INT6000 – DATABASE MANAGEMENT SYSTEMS

Module 6

Final Project: Grocery Management System

By

Jalajakshi Lali

Introduction

When I moved to USA recently, I observed the differences in the food requirements of people across different ethnicities. As I am from India, my diet predominantly contains long grain rice, lentils, Indian spices etc, which were not available in the stores close to the place I stay. Therefore I searched for stores which offer these products, but these were far-off. Most of the times, physically going to the store and looking for products is cumbersome and overwhelming. Additionally, not everyone gets the time to visit these stores every week for the groceries. There are a few other inconveniences I have observed as a customer. Often times when I order groceries online, the product is not listed as stocked out when unavailable and when I receive the order I get a replaced product. Therefore, I feel there is a need of creating a dynamic grocery management system which can mitigate these minor inconveniences and also it eases out the whole business process by making transactions effective and efficient.

Use Cases:

The users of this system will be Customers, Owner/Administrator, Store Employees. The app will have separate views for the admins, users and employees. In order to place order, the users will have to first sign up in the app. The users will have the option to purchase subscription. The customers with subscription will get a membership card. The other use cases include keeping track of products. The admin can track customer engagement and monitor the availability of products and restock them when unavailable. Once the Customer purchases a product, corresponding sales will be tracked and transaction details will be updated in payment section.

App’s Cost Model

The cost involved in developing the app would be database creation and maintenance. Basic reports can be exported from the app. Based on the reports extracted, the customers will be eligible for discounts, offers and deals with an option to take subscription. The deals include free delivery charges, low service charges, fast delivery options etc. The subscription will be time based and can be cancelled any time.

Business Analysis:

Customers: The app will be used by the customers primarily. The users have to register before they start to use the application. Registration is essential to identify each of the customer unambiguously. Information added during registration like name can be used to print on the bills.

Owner/ Administrator: Administrators are authorized to control the flow of data across systems. Therefore the admin privileges are only provided to the owner or a few people in the staff. Administrator can manipulate access to data available to the users. Admin can add or delete entries pertaining to product information. Allocate staff to pick items and pack them on receiving order. Access confidential customer information like customer card details etc.

Employees: The employees can make use of the system to check the order details. Mostly the employee access will be restricted to read only except for some cases where they would have to manipulate the transactions in case of refund or return. Employees can use the app to mark the order as completed and further packing it.

The entities involved in this system are

Customer: This entity holds all the information pertaining to the customer like customer ID (Primary Key), customer name, and contact details like phone number, email id and address

Membership Card: Details of Membership Card like Card ID (Primary Key), Card Number, Type and Expiry Date will be stored in this entity

Employee: The entity contains information like employee ID (Primary Key), name, hiring date, salary, job designation

Product: The product table would contain the product id(Primary Key), name, availability, and price

Sales: Data pertaining to Sales id(Primary Key), customer id(Foreign Key), product id(Foreign Key), Employee id(Foreign Key), quantity, and date of purchase will be available in this entity

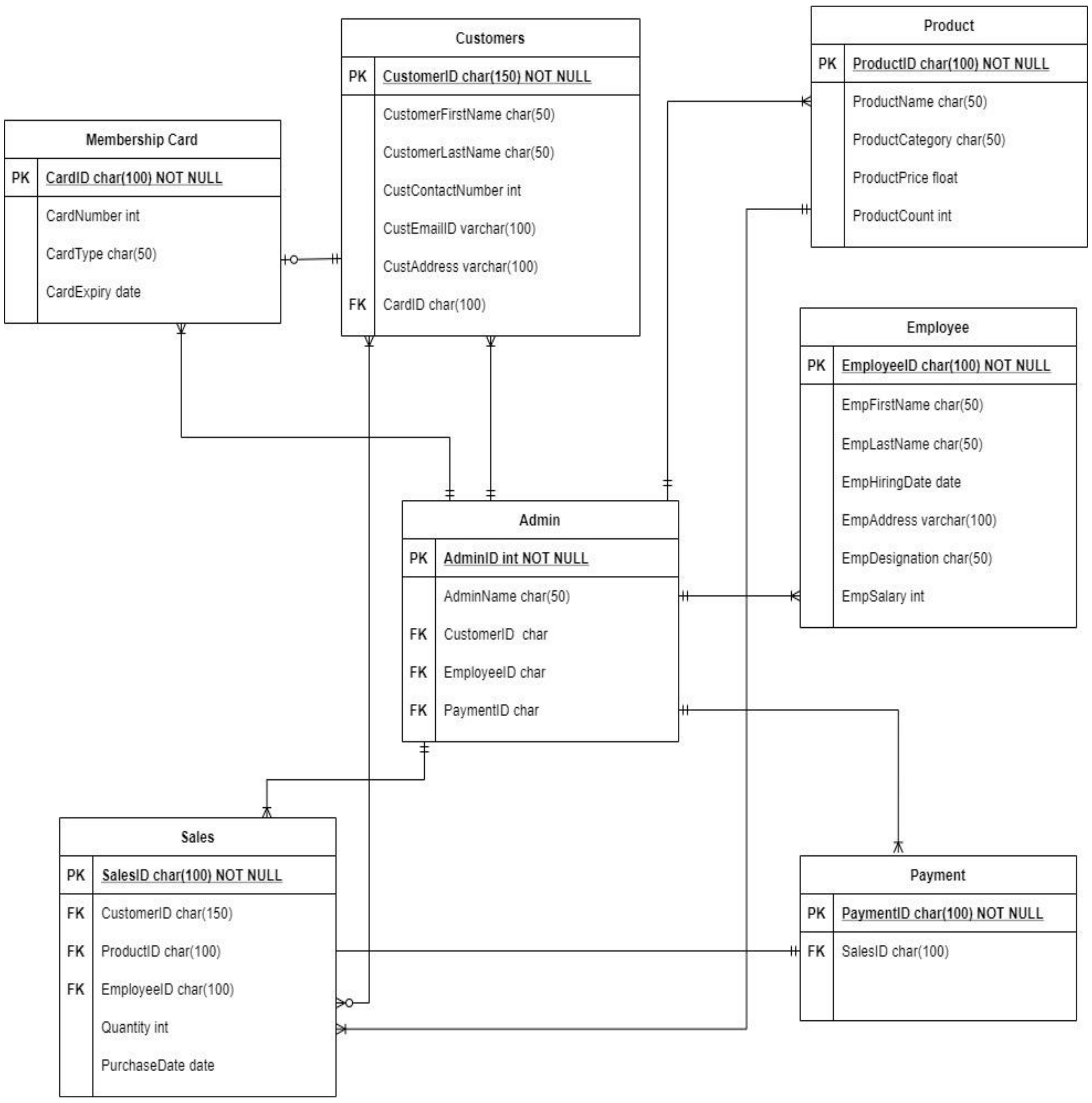
Payment: Payment Id (Primary Key) and Sales Id(Foreign Key) will be available in this entity

Admin: Admin acts as a master table, where details of Customer, Employee and Payment will be updated

Table Design and Analysis

The tables in this project are designed as per the following ERD

ERD

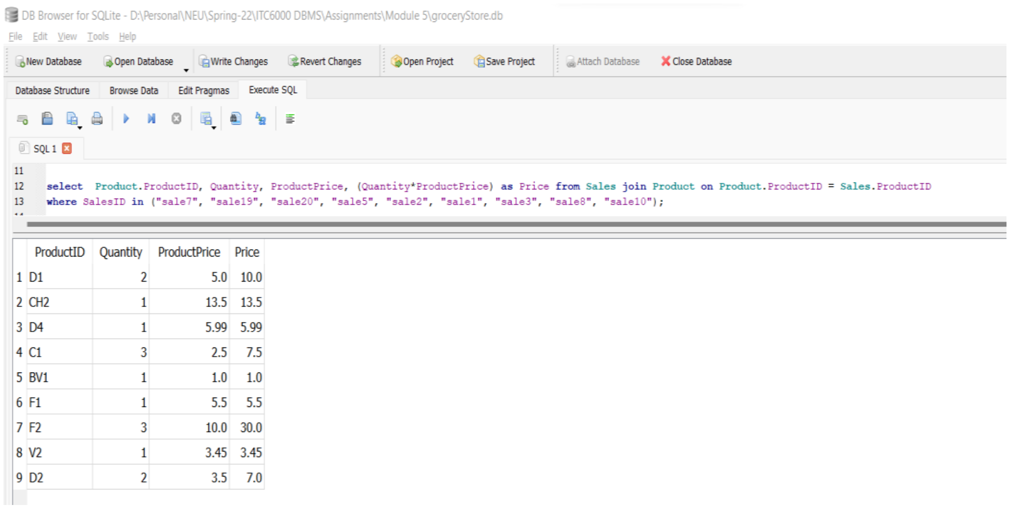


1. In total I would be creating 7 tables. Customer, Admin, Membership Card, Product, Employee, Payment, Sales
2. The Customer is the primary user. To order products customer needs to sign in and register. The details of registered customer will be stored in customer table
3. The product details available in grocery store can be fetched from product table. The admin can access the product stock details from this table.
4. The details filled by employees in employment form will be loaded into Employee table
5. Owner/Admin is the super user and admin has the privilege to access and manipulate data in these tables
6. On registering, the customer can take the subscription. The registered users with subscription will get a membership card. The details of this card will be stored in Membership Card details table
7. On purchase of products, the sales table will be updated, with details of customer id, product, quantity of products, and purchase date
8. After every transaction, payment table will be updated for every Sales

Database Implementation: In this project, I have created groceryStore.db and loaded data in the tables by importing csv files

Following are queries aligning with business rules

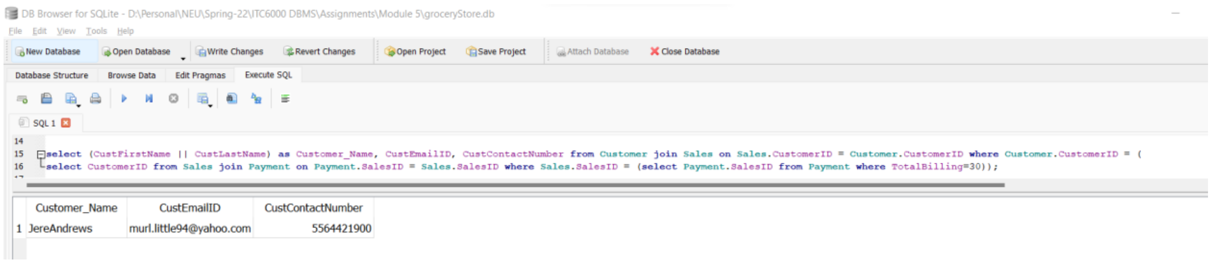
1. The following query provides detail on price of products ordered in particular sales



```
select  Product.ProductID, Quantity, ProductPrice, (Quantity*ProductPrice) as Price from Sales join Product on Product.ProductID = Sales.ProductID
where SalesID in ("sale7", "sale19", "sale20", "sale5", "sale2", "sale1", "sale3", "sale8", "sale10");
```

	ProductID	Quantity	ProductPrice	Price
1	D1	2	5.0	10.0
2	CH2	1	13.5	13.5
3	D4	1	5.99	5.99
4	C1	3	2.5	7.5
5	BV1	1	1.0	1.0
6	F1	1	5.5	5.5
7	F2	3	10.0	30.0
8	V2	1	3.45	3.45
9	D2	2	3.5	7.0

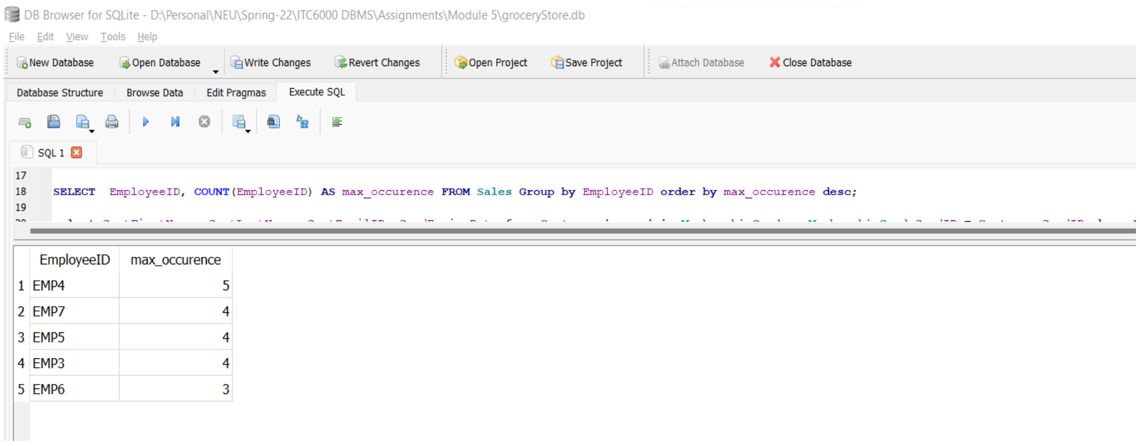
2. The following complex query is a join between customer, sales and payment table using which customer information can be retrieved when the total billing is equal to $30. This information can be used to provide special discounts to the customers who have made a purchase of $30.



```
select (CustFirstName || CustLastName) as Customer_Name, CustEmailID, CustContactNumber from Customer join Sales on Sales.CustomerID = Customer.CustomerID where Customer.CustomerID = (
select CustomerID from Sales join Payment on Payment.SalesID = Sales.SalesID where Sales.SalesID = (select Payment.SalesID from Payment where TotalBilling=30));
```

	Customer_Name	CustEmailID	CustContactNumber
1	JereAndrews	murl.little94@yahoo.com	5564421900

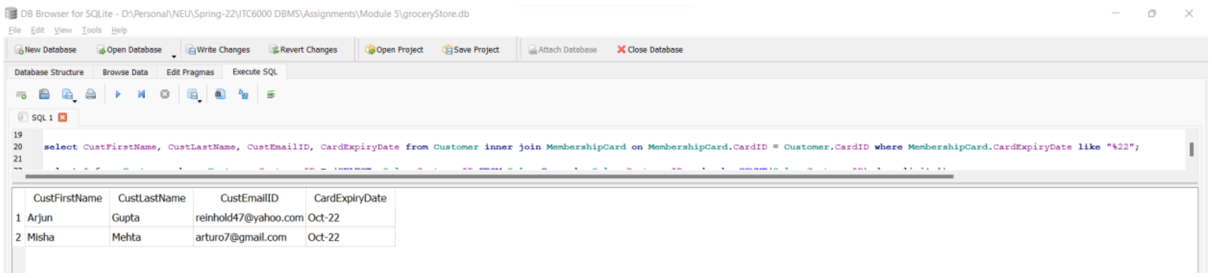
3. The following query can be used to check which employee is working on most orders. As per the data in my project Employee with id as EMP4 has worked on 5 orders which is the highest



```
SELECT  EmployeeID, COUNT(EmployeeID) AS max_occurence FROM Sales Group by EmployeeID order by max_occurence desc;
```

	EmployeeID	max_occurence
1	EMP4	5
2	EMP7	4
3	EMP5	4
4	EMP3	4
5	EMP6	3

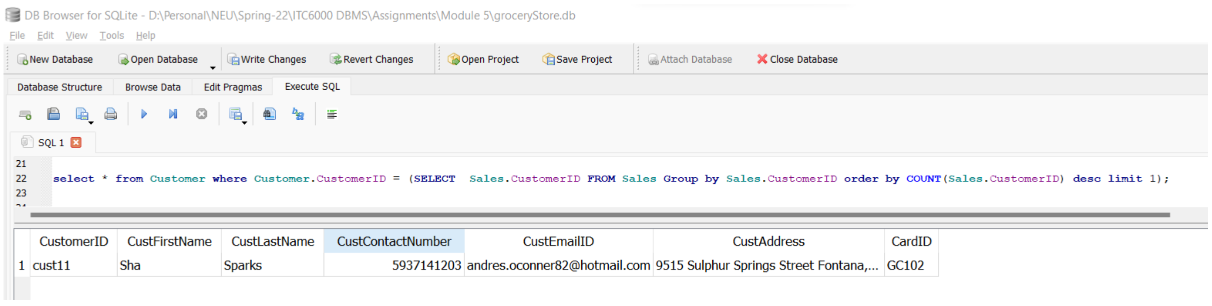
4. The following select query can be used to fetch information of customers with membership card whose expiry date is in the current year which is 2022. After retrieving this information the customers can be informed to update their subscription in order to provide the customers with a new card with renewed expiry date



```
select CustFirstName, CustLastName, CustEmailID, CardExpiryDate from Customer inner join MembershipCard on MembershipCard.CardID = Customer.CardID where MembershipCard.CardExpiryDate like "%22";
```

	CustFirstName	CustLastName	CustEmailID	CardExpiryDate
1	Arjun	Gupta	reinhold47@yahoo.com	Oct-22
2	Misha	Mehta	arturo7@gmail.com	Oct-22

5. The following query can be used to get information on the most loyal customer who has made many purchases from the grocery store. This information can be used to provide the customer with special offers and discounts



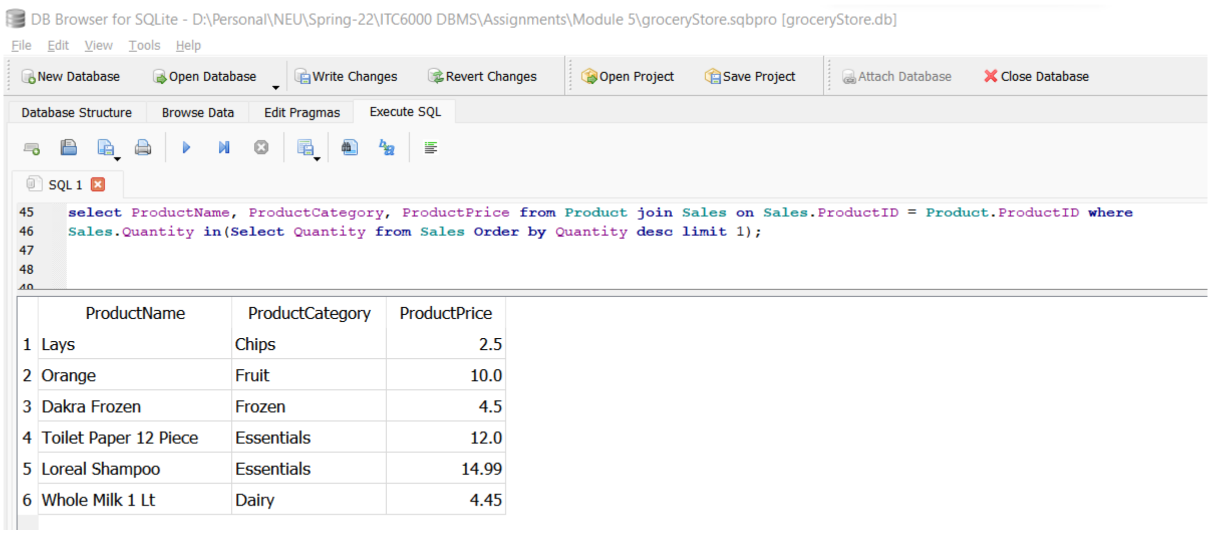
```
select * from Customer where Customer.CustomerID = (SELECT  Sales.CustomerID FROM Sales Group by Sales.CustomerID order by COUNT(Sales.CustomerID) desc limit 1);
```

	CustomerID	CustFirstName	CustLastName	CustContactNumber	CustEmailID	CustAddress	CardID
1	cust11	Sha	Sparks	5937141203	andres.oconner82@hotmail.com	9515 Sulphur Springs Street Fontana,...	GC102

Analytics, Reports, and Metrics

From the above queries, the report of employees who have worked on most orders can be retrieved using which Admin can raise their salary or promote them to a higher designation. The admin can get details of most loyal customer and offer deals and discounts.

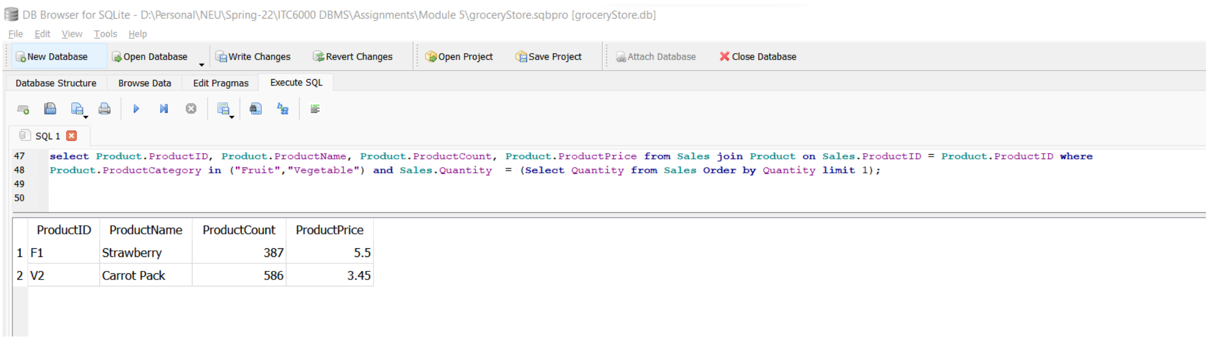
The following query can be executed to retrieve information regarding the most ordered products. Therefore as these products are most consumed these products can be stocked up more in the grocery store



```
select ProductName, ProductCategory, ProductPrice from Product join Sales on Sales.ProductID = Product.ProductID where
Sales.Quantity in(Select Quantity from Sales Order by Quantity desc limit 1);
```

	ProductName	ProductCategory	ProductPrice
1	Lays	Chips	2.5
2	Orange	Fruit	10.0
3	Dakra Frozen	Frozen	4.5
4	Toilet Paper 12 Piece	Essentials	12.0
5	Loreal Shampoo	Essentials	14.99
6	Whole Milk 1 Lt	Dairy	4.45

As the shelf life of fruits and vegetables is low, I was curious to check which fruit and vegetables have low sales. By executing the below query, we can retrieve details of the least ordered fruits and vegetables. It is clear that the Product count of Carrot pack is 586 which is high and as it is has low sales, the quantity for the same can be reduced from the inventory. Likewise, the sales of Strawberry is low accounting to the higher price, therefore the sales can be boosted if the price is reduced.



```sql
select Product.ProductID, Product.ProductName, Product.ProductCount, Product.ProductPrice from Sales join Product on Sales.ProductID = Product.ProductID where
Product.ProductCategory in ("Fruit","Vegetable") and Sales.Quantity  = (Select Quantity from Sales Order by Quantity limit 1);
```

	ProductID	ProductName	ProductCount	ProductPrice
1	F1	Strawberry	387	5.5
2	V2	Carrot Pack	586	3.45

Security Concerns

As the customers with subscription get a membership card which is a sensitive information, it becomes necessary that access to this table is restricted to the admin only. Sensitive information of customers like phone numbers, email id's are vulnerable and pose a security threat from hackers.

Architecture

Client Server Architecture: The application is built on single tiered architecture and is hosted on local instance of my machine. I have used DB Browser for SQLite to work on my project. The reports can be pulled by executing the queries aligned with the business rules.

Storage details: Windows 11, RAM: 8 GB, 64 bit operating system. The Space utilization for my project would be less than 4GB (as the application is small)

Project Wrap-up and Future Considerations

- Through this project I was able to build database based on single tier architecture which can be used to fetch reports
- I have learnt the concepts of Primary key and Foreign key and the usage
- Data can be loaded into tables by importing csv files
- Using join the information from multiple tables can be fetched at once
- In future, the application can be migrated to three tiered architecture with clear segregation between front end and back end

REFERENCES

- (2020, April 11). SQL Server CREATE TABLE: Creating a New Table in the Database. SQL Server Tutorial. https://www.sqlservertutorial.net/sql-server-basics/sql-server-create-table/
- GeeksforGeeks. (2021, October 28). How to Update Multiple Columns in Single Update Statement in SQL? https://www.geeksforgeeks.org/how-to-update-multiple-columns-in-single-update-statement-in-sql/
- Grocery Management System Project for Final Year. (2021, July 6). LovelyCoding.Org. https://www.lovelycoding.org/grossary-management-system/
- Import a CSV File Into an SQLite Table. (2022, April 3). SQLite Tutorial. https://www.sqlitetutorial.net/sqlite-import-csv/