SQL-Based Retail Inventory Management System

# 1. Project Summary

This project simulates a real-world retail inventory system using MySQL. It involves managing products, sales, suppliers, and store-wise inventory. The objective is to analyze inventory flow, sales performance, and automate operational tasks like restocking alerts and update tracking using SQL triggers.

# 2. Database Design

The database includes the following tables:

- Products (product\_id, product\_name, category\_id, price, reorder\_level)

- Sales (sale\_id, product\_id, quantity\_sold, total\_amount, sale\_date)

- Inventory (product\_id, store\_id, quantity\_on\_hand, last\_updated)

- Stores (store\_id, store\_name, location)

- Suppliers (supplier\_id, supplier\_name, contact\_info)

- Purchases (purchase\_id, supplier\_id, product\_id, quantity\_purchased, purchase\_date)

# 3. Sample Data

Example from the Products table:

| product\_id | product\_name | category\_id | price |

|------------|--------------|-------------|-------|

| 101 | Jeans | 1 | 1200 |

| 102 | T-shirt | 1 | 500 |

# 4. Key SQL Queries

## Query 1: Reorder Alerts

SELECT   
 p.product\_name,  
 s.store\_name,  
 i.quantity\_on\_hand,  
 p.reorder\_level  
FROM Inventory i  
JOIN Products p ON i.product\_id = p.product\_id  
JOIN Stores s ON i.store\_id = s.store\_id  
WHERE i.quantity\_on\_hand < p.reorder\_level;

This query identifies products in each store that are below their reorder level.

## Query 2: Top 5 Revenue Generating Products

SELECT   
 p.product\_name,  
 SUM(s.total\_amount) AS revenue  
FROM Sales s  
JOIN Products p ON s.product\_id = p.product\_id  
GROUP BY p.product\_id  
ORDER BY revenue DESC  
LIMIT 5;

This query calculates the total revenue from each product and displays the top 5 performers.

## Query 3: Trigger for Auto-Updating Inventory Timestamp

DELIMITER //  
CREATE TRIGGER update\_last\_updated  
BEFORE UPDATE ON Inventory  
FOR EACH ROW  
BEGIN  
 SET NEW.last\_updated = CURDATE();  
END //  
DELIMITER ;

This trigger automatically updates the 'last\_updated' column whenever inventory is modified.

# 5. Conclusion

Through this project, I strengthened my SQL skills in data modeling, complex querying, and automation using triggers. It simulates a real-world backend for inventory analysis and retail operations. This project lays a strong foundation for database-driven business intelligence.