

Project Report

Inventory and Warehouse Management System

Introduction

Efficient inventory and warehouse management is vital for companies to maintain accurate stock levels, optimize operations, and minimize manual errors. This project designs and implements a robust SQL backend for managing products, warehouses, suppliers, and stock quantities using PostgreSQL with pgAdmin4.

Abstract

The aim of this project is to build a relational database solution that tracks inventory levels, automates low stock alerts, and supports the transfer of stock between warehouses. Using triggers and stored procedures, critical inventory operations are automated to ensure consistency and integrity while providing clear reporting for timely decision-making.

Tools Used

- Database: PostgreSQL
- Interface: pgAdmin4
- Scripting: PL/pgSQL (procedures, triggers)

Steps Involved in Building the Project

1) Schema Modeling

- Suppliers: Stores supplier details.
- Warehouses: Contains warehouse names and locations.
- Products: Holds product data with reorder levels.
- Stock: Tracks product quantities in each warehouse.

2) Insert Sample Data

- Inserted at least 10 sample records per table to simulate real-world inventory.

3) Write Inventory Queries

- Query to display current stock across warehouses.
- Query to generate reorder alerts for products below their threshold.

4) Implement Trigger for Low Stock

- Created a `notify_low_stock` trigger function:
 - Executes after each insert/update on the stock table.
 - Compares new quantity with the product's reorder level.
 - Raises a notice if quantity is lower.

5) Create Stored Procedure for Stock Transfer

- Wrote transfer_stock procedure:
 - Checks if the source warehouse has enough stock.
 - Deducts quantity from source.
 - Inserts/updates destination warehouse's stock using ON CONFLICT.

6) Documentation and Testing

- Documented SQL schema, queries, trigger, and stored procedure.
- Verified the entire flow with test inserts, updates, and transfers.

Conclusion

This project demonstrates how a relational database can streamline inventory control and automate warehouse processes. By integrating triggers and stored procedures in PostgreSQL, we ensure data accuracy, automatic alerts, and reliable stock transfers. This approach forms a scalable base for further improvements like audit logging, user roles, or integration with front-end apps.

Deliverables

- ✓ SQL Schema
- ✓ Sample Data
- ✓ Queries
- ✓ Trigger
- ✓ Stored Procedure
- ✓ Complete Documentation

Prepared By: Darji Chintankumar Dineshchandra

Date: July 15, 2025

How to Use

1. Open pgAdmin4.
2. Run schema and data scripts.
3. Execute test queries.
4. Verify triggers and stored procedures.