### **Ecomm Web App Clone : : To-Do List for Schema Design Implementation**

#### **1. MongoDB Setup (Products)**

1. Create a products collection with the following schema:
   * name, category, price, attributes, stock, createdAt, updatedAt.
2. Populate the products collection with sample data representing various categories.

We have to create indexes on fields like category, price, and name for efficient querying

#### **2. PostgreSQL Setup (Users, Cart, Orders, etc.)**

1. Create a database named ecommerce-amazon-clone.
2. Create the following tables with relationships:
3. **Users Table:**sql  
   CREATE TABLE Users (

user\_id SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

email VARCHAR(150) UNIQUE NOT NULL,

password\_hash VARCHAR(255) NOT NULL,

address JSONB,

phone VARCHAR(15),

role ENUM('USER', 'ADMIN') DEFAULT 'USER',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

1. **Cart Table:**sql  
   CREATE TABLE Cart (

cart\_id SERIAL PRIMARY KEY,

user\_id INT REFERENCES Users(user\_id),

product\_id VARCHAR(24),

category VARCHAR(100),

quantity INT NOT NULL,

added\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

1. **Orders Table:**sql  
   CREATE TABLE Orders (

order\_id SERIAL PRIMARY KEY,

user\_id INT REFERENCES Users(user\_id),

total\_amount DECIMAL(10, 2) NOT NULL,

status ENUM('PENDING', 'COMPLETED', 'CANCELLED') DEFAULT

'PENDING',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

1. **OrderDetails Table:**sql  
   CREATE TABLE OrderDetails (

order\_detail\_id SERIAL PRIMARY KEY,

order\_id INT REFERENCES Orders(order\_id),

product\_id VARCHAR(24),

category VARCHAR(100),

quantity INT NOT NULL,

price DECIMAL(10, 2) NOT NULL

);

1. **Payments Table:**sql  
   CREATE TABLE Payments (

payment\_id SERIAL PRIMARY KEY,

order\_id INT REFERENCES Orders(order\_id),

payment\_method VARCHAR(50) NOT NULL,

payment\_status ENUM('SUCCESS', 'FAILED', 'PENDING') DEFAULT

'PENDING',

transaction\_id VARCHAR(100),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

Indexing in SQL ( For improved query search speed ) :

We should add indexes for frequently queried columns:  
sql  
CREATE INDEX idx\_users\_email ON Users (email);

CREATE INDEX idx\_cart\_user\_id ON Cart (user\_id);

CREATE INDEX idx\_orders\_user\_id ON Orders (user\_id);

Detailed schema design with relationships between tables :   
  
**1. Users Table**

#### **Schema Design**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| user\_id | SERIAL | PRIMARY KEY | Unique ID for the user. |
| name | VARCHAR(100) | NOT NULL | Name of the user. |
| email | VARCHAR(150) | UNIQUE, NOT NULL | Email address. |
| password\_hash | VARCHAR(255) | NOT NULL | Encrypted password. |
| address | JSONB | NULLABLE | Address stored in JSON format. |
| phone | VARCHAR(15) | NULLABLE | Phone number of the user. |
| role | ENUM | Default: 'USER', NOT NULL | Role of the user: 'USER', 'ADMIN'. |
| created\_at | TIMESTAMP | Default: CURRENT\_TIMESTAMP | Account creation time. |

### **2. Cart Table**

#### **Schema Design**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| cart\_id | SERIAL | PRIMARY KEY | Unique ID for the cart. |
| user\_id | INT | FOREIGN KEY REFERENCES Users(user\_id) | User to whom the cart belongs. |
| product\_id | VARCHAR(24) | NOT NULL | MongoDB ObjectId for the product. |
| category | VARCHAR(100) | NOT NULL | Product category. |
| quantity | INT | NOT NULL | Quantity of the product in cart. |
| added\_at | TIMESTAMP | Default: CURRENT\_TIMESTAMP | Time the product was added. |

#### **Cardinality Relationship**

* **Users (1) → Cart (Many):**One user can have many items in their cart, but each cart item belongs to one user.

### **3. Orders Table**

#### **Schema Design**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| order\_id | SERIAL | PRIMARY KEY | Unique ID for the order. |
| user\_id | INT | FOREIGN KEY REFERENCES Users(user\_id) | User who placed the order. |
| total\_amount | DECIMAL(10, 2) | NOT NULL | Total price of the order. |
| status | ENUM | Default: 'PENDING', NOT NULL | Status: 'PENDING', 'COMPLETED', 'CANCELLED'. |
| created\_at | TIMESTAMP | Default: CURRENT\_TIMESTAMP | Order creation time. |

#### **Cardinality Relationship**

* **Users (1) → Orders (Many):**One user can place multiple orders, but each order belongs to one user.

### **4. OrderDetails Table**

#### **Schema Design**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| order\_detail\_id | SERIAL | PRIMARY KEY | Unique ID for the order detail. |
| order\_id | INT | FOREIGN KEY REFERENCES Orders(order\_id) | Associated order. |
| product\_id | VARCHAR(24) | NOT NULL | MongoDB ObjectId for the product. |
| category | VARCHAR(100) | NOT NULL | Product category. |
| quantity | INT | NOT NULL | Quantity of the product ordered. |
| price | DECIMAL(10, 2) | NOT NULL | Price per product. |

#### **Cardinality Relationship**

* **Orders (1) → OrderDetails (Many):**One order can have multiple details, but each detail belongs to one order.

### **5. Payments Table**

#### **Schema Design**

| **Column** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| payment\_id | SERIAL | PRIMARY KEY | Unique ID for the payment. |
| order\_id | INT | FOREIGN KEY REFERENCES Orders(order\_id) | Associated order. |
| payment\_method | VARCHAR(50) | NOT NULL | Payment method (e.g., card, UPI). |
| payment\_status | ENUM | Default: 'PENDING', NOT NULL | Status: 'SUCCESS', 'FAILED', 'PENDING'. |
| transaction\_id | VARCHAR(100) | NULLABLE | ID of the transaction. |
| created\_at | TIMESTAMP | Default: CURRENT\_TIMESTAMP | Payment time. |

#### **Cardinality Relationship**

* **Orders (1) → Payments (1):**Each order can have only one payment associated with it.

### **6. Products Collection (MongoDB)**

#### **Schema Design**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| \_id | ObjectId | Unique ID for the product. |
| name | String | Name of the product. |
| category | String | Product category. |
| price | Number | Price of the product. |
| attributes | Object | Additional attributes (size, color, etc.). |
| stock | Number | Available stock. |
| createdAt | Date | Time of creation. |
| updatedAt | Date | Last updated time. |

#### **Cardinality Relationship**

* **Products (Independent):**Products are loosely coupled with other entities (referenced via product\_id).