Activity: Data Modeling Challenge Instructions

1. Identify the entities involved in the e-commerce platform, such as customers, products, categories, orders, and reviews.

- **Customers:** Users who register and make purchases.
- **Products:** Items for sale in the store.
- **Categories:** For organizing the products in group
- **Orders:** Transactions made by customers containing products.
- **Reviews:** Feedback from customers on products.

2. Determine the relationships between these entities and define the cardinality of each relationship (one-to-one, one-to-many, or many-to-many).
https://www.geeksforgeeks.org/cardinality-in-dbms/

Customers -> Products: many-to-many
Customers -> Orders: one-to-many
Customers -> Reviews: one-to-many

Products -> Categories: many-to-one
Products -> Orders: many-to-many
Products -> Reviews: one-to-many

3. Design a data model using either the embedded or reference approach based on the following requirements:
a. Customers can place multiple orders, and each order can contain multiple products.

b. Each product belongs to a specific category, and a category can have multiple products.

c. Customers can write reviews for products, and each product can have multiple reviews.

5. Decide on the key attributes for each entity and incorporate them into the data model.

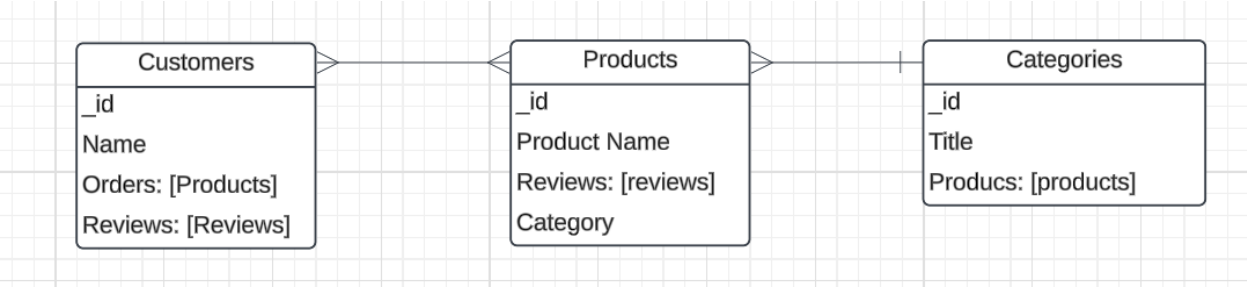
6. Consider the data access patterns and query requirements for the e-commerce platform. Ensure that the data model supports efficient and effective data retrieval.

7. Document your data model using an appropriate diagramming tool, such as a UML class diagram, to visually represent the entities, relationships, and attributes.

8. Write a brief explanation of your data model, highlighting the reasons for choosing the embedded or reference approach for each relationship.

I created a simple ERD model using Lucid Chart: https://lucid.app/lucidchart/5246404b-6f54-

40f7-a585-125f7c50b8ad/edit?viewport_loc=-94%2C-162%2C1707%2C743%2C0_0&invitationId=inv_583309f2-c813-4759-adb3-18c6d2aea70f



Explanation:

a) Customers -> Orders -> Products: **Referencing Approach** (I'll store the ObjectIDs of the products inside the Orders Key and then populate them to access the data for each document so I could use the other keys/attributes if needed.)

b) Products -> Category: **Referencing Approach** (I'll store the ObjectIDs of the products and then populate them to access the data for each document so I could use the other keys/attributes if needed.)

c) Customers -> Reviews: **Embedding Approach** (I'll embed the reviews along with the _id and name of the product since I don't need the other attributes)
Products -> Reviews: **Embedding Approach** (I'll embed the reviews along with the _id and the name of the owner since I don't need the other attributes)

Referencing Approach - if I might need all the attributes or keys of a document.
Embedding Approach – if I only need minimal attributes or keys of a document.