A General-Purpose Mercantile Database

Project Report

CSE 5720 (Section 61) - Database Systems – Fall 2023

Submitted to

Department of Computer Science and Engineering

California State University, San Bernardino, California

By

Matthew Borja

December 10, 2023

Table of Contents

1. Introduction

Long gone are the days of record-keeping on paper. In fact, computerized record-keeping has been a stable of the modern store longer than I have been alive, with the introduction of Structured Query Language (SQL) in 1974. The vast adoption of the Internet as a staple of everyday life has only reinforced the necessity of a well-managed record-keeping system. The ability to store, access, organize, and analyze data efficiently is at the heart of any successful enterprise, be it brick-and-mortar or online. Therefore, it is my objective to present a versatile solution: a well-managed database capable of servicing both virtual and physical storefronts. This is a product that will: maintain an organized and efficient electronic record of data pertaining to products and orders; track revenue data as well as operating costs to maximize profit margins; facilitate accurate sales projections; and pass on savings as well as track customer spending habits through the creation of a rewards program.

2. Business Rules

In preparation for this project, it makes the most sense to brainstorm the conceptual structure of the resulting database through the use of Business Rules. Business Rules are the common starting point in designing any database from scratch for a client and takes the form of a plain English description of the entities, their attributes, relationships to other entities, and the cardinality of that relationship. In this project, I chose to model this database for a grocery store setting. One such set of business rules could look like the following:

A grocery store company relies on an information system where products are the main part of the company's information system. Products are characterized by their UPC Code (unique), Name, Size, Department Number, Price, and Price Type. One or more Products are compiled into a list called an

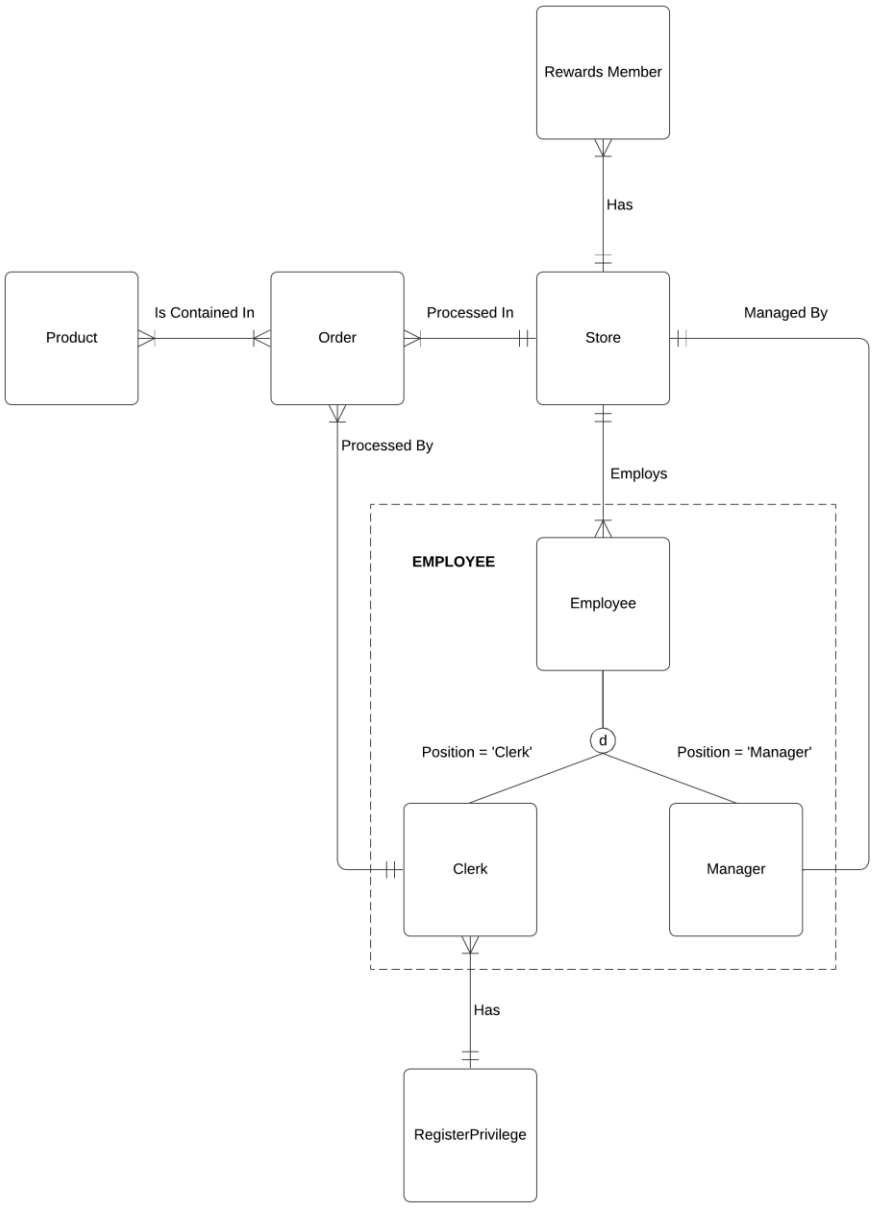
Order. An Order is characterized by their OrderID (unique), Date, Time, Rewards, Quantity, and Total. Orders are processed in a single Store. A Store is characterized by their StoreID (unique), Address, Telephone Number. Customers shop at a Store and generate Orders. Customers may sign up to be a part of a rewards program which will allow the Store to save their information and provide benefits such as discounts on various Products and including them in a mailing list of the latest deals. A Rewards Member is a customer that is characterized by their CustomerID (unique), Name, Address, Telephone Number, Cellphone Number, and E-mail. A Store employs Employees. Employees are characterized by their EmployeeID (unique), Name, Address, Telephone Number, Cellphone Number, Date of Hire, Position, and Salary. Employee has a partial specialization of Clerk or Manager. A Clerk is further characterized by ClerkID (unique). A Manager is further characterized by ManagerID (unique), and Privilege. Register Privilege keep track of who can use the registers and are characterized by RegisterPrivilegeID (unique), Username (ClerkID from Clerk), and Password. Clerks process Orders. A Store can have many Employees. Employees may only have one Store but can be loaned to other Stores as needed. A Store can have many Orders.

2.1. Methodology

To ensure the successful completion of this project, two components needed to be requisitioned: experience and data. Capitalizing on my past employment experience in the grocery industry, I have a foundational knowledge of the overall process that I am trying to simulate in the database. Furthermore, I would utilize the everyday grocery store items in my own house to populate the database with the necessary information and generate the rest with mock data, such as employee information, customer information, store information, etc.

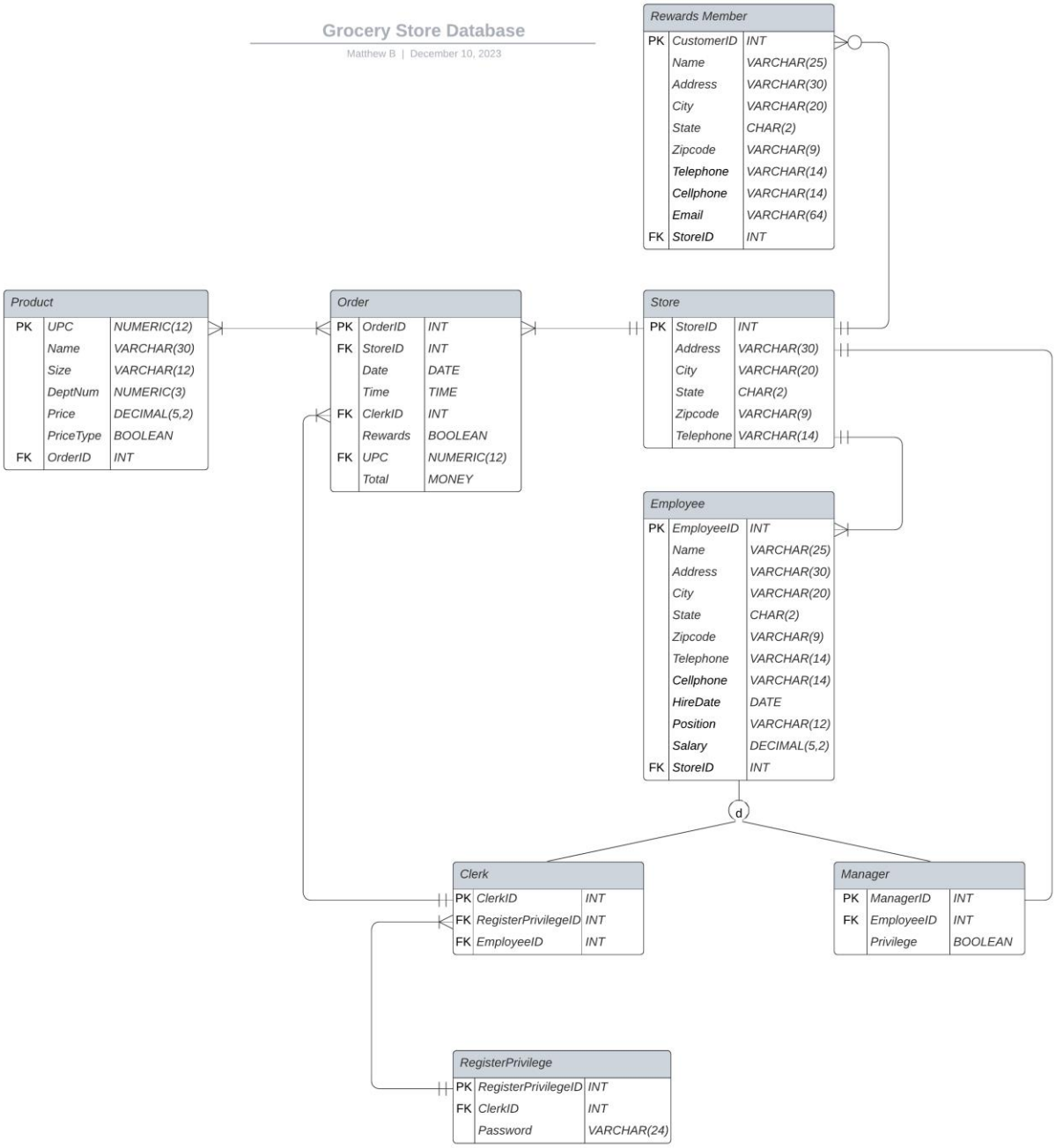
2.2. Defining Data Through Tables

Given the aforementioned business rules, the rough draft of the tables we will need for the database are illustrated as follows:



2.3. Further Defining Data Through Columns

Now with a clear segmentation of the data into tables, it is important to outline a more detailed illustration of the specifications given to us by the business rules.



2.4. Primary Keys

With consideration to primary keys, each table was determined to maintain their own unique IDs which were chosen to be their respective primary keys. Since every product is assigned a unique Universal Product Code (UPC), this made for a natural choice for the Product entity. The decision was made to enumerate all orders into unique IDs as you would find on a receipt with a transaction ID with OrderID being equivalent to transaction ID. Store numbers are commonplace in the business world, although this is irrelevant information to customers, therefore the StoreID uniquely identifies each store perfectly. Rewards Members are assigned a unique and incremented ID for which they can be identified and therefore this is their primary key. Employees are likewise assigned a unique ID number when they are hired and this serves as their primary key in the database. Furthermore, both Clerk and Manager have unique ID numbers to uniquely identify them as their primary keys. Lastly, a separate table is kept to maintain privileges to the registers on behalf of Clerks with a unique ID used as the primary key for this table.

2.5. Table Relationships

My assessment of the relationships between tables was illustrated in section 2.4. Here I will outline my reasoning from the ground up. One or many Products make up an Order. Likewise, a Product can be in many Orders. An Order takes place in a single Store and a Store processes many Orders in a day. One and only one Clerk is responsible for processing multiple Orders. A Store employs many Employees which are comprised of Clerks and Managers. A Store

has one and only one store Manager. A Store can have zero or many Rewards Members.

2.6. Implementation

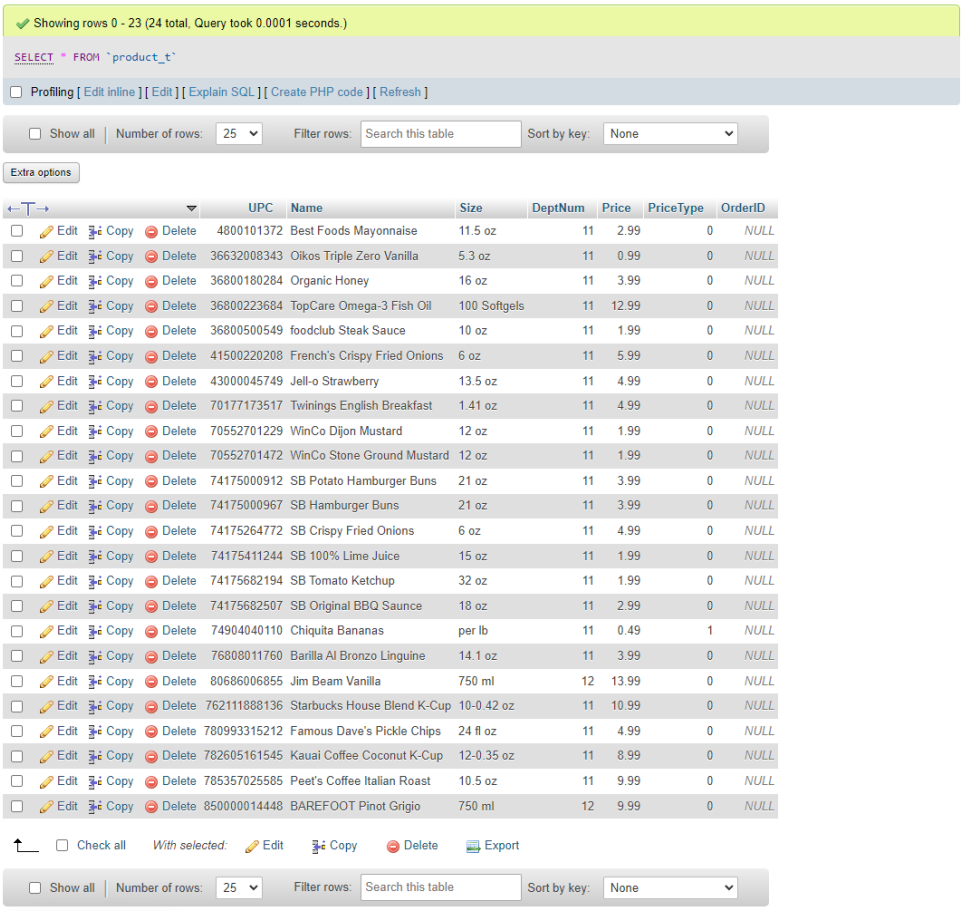
In the process of implementing my database blueprints, it became apparent that the Orders table primary key needed to be adjusted. Making the OrderID the primary key prevented multiple Products from populating the same order since by the rules of primary keys, there can be no duplicate entries under the same OrderID. By changing the primary key to include a composite key of (OrderID, UPC) this solved the problem, thus allowing one transaction to be made up of one or many products.

Furthermore, another refinement was needed to the Order table. This time, it was adding the foreign key to Rewards and changing it to an integer data type. By making this change, an order better reflected which one of our rewards members was conducting the purchase and giving the business a better way to analyze purchasing habits by the rewards members. In my experience, it is not uncommon for customers to come back and find fault with the transaction. A common SQL query that could be implemented in the software could look like the following:

This query is a way a manager can assist a customer with an order that did not satisfy the customer, they could easily look up the order through a variety of ways and resolve the problem.

Here is a compilation of mock data in the implemented database:

Products_t



Showing rows 0 - 23 (24 total, Query took 0.0001 seconds.)

```
SELECT * FROM `product_t`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

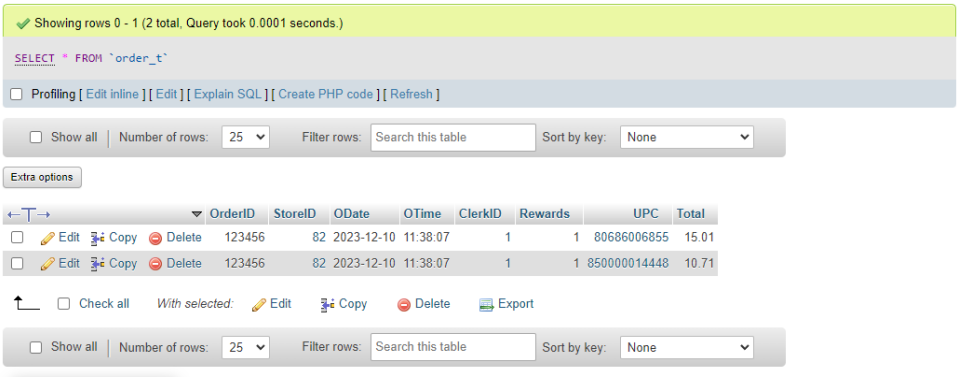
Extra options

	UPC	Name	Size	DeptNum	Price	PriceType	OrderID
Edit Copy Delete	4800101372	Best Foods Mayonnaise	11.5 oz	11	2.99	0	NULL
Edit Copy Delete	36632008343	Oikos Triple Zero Vanilla	5.3 oz	11	0.99	0	NULL
Edit Copy Delete	36800180284	Organic Honey	16 oz	11	3.99	0	NULL
Edit Copy Delete	36800223684	TopCare Omega-3 Fish Oil	100 Softgels	11	12.99	0	NULL
Edit Copy Delete	36800500549	foodclub Steak Sauce	10 oz	11	1.99	0	NULL
Edit Copy Delete	41500220208	French's Crispy Fried Onions	6 oz	11	5.99	0	NULL
Edit Copy Delete	43000045749	Jell-o Strawberry	13.5 oz	11	4.99	0	NULL
Edit Copy Delete	70177173517	Twinings English Breakfast	1.41 oz	11	4.99	0	NULL
Edit Copy Delete	70552701229	WinCo Dijon Mustard	12 oz	11	1.99	0	NULL
Edit Copy Delete	70552701472	WinCo Stone Ground Mustard	12 oz	11	1.99	0	NULL
Edit Copy Delete	74175000912	SB Potato Hamburger Buns	21 oz	11	3.99	0	NULL
Edit Copy Delete	74175000967	SB Hamburger Buns	21 oz	11	3.99	0	NULL
Edit Copy Delete	74175264772	SB Crispy Fried Onions	6 oz	11	4.99	0	NULL
Edit Copy Delete	74175411244	SB 100% Lime Juice	15 oz	11	1.99	0	NULL
Edit Copy Delete	74175682194	SB Tomato Ketchup	32 oz	11	1.99	0	NULL
Edit Copy Delete	74175682507	SB Original BBQ Saunce	18 oz	11	2.99	0	NULL
Edit Copy Delete	74904040110	Chiquita Bananas	per lb	11	0.49	1	NULL
Edit Copy Delete	76808011760	Barilla Al Bronzo Linguine	14.1 oz	11	3.99	0	NULL
Edit Copy Delete	80686006855	Jim Beam Vanilla	750 ml	12	13.99	0	NULL
Edit Copy Delete	762111888136	Starbucks House Blend K-Cup	10-0.42 oz	11	10.99	0	NULL
Edit Copy Delete	780993315212	Famous Dave's Pickle Chips	24 fl oz	11	4.99	0	NULL
Edit Copy Delete	782605161545	Kauai Coffee Coconut K-Cup	12-0.35 oz	11	8.99	0	NULL
Edit Copy Delete	785357025585	Peet's Coffee Italian Roast	10.5 oz	11	9.99	0	NULL
Edit Copy Delete	850000014448	BAREFOOT Pinot Grigio	750 ml	12	9.99	0	NULL

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Order_t



Showing rows 0 - 1 (2 total, Query took 0.0001 seconds.)

```
SELECT * FROM `order_t`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

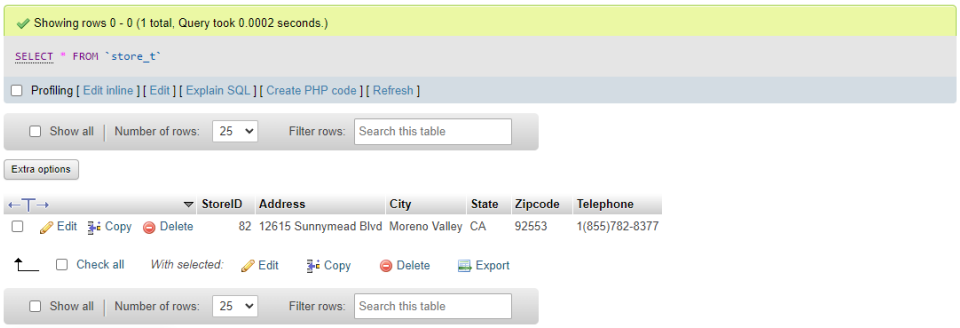
Extra options

	OrderID	StoreID	ODate	OTime	ClerkID	Rewards	UPC	Total
Edit Copy Delete	123456	82	2023-12-10	11:38:07	1	1	80686006855	15.01
Edit Copy Delete	123456	82	2023-12-10	11:38:07	1	1	850000014448	10.71

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Store_t



Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

```
SELECT * FROM `store_t`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

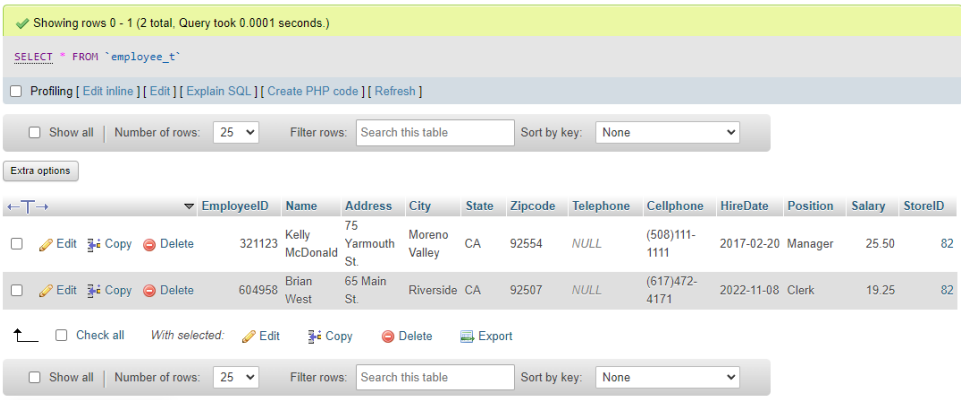
Extra options

	StoreID	Address	City	State	Zipcode	Telephone
Edit Copy Delete	82	12615 Sunnymead Blvd	Moreno Valley	CA	92553	1(855)782-8377

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

Employee_t



Showing rows 0 - 1 (2 total, Query took 0.0001 seconds.)

```
SELECT * FROM `employee_t`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

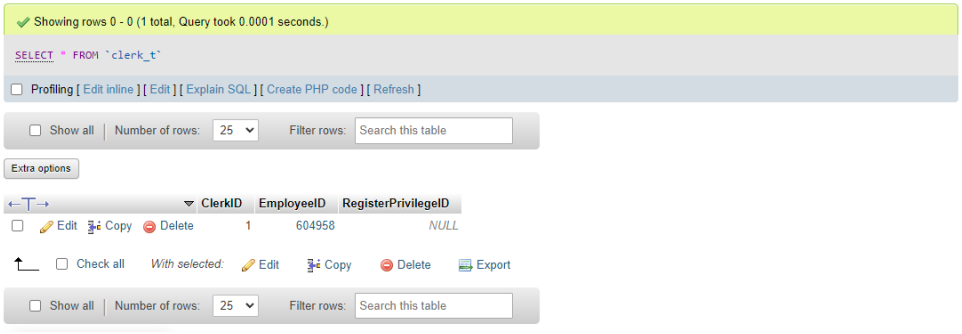
Extra options

	EmployeeID	Name	Address	City	State	Zipcode	Telephone	Cellphone	HireDate	Position	Salary	StoreID
Edit Copy Delete	321123	Kelly McDonald	75 Yarmouth St.	Moreno Valley	CA	92554	NULL	(508)111-1111	2017-02-20	Manager	25.50	82
Edit Copy Delete	604958	Brian West	65 Main St.	Riverside	CA	92507	NULL	(617)472-4171	2022-11-08	Clerk	19.25	82

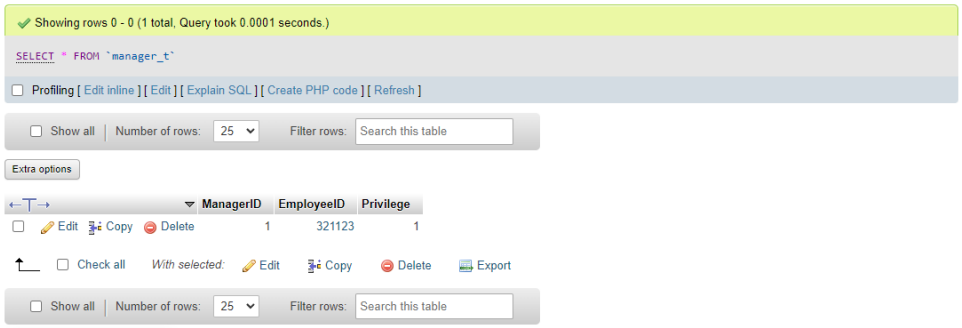
Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

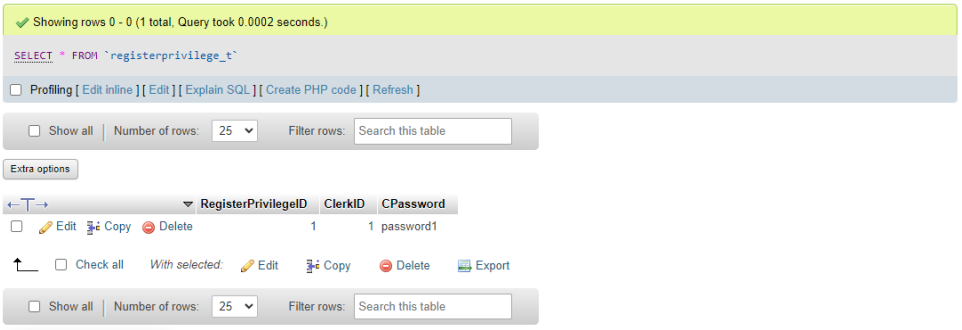
Clerk_t



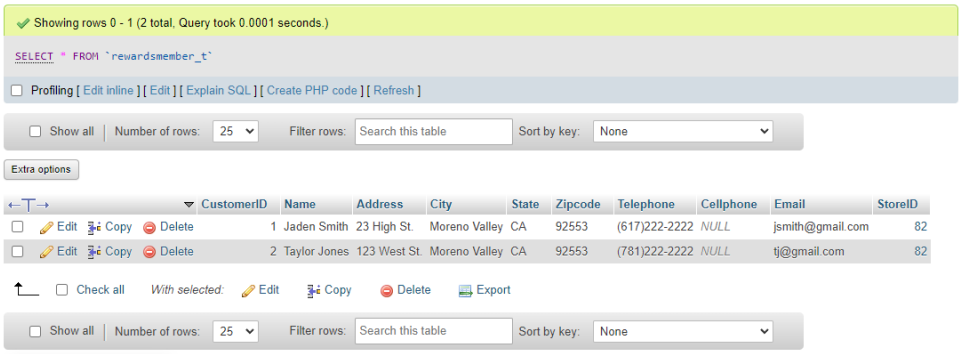
Manager_t



RegisterPrivilege_t



RewardsMember_t



2.7.Applying Normalization Rules

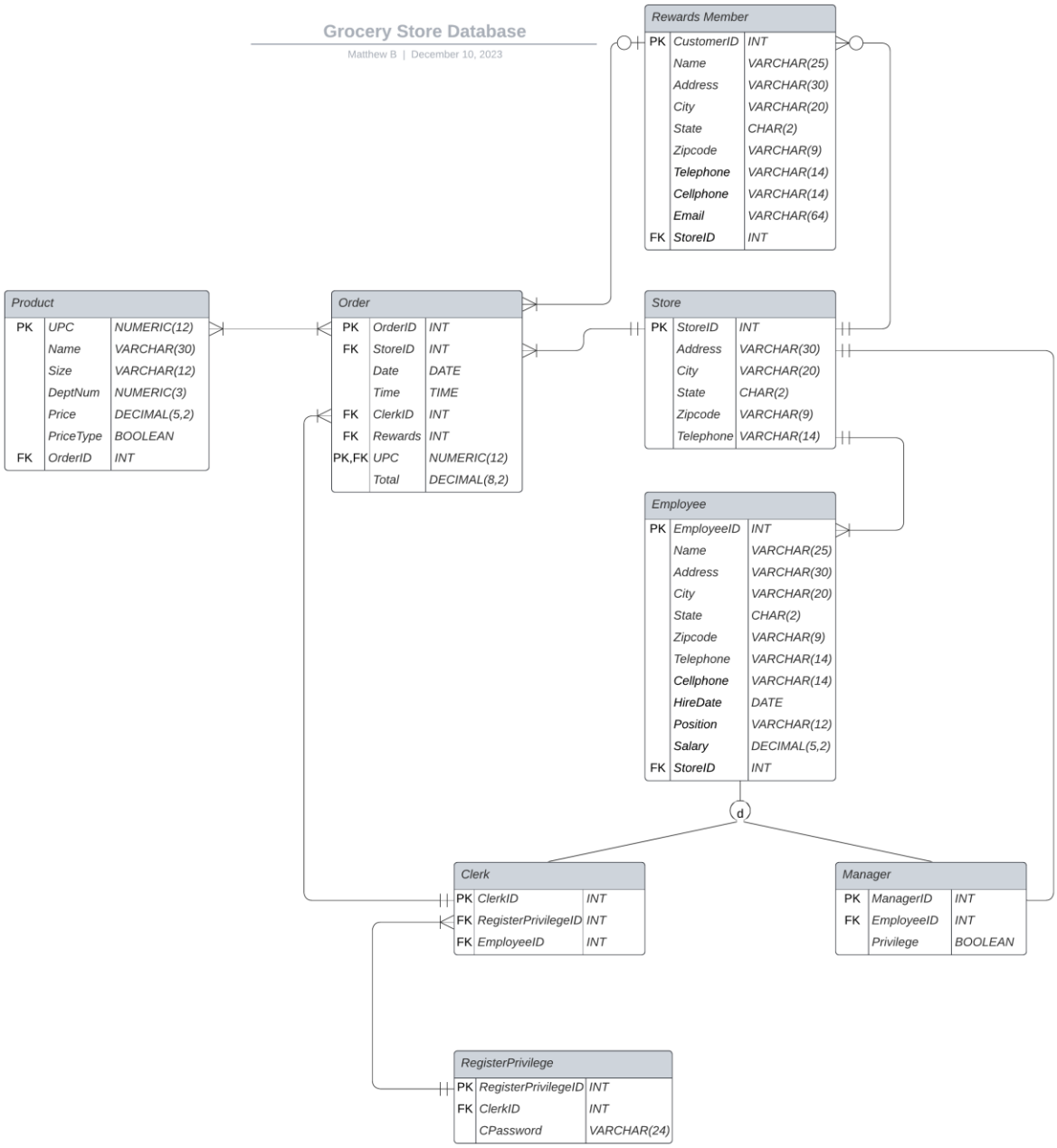
1NF: None of the tables use row order to convey information. Additionally, data types are strictly enforced among the various columns. Next, since all tables have primary keys, this maintains the rules of the first normal form. Lastly, multivalued attributes have been eliminated from each table.

2NF: Every non-key attribute in every table is fully functionally dependent on the entire primary key. This possible issue was addressed in the drafting phase, some new tables were made for proper dependencies among the data.

3NF: I could find no transitive dependencies among the non-key attributes.

4NF: I believe I addressed the fourth normal form by splitting up the clerk table with the registerprivilege table.

Therefore, the final draft of the ERD is as follows:


Grocery Store Database
Matthew B | December 10, 2023
Rewards Member
PK CustomerID INT
Name VARCHAR(25)
Address VARCHAR(30)
City VARCHAR(20)
State CHAR(2)
Zipcode VARCHAR(9)
Telephone VARCHAR(14)
Cellphone VARCHAR(14)
Email VARCHAR(64)
FK StoreID INT
Product
PK UPC NUMERIC(12)
Name VARCHAR(30)
Size VARCHAR(12)
DeptNum NUMERIC(3)
Price DECIMAL(5,2)
PriceType BOOLEAN
FK OrderID INT
Order
PK OrderID INT
FK StoreID INT
Date DATE
Time TIME
FK ClerkID INT
FK Rewards INT
PK,FK UPC NUMERIC(12)
Total DECIMAL(8,2)
Store
PK StoreID INT
Address VARCHAR(30)
City VARCHAR(20)
State CHAR(2)
Zipcode VARCHAR(9)
Telephone VARCHAR(14)
Employee
PK EmployeeID INT
Name VARCHAR(25)
Address VARCHAR(30)
City VARCHAR(20)
State CHAR(2)
Zipcode VARCHAR(9)
Telephone VARCHAR(14)
Cellphone VARCHAR(14)
HireDate DATE
Position VARCHAR(12)
Salary DECIMAL(5,2)
FK StoreID INT
d
Clerk
PK ClerkID INT
FK RegisterPrivilegeID INT
FK EmployeeID INT
Manager
PK ManagerID INT
FK EmployeeID INT
Privilege BOOLEAN
RegisterPrivilege
PK RegisterPrivilegeID INT
FK ClerkID INT
CPassword VARCHAR(24)