Online Pharmacy App

Josue Auguste

March 27, 2022

DATABASE DESIGN REVIEWED

Overview

After deep thinking, I realized that a relational database would be the number one choice for my application due to the fact that we will be dealing with highly normalized data (e.g. lots of relationships) and sometimes there will be many-to-many relationships. Also, since we are going to be changing the rows a lot because a lot of information is not permanent, using a relational database would provide easier access for cascading updates. PostgreSQL is the relational database that we finally opted for because PostgreSQL is very robust and is one of the most advanced open-source relational database systems. It is characterized by many features that help developers build applications, administrators to protect data integrity and help data analysts and data scientists to manage their data regardless of the size of the dataset.

All the data will be saved on a table format (rows and columns), the columns are known as the field of the table and the rows represent the record of specific item data. The rows hold different data within the same structure. The table is related to one another, the relation can be one-to-one, one-to-many, and many-to-many. Each table consists of a primary key constraint that uniquely identifies each record in a table. The information stored in the database will be accessible from the end-user through an API. Our preference goes toward the REST instead of the HTTP API.

DATA STRUCTURE

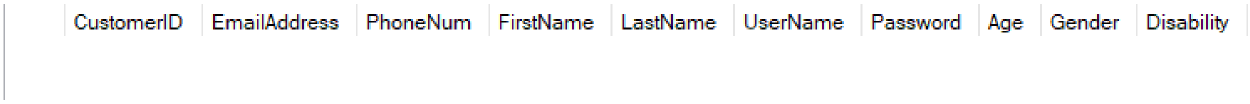
A database will be created with the following query then this database will be used to store the data.

```
CREATE DATABASE ONLINEPHARMACYAPP;
USE DATABASEONLINEPHARMACYAPP;
```

THE CUSTOMER TABLE

This table will the data for every single customer that would use this app to order medicine. The fields are clearly written below. The last four digits of the customer's social security will be stored and will serve as the primary key for the table.

```
CREATE TABLE Customer (
  CustomerID                      INT          PRIMARY KEY   IDENTITY,
  EmailAddress                VARCHAR(255)       DEFAULT NULL  UNIQUE,
  PhoneNum                    VARCHAR(255)       DEFAULT NULL,
  FirstName                   VARCHAR(225)       NOT NULL,
  LastName                    VARCHAR(225)       NOT NULL,
  UserName                    VARCHAR(225)       NOT NULL,
  Password                    VARCHAR(225)       NOT NULL,
  Age                         VARCHAR(225)       NOT NULL,
  Gender                      VARCHAR(225)       NOT NULL,
  Disability                    BOOLEAN          NOT NULL,
  );
```

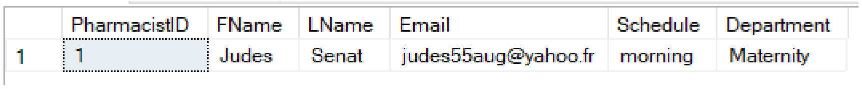


	CustomerID	EmailAddress	PhoneNum	FirstName	LastName	UserName	Password	Age	Gender	Disability

THE PHARMACIST TABLE

The pharmacist table will store all the information about the pharmacists and will enable to track any order by just knowing the pharmacy's name associated with a specific order. This table is in relation with the products and the customer tables.

```
CREATE TABLE Pharmacist (
  PharmacistID                    INT         PRIMARY KEY   IDENTITY,
  FName                         VARCHAR(10)      NOT NULL,
  LName                         VARCHAR(255)     NOT NULL,
  Email                         VARCHAR(200)     NOT NULL,
  Schedule                      VARCHAR(50)      NOT NULL,
  Department                    VARCHAR(50)      NOT NULL
  );
```

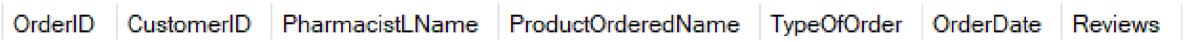


	PharmacistID	FName	LName	Email	Schedule	Department
1	1	Judes	Senat	judes55aug@yahoo.fr	morning	Maternity

The Order Table

The orders table contains all the completed orders that were made through the app. This table will get rid of any order older than three years. This table will be in relation with the customer table and the billing table as well.

```
CREATE TABLE Order (
  OrderID                         INT           PRIMARY KEY   IDENTITY,
  CustomerID                      INT           REFERENCES Customer (CustomerID),
  PharmacistLName               VARCHAR(10)     NOT NULL,
  ProductOrderedName            VARCHAR(10)     NOT NULL,
  TypeOfOrder                   TEXT            NOT NULL,
  OrderDate                     DATETIME        DEFAULT NULL,
  Reviews                        TEXT           DEFAULT NULL,
  );
```

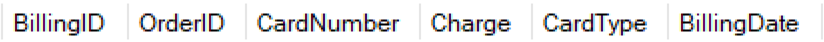


OrderID	CustomerID	PharmacistLName	ProductOrderedName	TypeOfOrder	OrderDate	Reviews

THE BILLING TABLE

Customers do not have to enter his billing info every time he/she is ordering medication. The app will ask for permission to save his/her payment method info. This table will be in relation with the CustomerOrderTrack table as it is shown in the references.

```
CREATE TABLE Billing (
  BillingID                      INT            PRIMARY KEY   IDENTITY,
  OrderID                        INT            REFERENCES Order (OrderID),
  CardNumber                     INT            NOT NULL,
  Charge                        MONEY           NOT NULL,
  CardType                     CHAR (20)        NOT NULL,
  BillingDate                  DATETIME         NOT NULL
  );
```

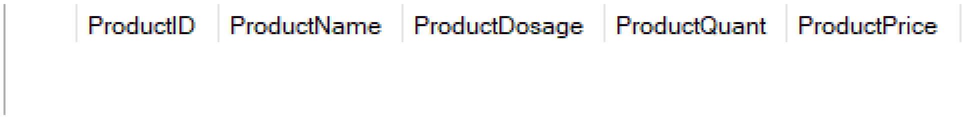


BillingID	OrderID	CardNumber	Charge	CardType	BillingDate

THE PRODUCT TABLE

This is considered to be the most important table in the database since it stores all the information about the products sold on the website. Customers will be able to search products through the (frontend) interface and the matching results should be shown on the screen. The rows of the table Products will constantly change in order to display the most recent information to the customers.

```
CREATE TABLE Product (
  ProductID                      INT           PRIMARY KEY   IDENTITY,
  ProductName                   CHAR (30)      NOT NULL,
  ProductDosage                 VARCHAR (50)   DEFAULT NULL,
  ProductQuant                   INT           NOT NULL,
  ProductPrice                  MONEY          NOT NULL,
  );
```

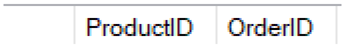


	ProductID	ProductName	ProductDosage	ProductQuant	ProductPrice

THE PRODUCTORDERED TABLE

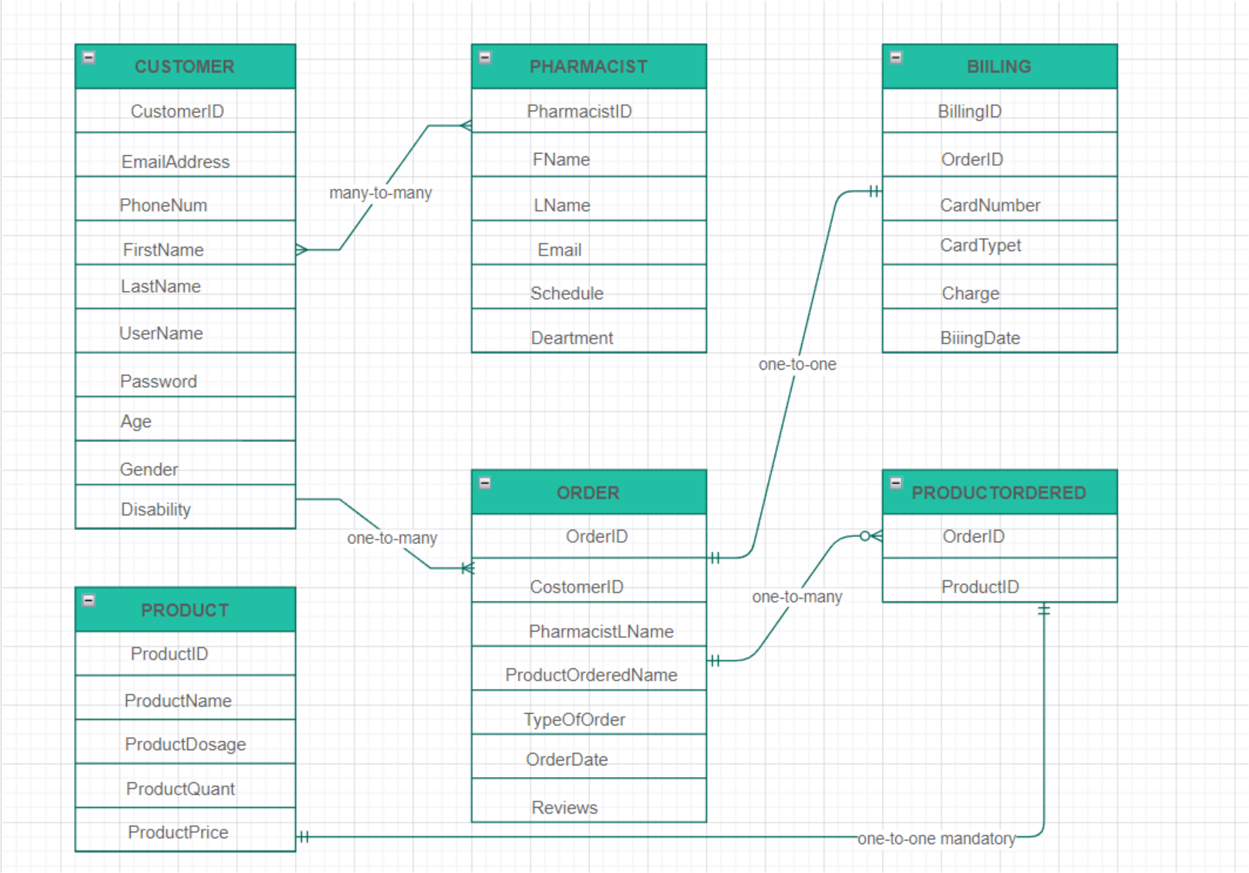
This table is a special one that gathered the primary keys from two different tables to for that unique table. It links the product and the order tables together. We needed this liaison between them because we notice that an order may have multiple products.

```
CREATE TABLE ProductOrdered (
  ProductID           INT       REFERENCES Product (ProductID),
  OrderID             INT       REFERENCES Order (OrderID) PRIMARY KEY (ProductID,
OrderID) IDENTITY);
```



	ProductID	OrderID

ENTITY RELATIONSHIP DIAGRAM



The image above is the entity-relationship diagram. It is shown the relation between many tables. There is a many-to-many- relationship between the customer and the pharmacist tables because a pharmacist can prepare orders for many customers while a customer may have a different pharmacist preparing his orders upon placing these orders. A one-to-many relationship exists between the customer and Order because one customer may place many orders. We notice that there is a one-to-one relationship between the billing table and the Order because each billing is associated with a specific order. The one-to-many relationship between the Order and the ProductOrdered tables can be explained by the fact that multiple products can be associated with one single order. Finally, it is important to point out the one-to-one relationship between the Product and the ProductOrdered tables which makes sense since each ordered item has to come from the product table.