Montreal Paint Store

Group 2 : Ahmed Tlili, Leon Petrinos, Mathilde Peruzzo

February 3, 2025

1 Requirements Analysis

1.1 Introduction

The purpose of this application is to model a retail company that sells paint and painting tools both online and in physical stores. The application allows the company to keep track of its stock and sales.

1.2 Data requirements

- Each store has a unique address and a phone number that we keep track of.
- Additionally, every store has employees. We track employees with their unique employee id, name and address.
- Every employee works on-site at a single store at a time. They might however switch to another store during their careers at the company.
- Additionally, every store is managed by one of the employees. An employee manages at most one store.
- Stores have a stock of products and we keep track of the quantity of each product.
- Every product is defined by its unique product id and unit price.
- There are two types of products sold: Paint and Tools.
- For paint products, we keep track of the base and color. For tool products, we keep track of their type (type can be Brush, Roller, Tape, ...)
- Customers should be able to purchase products. There are two types of purchases: In-Store purchases and Online purchases. Each purchase has a unique purchase id and a date. Since online purchases can be made at different delivery addresses, that address is tracked as part of the online purchase.
- We also keep track of customers that have made online purchases, noting their name and email.

- Every online purchase can only be made by one customer.

- Additionally, online purchases are handled by the stores. Every online purchase is handled by at least one of the stores, but it could be many if no store has all the required products in stock.

- We want to keep track of what store In-Store purchases were made in. We also want to record which employee handled each In-Store purchase (i.e. was at the cash register or sent a bill to the client).

- Purchases (both online and in-store) contain at least one type of product. We note the quantity of each product sold in a purchase.

- The customer might pay for their products at the time of their purchase, but they might also request for a bill to be sent. As prices change and customers might negotiate prices, the final amount to pay might not correspond to the sum of product unit prices. For these reason, we keep track of payments separately from purchases. Each payment has a date, an amount and payment details.

1.3 Functional requirements

The system should support the following functionalities:

1. Employee Management:
   - Add new employees with their details (address and name)
   - Update employee information
   - Reassign employees to different stores
   - At every store there is only one employee that is promoted to a manager
2. Store Management:
   - Add new stores with their details (address, phone number, manager)
   - Update store information.
   - Track inventory levels for each product in a store. Restock products and update stock quantities in case of a purchase
3. Product Management:
   - Add new products (paint and tools) to the inventory
   - Update product details like prices, as it may change over time. Deleting a product is also an option as it may be discontinued
   - Query the current stock of a particular product in a store
4. Customer and Purchase Management:
   - Register new customers for online purchases

   - Retrieve purchases that have not been paid yet
   - Have a complete log of purchases made that being either in-store or online
   - Retrieve purchases that have not been paid yet
   - Query purchases within a specified time period or by a certain customer

5. Reporting:
   - Must be able to query stores with low stock on specific items, Gross Income generated by each store

2 Relational model

2.1 Entities

- Store(s_address, phone_number, eID UNIQUE NOT NULL)

  Foreign key eID references Employee (for the manages relation)

- Employee(eID, e_address, name, s_address NOT NULL)

  Foreign key s_address references Store

- Product(pID, unit_price)

- Paint(pID, base, color)

  Foreign key pID references Product

- Tool(pID, type)

  Foreign key pID references Product.

- Customer(email, name)

- Payment(paID,puID UNIQUE NON NULL, amount, details, pa_date)

  Foreign key puID references Purchase.

- Purchase(puID, pu_date)

- Online(puID, email NOT NULL, address)

  Foreign key puID references Purchase.

  Foreign key email references Customer

- InStore(puID, eID, s_address)

  Foreign key puID references Purchase.

  Foreign key eID references Employee.

  Foreign key s_address references Store.

2.2 Relationships

- Stock(s_address, pID, quantity)

  Foreign key s_address references Store.

  Foreign key pID references Product.

- Contains(puID, pID, quantity)

  Foreign key puID references Purchase.

  Foreign key pID references Product.

- Handles(saddress, puID)

  Foreign key saddress references Store.

  Foreign key puID references Online.

3 Other Conditions

Things to check at the application level

1. In the E/R model:
   - nothing
2. In the relational model:
   - When an in-store purchase is made, we need to check that the employee registering it works at the store
   - A store has to have at least one employee
   - An employee manages at most one store
   - An online purchase has to be handled by at least one store
   - A purchase cannot be done with no items

4 Inspirations

- https://www.maadco.ca/
- https://www.claessens.com/

5 Work and collaboration

We met twice to discuss possible topics for this project, before finally choosing the current one. In those meetings, we also decided on the basic requirements and on a rough idea for the ER diagram.

We then separated the work as follows : Mathilde drew the ER diagram. Leon and Ahmed wrote the requirements and translated the ER diagram to the relational model. We kept helping each other and communicating during this phase.