DiasporaEat Ecommerce Database Design

Design By:
Victor Oluwamakin

Database Technology Choice: PostgreSQL (Relational Database)

Justification
1. Complex Relationships: The application has multiple interconnected entities (Users, Products, Orders, Cart) with many-to-many relationships (especially in order items).
2. Data Integrity: Critical for e-commerce transactions and inventory management.
3. ACID Compliance: Essential for financial transactions and order processing.
4. Normalized Data: Products, categories, and user information need to be normalized to avoid redundancy.
5. Frequent Updates: Regular updates to product inventory, order status, and cart contents.

Table Explanations and Usage Scenarios

1. **Users Table:**
Purpose: Stores user account information and authentication details

```
CREATE TABLE users (
   id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
   email VARCHAR(255) UNIQUE NOT NULL,
   password_hash VARCHAR(255) NOT NULL,
   first_name VARCHAR(100) NOT NULL,
   last_name VARCHAR(100) NOT NULL,
   phone_number VARCHAR(20) NOT NULL,
   created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
   updated_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);
```

Usage Scenario: When a user registers or logs in, their credentials are verified against this table.

2. **Addresses Table**
Purpose: Manages multiple shipping addresses for users

```
CREATE TABLE addresses (
   id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
   user_id UUID REFERENCES users(id),
   street VARCHAR(255) NOT NULL,
   city VARCHAR(100) NOT NULL,
   state VARCHAR(100) NOT NULL,
```

```
    country VARCHAR(100) NOT NULL,
    postal_code VARCHAR(20) NOT NULL,
    is_default BOOLEAN DEFAULT false,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);
```

Usage Scenario: When users add shipping addresses or select delivery locations during checkout.

3. Products Table
Purpose: Stores product information including inventory

```
CREATE TABLE products (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    category_id UUID REFERENCES categories(id),
    name VARCHAR(255) NOT NULL,
    description TEXT,
    price DECIMAL(10,2) NOT NULL,
    stock_quantity INTEGER NOT NULL,
    origin VARCHAR(100) NOT NULL,
    storage_instructions TEXT,
    nutritional_info JSONB,
    is_active BOOLEAN DEFAULT true,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);
```

Usage Scenario: Displayed on product listing and detail pages, used for inventory management.

4. **Categories Table**
Purpose: Organizes products into logical groups

```
CREATE TABLE categories (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    name VARCHAR(100) NOT NULL,
    description TEXT,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);
```

Usage Scenario: Used for product filtering and navigation.

5. Cart and CartItems Tables
Purpose: Manages shopping cart functionality

```
CREATE TABLE cart (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    user_id UUID REFERENCES users(id),
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE cart_items (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    cart_id UUID REFERENCES cart(id),
    product_id UUID REFERENCES products(id),
    quantity INTEGER NOT NULL,
    price_at_time DECIMAL(10,2) NOT NULL,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);
```

Usage Scenario: Tracks items added to cart before checkout.

6. **Orders and OrderItems Tables**
Purpose: Records customer orders and their items

```
CREATE TABLE orders (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    user_id UUID REFERENCES users(id),
    status VARCHAR(50) NOT NULL,
    total_amount DECIMAL(10,2) NOT NULL,
    payment_status VARCHAR(50) NOT NULL,
    shipping_address JSONB NOT NULL,
    estimated_delivery TIMESTAMP WITH TIME ZONE,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE order_items (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    order_id UUID REFERENCES orders(id),
    product_id UUID REFERENCES products(id),
    quantity INTEGER NOT NULL,
    price_at_time DECIMAL(10,2) NOT NULL,
```

```
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);
```

Usage Scenario: Created when orders are placed, used for order history and tracking.

7. Wishlists and WishlistItems Table

Purpose: Allows users to save products they are interested in.

```
CREATE TABLE wishlists (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    user_id UUID REFERENCES users(id),
    name VARCHAR(100) NOT NULL,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE wishlist_items (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    wishlist_id UUID REFERENCES wishlists(id),
    product_id UUID REFERENCES products(id),
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);
```

Usage Scenario (Stretch Feature): Users can create wishlists to save products they are interested in. This allows them to easily find and purchase these products later.

Database Indexes

```
-- Performance optimization indexes
CREATE INDEX idx_products_category ON products(category_id);
CREATE INDEX idx_cart_items_cart ON cart_items(cart_id);
CREATE INDEX idx_order_items_order ON order_items(order_id);
CREATE INDEX idx_orders_user ON orders(user_id);
CREATE INDEX idx_products_name ON products(name);
CREATE INDEX idx_wishlist_items_wishlist ON wishlist_items(wishlist_id);
```

Data Access Patterns

1. **Product Browsing:**

```
SELECT p.*, c.name as category_name
FROM products p
JOIN categories c ON p.category_id = c.id
WHERE p.is_active = true
ORDER BY p.created_at DESC;
```

2.**Cart Management:**

```
SELECT p.*, ci.quantity, ci.price_at_time
FROM cart_items ci
JOIN products p ON ci.product_id = p.id
WHERE ci.cart_id = :cart_id;
```

3. **Order History:**

```
SELECT o.*, oi.quantity, p.name
FROM orders o
JOIN order_items oi ON o.id = oi.order_id
JOIN products p ON oi.product_id = p.id
WHERE o.user_id = :user_id;
```

4. **Wishlist Management**

```
SELECT p.*
FROM wishlist_items wi
JOIN products p ON wi.product_id = p.id
WHERE wi.wishlist_id = :wishlist_id;
```

Changes from Previous Version

1. Change the database design from MongoDb to postgres
2. Added a "Wishlists" and "WishlistItems" table to support a stretch feature that allows users to save products they are interested in.
3. Updated the Entity-Relationship Diagram to include the new Wishlists and WishlistItems tables.
4. Added explanations for the Wishlists and WishlistItems tables, including their purpose and usage scenarios.
5. Added a new data access pattern for managing wishlists.
6. Updated the database indexes to include an index on the wishlist_items table.

Entity-Relationship Diagram

```
erDiagram

    Users ||--o{ Orders : places

    Users ||--o{ Addresses : has

    Users ||--o{ Cart : owns

    Orders ||--|{ OrderItems : contains

    Products ||--o{ OrderItems : "ordered in"

    Products ||--o{ CartItems : "added to"

    Cart ||--|{ CartItems : contains

    Products ||--o{ ProductImages : has

    Products }|--|| Categories : "belongs to"

    Users ||--o{ Wishlists : has

    Wishlists ||--|{ WishlistItems : contains


    Users {

        UUID id PK

        string email UK
```

```
    string password_hash
    string first_name
    string last_name
    string phone_number
    timestamp created_at
    timestamp updated_at
}

Addresses {
    UUID id PK
    UUID user_id FK
    string street
    string city
    string state
    string country
    string postal_code
    boolean is_default
    timestamp created_at
}

Products {
    UUID id PK
    UUID category_id FK
    string name
```

```
    text description
    decimal price
    int stock_quantity
    string origin
    text storage_instructions
    jsonb nutritional_info
    boolean is_active
    timestamp created_at
    timestamp updated_at
}

Categories {
    UUID id PK
    string name
    string description
    timestamp created_at
}

ProductImages {
    UUID id PK
    UUID product_id FK
    string url
    int display_order
    timestamp created_at
```

```
}

Cart {
    UUID id PK
    UUID user_id FK
    timestamp created_at
    timestamp updated_at
}

CartItems {
    UUID id PK
    UUID cart_id FK
    UUID product_id FK
    int quantity
    decimal price_at_time
    timestamp created_at
    timestamp updated_at
}

Orders {
    UUID id PK
    UUID user_id FK
    string status
    decimal total_amount
```

```
    string payment_status
    jsonb shipping_address
    timestamp estimated_delivery
    timestamp created_at
    timestamp updated_at
}

OrderItems {
    UUID id PK
    UUID order_id FK
    UUID product_id FK
    int quantity
    decimal price_at_time
    timestamp created_at
}

Wishlists {
    UUID id PK
    UUID user_id FK
    string name
    timestamp created_at
    timestamp updated_at
}
```

```
WishlistItems {
    UUID id PK
    UUID wishlist_id FK
    UUID product_id FK
    timestamp created_at
    timestamp updated_at
}
```

Entity-Relationship Diagram