Inventory and Billing System using MySQL

Author: RISHIKA REDDY

Date: June 19, 2025
Contact: reddyrishika0912@gmail.com

Table of Contents

---

1. Introduction

The Inventory and Billing System is a sophisticated MySQL-based database solution developed to manage inventory, supplier relationships, customer transactions, and billing processes. Initiated on June 12, 2025, this project was completed within a 7-day timeline, culminating on June 19, 2025. The system addresses core business needs while incorporating advanced features to enhance operational efficiency and data analysis.

1.1 Objectives

- Ensure data integrity through structured schema design.
- Automate stock updates and billing operations via triggers.
- Provide analytical insights through procedures and views.
- Maintain an auditable record of all changes.

1.2 Scope

The system encompasses core functionalities such as product management, stock tracking, and invoicing, supplemented by bonus features including role-based access, product batch tracking, multi-warehouse support, and monthly analytics.

---

2. Technical Overview

2.1 Database Schema

The inventory_billing database is structured with the following tables:

- products (primary key: product_id, fields: name, sku, unit_price, min_stock_level).
- stock_movements (primary key: movement_id, foreign key: product_id).
- suppliers and customers (primary keys: supplier_id, customer_id).
- purchase_orders and sales_orders (primary keys: po_id, so_id, foreign keys: supplier_id, customer_id, product_id).
- invoices (primary key: invoice_id, foreign key: so_id).
- audit_logs (primary key: log_id, fields: table_name, action, record_id).
- Bonus Tables: roles (primary key: role_id), users (primary key: user_id, foreign key: role_id), product_batches (primary key: batch_id, foreign key: product_id), warehouses (primary key: warehouse_id), stock_locations (primary key: location_id, foreign keys: product_id, warehouse_id).

2.2 Implemented Features

- **Core Features**:
    - Product Management: View and manage product catalog.
    - Stock Movement Tracking: Record and monitor inventory changes.
    - Supplier and Customer Management: Maintain vendor and client data.
    - Purchase and Sales Order Processing: Handle procurement and sales.

    - Billing and Invoicing: Generate and track invoices.
    - Inventory Valuation: Calculate stock value using FIFO method.
    - Stock Alerts: Notify on low stock levels.
    - Audit Logging: Track all database modifications.
- **Bonus Features**:
    - Role-Based Access Control: Define user permissions.
    - Product Batches with Expiry Dates: Manage batch-level inventory.
    - Multiple Warehouse Support: Distribute stock across locations.
    - Monthly Analytics: Summarize sales and purchase data.

2.3 Tools and Technologies

- **Database Engine**: MySQL 8.0+ with InnoDB for transaction support.
- **Development Tool**: MySQL Workbench for design and execution.
- **Version Control**: Git (optional for change tracking).

---

3. Installation and Configuration

3.1 Prerequisites

- MySQL Server (version 8.0 or higher) installed and running.
- MySQL Workbench (latest version recommended).
- Administrative privileges on the MySQL instance.

3.2 Setup Instructions

1. **Database Initialization**:
    - Launch MySQL Workbench and establish a connection (e.g., "connection1").
    - Execute schema.sql from the schema/ directory to create the inventory_billing database and tables.
2. **Data Population**:
    - Run sample_data.sql from the data/ directory to populate tables with sample data.
3. **Feature Deployment**:
    - Execute triggers.sql, procedures.sql, and views.sql from their respective directories to enable automation, procedures, and views.
4. **Verification**:
    - Confirm table creation in the "Schemas" pane of MySQL Workbench.
    - Test with a sample query (e.g., SELECT * FROM vw_current_stock_levels;).
5. **Submission**:
    - Compress the project directory using:

4. Usage Guidelines

4.1 Core Feature Operations

- **Product Management**: Retrieve product details with SELECT * FROM products; or check stock status via vw_current_stock_levels.
- **Stock Movements**: Monitor changes with SELECT * FROM stock_movements; or trigger updates by modifying purchase_orders.
- **Supplier/Customer Management**: Access data using SELECT * FROM suppliers; or SELECT * FROM customers;.
- **Purchase Orders**: Update received quantities (e.g., UPDATE purchase_orders SET quantity_received = 40 WHERE po_id = 2;) to test stock triggers.
- **Sales Orders and Billing**: Insert sales (e.g., INSERT INTO sales_orders VALUES (..., '2025-06-19', ...);) and view invoices with vw_sales_order_summary.
- **Inventory Valuation**: Execute CALL sp_calculate_fifo_valuation(); for valuation reports.
- **Stock Alerts**: Run CALL sp_update_stock_alerts(); to log low stock notifications.
- **Audit Logs**: Review changes with SELECT * FROM vw_audit_log_summary;.

4.2 Bonus Feature Operations

- **Role-Based Access**: Manage permissions with SELECT * FROM roles; and assign users via users table.
- **Product Batches**: Check expiry status with SELECT * FROM vw_product_batch_status;.
- **Multiple Warehouses**: View distribution with SELECT * FROM vw_warehouse_stock;.
- **Monthly Analytics**: Generate reports with CALL sp_generate_monthly_sales_analytics(); and CALL sp_generate_monthly_purchase_analytics();.

---

5. Limitations and Future Enhancements

5.1 Limitations

- Triggers currently assign stock updates to the first warehouse; dynamic warehouse selection is not implemented.
- Inventory valuation (FIFO) relies on unit_price as a cost proxy; batch-specific costs are not tracked.
- The bonus feature "Integration with an external API for real-time stock updates" has not been implemented due to time constraints and project scope limitations.

5.2 Future Enhancements

- Integrate a real-time API for stock updates.
- Implement batch-specific cost tracking for accurate valuation.
- Enhance trigger logic to support dynamic warehouse assignment based on order data.

---

6. References

- MySQL Official Documentation: https://dev.mysql.com/doc/

---

7. Author Information

- **Name**: Rishika Reddy
- **College**: KL University.in
- **Student ID**: 2200033131
- **Date**: June 19, 2025
- **Contact**: reddyrishika0912@gmail.com

---

8. Version History

- **v1.0**: Initial release with all core and bonus features implemented, June 19, 2025.