Adam Cunningham
Moses Maugans
Christopher Crabtree
Julien Galons
Project 4, csc460, Dr. McCann, Database design
Due 12/8/20 (one late day used)

Conceptual Database Design

Our database design followed the specifications in the outline provided by the TAs.

It follows the following restrictions:
When adding a member, the ID and phone number cannot be null
When updating, the user can update everything except the id
When deleting, the entire row is deleted

When adding a supply record, the supply ID cannot be null.
The date cannot be ahead of the current date.

Sales are initialized with a non-empty ID and a price of 0.
When a subsale with a Sale's ID is added, it updates the Sale's totalPrice

Query Description

Our custom query was "What is the lifetime savings of a specific member."
It is important for a business to maintain members (80% of profit comes from 20% of customers), So loyalty and continued business relationships are key. The lifetime savings is a convincing bit of data to demonstrate how being a member is benefiting that member. People generally enjoy seeing their savings at the bottom of a receipt.

Query execution (to explain how more than 1 table was used):
We used an employee lookup similar to the Member id/phone search. Then checked sales with that member, and used subsales of that sale to tally the total spent, and used product retail price information to calculate the difference between the member's spending and the total retail price had they not been a member.

Database Schema

Member (ID, firstName, lastName, DOB, address, phoneNumber, rewardPoints, active)

Employee (ID, firstName, lastName, gender, address, phoneNumber, groupType, salary)

Product (ID, name, retailPrice, category, membershipDiscount, quantity)

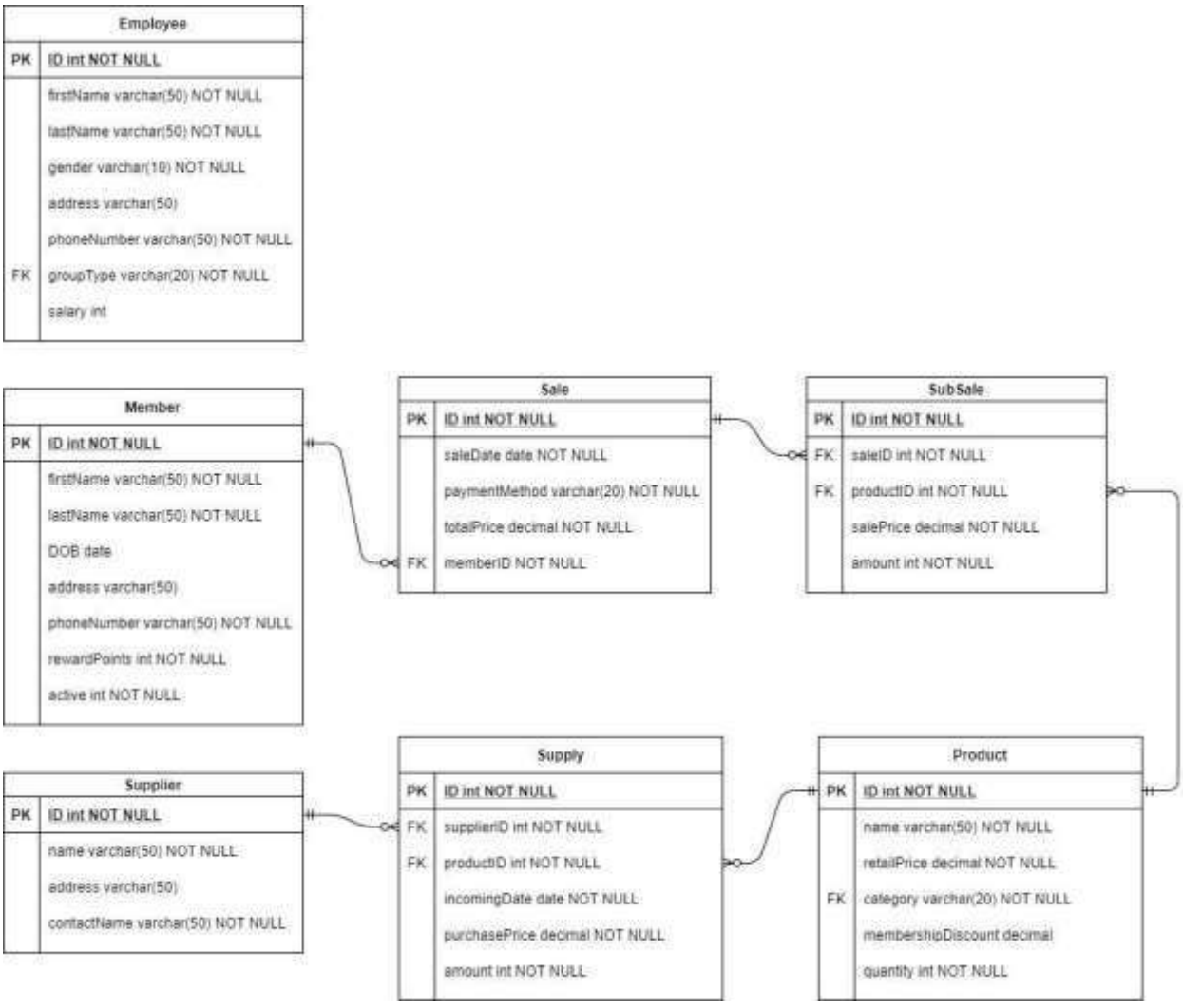
Supply (ID, supplierID, productID, incomingDate, purchasePrice, amount)

Supplier (ID, name, address, contactName)

Sale (ID, saleDate, paymentMethod, totalPrice, memberID)

SubSale (ID, productID, saleID, salePrice, amount)

Database Relations E-R Model



Relational Normalization Analysis

None of the relations in the database have list attributes. Each attribute value is a single value. This implies the database is in 1NF.

The following tables contain the following Function Dependencies

Member: {ID → firstName, ID → lastName, ID → DOB, ID → address, ID → phoneNumber, ID → rewardPoints, ID → active}

Employee: {ID → firstName, ID → lastName, ID → gender, ID → address, ID → phoneNumber, ID → groupType, ID → salary}

Product: {ID → name, ID → retailPrice, ID → category, ID → membershipDiscount, ID → quantity}

Supply: {ID → supplierID, ID → productID, ID → incomingDate, ID → purchasePrice, ID → amount}

Supplier: {ID → name, ID → address, ID → contactName}

Sale: {ID → saleDate, ID → paymentMethod, ID → totalPrice, ID → memberID}

SubSale: {ID → productID, ID → saleID, ID → salePrice, ID → amount}

For each Relation, the ID attribute functionally determines all other attributes. By definition, each ID attribute for each relation is a candidate key, the only candidate key of each relation, or the primary key of each relation. As such, the database meets the condition for 2NF. Notice there exists no functional dependencies among the relations of which a nonprimary attribute is the determinant, eliminating the possibility for transitive dependencies. Therefore, the database is in 3NF.