RETAIL DB

Database Design Decisions and Assumptions

Maheep Isher Singh Chawla
100909435

Table of Contents

Documentation of Database Design Decisions and Assumptions

Identification and Selection of Entities

According to the business needs, the main entities are as follows:

- **Customers**: The buyers who purchase goods from the store online.
- **Products**: The items for sale.
- **Orders**: Each purchase by the customer. A customer may have multiple orders, and an order may also have many products.
- **Inventory**: It maintains the stock of each product.
- **Reviews**: It stores reviews left by customers for the product that they bought along with its rating.

This structure was selected because it reflects common e-commerce operations and follows the relational model, ensuring that data is consistent and the relationships are normalized.

Attributes and Constraints

Each entity has been designed to include essential attributes to capture the necessary information without causing redundancy.

1. **Customers**:
   - customerID as the primary key for the unique identification of each customer.
   - name, emailid, and address to capture the basic details of customers.
   - Constraint on name and customerID not to be duplicated.

2. **Products**:
   - product_ID as the primary key to uniquely identify each product.
   - The product name, product category, and description were included to capture the information of the product. Price was included so that each product would have some cost associated with it.

3. **Orders**:
   - order_ID would be the primary key to uniquely identify the orders.
   - customerID as the foreign key to associate each order with a customer.
   - product_id as the foreign key to identify the product ordered; this allows every order to be associated with multiple products.
   - total_amount for recording the calculated total of each order, and order_quantity for the number of items ordered.

- Constraints on customerID and product_id ensure that orders only involve existing customers and products.

4. **Inventory**:
   - product_id as the primary key and foreign key, which ensures that each inventory entry will be about an existing product.
   - stock_quantity to maintain the number of each product in stock, which can also support operations related to stock management.

5. Reviews:
   - review_ID- a unique identifier for each review.
   - Product_id and customerID were added as foreign keys for the association of reviews with specific customers and products.
   - Rating and Review added for feedback details by adding constraints on Rating for valid selections, such as a 1-5 rating scale.
   - Ensures that every review will be traceable to only one unique combination of customer and product.

Relationship and ER Diagram Decisions

The relationships were drawn according to logical relationships present in an e-commerce environment:

One-to-Many Relationships:

- Between Customers and Orders: One customer can place multiple orders, but each order will be associated with one and only one customer.
- Between Products and Reviews: One product can have multiple reviews, yet each review will be about one product alone.

Many-to-Many Relationship:

- Between Orders and Products: Each order may be associated with multiple products, and each product can be part of multiple orders. To this end, the presence of product_id in the Orders table facilitated a simplified many-to-many relationship without the need to explicitly create a joining table. In the extended model, this relationship could be more accurately implemented with a separate Order_Items table.

Data Integrity and Normalization

The schema has been developed to satisfy Third Normal Form (3NF):

- No redundant information, like in Orders, has customers' information. The Info is referred from the foreign keys.
- Dependencies removed, example: product information is only stored in Products, not in Orders.
- Foreign key constraints are used for Referential integrity.
- Orders and Reviews point to existing customers and products.
- Inventory and Products participate in a one-to-one relationship using the foreign key product_id.

Assumptions Made

The following assumptions were used to make designing smooth:

- **Unique Identifiers**: Every entity should have a unique identifier. This is taken care of with the primary keys that are there for each entity, for example, customerID, product_ID, order_ID, and review_ID.
- **Product Categories**: Products have distinct categories they could fall into, such as "Electronics," "Clothing," etc., which is represented as a simple text attribute, product_category, in Products.
- **Review Ratings**: For simplicity, it has been assumed that the rating scale varies from 1 to 5.
- **Order Structure**: Orders represents one line in an order, not the order as a whole; this makes the data model simple but might be extended for a more complicated order handling-processing by adding, for instance, an Order_Items table.
- **Inventory Management**: We assume here that the stock is updated separately and no items ordered have been put on hold. Hence, inventory only shows the available stock at any given time.

Business Intelligence Views

SQL views are designed and can facilitate common business queries like the average rating of each product based on customer reviews.

- **Product_Average_Ratings** will return the average ranking for each product based on customer reviews. This could provide business insights into customer satisfaction with different products.
- **Customer_Total_Spending** returns the spending of every customer across all orders. As an example, this can be utilized for extracting high-value customers.

These views were developed to supply useful insights without complex querying into the operational database.