**Database Schema**

**E-commerce-Admin-API**

The Inventory Management System is a Apis based application that helps businesses manage their product inventory efficiently. It provides features for adding and updating products, tracking sales, analyzing revenue, and monitoring inventory levels. Users can filter and search for products, view sales data by date range, and receive low stock alerts. The system also maintains a history of inventory changes for auditing purposes.

**Database Schema**

we are using mysql relational database .workbench is used to as a relational dbms tool. To run apis in vs code we are using thunderclient.it can be easily used by installing its extension.

**Product and Inventory:**

**Relationship Type:**

One-to-One

**Explanation:** Each product in the "products" table has a one-to-one relationship with an inventory record in the "inventory" table. This relationship allows you to associate each product with its current inventory status. For example, if you have a product called "Widget A," the "inventory" table will contain a record for "Widget A" that tracks its current quantity in stock and the last time it was updated.

**Product and Sale:**

**Relationship Type:**

One-to-Many

Explanation: Each product in the "products" table can have multiple sales records associated with it in the "sales" table. This one-to-many relationship allows you to record all the sales transactions for a particular product. For example, if you sell "Widget A" multiple times, each sale will be recorded in the "sales" table, and they will all be associated with "Widget A."

**Product and InventoryChangeLog:**

**Relationship Type:**

One-to-Many

Explanation: Each product in the "products" table can have multiple entries in the "inventory\_change\_logs" table. This one-to-many relationship is used to log changes in inventory for a specific product. Whenever the quantity of a product in stock changes, an entry is added to the "inventory\_change\_logs" table, indicating the change and when it occurred.

**Product and Category:**

**Relationship Type:**

Many-to-One

Explanation: The "products" table is related to the "categories" table in a many-to-one relationship. This means that multiple products can belong to the same category, but each product can only belong to one category.

**Detailed Schema:**

**Products**

* id
* name
* description
* price
* category\_id

**Sale**

* id
* product\_id
* quantity
* sale\_date

**Inventory**

* id
* Product\_id
* quantity
* last\_updated

**Category**

* id
* name

**InventoryChangeLog**

* id
* product\_id
* quantity\_change
* timestamp

**Additional Questions**

**If there are libraries essential for the project configuration, list them and explain the reason.**

Sqlalchemy is essential for interacting with database in python. It makes it easy to run different queries with minimal and easy syntax.In this project we are using sql alchemy to interact with mysql

**The timezone of the data and the user's timezone may be different from your located timezone. What should be considered when handling Datetime using Python and Database? (Explanation or code example)**

We can use pytz library to handle the difference in timezones of user and database administrator