PHARM858C

George’s Pharmacy

Lovely Fernandez | Paris Le
C20305696 | C20401536

Table of Contents

ERD Diagram

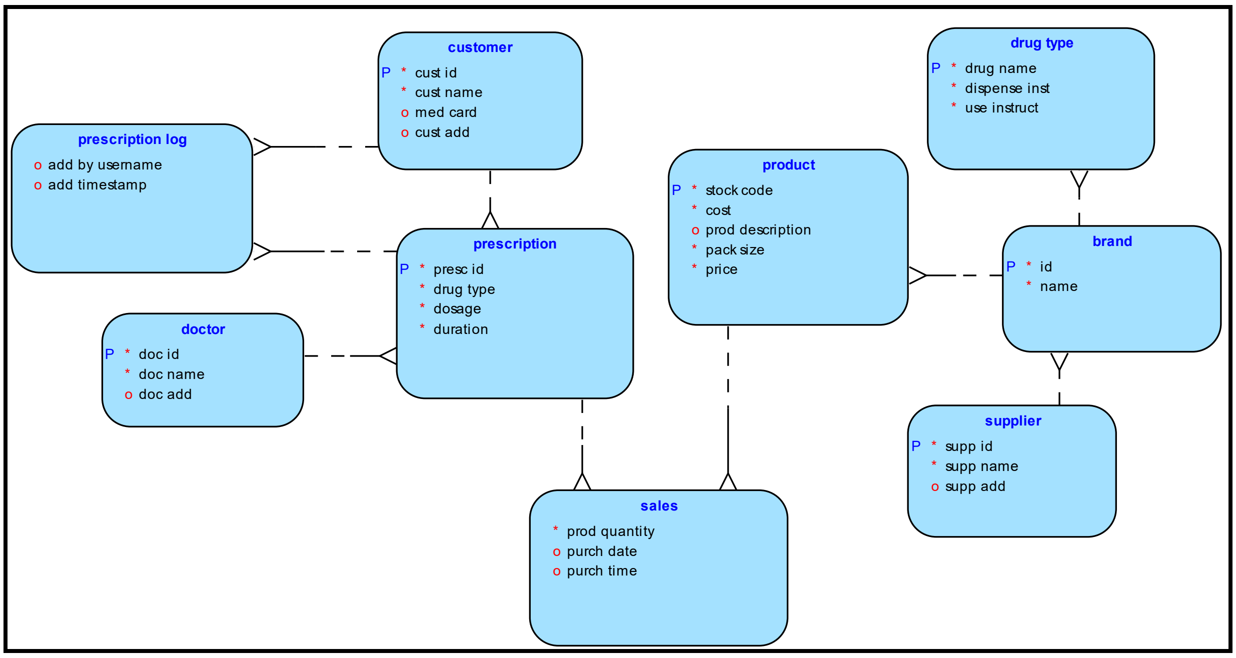

Figure 1: Logical Diagram of Pharm858C Database

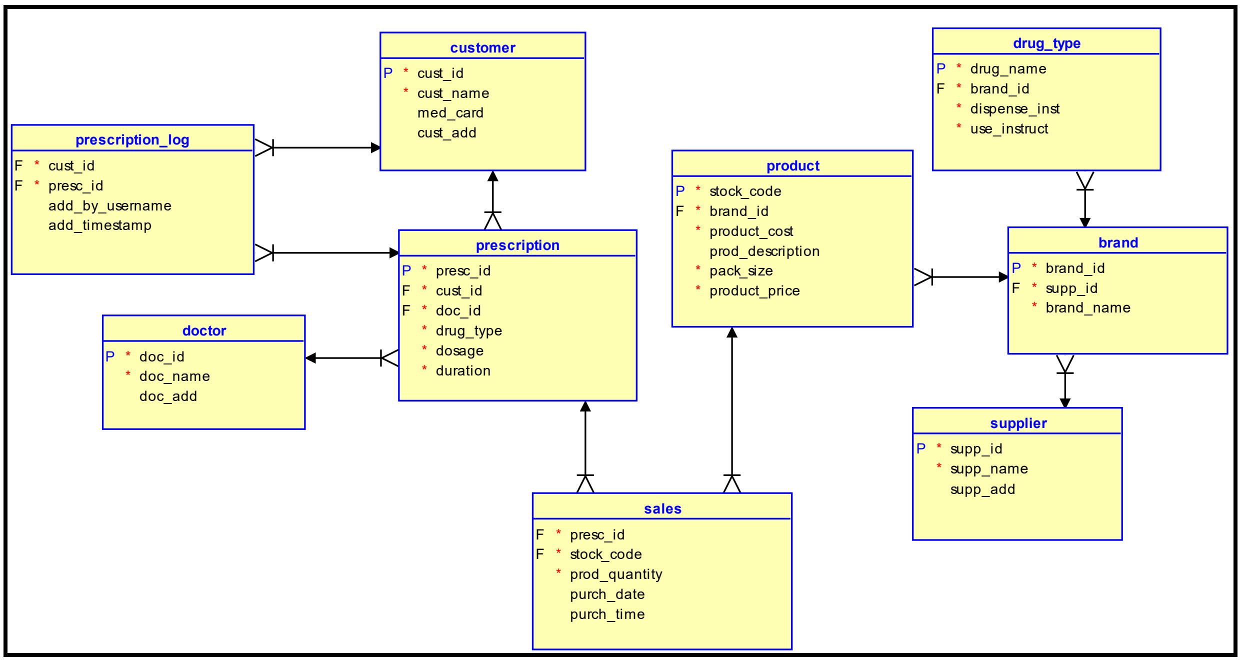

Figure 2: Relational Diagram of Pharm858C Database

Relationships between entities

Brand and Pack Sizes: This has a 1:N (one-to-many) relationship type. A brand can have multiple pack sizes, for example, a packet of 12 or 24 for Paracetamol.
Brand and Supplier: A supplier can supply many brands but each brand will only have one supplier. This relationship is also 1:N.

Customer and Prescription: This is a 1:N relationship where each customer can have different prescriptions but each prescription is only associated with one customer.
Prescription and Drug: This 1:N relationship expresses that each prescription can contain multiple drugs but each prescribed drug is connected to one prescription.
Drug and Product: This is a many-to-one relationship (N:1) where each drug prescribed follows a specific product of the Pharmacy's stock.
Sales and Prescription: This is a 1:1 relationship. Each sale is associated with one prescription. The relationship ensures that every sale is connected to a particular prescription.

Roles and Responsibility

Individual Role

Within Pharm858C's database, there are two roles – George and Kevin - divided between myself and my partner, Paris. I will be taking George's role who is responsible for processing the customer's prescription unto the system which includes; adding the prescription information unto the customer's file, creating a new file if it is a new customer, adding the prescribed drugs including the dosage and duration, and recording any transactions made unto the prescription log which saves the user who updated the system and date of when it was done.

Program Files

Here are the program files related to my role as George. I have written a procedure for adding a prescription as well as a log book which documents every transaction made using *addprescription()*.

1. Creating the DB for Pharm858C | DB_Pharm858C.sql
   I was responsible for creating the tables and Paris had populated these tables to create the database.

2. Creating the procedure for adding a prescription | addprescription.sql
   As per my role, I had created a procedure for adding a prescription into the system. This takes a prescription into the Prescription Book (table) and adds the customer's information including name, address, medical card ID, doctor information and the drugs they are prescribed with. Error checking is implemented here to avoid duplicate values of customers or drugs that have already been record on their profile.
   We can call this procedure via python using the *addprescription.py* file.

3. Adding appropriate functions | functions.sql
   This file contains 4 different functions that supports the add prescription procedure.

   1. customer_exists()
      This checks if the customer already exists and if not, to create a new index card to add the prescription to.
   2. Doctor_exists()
      This checks if the doctor exists in the system and if no match is made, system will alert that the doctor does not exist.
   3. Drug_exists_cust()
      This checks if the drug already exists under the customer profile, if so, it will not add another entry. This is implemented to avoid duplicated inputs of the same drug.
   4. Drug_exists_drugbook()
      If the prescription only contains the brand of the drug, the system will check the drug type associated with the brand and if it does not exist, it will raise an error.

4. Implementing a prescription log | triggerAddPresc.sql

A trigger is implemented unto the procedure for adding a prescription. It will save all transaction made, recording the name of the customer, the new prescription id, the user (staff) who added the prescription and the date the entry was made.

5. Permissions (GRANTs) | adding_grants_addprescription.sql

This contains the permissions to allow a user to access *addprescription()*. I have granted insert, select and update to a limited set of tables to allow user to access this procedure and update the information to the appropriate tables.