

Carmen Municipal College

A.Y. 2014-2025

**StoreStocker**

**CC105 – Information Management**

**Midterm Project**

**MANDIN, JOHN MICHAEL**

**MAÑING, JOHN LINDSAY FLORENCE**

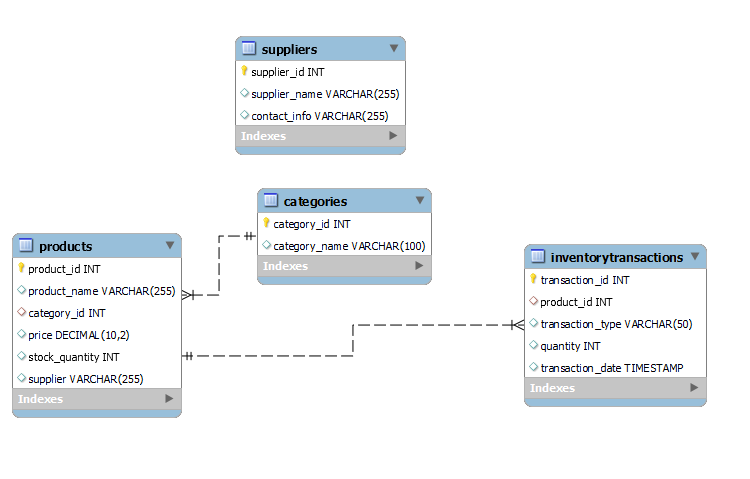
**I. Introduction**

**StoreStocker** is a comprehensive, multi-platform mobile application designed to streamline the inventory management process for grocery store owners and managers. The system aims to help users efficiently track, update, and organize their store's inventory, ensuring optimal stock levels and smooth operations.

The application allows users to manage multiple product categories (e.g., fresh produce, dairy, canned goods, snacks, etc.) and provides features to add new items, update stock quantities, and monitor sales. Grocery store managers can easily track product availability, receive notifications for low stock levels, and make necessary adjustments to keep inventory up to date.

Users can filter and search through inventory based on product name, category, or stock status. The system also allows for easy data modification, so any discrepancies or updates can be quickly addressed. Additionally, users have the option to delete any outdated or expired items from the inventory, freeing up space and ensuring that the system remains organized and efficient.

**II. Entity Relationship Diagram**



**Database**

**Table Name: CATEGORIES**

This table contains the categories of the products in the grocery store.

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Type (size)** | **Description** |
| category\_id | INT | Unique identifier for the product category. |
| category\_name | VARCHAR(100) | Name of the product category (e.g., Fresh Produce, Dairy, etc.). |

**Table Name: PRODUCTS**

This table contains the products in the store's inventory.

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Type (size)** | **Description** |
| product\_id | INT | Unique identifier for the product. |
| product\_name | VARCHAR(255) | Name of the product. |
| category\_id | INT | Foreign key that references the category\_id in the CATEGORIES table. |
| price | DECIMAL(10, 2) | Price of the product. |
| stock\_quantity | INT | Number of units of the product in stock. |
| supplier | VARCHAR(255) | Supplier of the product. |

**Table Name: INVENTORYTRANSACTIONS**

This table records inventory transactions, including stock updates, purchases, or sales of products.

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Type (size)** | **Description** |
| transaction\_id | INT | Unique identifier for the transaction. |
| product\_id | INT | Foreign key that references the product\_id in the PRODUCTS table. |
| transaction\_type | VARCHAR(50) | Type of transaction (e.g., 'Restock', 'Sale', 'Purchase'). |
| quantity | INT | Quantity involved in the transaction. |
| transaction\_date | TIMESTAMP | The date and time when the transaction occurred. |

**Table Name: SUPPLIERS**

This table stores information about the suppliers of the products.

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Type (size)** | **Description** |
| supplier\_id | INT | Unique identifier for the supplier. |
| supplier\_name | VARCHAR(255) | Name of the supplier. |
| contact\_info | VARCHAR(255) | Contact information for the supplier. |

***IV.*Functionalities**

**Creating table:** this code shows the creation of the tables

CREATE DATABASE StoreStocker;

USE StoreStocker;

CREATE TABLE Categories (

category\_id INT AUTO\_INCREMENT PRIMARY KEY,

category\_nameVARCHAR(100)

) ENGINE=InnoDB;

INSERT INTO Categories (category\_name)

VALUES ('Fresh Produce'),

('Dairy'),

('Canned Goods'),

('Snacks'),

('Beverages');

CREATE TABLE Products (

product\_id INT AUTO\_INCREMENT PRIMARY KEY,

product\_nameVARCHAR(255),

category\_id INT,

price DECIMAL(10, 2),

stock\_quantity INT,

supplier VARCHAR(255),

FOREIGN KEY (category\_id)

REFERENCES Categories(category\_id)

ON DELETE CASCADE

) ENGINE=InnoDB;

INSERT INTO Products (product\_name, category\_id, price, stock\_quantity, supplier)

VALUES ('Apple', 1, 0.99, 100, 'Fresh Farms'),

('Milk', 2, 1.49, 50, 'Dairy Co'),

('Canned Beans', 3, 0.79, 200, 'Canned Goods Inc'),

('Chips', 4, 2.49, 150, 'Snackables Ltd'),

('Soda', 5, 1.99, 120, 'Beverages Corp');

CREATE TABLE InventoryTransactions (

transaction\_id INT AUTO\_INCREMENT PRIMARY KEY,

product\_id INT,

transaction\_typeVARCHAR(50), -- 'Purchase' or 'Sale' or 'Restock'

quantity INT,

transaction\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (product\_id)

REFERENCES Products(product\_id)

ON DELETE CASCADE

) ENGINE=InnoDB

INSERT INTO InventoryTransactions (product\_id, transaction\_type, quantity)

VALUES (1, 'Restock', 50),

(2, 'Sale', 20),

(3, 'Restock', 100),

(4, 'Sale', 30),

(5, 'Restock', 40);

CREATE TABLE Suppliers (

supplier\_id INT AUTO\_INCREMENT PRIMARY KEY,

supplier\_nameVARCHAR(255),

contact\_infoVARCHAR(255)

) ENGINE=InnoDB;

INSERT INTO Suppliers (supplier\_name, contact\_info)

VALUES ('Fresh Farms', '123-456-7890'),

('Dairy Co', '987-654-3210'),

('Canned Goods Inc', '555-123-4567'),

('Snackables Ltd', '444-789-1234'),

('Beverages Corp', '666-222-8888')

SELECT p.product\_name, c.category\_name, p.price, p.stock\_quantity

FROM Products p

JOIN Categories c ON p.category\_id = c.category\_id;

SELECT p.product\_name, it.transaction\_type, it.quantity, it.transaction\_date

FROM InventoryTransactions it

JOIN Products p ON it.product\_id = p.product\_id

ORDER BY it.transaction\_date DESC;