Inventory Management System

SQL Mini Project

Submitted by:

Group 5

Advisor: Dr. Prasenjit Choudhury

08.04.2025 (V1.0)

Contents

1 Team Details

Roll Number	Registration Number	Name
23CS8021	23U10094	SHIVAM ROY
23CS8022	23U10097	VENKATA DURGA BHAVANI KALLA
23CS8023	23U10100	KHUSI SHAW
23CS8024	23U10105	SIDDHANT SINGH
23CS8025	23U10109	SAPTARSHI DHIBAR

2 Problem Statement

The objective of this project is to create a simple Inventory Management System for a retail shop. The system will maintain **products**, **suppliers**, and **sales transactions** to support business operations.

Database Tables

The following tables will be used to hold the system's data:

- **Products**: (ProductID, Name, Price, Stock, SupplierID)

- **Suppliers**: (SupplierID, SupplierName, Contact)

- **Sales**: (SaleID, ProductID, QuantitySold, SaleDate, TotalAmount)

Queries

The system will support the following queries:

1. List all products with stock less than 10.

2. Get details of suppliers providing a particular product.

3. Get total sales for a particular month.

4. Reduce stock after a sale.

5. Get best-selling products.

6. Get sales made within the last 7 days.

7. Remove a product that has been discontinued from stock.

8. Get the most revenue-producing products.

9. Determine the least-selling products.

10. Identify the suppliers that provide the greatest number of products.

3 Purpose of the Project

The purpose of this project is to create a simple **Inventory Management System (IMS)** to improve the flow of goods in a retail store such that there are sufficient stocks but minimal excess.

Objectives

The principal tasks are:

- **Track Products**: Keep product information (ID, name, price, quantity).

- **Manage Suppliers**: Maintain supplier details.

- **Track Sales**: Record sales and produce reports.

- **Automate Updates**: Real-time adjustments to inventory after sale.

- **Reports and Queries**: Report sales and query inventory levels.

Significance

Proper inventory management avoids stockouts, overstocking, and enhances supplier relationships, leading to well-informed decisions and maximum profitability.

Scope

The system will:

- Maintain products and suppliers.

- Automate sales tracking and updating inventory.

- Give reports and simple queries.

Developed using SQL, it can be accessed through a text or web interface.

Impact

The IMS will enhance efficiency, enable improved decision-making, maximize stock, and enhance supplier relationships, leading to overall store performance improvement.

4 Introduction

Inventory management is a very important aspect of any retail company, as it has direct impact on the store's capability to deliver goods to its consumers, achieve operational efficiency, and maximize profits. An inefficient inventory system can cause unnecessary inventory leading to high holding costs and even wastage, or at its worst, stockouts that may infuriate consumers and result in missed sales.

The **Inventory Management System (IMS)** of a retail shop is intended to overcome these problems by offering an uninterrupted means of monitoring products, suppliers, and sales transactions. This project is intended to create a system that not only monitors products and their availability but also updates in real-time based on sales operations, so inventory records are always up to date.

In a store setting, it can be a labor-intensive and error-ridden function to track products, suppliers, and sales manually. Sometimes, companies resort to using spreadsheets or simple systems, which are susceptible to errors and inefficiencies. The suggested IMS will automate principal functionalities and it will be simpler to track levels of stock, patterns of sales, and the performance of suppliers, minimising errors and the overall inefficiency of the store.

Technological Approach

The project will utilize **SQL** to develop and maintain the database, where product, supplier, and sales data will be stored. SQL's power and flexibility make it a perfect fit for managing queries and making sure the system is scalable as the business expands. The front-end of the system can be developed as a simple console-based program for the time being, but the system is easily scalable to a web-based interface, thus making it flexible for future growth.

The system will also have several helpful features like querying low stock products, fetching best-selling products, and creating reports on sales within certain time ranges. This will help store managers make decisions in a timely manner.

In the following sections, we will explore the database structure, queries, and technical implementation in greater detail.

5 Implementation

The system is implemented using SQL for database management.

Database Design

The following tables were created to manage the information:

- **Suppliers**: Stores supplier information, including SupplierID, SupplierName, and Contact.

```
CREATE TABLE Suppliers (
    SupplierID INT PRIMARY KEY,
    SName VARCHAR(100),
    Contact VARCHAR(15)
);

INSERT INTO Suppliers(SupplierID, SName, Contact) VALUES
(101, 'Reliance Digital', '9876543210'),
(102, 'Croma', '9123456789'),
(103, 'Big C Mobiles', '9988776655'),
(104, 'Poorvika Mobiles', '9011223344'),
(105, 'Vijay Sales', '9090909090'),
(106, 'Sangeetha Mobiles', '8888888888'),
(107, 'Spice Digital', '8777666555');
```

- **Products**: Stores product information, including ProductID, Name, Price, Stock, and SupplierID.

```
CREATE TABLE Product (
    ProductID INT PRIMARY KEY,
    PName VARCHAR(100),
    Price DECIMAL(10,2),
    Stock INT,
    SupplierID INT,
    FOREIGN KEY(SupplierID) REFERENCES Suppliers(SupplierID)
);
```

```
INSERT INTO Product(ProductID, PName, Price, Stock, SupplierID) VALUES
(1, 'Laptop A1', 45000.00, 10, 101),
(2, 'Wireless Mouse', 700.00, 40, 102),
(3, 'Web Camera', 1200.00, 15, 101),
(4, 'Bluetooth Speaker', 1500.00, 25, 103),
(5, 'Smart Watch X', 2200.00, 8, 102),
(6, 'Phone Charger', 300.00, 50, 104),
(7, 'Earphones', 500.00, 60, 105),
(8, 'Hard Disk 1TB', 3500.00, 12, 101),
(9, 'Power Adapter', 900.00, 30, 106),
(10, 'Computer Mouse', 600.00, 70, 102),
(11, 'External HDD 2TB', 5200.00, 9, 101),
(12, 'Mini Drone', 6800.00, 4, 107),
(13, 'Smart Mug', 800.00, 20, 106),
(14, 'Fitness Band', 1100.00, 11, 105),
(15, 'VR Headset', 7500.00, 3, 103);
```

- **Sales**: Tracks sales transactions, including SaleID, ProductID, QuantitySold, SaleDate, and TotalAmount.

```
CREATE TABLE Sales (
    SaleID INT PRIMARY KEY,
    ProductID INT,
    QuantitySold INT,
    SaleDate DATE,
    FOREIGN KEY(ProductID) REFERENCES Product(ProductID)
);

INSERT INTO Sales(SaleID, ProductID, QuantitySold, SaleDate) VALUES
(201, 1, 2, '2024-12-01'),
(202, 2, 4, '2024-12-02'),
(203, 3, 1, '2024-12-03'),
(204, 4, 3, '2024-12-03'),
(205, 5, 2, '2024-12-04'),
(206, 6, 5, '2024-12-05'),
(207, 7, 3, '2024-12-06'),
(208, 8, 2, '2024-12-07'),
```

```
(209, 9, 2, '2024-12-07'),
(210, 10, 6, '2024-12-08'),
(211, 11, 1, '2024-12-09'),
(212, 12, 2, '2024-12-10'),
(213, 13, 4, '2024-12-10'),
(214, 14, 3, '2024-12-11'),
(215, 15, 1, '2024-12-12'),
(216, 1, 5, '2024-09-01'),
(217, 4, 2, '2025-03-04'),
(218, 6, 2, '2025-04-03');
```

SQL Queries

Below are the SQL queries that were implemented to handle different functionalities:

- **Query to List Products with Stock Below 10**:

```
SELECT ProductID, PName
FROM Product
WHERE Stock < 10;
```

- **Query to Fetch Supplier Information for a Particular Product**:

```
SELECT s.SupplierID, s.SName, p.PName
FROM Suppliers s
JOIN Product p ON s.SupplierID = p.SupplierID;
```

- **Query to Find Total Sales for a Specific Month**:

```
SELECT MONTH(SaleDate) AS month, COUNT(s.QuantitySold) AS Nofsales
FROM Sales s
GROUP BY MONTH(SaleDate);
```

- **Query to Update Stock After a Sale**:

```
DELIMITER $$

CREATE TRIGGER update_stock_after_sale
AFTER INSERT ON Sales
FOR EACH ROW
BEGIN
    UPDATE Product
    SET Stock = Stock - NEW.QuantitySold
    WHERE ProductID = NEW.ProductID;
END$$

DELIMITER ;
```

- **Query to Get the Top-Selling Products**:

```
SELECT p.ProductID, p.PName
FROM Product p
JOIN Sales s ON s.ProductID = p.ProductID
WHERE s.QuantitySold = (SELECT MAX(QuantitySold) FROM Sales);
```

- **Query to Find Sales Made in the Last 7 Days**:

```
SELECT *
FROM Sales
WHERE SaleDate >= CURDATE() - INTERVAL 7 DAY;
```

- **Query to Remove a Discontinued Product from the Inventory**:

```
delimiter $$
create trigger after_supplier_delete
before delete on Suppliers
for each row
begin
    delete from Sales
    where ProductID in(
        select ProductID from Product where SupplierID=old.SupplierID
```

```
        );
        delete from Product
        where SupplierID=old.SupplierID;
    end $$
    delimiter ;
    delete from Suppliers
    where SupplierID=107;
```

- **Query to Retrieve Products with the Highest Revenue**:

```
SELECT p.ProductID, p.PName, SUM(s.QuantitySold * p.Price) AS TotalRevenue
FROM Product p
JOIN Sales s ON s.ProductID = p.ProductID
GROUP BY p.ProductID
ORDER BY TotalRevenue DESC
LIMIT 1;
```

- **Query to Identify the Least Sold Products**:

```
SELECT p.ProductID, p.PName, SUM(s.QuantitySold) AS QuantitySold
FROM Product p
JOIN Sales s ON s.ProductID = p.ProductID
GROUP BY p.ProductID
ORDER BY QuantitySold ASC
LIMIT 1;
```

- **Query to Find Suppliers Who Supply the Most Products**:

```
SELECT s.SupplierID, s.SName, COUNT(p.PName) AS No_of_Products_Given
FROM Suppliers s
JOIN Product p ON s.SupplierID = p.SupplierID
GROUP BY s.SupplierID
ORDER BY No_of_Products_Given DESC
LIMIT 1;
```

The complete project is available at
`https://github.com/mrshivamroy/Inventory-Management-System`.

6 Results and Discussions

Results

The following SQL queries were run to fetch and examine different parameters of the database. Below are the output snapshots of the tables and the outcomes of each query :
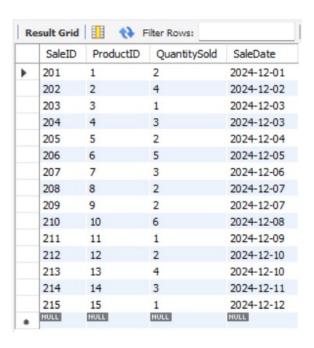

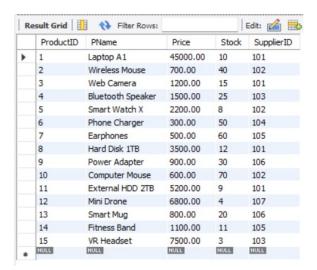Figure 6.1: Suppliers Table

Figure 6.2: Products Table

Figure 6.3: Sales Table

Figure 6.4: Results of Query to List Products with Stock Below 10

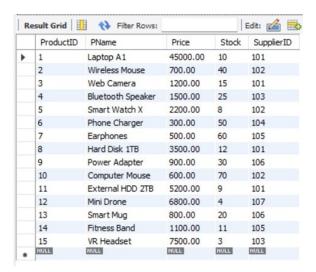Figure 6.5: Results of Query to Fetch Supplier Information for a Particular Product
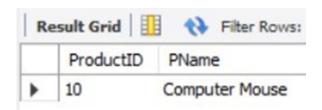

Figure 6.6: Results of Query to Find Total Sales for a Specific Month

Figure 6.7: Product Table before applying the query 4

Figure 6.8: Results of Query to Update Stock After a Sale
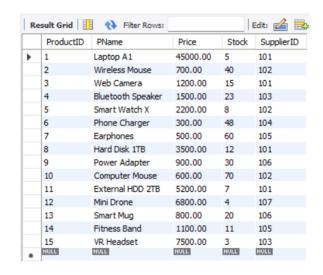

Figure 6.9: Results of Query to Get the Top-Selling Products

Figure 6.10: Results of Query to Find Sales Made in the Last 7 Days
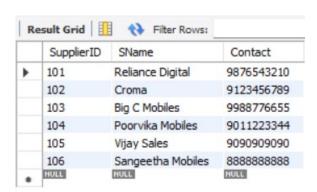
Figure 6.11: Results of Query to Remove a Discontinued Product from the Inventory Table Suppliers
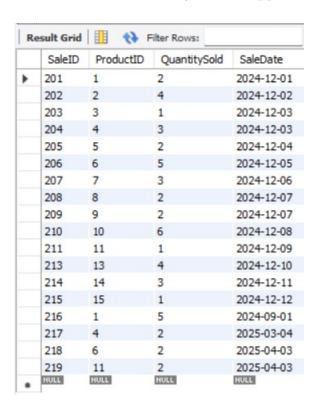

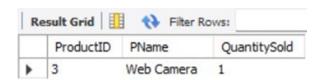Figure 6.12: Results of Query to Remove a Discontinued Product from the Inventory Table Products

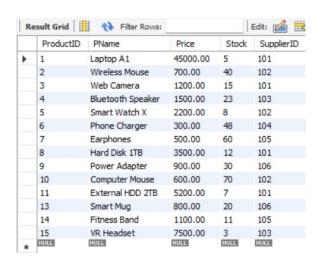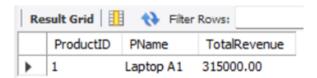Figure 6.13: Results of Query to Remove a Discontinued Product from the Inventory Table Sales

Figure 6.14: Results of Query to Retrieve Products with the Highest Revenue

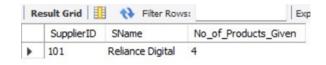Figure 6.15: Results of Query to Identify the Least Sold Products

Figure 6.16: Results of Query to Find Suppliers Who Supply the Most Products

7 Conclusion

The Inventory Management System implemented in this project makes inventory tracking, sales management, and supplier relations easier for retail stores, providing a scalable solution to universal problems.

Key Achievements

The system achieves the following:

- **Real-time Inventory Updates**: Maintains accurate records with automatic updates upon sales.

- **Supplier Management**: Facilitates supplier tracking and communication.

- **Sales Reporting**: Tracks sales, calculates revenues, and detects best-selling products.

- **Query Functionality**: Provides instant access to important information, such as low-stock products.

Impact on Retail Business Operations

The system saves employee time spent on stock, avoids stockouts, and offers actionable intelligence for improved buying decisions, which in turn maximizes profitability.

Future Work and Enhancements

Future enhancements are:

- **User Interface**: A GUI to enhance usability.

- **Cloud Integration**: Cloud connectivity for multi-location companies.

- **Advanced Analytics**: Such as predictive analytics and dynamic pricing.

- **Mobile Access**: Mobile application for remote inventory management.

Final Thoughts

This project brings retail operations into the modern era by automating key processes, eliminating errors, and enhancing profitability. It offers an essential tool for successful inventory control in the competitive retail market today.

8 Bibliography

- Silberschatz, A., Korth, H. F., Sudarshan, S. *Database System Concepts*, 7th Edition, McGraw-Hill Education, 2019.

- W3Schools. *SQL Tutorial*, `https://www.w3schools.com/sql/`, 2020.

- MySQL Documentation. *MySQL 8.0 Reference Manual*, `https://dev.mysql.com/doc/refman/8.0/en/`, 2020.

- Stack Overflow. *Database-related Questions and Discussions*, `https://stackoverflow.com/questions/tagged/database`, Accessed: 2022.