Inventory Management System - SQL Script Explanation

## Introduction / Project Overview

This project entails the development of a browser-based Inventory Management System tailored for small-scale retailers and shop owners. The system simplifies the process of tracking and maintaining stock levels by offering capabilities such as adding new items, modifying product information, removing outdated entries, and reviewing the current inventory. Additionally, it includes a built-in mechanism to alert users when the quantity of a product drops below a predefined threshold.

## Objective / Purpose

The primary goals of this system are as follows:

* + Facilitate the structured storage and management of product information using a relational database.
  + Provide basic functionality to insert, modify, delete, and retrieve product data through a user-friendly interface.
  + Implement an automatic alert system for low stock levels to aid timely reordering.

## TABLES

## Categories Table

CREATE TABLE Categories (

id INT PRIMARY KEY IDENTITY(1,1),

name NVARCHAR(50) NOT NULL,

description NVARCHAR(MAX)

);

**Purpose:** Stores different product categories (e.g., Electronics, Furniture).

- id: Unique identifier.

- name: Category name.

- description: Optional details about the category.

## Suppliers Table

CREATE TABLE Suppliers (

id INT PRIMARY KEY IDENTITY(1,1),

name NVARCHAR(100) NOT NULL,

contact\_number NVARCHAR(20),

email NVARCHAR(100),

address NVARCHAR(MAX)

);

Purpose: Stores supplier information.

- Includes contact number, email, and address.

## Users Table

CREATE TABLE Users (

id INT PRIMARY KEY IDENTITY(1,1),

username NVARCHAR(50) NOT NULL UNIQUE,

password NVARCHAR(255) NOT NULL,

role NVARCHAR(10) NOT NULL CHECK (role IN ('admin', 'manager', 'staff')),

email NVARCHAR(100) UNIQUE,

created\_at DATETIME DEFAULT GETDATE()

);

Purpose: Stores system user credentials and roles.

- Role is restricted to: admin, manager, or staff.

## Products Table

CREATE TABLE Products (

id INT PRIMARY KEY IDENTITY(1,1),

name NVARCHAR(100) NOT NULL,

category\_id INT,

supplier\_id INT,

quantity INT NOT NULL DEFAULT 0,

reorder\_level INT DEFAULT 0,

price DECIMAL(10,2) NOT NULL,

created\_at DATETIME DEFAULT GETDATE(),

CONSTRAINT FK\_Products\_Categories FOREIGN KEY (category\_id) REFERENCES Categories(id),

CONSTRAINT FK\_Products\_Suppliers FOREIGN KEY (supplier\_id) REFERENCES Suppliers(id)

);

Purpose: Central table to manage products.

- Links to Categories and Suppliers.

## Customers Table

CREATE TABLE Customers (

id INT PRIMARY KEY IDENTITY(1,1),

name NVARCHAR(100) NOT NULL,

email NVARCHAR(100),

phone NVARCHAR(20),

address NVARCHAR(MAX)

);

Purpose: Stores information about customers.

## Stock\_Log Table

CREATE TABLE Stock\_Log (

id INT PRIMARY KEY IDENTITY(1,1),

product\_id INT NOT NULL,

change\_type NVARCHAR(10) NOT NULL CHECK (change\_type IN ('in', 'out')),

quantity\_changed INT NOT NULL,

changed\_by INT,

change\_date DATETIME DEFAULT GETDATE(),

notes NVARCHAR(MAX),

CONSTRAINT FK\_StockLog\_Products FOREIGN KEY (product\_id) REFERENCES Products(id),

CONSTRAINT FK\_StockLog\_Users FOREIGN KEY (changed\_by) REFERENCES Users(id)

);

Purpose: Tracks inventory changes (stock in/out).

## Purchases Table

CREATE TABLE Purchases (

id INT PRIMARY KEY IDENTITY(1,1),

product\_id INT NOT NULL,

supplier\_id INT NOT NULL,

quantity INT NOT NULL,

purchase\_date DATETIME DEFAULT GETDATE(),

unit\_cost DECIMAL(10,2) NOT NULL,

total\_cost DECIMAL(10,2) NOT NULL,

CONSTRAINT FK\_Purchases\_Products FOREIGN KEY (product\_id) REFERENCES Products(id),

CONSTRAINT FK\_Purchases\_Suppliers FOREIGN KEY (supplier\_id) REFERENCES Suppliers(id)

);

Purpose: Logs product purchases.

## Sales Table

CREATE TABLE Sales (

id INT PRIMARY KEY IDENTITY(1,1),

product\_id INT NOT NULL,

customer\_id INT,

quantity\_sold INT NOT NULL,

sale\_price DECIMAL(10,2) NOT NULL,

total\_price DECIMAL(10,2) NOT NULL,

sale\_date DATETIME DEFAULT GETDATE(),

CONSTRAINT FK\_Sales\_Products FOREIGN KEY (product\_id) REFERENCES Products(id),

CONSTRAINT FK\_Sales\_Customers FOREIGN KEY (customer\_id) REFERENCES Customers(id)

);

Purpose: Records product sales.

## Returns Table

CREATE TABLE Returns (

id INT PRIMARY KEY IDENTITY(1,1),

sale\_id INT NOT NULL,

product\_id INT NOT NULL,

quantity\_returned INT NOT NULL,

return\_reason NVARCHAR(MAX),

return\_date DATETIME DEFAULT GETDATE(),

CONSTRAINT FK\_Returns\_Sales FOREIGN KEY (sale\_id) REFERENCES Sales(id),

CONSTRAINT FK\_Returns\_Products FOREIGN KEY (product\_id) REFERENCES Products(id)

);

Purpose: Tracks returned products.

## Payments Table

CREATE TABLE Payments (

id INT PRIMARY KEY IDENTITY(1,1),

customer\_id INT NOT NULL,

amount\_paid DECIMAL(10,2) NOT NULL,

payment\_date DATETIME DEFAULT GETDATE(),

payment\_method NVARCHAR(50) NOT NULL,

CONSTRAINT FK\_Payments\_Customers FOREIGN KEY (customer\_id) REFERENCES Customers(id)

);

Purpose: Records payments from customers.

# Main Features

### Add Product:

Enables users to insert new product records into the inventory.

### Edit Product Information:

Allows modification of existing product details as needed.

### View Products:

Displays all inventory items with relevant stock and category data.

### Delete Product:

Removes discontinued or unnecessary products from the system.

### Low Stock Warning:

Highlights items below their reorder level to prompt restocking.

# Expected outcomes

Upon successful completion, this project is expected to deliver:

* + A fully operational web application for small-scale inventory management.
  + Secure and structured storage of inventory data within a MySQL database.
  + An intuitive front-end interface that promotes ease of use.
  + Enhanced understanding of PHP-MySQL integration and the CRUD application model in practice.

# Conclusion

This inventory management system is designed not only to assist shopkeepers in maintaining real-time stock control but also to serve as a practical learning exercise for connecting server- side scripts with a relational database. It reflects key concepts in database operations and user interaction through a web-based interface, making it a strong foundation for more complex inventory solutions in the future