MHD Dips

Database Design

Ed Davis
March 21, 2024

Overview

I will be using Postgres database to persist all the data related to my e-commerce application. I chose this database mainly for two reasons. First, there are several relationships between tables. For that reason, I believe that an SQL database would be the best fit. Secondly, I chose Postrgres in particular because it is the database that I am most familiar with. That was something I considered heavily due to the time constraints to deliver the product.

The web application will interact with the Postgres database through REST APIs. The frontend will send GET, POST, UPDATE, and DELETE requests to the backend/service layer that will be set up to handle those requests and send them to the database. Which will help my project stay in line with industry standards.

Database Specifications

Since I will be using an SQL database, I will be using unique IDs as primary keys for my tables to make it easier to retrieve data. I will also be using unique restraints, and required entries on certain columns to help with data integrity and security. Those columns will include things such as names. Below is a breakdown of each table. This will be followed by an illustration of the relationships between tables.

Products

<table-ID> PK
Image – String
Name – String
Quantity – INT
Price – Money
Featured – Boolean
Account-ID FK

OrderDetails

<table-ID> PK
Product_ ID – INT – FK
Quantity – INT
Total Price – Money
Account_ID – INT – FK
Shipper_ ID – INT _ FK

Reviews

<table-ID> PK
Rating – INT
Comment – String
Account_ID – FK

Shippers

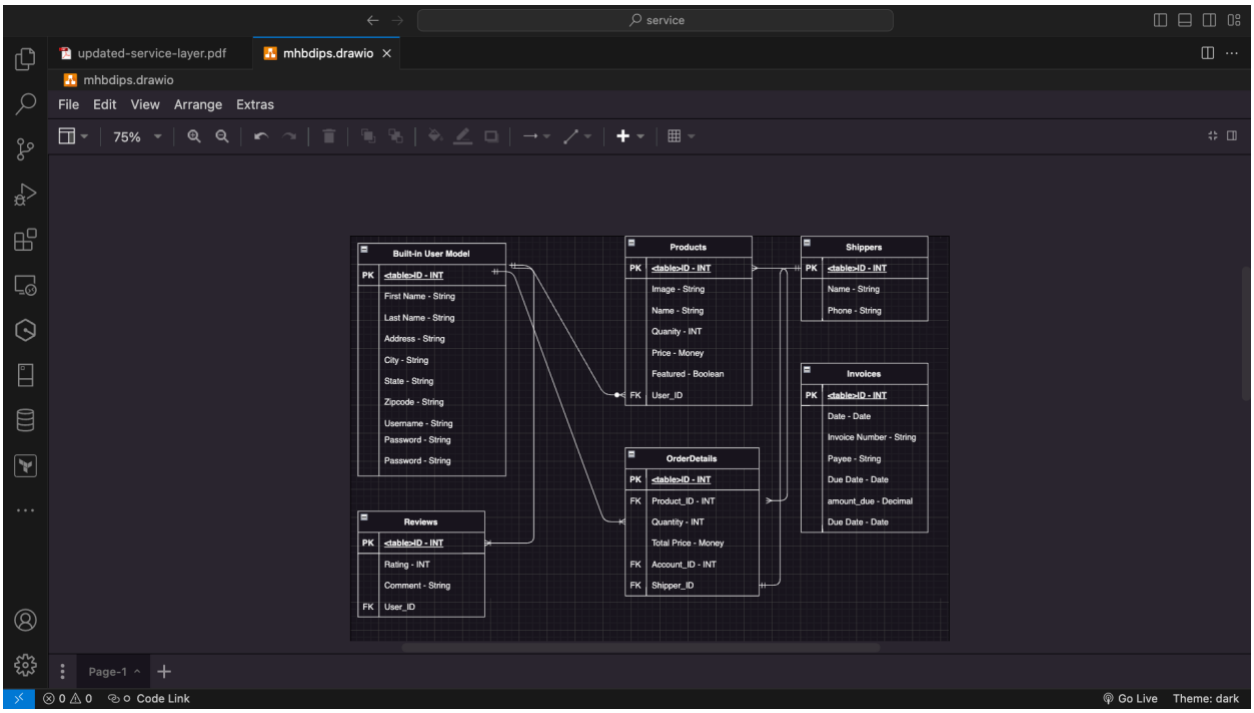
<table-ID> PK
Name – String
Phone – String

Invoices

<table-ID> PK
Date – Date

Invoice Number – String
Payee – String
Due Date – Date
Date Paid - Date

Relationship Diagram



Purpose, Implementations, and Interactions

Products

Purpose:

The product table in my application will be used to store information about the products available within the store. The owner will be able to perform CRUD operations on the products from the admin site.

Implementation:

The product table will be created in the models. Once created the admin site will be registered in the backend and will provide an interface for the owner to interact with the database.

Interaction:

The owner will be able to use the admin site to update product information, upload images of the products, and keep an eye on their quantities.

OrderDetails

Purpose:

The order details table in my application will be used to store information about the products purchased. This is the linking table. This table has several relationships with other tables within the database.

Implementation:

The backend will receive a POST request about the information that a customer's order as they begin to checkout. The backend will save this information and send it to the database. The order details table will also be given the product id, account id, and shipper id as foreign keys to help keep things organized.

Interaction:

Once customers are satisfied with the products that have been put in their cart, they will fill out the information on the shopping cart page and submit it to proceed to the next step. At that point, they will send the POST request to the backend.

Reviews

Purpose:

The reviews table in my application will be used to render comments that users of the application have made about products that they have purchased.

Implementation:

When a customer wants to leave a review, their account will be used as the foreign key for the reviews table. This will allow the customer only to have to provide a rating and comment without having to supply their name.

Interaction:

The reviews table in my application will be used to render comments that users of the application have made about products that they have purchased.

Shippers

Purpose:

The shippers' table within my application will be used to keep track of which shipper has been assigned an order to deliver. This table is meant for the owner. She will be the only person that has a view of this table.

Implementation:

The shippers id will be set as the foreign key on the order details table. This will allow the owner to keep track of which orders have been given to which shipping partner.

Interaction:

The owner will be able to use the admin site to attach a shipper to an order.

Invoices

Purpose:

The invoices table is another table that is only meant for the owner of the e-commerce store. This table is designed to help the owner keep track of invoices from shippers. Normal Users of the application will not have visibility of the table.

Implementation:

The invoices table will be created using models. The admin site will be registered which will provide an interface for the owner to interact with the database.

Interaction:

The owner will use the admin interface to enter and view information about her invoices.