER	ELECTRONICS	10000.00	20000.00	29



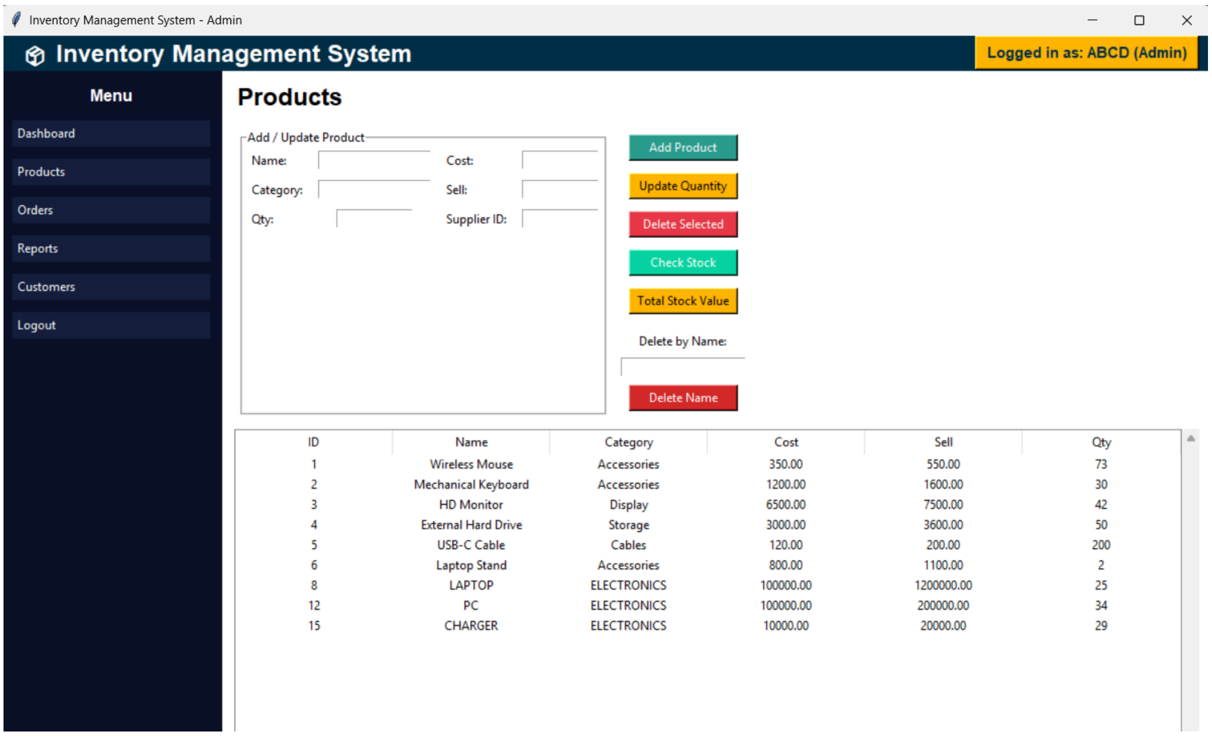

Inventory Management System - Admin
Inventory Management System
Logged in as: ABCD (Admin)
Menu
Orders
View Sales Log (Trigger)
Add Sales Order
Submit

Customer Name: Ananya Gupta

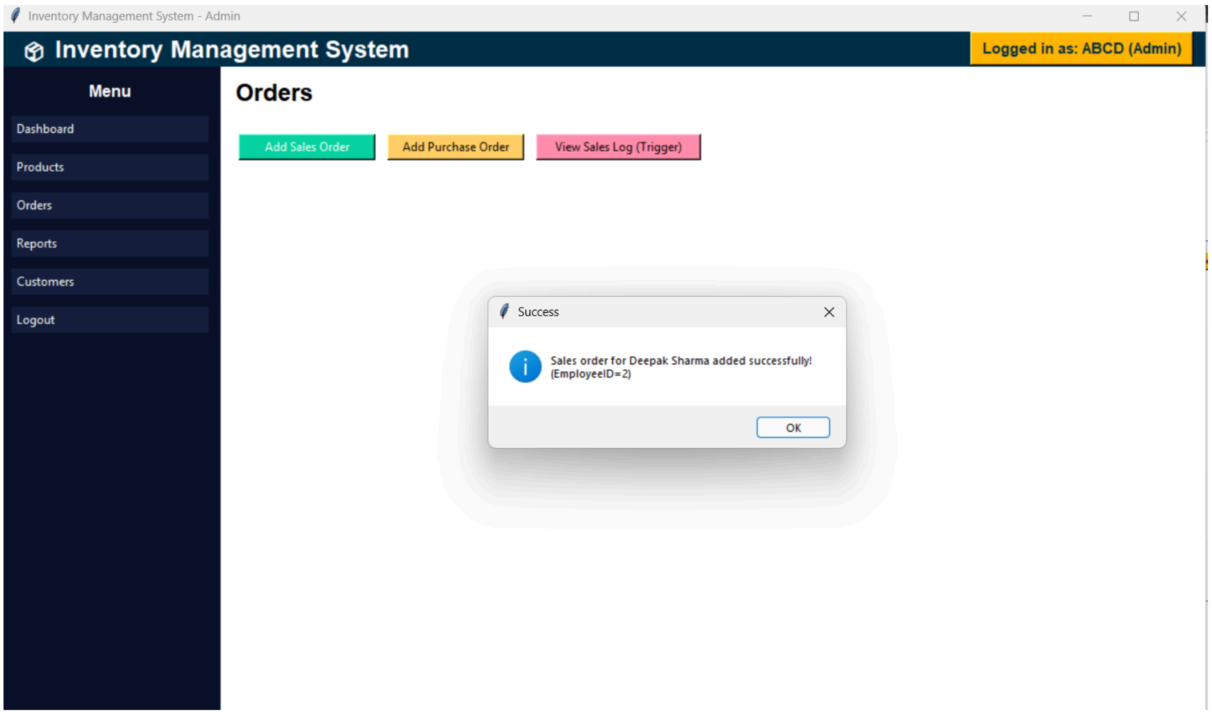
Product ID: 6

Quantity: 78

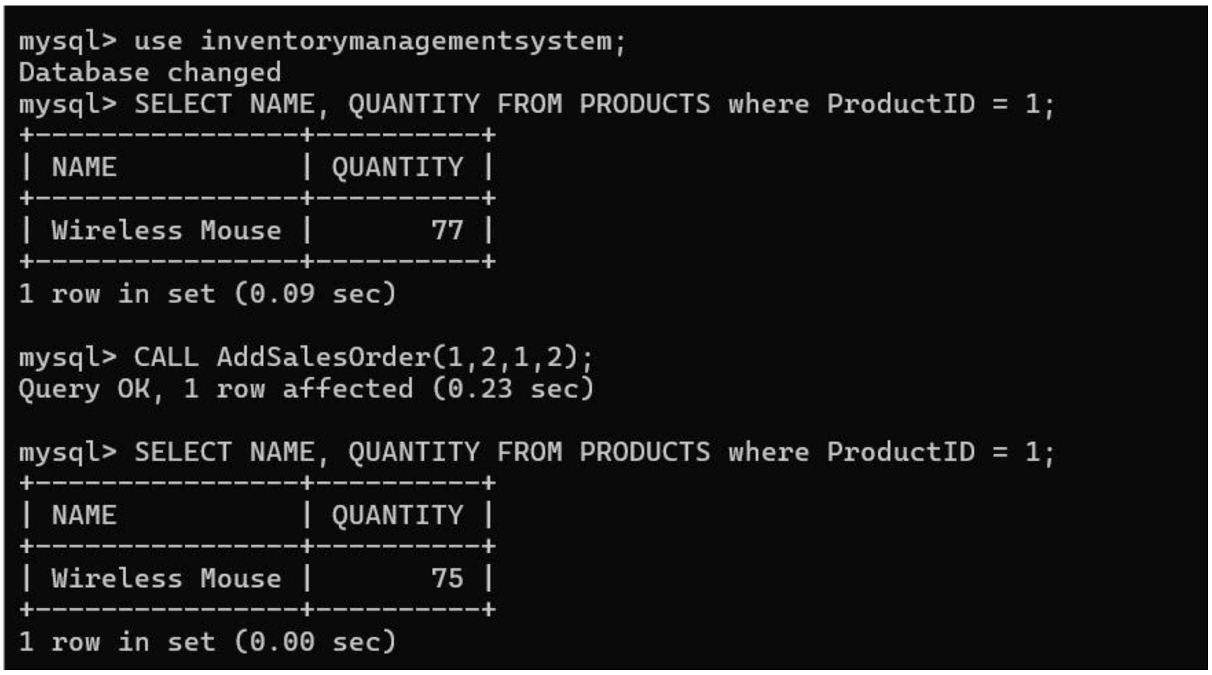
AFTER SALES ORDER –



ID	Name	Category	Cost	Sell	Qty
1	Wireless Mouse	Accessories	350.00	550.00	73
2	Mechanical Keyboard	Accessories	1200.00	1600.00	30
3	HD Monitor	Display	6500.00	7500.00	42
4	External Hard Drive	Storage	3000.00	3600.00	50
5	USB-C Cable	Cables	120.00	200.00	200
6	Laptop Stand	Accessories	800.00	1100.00	2
8	LAPTOP	ELECTRONICS	100000.00	1200000.00	25
12	PC	ELECTRONICS	100000.00	200000.00	34
15	CHARGER	ELECTRONICS	10000.00	20000.00	29

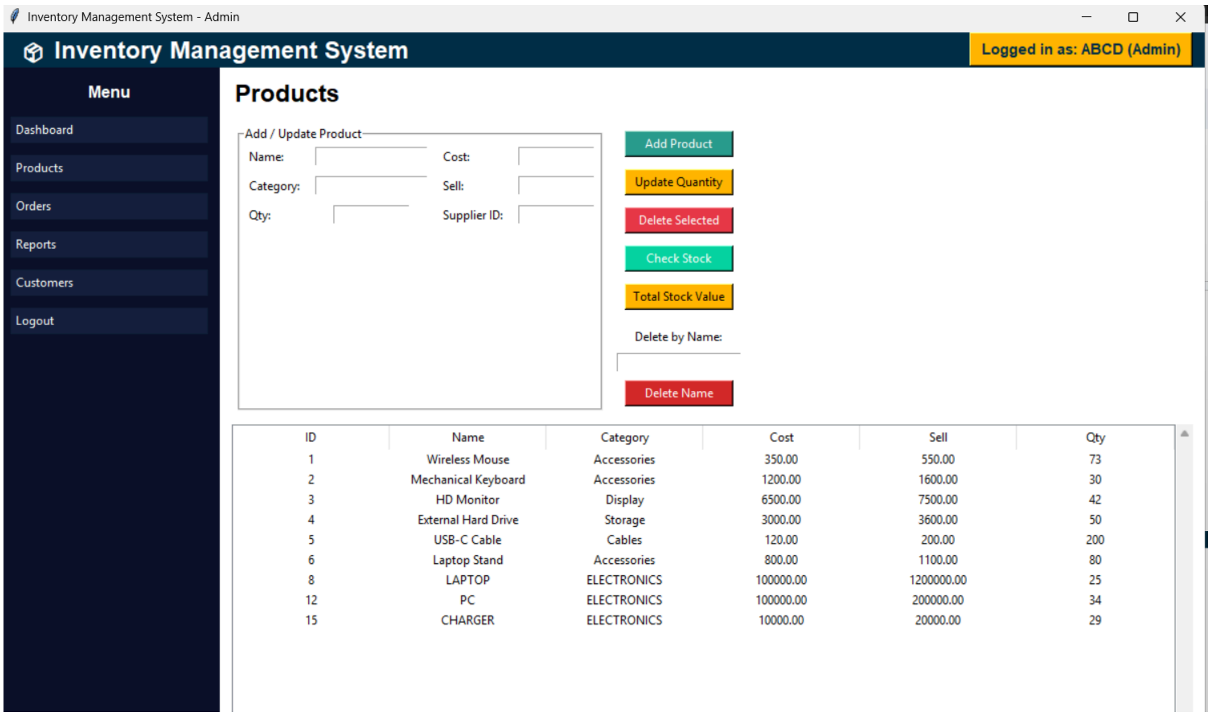


Sales order for Deepak Sharma added successfully! (EmployeeID=2)

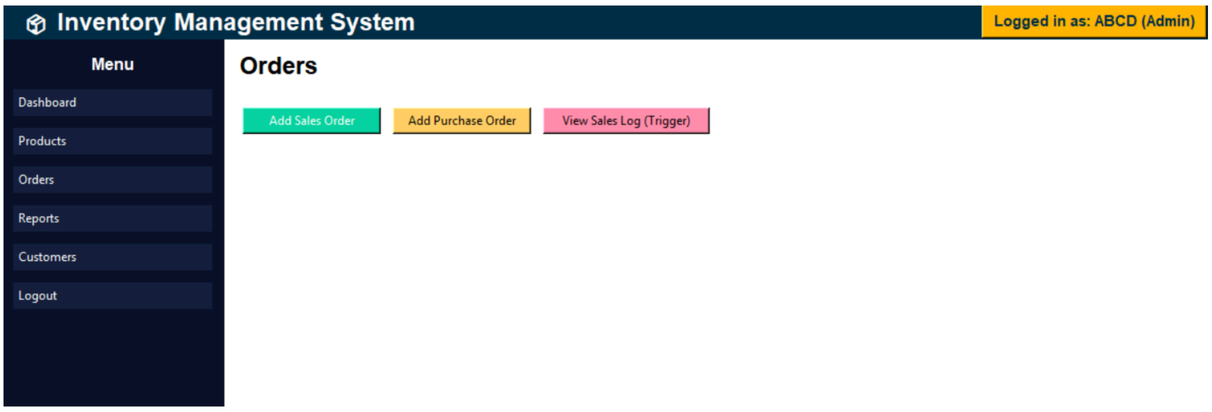


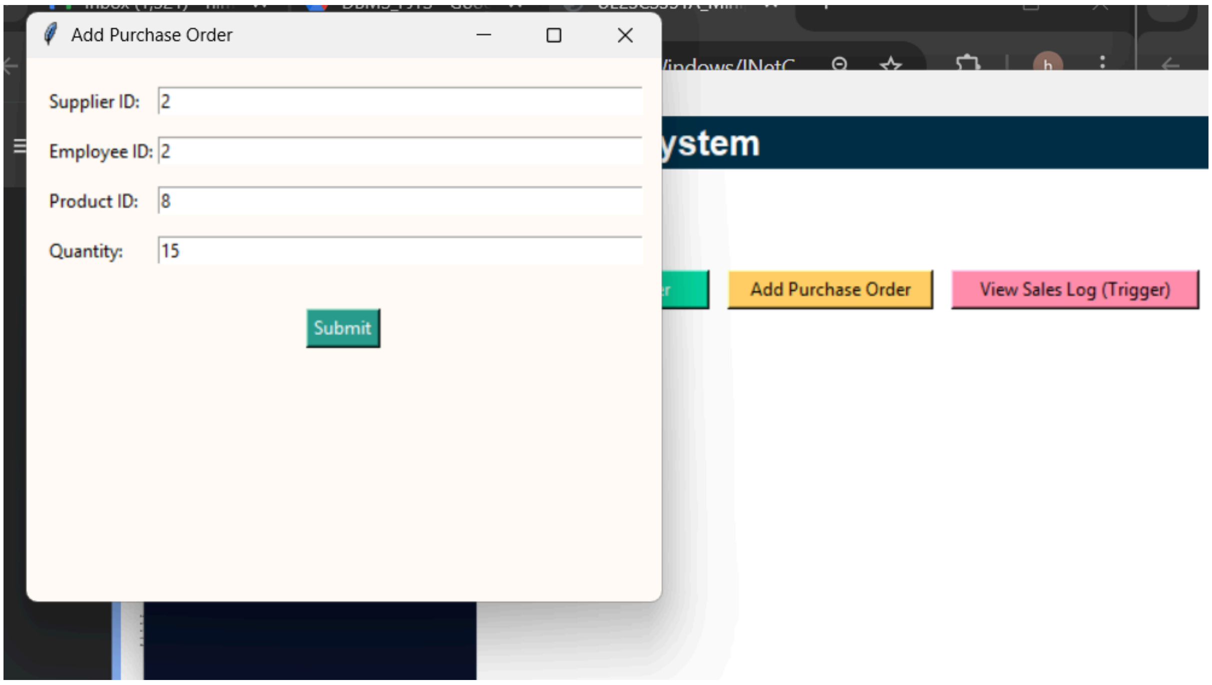
```
mysql> use inventorymanagementsystem;
Database changed
mysql> SELECT NAME, QUANTITY FROM PRODUCTS where ProductID = 1;
+-----------------+----------+
| NAME            | QUANTITY |
+-----------------+----------+
| Wireless Mouse |       77 |
+-----------------+----------+
1 row in set (0.09 sec)

mysql> CALL AddSalesOrder(1,2,1,2);
Query OK, 1 row affected (0.23 sec)

mysql> SELECT NAME, QUANTITY FROM PRODUCTS where ProductID = 1;
+-----------------+----------+
| NAME            | QUANTITY |
+-----------------+----------+
| Wireless Mouse |       75 |
+-----------------+----------+
1 row in set (0.00 sec)
```

ADD PURCHASE ORDER :



ID	Name	Category	Cost	Sell	Qty
1	Wireless Mouse	Accessories	350.00	550.00	73
2	Mechanical Keyboard	Accessories	1200.00	1600.00	30
3	HD Monitor	Display	6500.00	7500.00	42
4	External Hard Drive	Storage	3000.00	3600.00	50
5	USB-C Cable	Cables	120.00	200.00	200
6	Laptop Stand	Accessories	800.00	1100.00	80
8	LAPTOP	ELECTRONICS	100000.00	1200000.00	25
12	PC	ELECTRONICS	100000.00	200000.00	34
15	CHARGER	ELECTRONICS	10000.00	20000.00	29



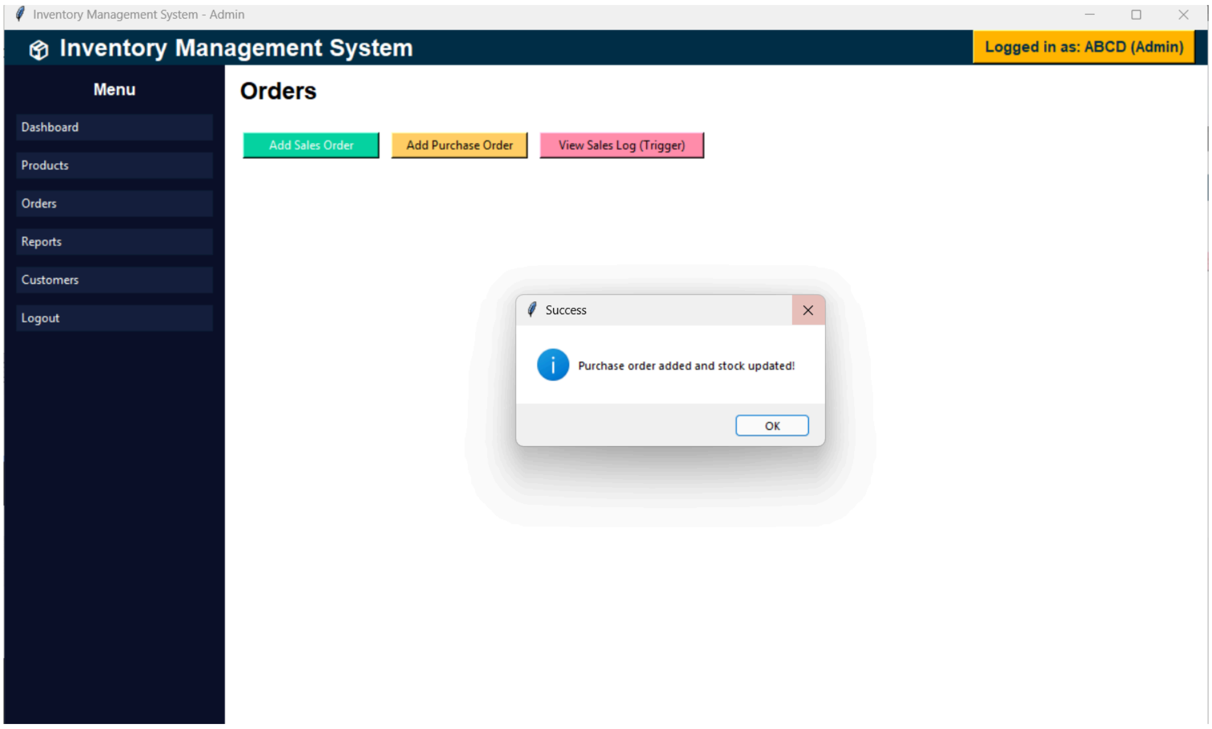

Add Purchase Order

Supplier ID: 2

Employee ID: 2

Product ID: 8

Quantity: 15


Inventory Management System - Admin

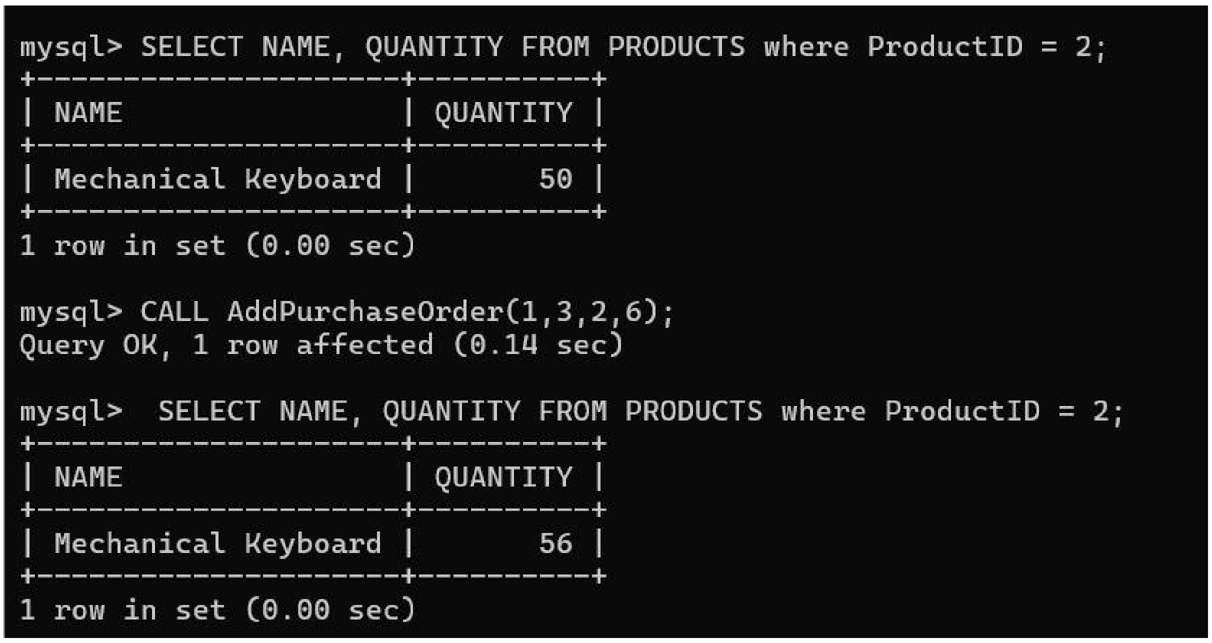
Inventory Management System

Logged in as: ABCD (Admin)

Orders

Success

Purchase order added and stock updated!

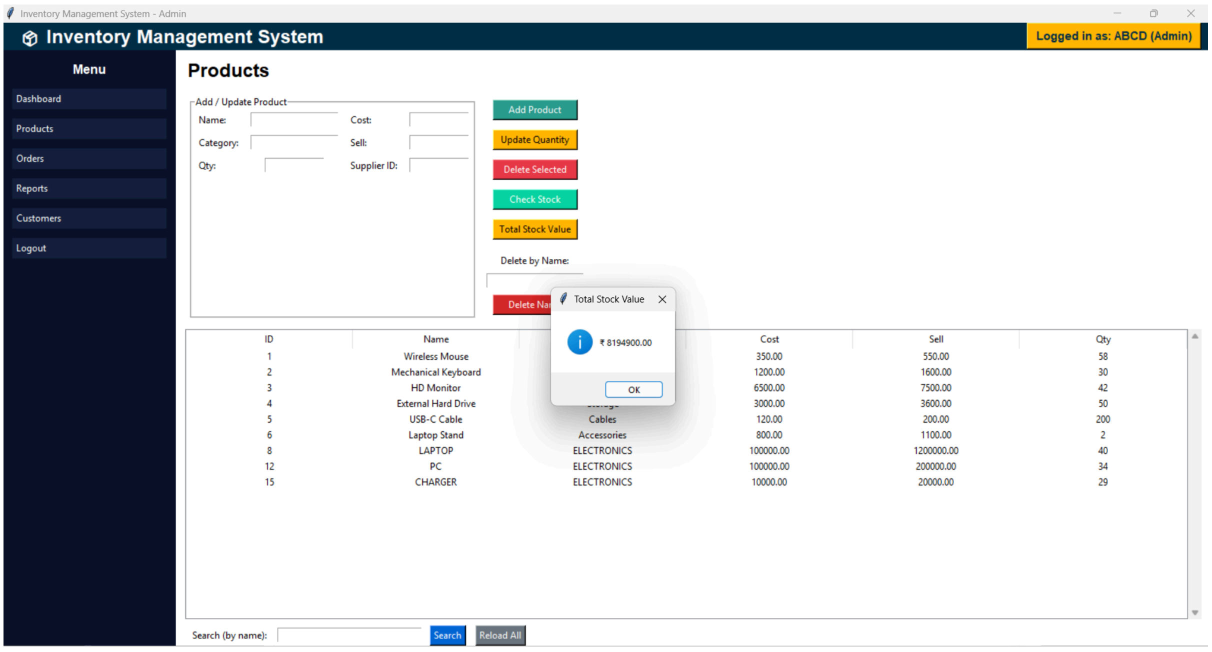


```
mysql> SELECT NAME, QUANTITY FROM PRODUCTS where ProductID = 2;
+---------------------+----------+
| NAME                | QUANTITY |
+---------------------+----------+
| Mechanical Keyboard |       50 |
+---------------------+----------+
1 row in set (0.00 sec)

mysql> CALL AddPurchaseOrder(1,3,2,6);
Query OK, 1 row affected (0.14 sec)

mysql>  SELECT NAME, QUANTITY FROM PRODUCTS where ProductID = 2;
+---------------------+----------+
| NAME                | QUANTITY |
+---------------------+----------+
| Mechanical Keyboard |       56 |
+---------------------+----------+
1 row in set (0.00 sec)
```

Functions:

3. Calculate total stock value

```
DELIMITER //

CREATE FUNCTION GetTotalStockValue()

RETURNS DECIMAL(15,2)

DETERMINISTIC

BEGIN

   DECLARE total_value DECIMAL(15,2);

   SELECT SUM(Quantity * CostPrice) INTO total_value FROM Products;

   RETURN total_value;

END //

DELIMITER ;
```



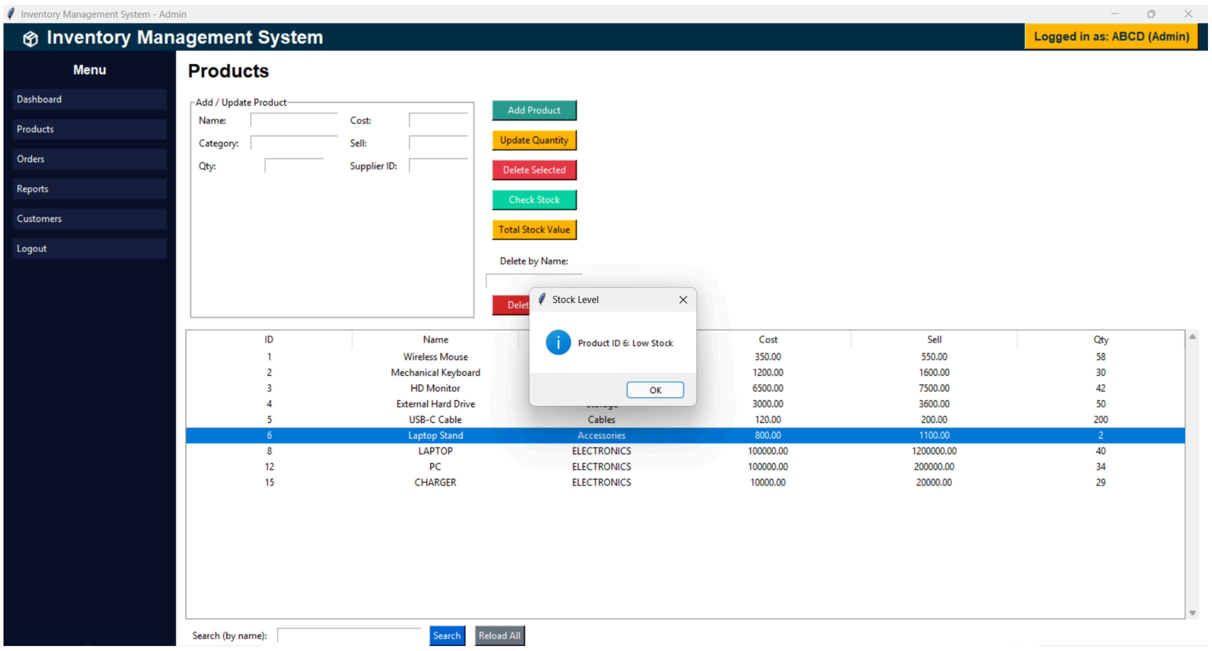
₹ 8194900.00

ID	Name		Cost	Sell	Qty
1	Wireless Mouse		350.00	550.00	58
2	Mechanical Keyboard		1200.00	1600.00	30
3	HD Monitor		6500.00	7500.00	42
4	External Hard Drive	[illegible]	3000.00	3600.00	50
5	USB-C Cable	Cables	120.00	200.00	200
6	Laptop Stand	Accessories	800.00	1100.00	2
8	LAPTOP	ELECTRONICS	100000.00	1200000.00	40
12	PC	ELECTRONICS	100000.00	200000.00	34
15	CHARGER	ELECTRONICS	10000.00	20000.00	29

4.Check stock level per product

```
DELIMITER //

CREATE FUNCTION CheckStockLevel(p_productID INT)

RETURNS VARCHAR(50)

DETERMINISTIC

BEGIN

    DECLARE stock INT;

    SELECT Quantity INTO stock FROM Products WHERE ProductID = p_productID;

    IF stock < 10 THEN

        RETURN 'Low Stock';

    ELSEIF stock BETWEEN 10 AND 50 THEN

        RETURN 'Moderate Stock';

    ELSE

        RETURN 'In Stock';

    END IF;

END //
```

DELIMITER ;



Inventory Management System - Admin

Inventory Management System

Logged in as: ABCD (Admin)

Products

Stock Level

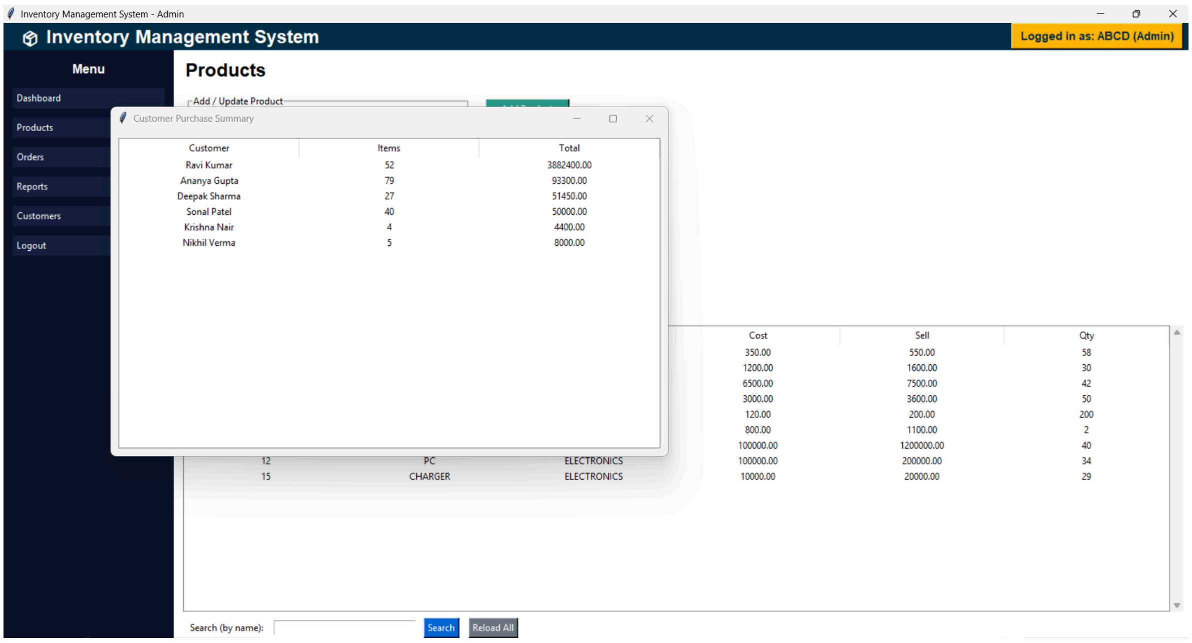
Product ID 6: Low Stock

ID	Name		Cost	Sell	Qty
1	Wireless Mouse		350.00	550.00	58
2	Mechanical Keyboard		1200.00	1600.00	30
3	HD Monitor		6500.00	7500.00	42
4	External Hard Drive	[illegible]	3000.00	3600.00	50
5	USB-C Cable	Cables	120.00	200.00	200
6	Laptop Stand	Accessories	800.00	1100.00	2
8	LAPTOP	ELECTRONICS	100000.00	1200000.00	40
12	PC	ELECTRONICS	100000.00	200000.00	34
15	CHARGER	ELECTRONICS	10000.00	20000.00	29

5. To calculate total sales amount-Function

```
DELIMITER //

CREATE FUNCTION GetTotalSalesAmountByCustomer(p_customerID INT)

RETURNS DECIMAL(15,2)

DETERMINISTIC

BEGIN

   DECLARE total_amount DECIMAL(15,2);



   SELECT SUM(sod.Quantity * sod.UnitPrice) INTO total_amount

   FROM SalesOrders so

   JOIN SalesOrderDetails sod ON so.SOID = sod.SOID

   WHERE so.CustomerID = p_customerID;



   RETURN IFNULL(total_amount, 0);

END //
```

DELIMITER ;



Inventory Management System - Admin

Inventory Management System

Logged in as: ABCD (Admin)

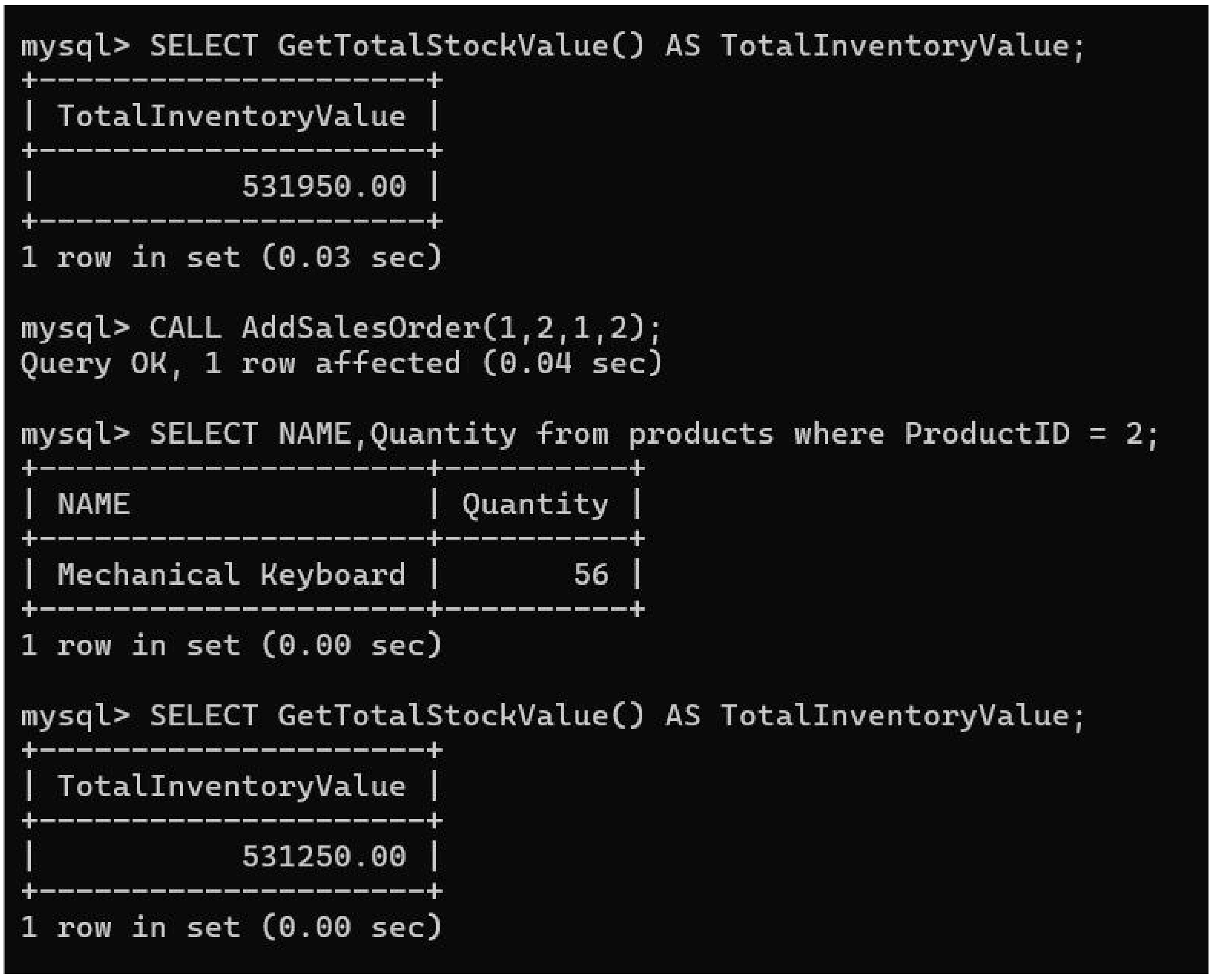
Products

Add / Update Product

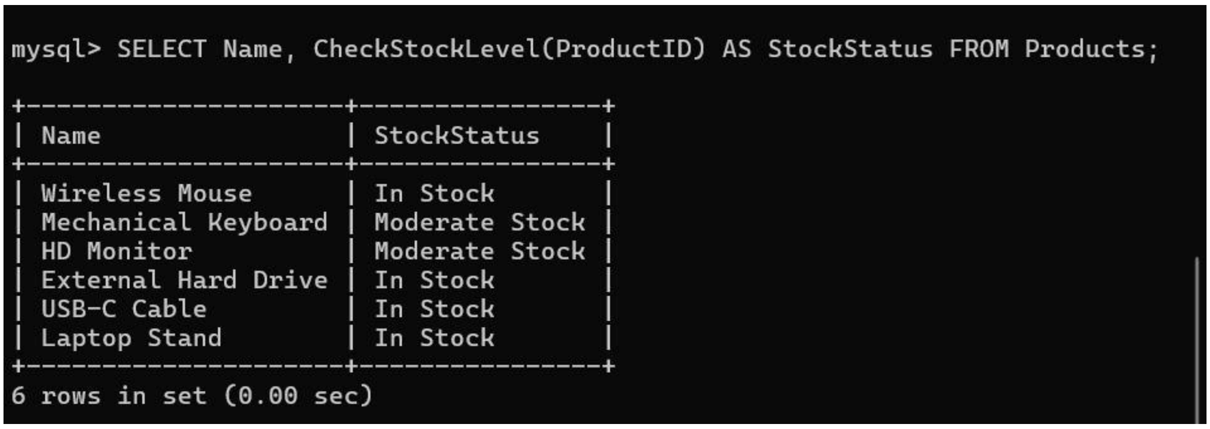
Customer Purchase Summary

Customer	Items	Total
Ravi Kumar	52	3882400.00
Ananya Gupta	79	93300.00
Deepak Sharma	27	51450.00
Sonal Patel	40	50000.00
Krishna Nair	4	4400.00
Nikhil Verma	5	8000.00

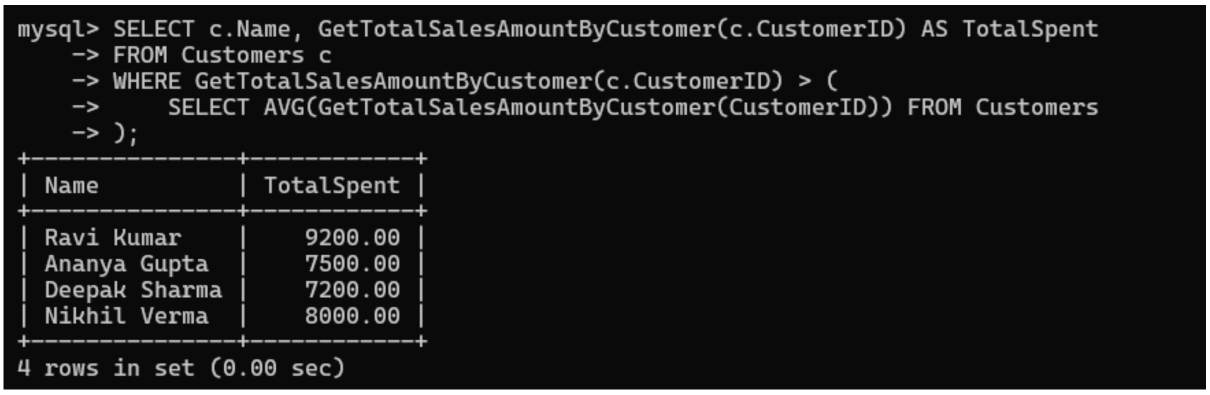
			Cost	Sell	Qty
			350.00	550.00	58
			1200.00	1600.00	30
			6500.00	7500.00	42
			3000.00	3600.00	50
			120.00	200.00	200
			800.00	1100.00	2
			100000.00	1200000.00	40
12	PC	ELECTRONICS	100000.00	200000.00	34
15	CHARGER	ELECTRONICS	10000.00	20000.00	29



```
mysql> SELECT GetTotalStockValue() AS TotalInventoryValue;
+---------------------+
| TotalInventoryValue |
+---------------------+
|           531950.00 |
+---------------------+
1 row in set (0.03 sec)

mysql> CALL AddSalesOrder(1,2,1,2);
Query OK, 1 row affected (0.04 sec)

mysql> SELECT NAME,Quantity from products where ProductID = 2;
+---------------------+----------+
| NAME                | Quantity |
+---------------------+----------+
| Mechanical Keyboard |       56 |
+---------------------+----------+
1 row in set (0.00 sec)

mysql> SELECT GetTotalStockValue() AS TotalInventoryValue;
+---------------------+
| TotalInventoryValue |
+---------------------+
|           531250.00 |
+---------------------+
1 row in set (0.00 sec)
```



```
mysql> SELECT Name, CheckStockLevel(ProductID) AS StockStatus FROM Products;

+---------------------+----------------+
| Name                | StockStatus    |
+---------------------+----------------+
| Wireless Mouse      | In Stock       |
| Mechanical Keyboard | Moderate Stock |
| HD Monitor          | Moderate Stock |
| External Hard Drive | In Stock       |
| USB-C Cable         | In Stock       |
| Laptop Stand        | In Stock       |
+---------------------+----------------+
6 rows in set (0.00 sec)
```



```
mysql> SELECT c.Name, GetTotalSalesAmountByCustomer(c.CustomerID) AS TotalSpent
    -> FROM Customers c
    -> WHERE GetTotalSalesAmountByCustomer(c.CustomerID) > (
    ->     SELECT AVG(GetTotalSalesAmountByCustomer(CustomerID)) FROM Customers
    -> );
+---------------+------------+
| Name          | TotalSpent |
+---------------+------------+
| Ravi Kumar    |    9200.00 |
| Ananya Gupta  |    7500.00 |
| Deepak Sharma |    7200.00 |
| Nikhil Verma  |    8000.00 |
+---------------+------------+
4 rows in set (0.00 sec)
```

Triggers:

6. Create Sales Log Table (for trigger)

```
CREATE TABLE IF NOT EXISTS SalesLog (
    LogID INT AUTO_INCREMENT PRIMARY KEY,
    ProductID INT,
    Quantity INT,
    ActionTime DATETIME,
    Message VARCHAR(255)
);
```

7.

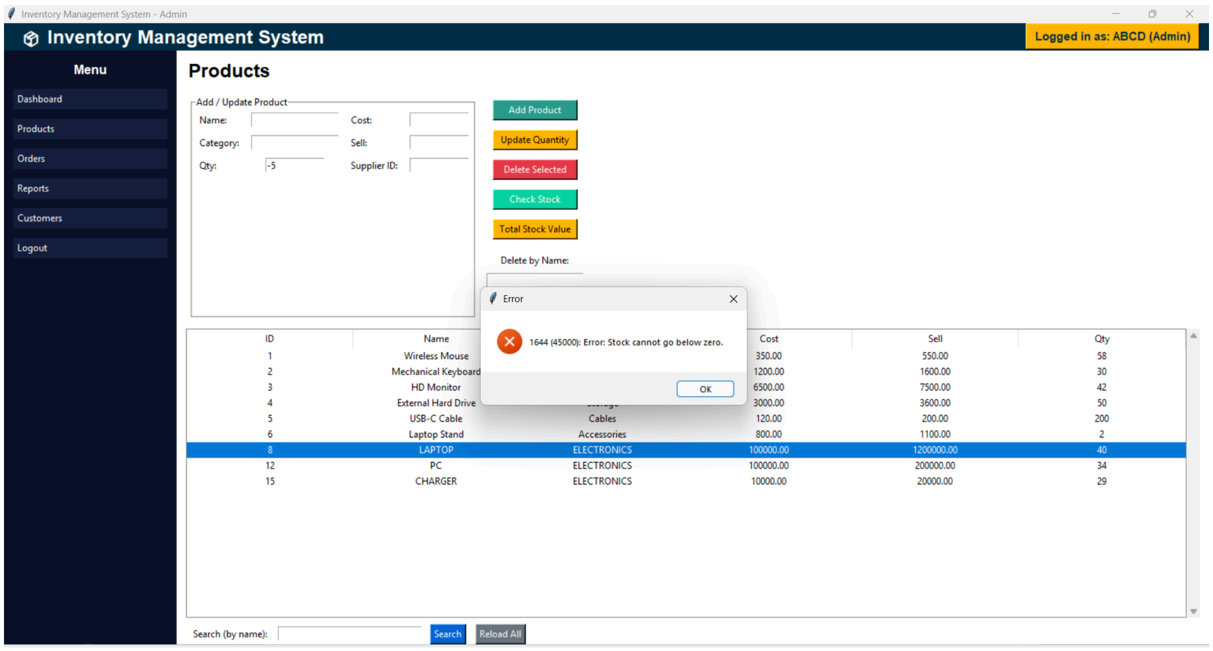
(Log every sales order detail insert)

```
DELIMITER //
CREATE TRIGGER AfterSalesOrderInsert
AFTER INSERT ON SalesOrderDetails
FOR EACH ROW
BEGIN
    INSERT INTO SalesLog (ProductID, Quantity, ActionTime, Message)
    VALUES (NEW.ProductID, NEW.Quantity, NOW(),
            CONCAT('Sold ', NEW.Quantity, ' units of Product ID ', NEW.ProductID));
```

```
END //

DELIMITER ;
```

8. Prevent negative stock

```
DELIMITER //

CREATE TRIGGER PreventNegativeStock

BEFORE UPDATE ON Products

FOR EACH ROW

BEGIN

   IF NEW.Quantity < 0 THEN

      SIGNAL SQLSTATE '45000'

      SET MESSAGE_TEXT = 'Error: Stock cannot go below zero.';

   END IF;

END //

DELIMITER ;
```



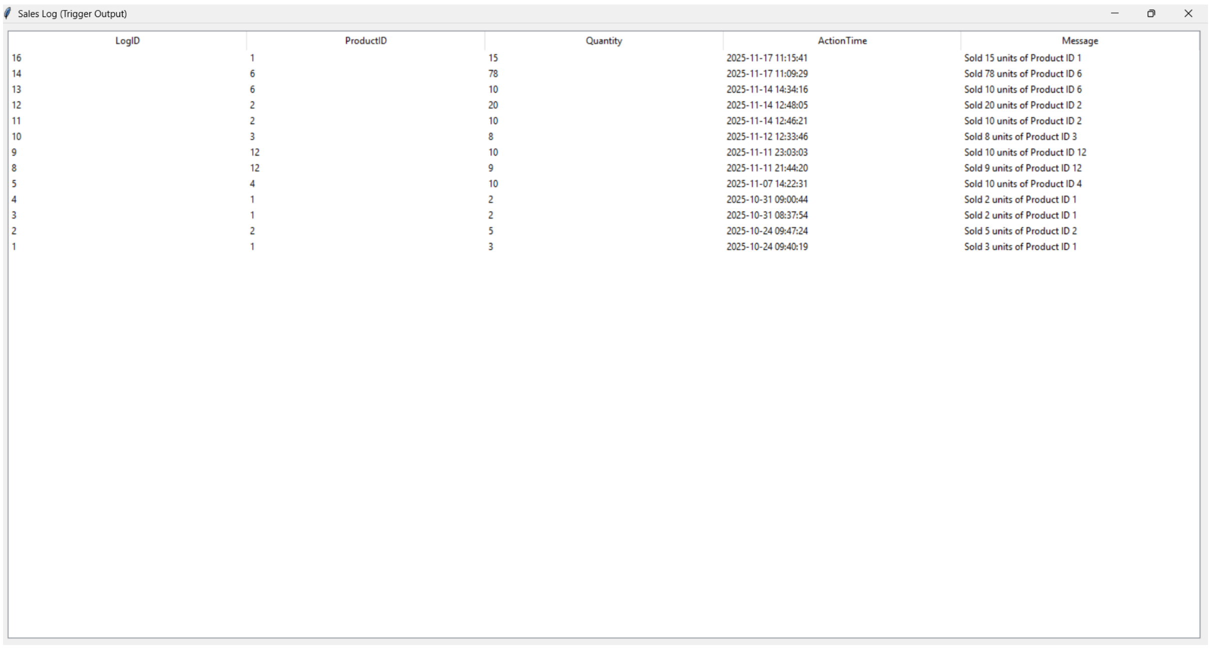
Qty: -5

Error

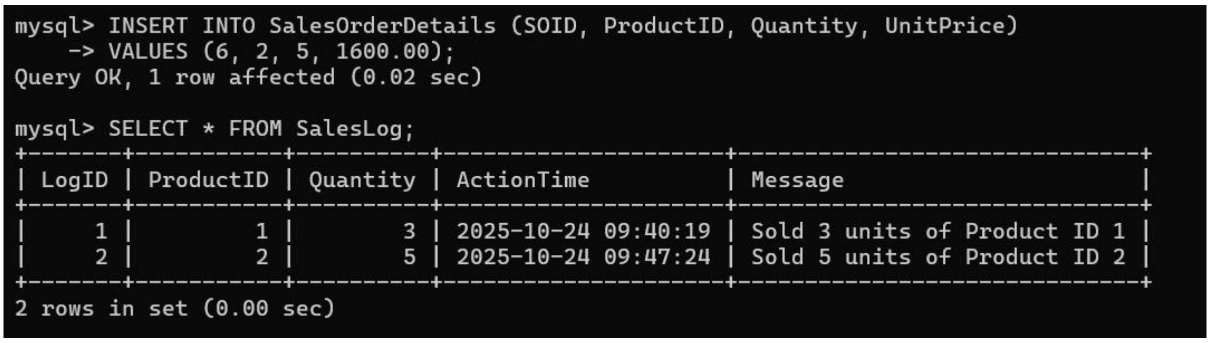
1644 (45000): Error: Stock cannot go below zero.

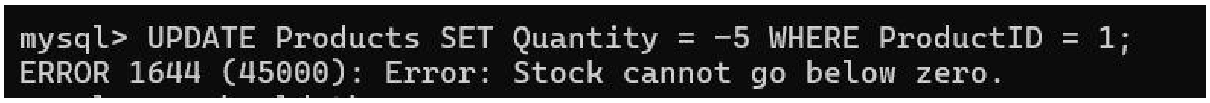
OK

ID	Name		Cost	Sell	Qty
1	Wireless Mouse	[illegible]	350.00	550.00	58
2	Mechanical Keyboard	[illegible]	1200.00	1600.00	30
3	HD Monitor	[illegible]	6500.00	7500.00	42
4	External Hard Drive	[illegible]	3000.00	3600.00	50
5	USB-C Cable	Cables	120.00	200.00	200
6	Laptop Stand	Accessories	800.00	1100.00	2
8	LAPTOP	ELECTRONICS	100000.00	1200000.00	40
12	PC	ELECTRONICS	100000.00	200000.00	34
15	CHARGER	ELECTRONICS	10000.00	20000.00	29

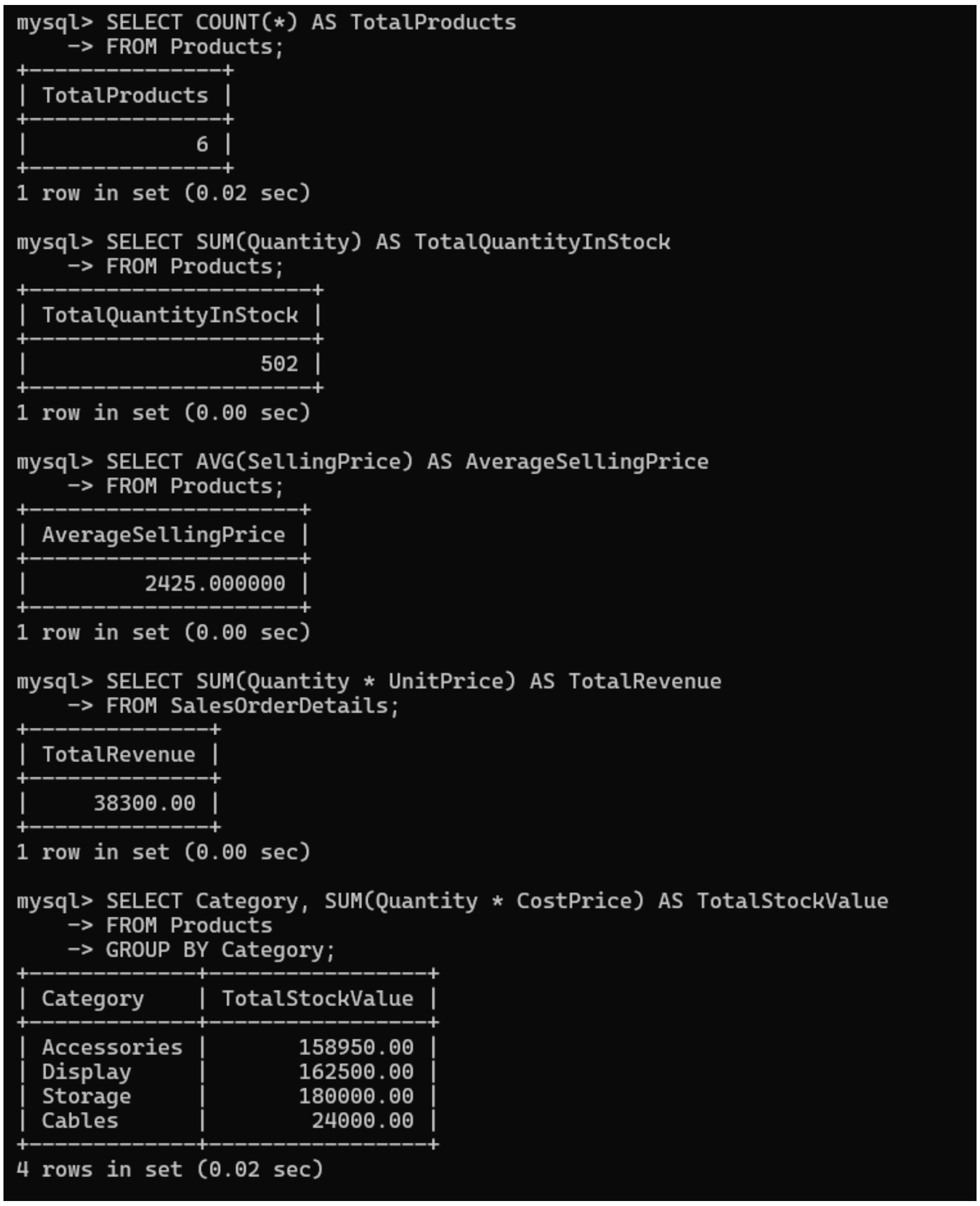

Sales Log (Trigger Output)

LogID	ProductID	Quantity	ActionTime	Message
16	1	15	2025-11-17 11:15:41	Sold 15 units of Product ID 1
14	6	78	2025-11-17 11:09:29	Sold 78 units of Product ID 6
13	6	10	2025-11-14 14:34:16	Sold 10 units of Product ID 6
12	2	20	2025-11-14 12:48:05	Sold 20 units of Product ID 2
11	2	10	2025-11-14 12:46:21	Sold 10 units of Product ID 2
10	3	8	2025-11-12 12:33:46	Sold 8 units of Product ID 3
9	12	10	2025-11-11 23:03:03	Sold 10 units of Product ID 12
8	12	9	2025-11-11 21:44:20	Sold 9 units of Product ID 12
5	4	10	2025-11-07 14:22:31	Sold 10 units of Product ID 4
4	1	2	2025-10-31 09:00:44	Sold 2 units of Product ID 1
3	1	2	2025-10-31 08:37:54	Sold 2 units of Product ID 1
2	2	5	2025-10-24 09:47:24	Sold 5 units of Product ID 2
1	1	3	2025-10-24 09:40:19	Sold 3 units of Product ID 1


```
mysql> INSERT INTO SalesOrderDetails (SOID, ProductID, Quantity, UnitPrice)
    -> VALUES (6, 2, 5, 1600.00);
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM SalesLog;
+-------+-----------+----------+---------------------+----------------------------+
| LogID | ProductID | Quantity | ActionTime          | Message                    |
+-------+-----------+----------+---------------------+----------------------------+
|     1 |         1 |        3 | 2025-10-24 09:40:19 | Sold 3 units of Product ID 1 |
|     2 |         2 |        5 | 2025-10-24 09:47:24 | Sold 5 units of Product ID 2 |
+-------+-----------+----------+---------------------+----------------------------+
2 rows in set (0.00 sec)
```

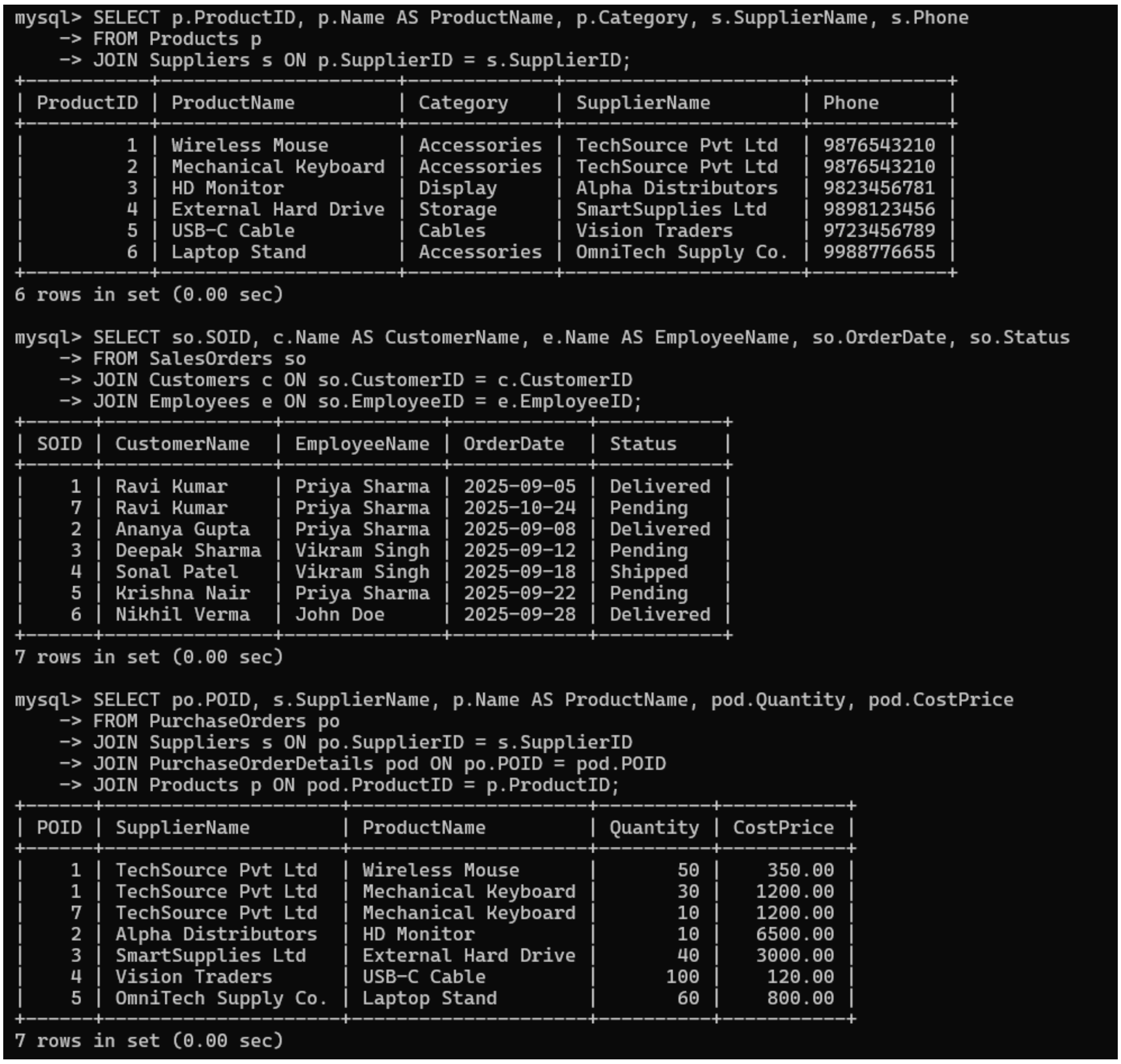

```
mysql> UPDATE Products SET Quantity = -5 WHERE ProductID = 1;
ERROR 1644 (45000): Error: Stock cannot go below zero.
```

AGGREGATED QUERIES :

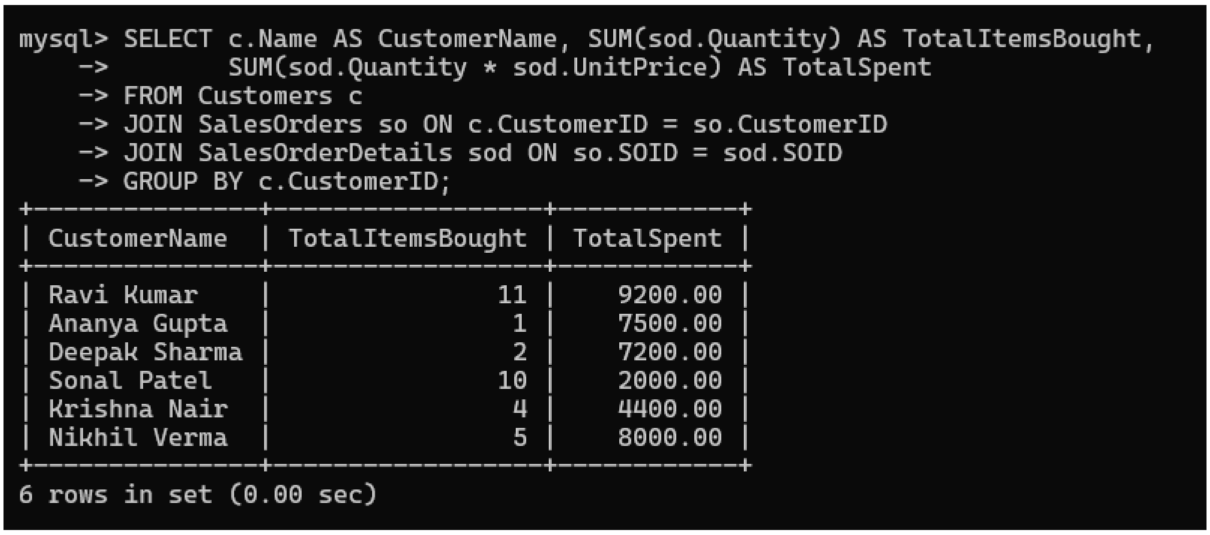


```
mysql> SELECT COUNT(*) AS TotalProducts
    -> FROM Products;
+---------------+
| TotalProducts |
+---------------+
|             6 |
+---------------+
1 row in set (0.02 sec)

mysql> SELECT SUM(Quantity) AS TotalQuantityInStock
    -> FROM Products;
+----------------------+
| TotalQuantityInStock |
+----------------------+
|                  502 |
+----------------------+
1 row in set (0.00 sec)

mysql> SELECT AVG(SellingPrice) AS AverageSellingPrice
    -> FROM Products;
+---------------------+
| AverageSellingPrice |
+---------------------+
|         2425.000000 |
+---------------------+
1 row in set (0.00 sec)

mysql> SELECT SUM(Quantity * UnitPrice) AS TotalRevenue
    -> FROM SalesOrderDetails;
+--------------+
| TotalRevenue |
+--------------+
|     38300.00 |
+--------------+
1 row in set (0.00 sec)

mysql> SELECT Category, SUM(Quantity * CostPrice) AS TotalStockValue
    -> FROM Products
    -> GROUP BY Category;
+-------------+-----------------+
| Category    | TotalStockValue |
+-------------+-----------------+
| Accessories |       158950.00 |
| Display     |       162500.00 |
| Storage     |       180000.00 |
| Cables      |        24000.00 |
+-------------+-----------------+
4 rows in set (0.02 sec)
```

JOINS :

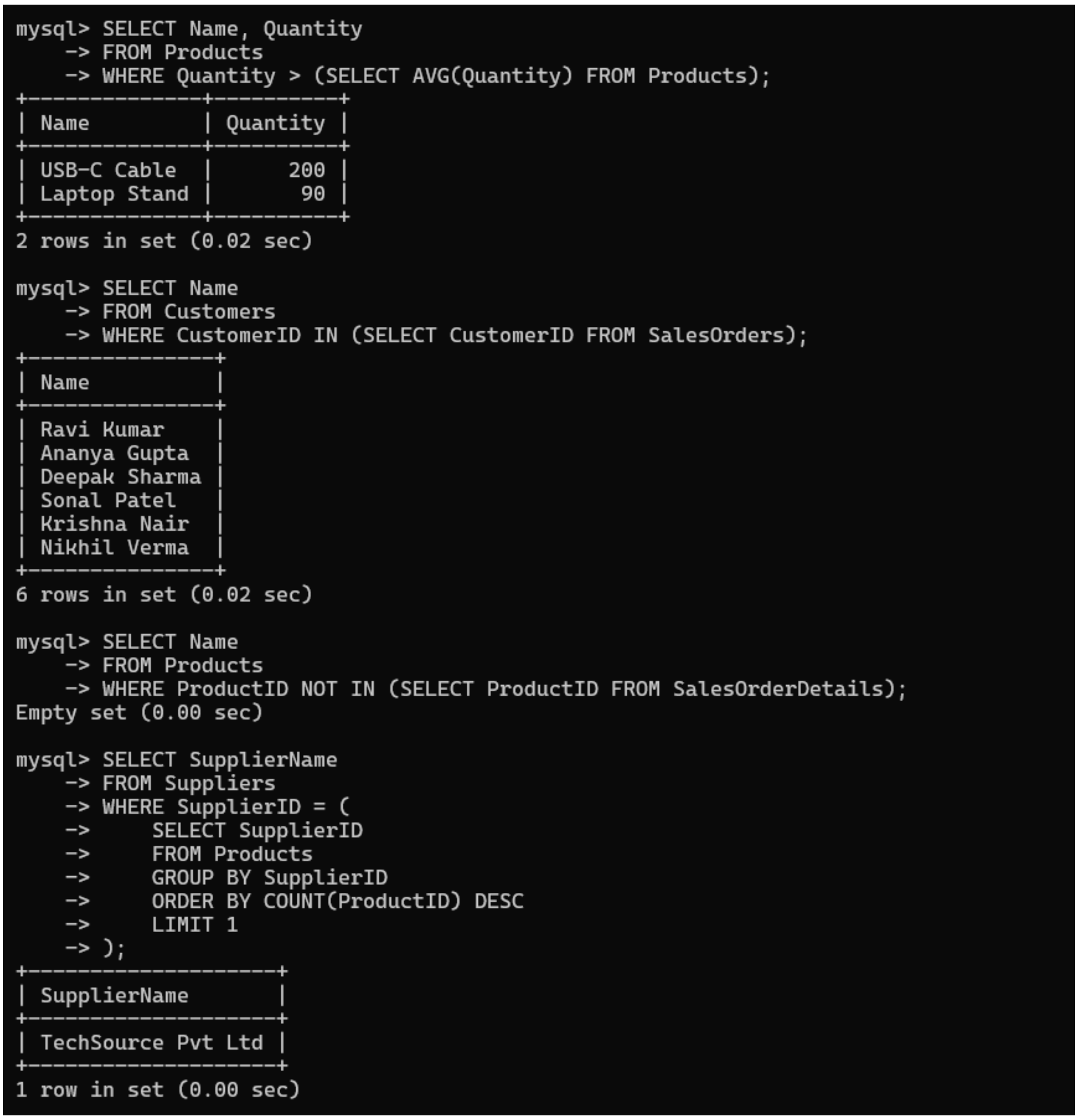


```
mysql> SELECT p.ProductID, p.Name AS ProductName, p.Category, s.SupplierName, s.Phone
    -> FROM Products p
    -> JOIN Suppliers s ON p.SupplierID = s.SupplierID;
+-----------+---------------------+-------------+---------------------+------------+
| ProductID | ProductName         | Category    | SupplierName        | Phone      |
+-----------+---------------------+-------------+---------------------+------------+
|         1 | Wireless Mouse      | Accessories | TechSource Pvt Ltd  | 9876543210 |
|         2 | Mechanical Keyboard | Accessories | TechSource Pvt Ltd  | 9876543210 |
|         3 | HD Monitor          | Display     | Alpha Distributors  | 9823456781 |
|         4 | External Hard Drive | Storage     | SmartSupplies Ltd   | 9898123456 |
|         5 | USB-C Cable         | Cables      | Vision Traders      | 9723456789 |
|         6 | Laptop Stand        | Accessories | OmniTech Supply Co. | 9988776655 |
+-----------+---------------------+-------------+---------------------+------------+
6 rows in set (0.00 sec)

mysql> SELECT so.SOID, c.Name AS CustomerName, e.Name AS EmployeeName, so.OrderDate, so.Status
    -> FROM SalesOrders so
    -> JOIN Customers c ON so.CustomerID = c.CustomerID
    -> JOIN Employees e ON so.EmployeeID = e.EmployeeID;
+------+---------------+--------------+------------+-----------+
| SOID | CustomerName  | EmployeeName | OrderDate  | Status    |
+------+---------------+--------------+------------+-----------+
|    1 | Ravi Kumar    | Priya Sharma | 2025-09-05 | Delivered |
|    7 | Ravi Kumar    | Priya Sharma | 2025-10-24 | Pending   |
|    2 | Ananya Gupta  | Priya Sharma | 2025-09-08 | Delivered |
|    3 | Deepak Sharma | Vikram Singh | 2025-09-12 | Pending   |
|    4 | Sonal Patel   | Vikram Singh | 2025-09-18 | Shipped   |
|    5 | Krishna Nair  | Priya Sharma | 2025-09-22 | Pending   |
|    6 | Nikhil Verma  | John Doe     | 2025-09-28 | Delivered |
+------+---------------+--------------+------------+-----------+
7 rows in set (0.00 sec)

mysql> SELECT po.POID, s.SupplierName, p.Name AS ProductName, pod.Quantity, pod.CostPrice
    -> FROM PurchaseOrders po
    -> JOIN Suppliers s ON po.SupplierID = s.SupplierID
    -> JOIN PurchaseOrderDetails pod ON po.POID = pod.POID
    -> JOIN Products p ON pod.ProductID = p.ProductID;
+------+---------------------+---------------------+----------+-----------+
| POID | SupplierName        | ProductName         | Quantity | CostPrice |
+------+---------------------+---------------------+----------+-----------+
|    1 | TechSource Pvt Ltd  | Wireless Mouse      |       50 |    350.00 |
|    1 | TechSource Pvt Ltd  | Mechanical Keyboard |       30 |   1200.00 |
|    7 | TechSource Pvt Ltd  | Mechanical Keyboard |       10 |   1200.00 |
|    2 | Alpha Distributors  | HD Monitor          |       10 |   6500.00 |
|    3 | SmartSupplies Ltd   | External Hard Drive |       40 |   3000.00 |
|    4 | Vision Traders      | USB-C Cable         |      100 |    120.00 |
|    5 | OmniTech Supply Co. | Laptop Stand        |       60 |    800.00 |
+------+---------------------+---------------------+----------+-----------+
7 rows in set (0.00 sec)
```

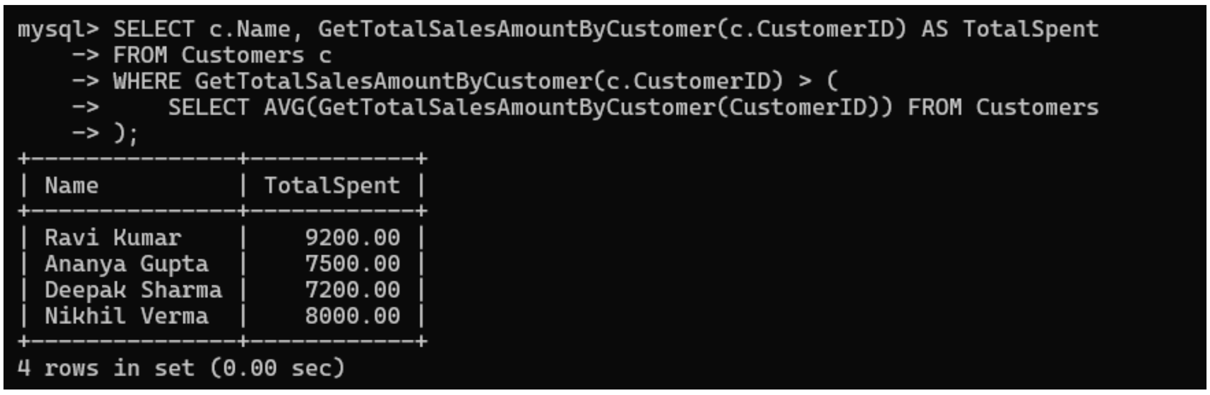


```
mysql> SELECT c.Name AS CustomerName, SUM(sod.Quantity) AS TotalItemsBought,
    ->        SUM(sod.Quantity * sod.UnitPrice) AS TotalSpent
    -> FROM Customers c
    -> JOIN SalesOrders so ON c.CustomerID = so.CustomerID
    -> JOIN SalesOrderDetails sod ON so.SOID = sod.SOID
    -> GROUP BY c.CustomerID;
+---------------+------------------+------------+
| CustomerName  | TotalItemsBought | TotalSpent |
+---------------+------------------+------------+
| Ravi Kumar    |               11 |    9200.00 |
| Ananya Gupta  |                1 |    7500.00 |
| Deepak Sharma |                2 |    7200.00 |
| Sonal Patel   |               10 |    2000.00 |
| Krishna Nair  |                4 |    4400.00 |
| Nikhil Verma  |                5 |    8000.00 |
+---------------+------------------+------------+
6 rows in set (0.00 sec)
```

NESTED QUERIES :



```
mysql> SELECT Name, Quantity
    -> FROM Products
    -> WHERE Quantity > (SELECT AVG(Quantity) FROM Products);
+--------------+----------+
| Name         | Quantity |
+--------------+----------+
| USB-C Cable  |      200 |
| Laptop Stand |       90 |
+--------------+----------+
2 rows in set (0.02 sec)

mysql> SELECT Name
    -> FROM Customers
    -> WHERE CustomerID IN (SELECT CustomerID FROM SalesOrders);
+---------------+
| Name          |
+---------------+
| Ravi Kumar    |
| Ananya Gupta  |
| Deepak Sharma |
| Sonal Patel   |
| Krishna Nair  |
| Nikhil Verma  |
+---------------+
6 rows in set (0.02 sec)

mysql> SELECT Name
    -> FROM Products
    -> WHERE ProductID NOT IN (SELECT ProductID FROM SalesOrderDetails);
Empty set (0.00 sec)

mysql> SELECT SupplierName
    -> FROM Suppliers
    -> WHERE SupplierID = (
    ->     SELECT SupplierID
    ->     FROM Products
    ->     GROUP BY SupplierID
    ->     ORDER BY COUNT(ProductID) DESC
    ->     LIMIT 1
    -> );
+---------------------+
| SupplierName        |
+---------------------+
| TechSource Pvt Ltd  |
+---------------------+
1 row in set (0.00 sec)
```



```
mysql> SELECT c.Name, GetTotalSalesAmountByCustomer(c.CustomerID) AS TotalSpent
    -> FROM Customers c
    -> WHERE GetTotalSalesAmountByCustomer(c.CustomerID) > (
    ->     SELECT AVG(GetTotalSalesAmountByCustomer(CustomerID)) FROM Customers
    -> );
+---------------+------------+
| Name          | TotalSpent |
+---------------+------------+
| Ravi Kumar    |    9200.00 |
| Ananya Gupta  |    7500.00 |
| Deepak Sharma |    7200.00 |
| Nikhil Verma  |    8000.00 |
+---------------+------------+
4 rows in set (0.00 sec)
```