

---

## Core Concept

You're building a system that helps **track products, stock levels, suppliers, and inventory movement**. The idea is to manage all products coming into your business (stock-in) and going out (sales, damaged items, etc.)—while keeping everything organized in a central database.

---

## Core Tables You'll Need

### 1. Products Table

This will store all your items like keyboards, monitors, etc.

- **Fields:** Name, Category, Supplier, Quantity, Price, Reorder Level (to alert low stock), Dates
  - **Purpose:** Manage basic product information and keep track of stock level.
- 

### 2. Categories Table

Products will be grouped under categories (e.g., Electronics, Furniture).

- **Fields:** Name, Description
  - **Purpose:** Helps organize and filter products easily.
-

### 3. Suppliers Table

Stores info about vendors who provide products.

- **Fields:** Name, Contact Info, Address
  - **Purpose:** You'll know where each product came from and whom to contact for restocking.
- 

### 4. Stock-In Table

Whenever new stock is added (e.g., 30 new mice), it's logged here.

- **Fields:** Product, Quantity, Date, Supplier, Notes
  - **Purpose:** Helps trace incoming inventory and its source.
- 

### 5. Stock-Out Table

This logs when stock is sold or removed (e.g., sold to customers, damaged, etc.)

- **Fields:** Product, Quantity, Date, Customer/Purpose
  - **Purpose:** Tracks inventory going out of the system.
- 

### 6. Users Table

Admins and staff login accounts.

- **Fields:** Username, Password (encrypted), Role (Admin or Staff)
  - **Purpose:** Control who can access and modify what.
-

## **Key Functionalities You Should Include**

### **Product Management**

- Add new items
  - Edit product info
  - Delete (with checks – only if not linked to stock)
  - Search and filter (by name, category, stock level)
- 

### **Stock Management**

- **Stock-In:** Record every time you receive products.
  - **Stock-Out:** Record whenever products are sold, used, or lost.
  - **Live Quantity Tracking:** Every stock-in or out updates product quantity automatically.
  - **Low Stock Alerts:** System warns when quantity falls below reorder level.
- 

### **Reporting**

- **Daily/Monthly Sales Report**
  - **Current Inventory Summary**
  - **Product-wise Stock Movement**
  - **Supplier-wise Purchases**
  - **Out-of-Stock Products**
-

## **Advanced Functionalities (Recommended)**

### **1. Barcode Scanning Support**

- Helps scan product codes for faster check-in/out.

### **2. Multiple Warehouse Locations**

- Add a new locations table and track stock per warehouse or store.

### **3. Audit Trail (Who did what)**

- Log every change (added stock, deleted product, etc.) by user.

### **4. Export to Excel/PDF**

- Enable admins to download reports.

### **5. Stock Adjustment Option**

- Manually correct stock if physical count doesn't match the system.

### **6. Damaged/Returned Product Log**

- Track inventory that is returned or no longer usable.

### **7. Role-Based Access Control**

- Admins can do everything; staff can only view/add limited data.

### **8. Bulk Import of Products**

- Upload via CSV to avoid manual data entry.

---

## ⚙️ Useful SQL Features You'll Be Using

- **JOIN:** To combine data across tables (e.g., get product + supplier info).
- **GROUP BY:** To make summary reports (e.g., total stock sold this month).
- **TRIGGERS (optional):** Automatically update log/history on certain actions.
- **VIEWS:** Predefined queries for reporting dashboards.
- **FOREIGN KEYS:** Link tables properly and maintain data consistency.

---

## 🔑 Example: How the System Flows

1. You **add a product** → It's saved in the products table.
2. You **receive stock** → stock\_in is updated → product quantity increases.
3. You **sell/remove stock** → stock\_out is updated → product quantity decreases.
4. You **view reports** or get **low stock alerts**.
5. Admin can add/edit users, see reports, and maintain data security.