Ecommerce Databases

An E-Commerce database is used to store information about various entities in an E-Commerce website. We will look at a few E-Commerce databases below:

Here, we will look at an e-commerce database designed by Biswanath Giri. [1] According to him, an e-commerce website has the following requirements: [1]

- User Management
- Product Catalog
- Inventory Management
- Orders & transactions
- Shopping Cart
- Reviews & Ratings
- Payment Processing

The Entity-Relationship Diagram should consist of the following tables: [1]

- Users
- Products
- Categories
- Orders
- OrderItems
- CartItems
- Reviews
- Payments

The database needs to be normalized to 3NF. [1]
It has the following tables; [1]

- Users (**user_id**, username, password, email, firstname, lastname, created_at)
- Products (**product_id**, name, description, price, stock_quantity, **category_id**, created_at)
- Categories (**category_id**, category_name)
- Orders (**order_id**, **user_id**, total_amount, status, created_at)
- OrderItems (**order_item_id**, **order_id**, **product_id**, quantity, price)
- CartItems (**cart_item_id**, **user_id**, **product_id**, quantity)
- Reviews (**review_id**, **user_id**, **product_id**, rating, comment, created_at)
- Payments (**payment_id**, **order_id**, payment_method, amount, status, created_at)

We will look at an E-Commerce Database design made by Fabric Inc. A good ecommerce database has the following features: It should be **simple** and **functional**, should have **high performance** and should have **high availability & scalability**. [2]

- **Simple and Functional**: The database table structure is simple but covers all the required functionality without compromising the user experience. [2]

- **High Performance:** Database queries execute quickly to facilitate live customer interactions and support a frictionless shopping experience. Therefore, the selected database should have good indexing and performance optimization options. [2]

- **High Availability & Scalability**: A good database design is highly available with automatic snapshots and enables automatic scaling to support future platform growth as well as sudden traffic spikes. [2]

An ecommerce site should have the following key parts:

- **Database Scope**: The DB structure depends on the functionality of the ecommerce site. [2]
- **Database Type**: It can be a relational database or a NoSQL database or be a hybrid approach. [2]
- **Database Infrastructure**: It can be unmanaged - hosted & managed by us or it can be **unmanaged** (Eg - Amazon RDS / DynamoDB). [2]

The main consideration for designing a database depends on 2 parts: **Core Functions** and **Additional Functions**. The **Core Functions** of a database are for doing the day-to-day operations of a database. Eg - user management, product and inventory management, shopping cart function, payment management, and shipping/logistics management. [2]

The **Additional Functions** are those that are NICE TO HAVE. Eg - marketing functions, help desk and support, advanced analytics, and third-party integrations. The **Core Functions** of the e-commerce platform are covered by the following tables in the database: [2]

ProductCategory (**id**, name, description, created_at, modified_at, *deleted_at*)
ProductInventory (**id**, quantity, created_at, modified_at, deleted_at)
Discount (**id**, name, description, discount_percent, active, created_at, modified_at, deleted_at)
Product (**id**, name, description, sku, category_id, **inventory_id**, price, **discount_id**, created_at, modified_at, *deleted_at*)
OrderDetails (**id**, **user_id**, total, **payment_id**, created_at, modified_at)
OrderItems (**id**, **order_id**, **product_id**, quantity, created_at, modified_at)
PaymentDetails (**id**, **order_id**, amount, provider, status, created_at, modified_at)
ShoppingSession (**id**, **user_id**, total, created_at, modified_at)
CartItem (**id**, **session_id**, **product_id**, quantity, created_at, modified_at)
User (**id**, username, password, first_name, last_name, telephone, created_at, modified_at)
UserAddress (**id**, **user_id**, address_line1, address_line2, city, postal_code, country, telephone, mobile)
UserPayment (**id**, **user_id**, payment_type, provider, account_no, expiry)

User Management is done by User, UserPayment and UserAddress tables.
Product Management is done by Product, ProductCategory, ProductInventory and Discount tables. [2]
The Shopping Process involves the user adding items to cart, checking them out and then paying for them using a payment provider. The ShoppingSession and CartItem tables are temporary in nature. They only store the shopping session information of the current user till the order is confirmed and the data is moved to permanent storage tables. [2]

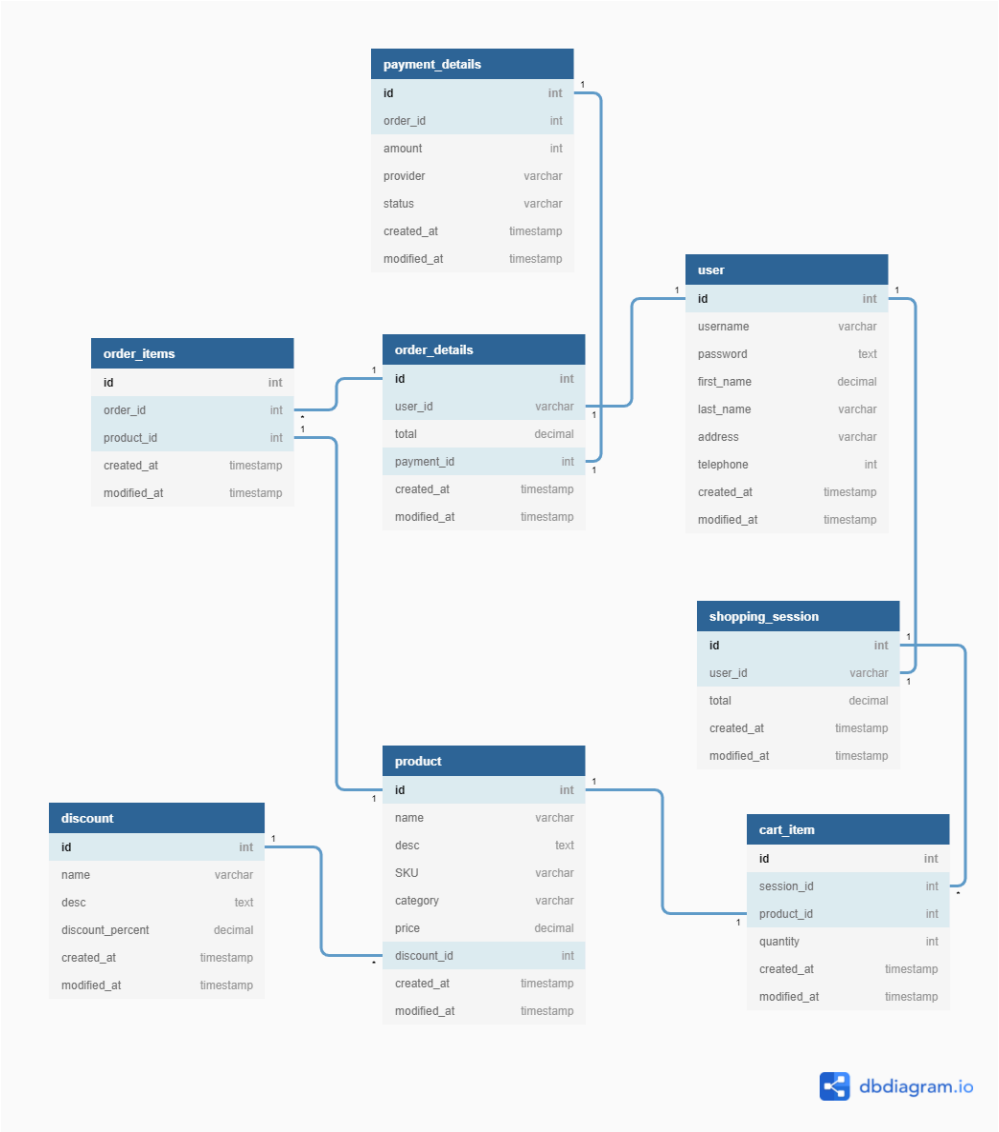
Designing A Shopping Cart Database
A Shopping Cart Database should be highly available, fault tolerant and highly responsive to provide customers with a smooth shopping experience. The process of designing a shopping cart database can be divided into 3 main components: [3]

- **Static Data**
- **Session Data**
- **Processed Data**

The **Static Data** component includes the info that the customer will need to retrieve while working with the Shopping Cart Database. This data is stored in the following type of tables: [3]

- **Product Table**
- **Discount Table**
- **User Table**

The **Session Data** includes all the interactions that the customer has with the shopping cart. It is the most important part of the shopping cart database. This is the **ShoppingSession** table and the **CartItem** table. [3]



The Static Data

The **Product**, **Inventory** and **Pricing** tables will get SELECT queries only when the user adds items to the cart. The **Product** table is updated when a purchase is completed. The **Product** and **Inventory** tables should be updated by the administrators. The **Users** table is updated either when a new user is created or when a user wishes to change his details. [3]

The Session Data

The tables handling session data are very active. The **ShoppingCart** and **CartItem** tables are temporary storage for the interactions between the customer and the shopping cart. When a customer visits Ecommerce Platform, then a new session is created in the **ShoppingSession** table and each item is added to the **CartItems** table linked to the specific session. Thus, the shopping cart's state is captured regardless of the customer interactions in the e-commerce platform. [3]

Processed Data

This data contains the completed order details and the corresponding payment details. When a transaction is completed, then the relevant data set is moved from the **ShoppingSession** table to the **OrderDetails** table and the **CartItems** table to the **OrderItems** table. The respective data is then deleted from the **ShoppingSession** and the **OrderDetails** tables. [3]

REFERENCES

[1] Designing the database schema for a new e-commerce platform and considering factors like normalization, denormalization, indexing, and performance optimization to ensure efficient storage and retrieval of data. | by Biswanath Giri | Medium

[2] What's an Example of Good E-Commerce Database Design?

[3] How Do You Design a Shopping Cart Database for E-Commerce?