Information Systems 3
T3
Assignment

Due 17 August 2025

Uwaiz Laher - 221270191
Muhammad Siddeeq Rabin - 221084096
Strathmore York Fynn - 221547290

Catalog

Store Inventory Management System

Introduction

This document is made to inform the reader of the most important details relating to the Store Inventory Management System. It is meant to serve as a companion piece with the system itself. The system was designed to be used by store managers, giving them a wide overview of the store from the various items used for profit to the employees and regular customers.

At its core the system stores data relating to the stores inventory. Accompanying this it also stores data relating to the stores warehouse, contact and shift information for each and every employee, contact information for every regular customer (that has made an account with the store) and each and every sale.

Problem Statement

Smaller stores are are still using more archaic methods of inventory taking using physical methods such as notebooks, stock cards and other pen and paper systems.. This method is slow and time consuming, costing the store time and by extension profit. As the store grows, their selection of items expands, the amount of employees grow and a customer base is formed which slowly cultivates.. This only heightens the amount of work required to ensure that the store inventory is accurate and employee work times do not contradict themselves.

Restocking the store also takes far too much time. The quantity of a product is difficult to track when stock is only taken after hours using a notebook and so orders from the stores warehouse can only be done at the end of the day instead of immediately when a products quantity drops to a minimal amount. This issue causes deliveries from the warehouse to arrive at inopportune times and the following unloading, packing and restocking must be delayed until there are employees that can be assigned to this duty.

System Aim

The systems purpose is to streamline the store stock management process. Users(Store Managers) will be able to view the stores entire current inventory, information regarding their employees(Contact information along with Shift information), Warehouse stock and also information regarding regular customers.

Along with allowing the stores inventory to be viewed with far more ease the system will also allow for orders to be automatically sent to the warehouse when the quantity of a product drops below a certain amount. This will eliminate the need for a manual order to be issued by the manager. The warehouse orders delivery time will be view-able, allowing the manager time to assign employees to unpack and restock when it arrives.. The order status(whether it is pending or complete) will also be visible to the Manager.

Outside of the systems inventory management functions it also keeps track of employee data. This information revolves around employees themselves (including managers) and their shifts. Employee contact information will be stored allowing management to contact any employee if there is a disruption and the employee has not called in ahead. Employees will also be able to view their shift time and in which area of the store they will be assigned in for that shift (Food, Clothing and Home Decoration).

With this system outdated methods of inventory management will be rendered obsolete.

System Scope

The Store Inventory Management System is designed for tracking and managing the inventory of a singular store which sells 3 categories of products. These categories are Groceries, Clothing and Home decoration items. The system allows the stores managers to view inventory levels at any time, employee/customer information(Contact information for both and shift information for employees). Warehouse inventory and orders from the warehouse will also be view-able from the system though only managers will be able to view this data.

In Scope

- Adding, editing and deletion of records
- Tracking of sales transactions
- Tracking of product quantity
- Automatic warehouse order issuing
- Tracking of warehouse deliveries

Out of Scope

- Mobile Integration
- Offline functionality
- Online ordering

Requirements (Functional and Non-functional)

Functional

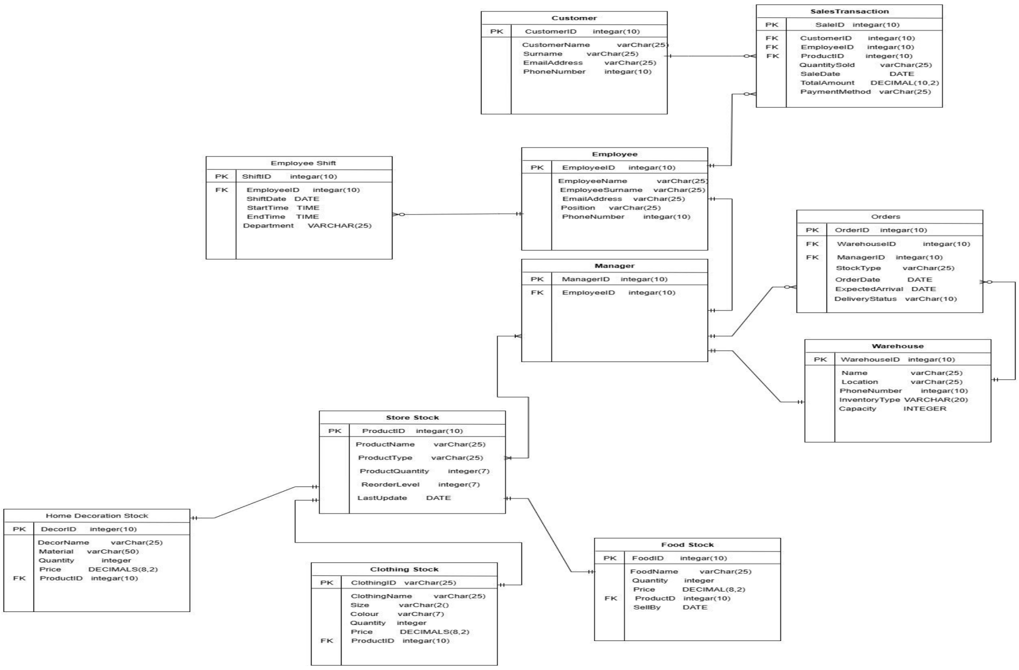
- Inventory Tracking
- Management of Products
- Management of Employees
- Tracking of Employee shifts
- Automatic order issuing when product quantity is low

Non-Functional

- System Performance
- Usability
- Scalability
- Reliability

Database Design

ERD



1. Every Employee must have a name, surname and position.
2. A sale must be linked to a valid customer and employee.
3. A Order is auto-generated when a products quantity drops below a certain number.
4. Each warehouse only holds products from 1 part of the stores product categories.
5. Store inventory must be updated whenever a sale is made.

Implementation

The system was created in MySQLiteStudio over a number of days. The system features 11 tables all written and executed within the database engine. The 11 tables are:

- Employee
- Manager
- Warehouse
- Store Stock
- Clothing Stock
- Food Stock
- Home Decoration Stock
- Orders
- Sales Transaction
- Employee Shifts
- Customers

Table Statements

Employee
Stores Employee Detail

```
CREATE TABLE Employee (
    EmployeeID INTEGER PRIMARY KEY AUTOINCREMENT,
    EmployeeName VARCHAR(25) NOT NULL,
    EmployeeSurname VARCHAR(25) NOT NULL,
    EmailAddress VARCHAR(25),
    Position VARCHAR(25) NOT NULL,
    PhoneNumber VARCHAR(10)
 );
```

Manager
Creates Manager table

```
CREATE TABLE Manager (
    ManagerID INTEGER PRIMARY KEY AUTOINCREMENT,
    EmployeeID INTEGER NOT NULL UNIQUE,
    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
);
```

Warehouse
Creates Warehouse table and defines its inventory type

```
CREATE TABLE Warehouse (
    WarehouseID INTEGER PRIMARY KEY AUTOINCREMENT,
    Name VARCHAR(25) NOT  NULL,
    Location VARCHAR(25) NOT NULL,
    PhoneNumber VARCHAR(10),
    InventoryType VARCHAR(25)CHECK (InventoryType IN ('Food','Clothing','Home Decor')),
    Capacity INTEGER NOT NULL
);
```

Store Stock
Tracks store inventory and reorder amount

```
CREATE TABLE StoreStock (
   ProductID INTEGER PRIMARY KEY AUTOINCREMENT,
   ProductName VARCHAR(25),
   ProductType VARCHAR(25)CHECK (ProductType IN ('Food','Clothing','Home Decor')),
   ProductQuantity INTEGER,
   ReorderLevel INTEGER,
   LastUpdate DATE
);
```

Clothing Stock
Tracks store inventory and reorder amount

```
CREATE TABLE ClothingStock (
   ClothingID INTEGER PRIMARY KEY AUTOINCREMENT,
   ClothingName VARCHAR(25),
   Size VARCHAR(25),
   Colour VARCHAR(25),
   Quantity INTEGER,
   Price DECIMAL(8,2),
   ProductID INTEGER,
   FOREIGN KEY (ProductID) REFERENCES StoreStock(ProductID)
);
```

Food Stock
Tracks store inventory and reorder amount

```
CREATE TABLE FoodStock (
   FoodID INTEGER PRIMARY KEY AUTOINCREMENT,
   FoodName VARCHAR(25),
   Quantity INTEGER,
   Price DECIMAL(8,2),
   ProductID INTEGER,
   SellBy DATE,
   FOREIGN KEY (ProductID) REFERENCES StoreStock(ProductID)
);
```

Home Decoration Stock
Tracks store inventory and reorder amount

```
CREATE TABLE HomeDecoStock (
   DecorID INTEGER PRIMARY KEY AUTOINCREMENT,
   DecorName VARCHAR(25),
   Material VARCHAR(25),
   Quantity INTEGER,
   Price DECIMAL(8,2),
   ProductID INTEGER,
   FOREIGN KEY (ProductID) REFERENCES StoreStock(ProductID)
);
```

Orders
Stores automatic and manually created orders

```
CREATE TABLE Orders (
   OrderID        INTEGER PRIMARY KEY AUTOINCREMENT,
   WarehouseID    INTEGER NOT NULL,
   ManagerID      INTEGER NOT NULL,
   StockType      VARCHAR(25) NOT NULL CHECK (StockType IN ('Food','Clothing','Home
Decor')),
   OrderDate      DATE NOT NULL DEFAULT (DATE('now')),
   ExpectedArrival DATE,
   DeliveryStatus VARCHAR(10) NOT NULL DEFAULT 'Pending',
   FOREIGN KEY (WarehouseID) REFERENCES Warehouse(WarehouseID),
   FOREIGN KEY (ManagerID)   REFERENCES Manager(ManagerID)
);
```

Sales Transaction
Tracks store sales

```
CREATE TABLE SalesTransactions (
   SaleID INTEGER PRIMARY KEY AUTOINCREMENT,
   CustomerID INTEGER NOT NULL,
   EmployeeID INTEGER NOT NULL,
   ProductID INTEGER NOT NULL,
   QuantitySold INTEGER NOT NULL,
   SaleDate DATE NOT NULL,
   TotalAmount DECIMAL(10,2) NOT NULL,
   PaymentMethod VARCHAR(25) NOT NULL CHECK (PaymentMethod IN ('EFT','Card','Cash')),
   FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
   FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
   FOREIGN KEY (ProductID) REFERENCES StoreStock(ProductID)
);
```

Employee Shifts
Managers epmployee shifts

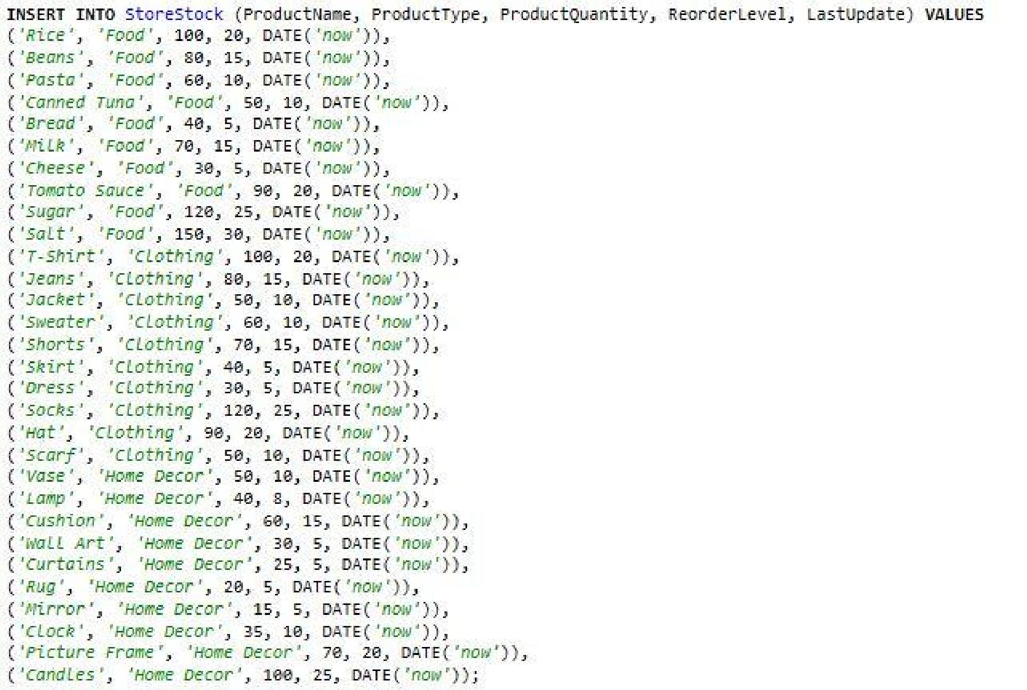
```
CREATE TABLE EmployeeShift (
   ShiftID INTEGER PRIMARY KEY AUTOINCREMENT,
   EmployeeID INTEGER,
   ShiftDate DATE NOT NULL,
   StartTime TIME NOT NULL,
   EndTime TIME NOT NULL,
   Department VARCHAR(25)  NOT NULL CHECK (Department IN ('Food','Clothing','Home
Decor')),
   FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
);
```

Customers
Keeps track of customers

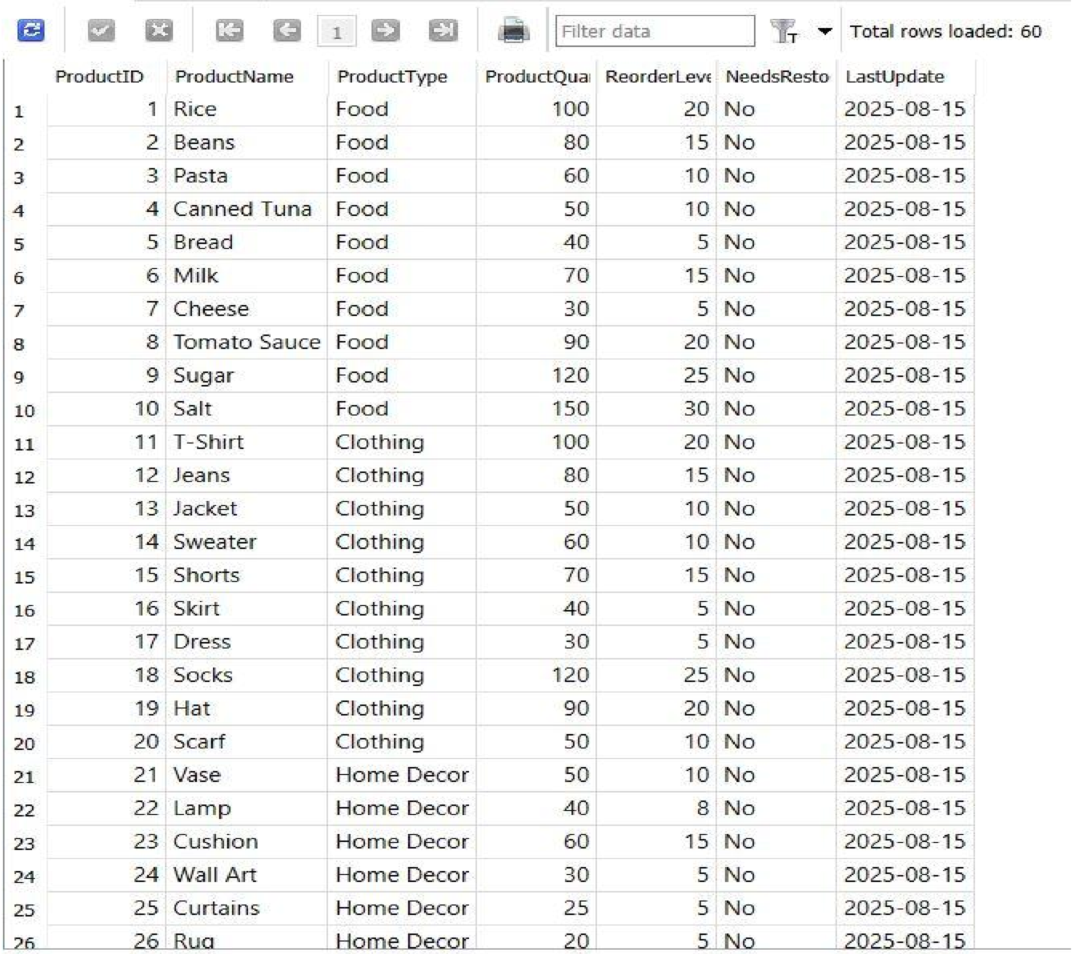
```
CREATE TABLE Customer (
   CustomerID INTEGER PRIMARY KEY AUTOINCREMENT,
   CustomerName       VARCHAR(25) NOT NULL,
   CustomerSurname    VARCHAR(25) NOT NULL,
   EmailAddress VARCHAR(50),
   PhoneNumber  VARCHAR(15)
);
```

Data Population

Each table has a minimum of 10 entries except for the ones which are auto-generated. Examples:



```
INSERT INTO StoreStock (ProductName, ProductType, ProductQuantity, ReorderLevel, LastUpdate) VALUES
('Rice', 'Food', 100, 20, DATE('now')),
('Beans', 'Food', 80, 15, DATE('now')),
('Pasta', 'Food', 60, 10, DATE('now')),
('Canned Tuna', 'Food', 50, 10, DATE('now')),
('Bread', 'Food', 40, 5, DATE('now')),
('Milk', 'Food', 70, 15, DATE('now')),
('Cheese', 'Food', 30, 5, DATE('now')),
('Tomato Sauce', 'Food', 90, 20, DATE('now')),
('Sugar', 'Food', 120, 25, DATE('now')),
('Salt', 'Food', 150, 30, DATE('now')),
('T-Shirt', 'Clothing', 100, 20, DATE('now')),
('Jeans', 'Clothing', 80, 15, DATE('now')),
('Jacket', 'Clothing', 50, 10, DATE('now')),
('Sweater', 'Clothing', 60, 10, DATE('now')),
('Shorts', 'Clothing', 70, 15, DATE('now')),
('Skirt', 'Clothing', 40, 5, DATE('now')),
('Dress', 'Clothing', 30, 5, DATE('now')),
('Socks', 'Clothing', 120, 25, DATE('now')),
('Hat', 'Clothing', 90, 20, DATE('now')),
('Scarf', 'Clothing', 50, 10, DATE('now')),
('Vase', 'Home Decor', 50, 10, DATE('now')),
('Lamp', 'Home Decor', 40, 8, DATE('now')),
('Cushion', 'Home Decor', 60, 15, DATE('now')),
('Wall Art', 'Home Decor', 30, 5, DATE('now')),
('Curtains', 'Home Decor', 25, 5, DATE('now')),
('Rug', 'Home Decor', 20, 5, DATE('now')),
('Mirror', 'Home Decor', 15, 5, DATE('now')),
('Clock', 'Home Decor', 35, 10, DATE('now')),
('Picture Frame', 'Home Decor', 70, 20, DATE('now')),
('Candles', 'Home Decor', 100, 25, DATE('now'));
```



	ProductID	ProductName	ProductType	ProductQua	ReorderLeve	NeedsResto	LastUpdate
1	1	Rice	Food	100	20	No	2025-08-15
2	2	Beans	Food	80	15	No	2025-08-15
3	3	Pasta	Food	60	10	No	2025-08-15
4	4	Canned Tuna	Food	50	10	No	2025-08-15
5	5	Bread	Food	40	5	No	2025-08-15
6	6	Milk	Food	70	15	No	2025-08-15
7	7	Cheese	Food	30	5	No	2025-08-15
8	8	Tomato Sauce	Food	90	20	No	2025-08-15
9	9	Sugar	Food	120	25	No	2025-08-15
10	10	Salt	Food	150	30	No	2025-08-15
11	11	T-Shirt	Clothing	100	20	No	2025-08-15
12	12	Jeans	Clothing	80	15	No	2025-08-15
13	13	Jacket	Clothing	50	10	No	2025-08-15
14	14	Sweater	Clothing	60	10	No	2025-08-15
15	15	Shorts	Clothing	70	15	No	2025-08-15
16	16	Skirt	Clothing	40	5	No	2025-08-15
17	17	Dress	Clothing	30	5	No	2025-08-15
18	18	Socks	Clothing	120	25	No	2025-08-15
19	19	Hat	Clothing	90	20	No	2025-08-15
20	20	Scarf	Clothing	50	10	No	2025-08-15
21	21	Vase	Home Decor	50	10	No	2025-08-15
22	22	Lamp	Home Decor	40	8	No	2025-08-15
23	23	Cushion	Home Decor	60	15	No	2025-08-15
24	24	Wall Art	Home Decor	30	5	No	2025-08-15
25	25	Curtains	Home Decor	25	5	No	2025-08-15
26	26	Rug	Home Decor	20	5	No	2025-08-15

Conclusion

The system successfully manages retail store operations by managing and logging all employees, managers, warehouses, inventory levels and sales transactions. It also automates the creation of warehouse orders when a products quantity is lowered to a certain number.

Limitations

- The system is limited to predefined and limited payment methods.
- The system does not have a user friendly interface for ease of use.

Future Improvements

- Development of a proper front-end user interface.
- Extending the amount of payment methods