

**NMCNPM - Quản lý nhà hàng - Group 14**

# TÀI LIỆU THIẾT KẾ



Bộ môn Công nghệ phần mềm  
Khoa Công nghệ thông tin  
Đại học Khoa học tự nhiên TP HCM

# MỤC LỤC

<b>1</b>	<b>Bảng đánh giá thành viên .....</b>	<b>2</b>
<b>2</b>	<b>Mô hình quan niệm .....</b>	<b>3</b>
<b>3</b>	<b>Thiết kế kiến trúc .....</b>	<b>4</b>
<b>4</b>	<b>Thiết kế dữ liệu.....</b>	<b>7</b>

# TÀI LIỆU THIẾT KẾ

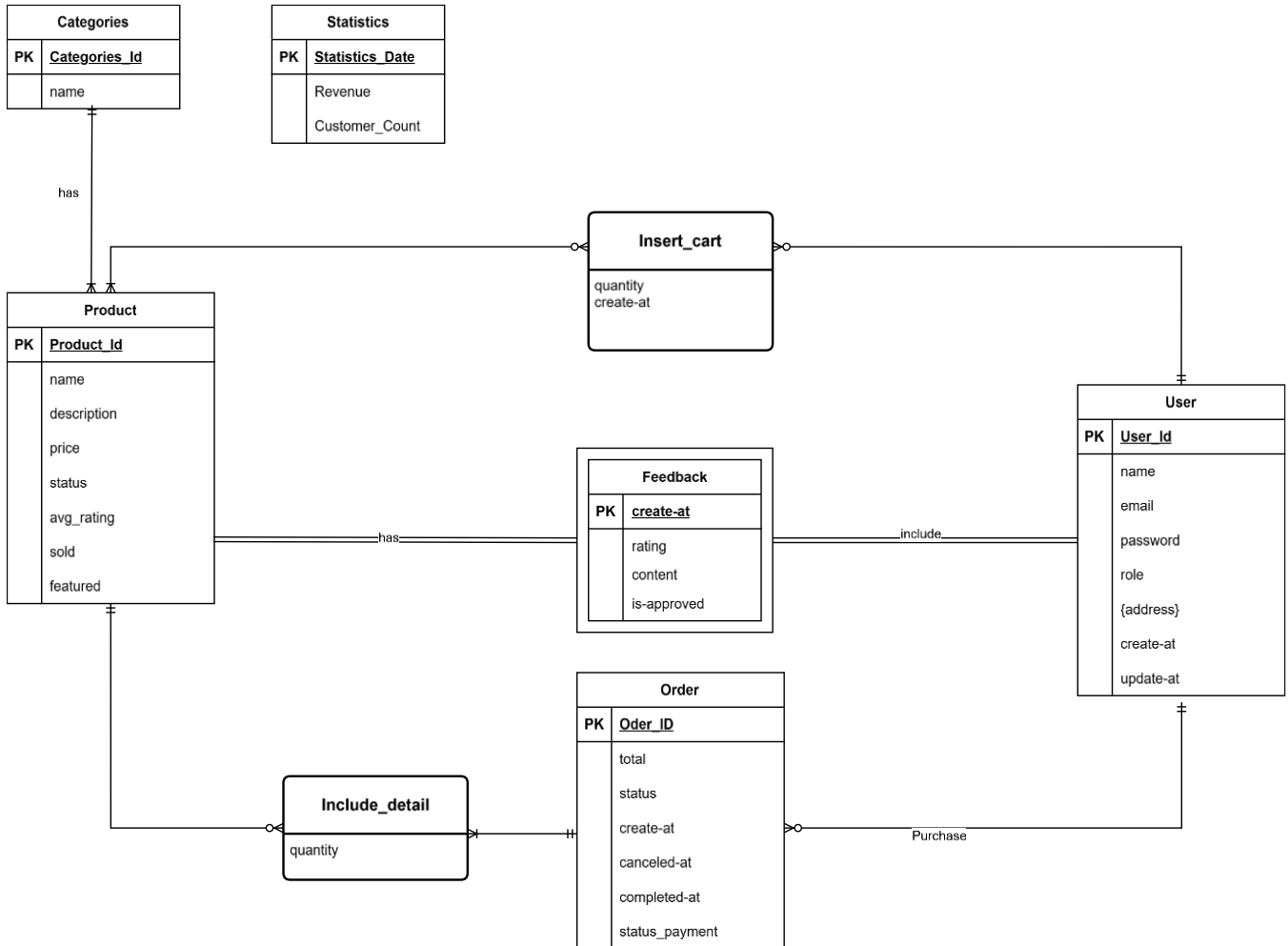
Tài liệu tập trung vào các chủ đề:

- ✓ Tạo ra tài liệu thiết kế phần mềm.
- ✓ Hoàn chỉnh tài liệu thiết kế phần mềm với các nội dung:
  - Mô hình quan niệm
  - Thiết kế kiến trúc
  - Thiết kế dữ liệu
  - Thiết kế giao diện người dùng
- ✓ Đọc hiểu tài liệu thiết kế phần mềm.

# 1 Bảng đánh giá thành viên

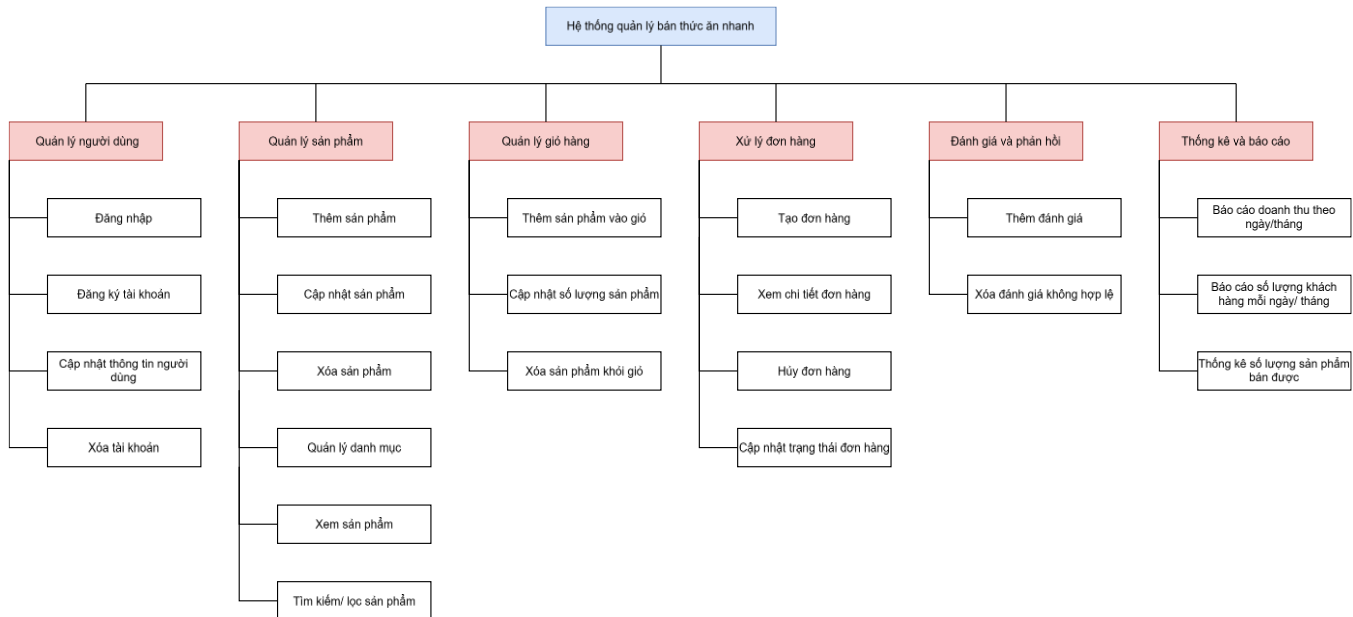
MSSV	Họ Tên	% đóng góp (tối đa 100%)	Chữ ký
22120167	Mạnh Trọng Kiên	100%	
22120215	Nguyễn Thị Mỹ	100%	
22120224	Trần Hoàng Kim Ngân	100%	
22120258	Quách Quỳnh Như	100%	
22120265	Nguyễn Thanh Phong	100%	

# 2 Mô hình quan niệm



Sơ đồ thể hiện các thực thể ngữ nghĩa trong trang web bán hàng.

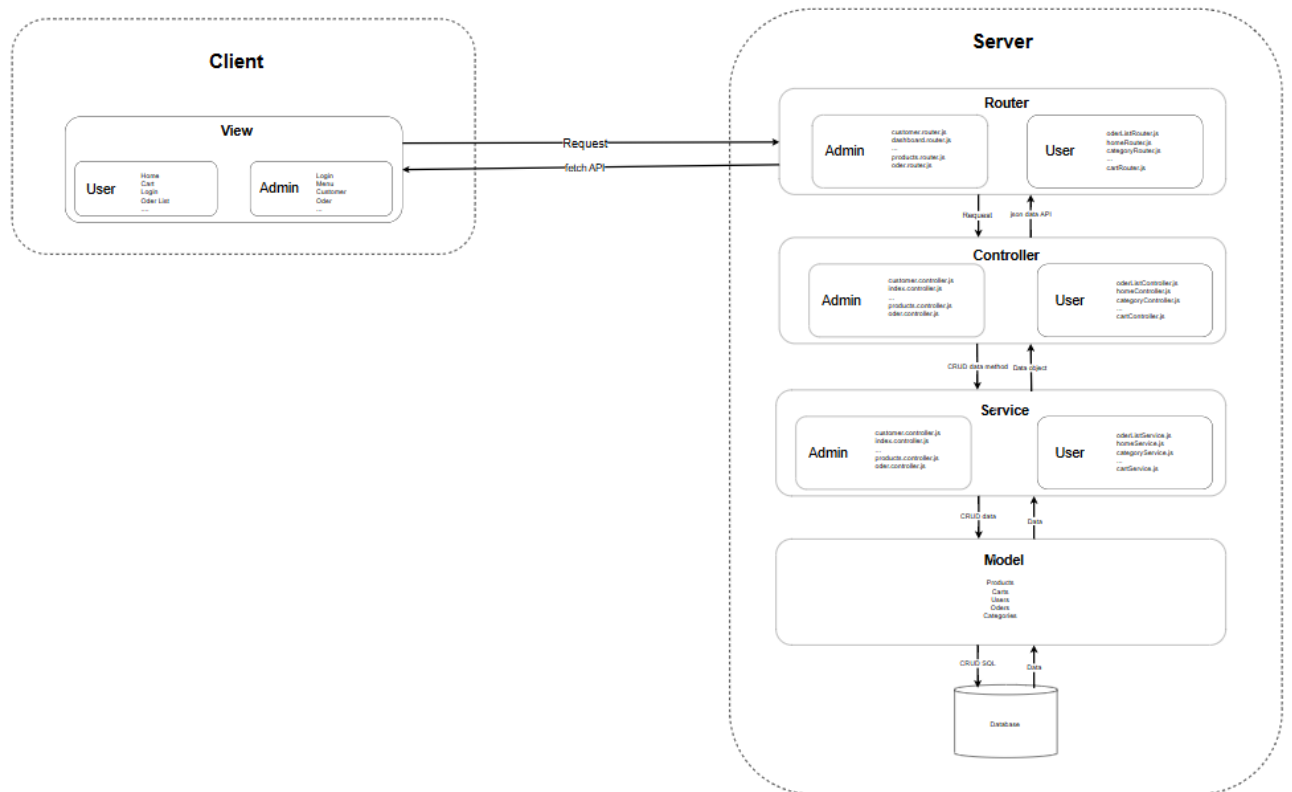
# 3 Thiết kế kiến trúc



## Sơ đồ phân rã theo chiều dọc phần mềm

### Các Thành Phần Trong Hệ Thống Quản Lý Bán Hàng Online

- **Quản lý người dùng:** gồm các chức năng liên quan đến người dùng: đăng nhập, đăng xuất, đăng ký tài khoản mới và xóa tài khoản.
- **Quản lý sản phẩm:** bao gồm các công việc và yêu cầu liên quan đến sản phẩm.
- **Quản lý giỏ hàng:** thực hiện công việc về giỏ hàng như thêm, xóa,...
- **Xử lý đơn hàng:** Quản lý quy trình tạo đơn hàng, cập nhật trạng thái đơn hàng (đang xử lý, đã giao, đã hủy) và xử lý các yêu cầu liên quan đến đơn hàng.
- **Đánh giá và phản hồi:** thực hiện liên quan đến đánh giá, phản hồi của khách hàng.
- **Thống kê và báo cáo:** thực hiện các báo cáo về doanh thu, số sản phẩm, khách hàng.



### Sơ đồ kiến trúc tổng thể theo mô hình Client – Server và MVC

#### 1. Kiến trúc áp dụng:

**MVC (Model-View-Controller):** Hệ thống áp dụng mẫu thiết kế MVC để phân chia các thành phần chức năng của ứng dụng.

- **Controller:** Sử dụng **Express.js** và **Node.js**. Các controller như `categoryController.js`, `products.controller.js`, và `createItem.controller.js`,... đảm nhận các yêu cầu từ người dùng, cập nhật **Model** và chọn **View**. **Express.js** tiếp nhận và xử lý các yêu cầu HTTP từ client, chuyển hướng yêu cầu đến các controller tương ứng để xử lý.
- **View:** Hiển thị dữ liệu từ **Model** cho người dùng, dùng **HTML**, kết hợp với **EJS** để render nội dung động. **TailwindCSS** được sử dụng để thiết kế giao diện, và **JavaScript** sẽ đảm nhận các chức năng như thêm, xóa món ăn, hoặc xem trước hình ảnh trong quá trình tương tác của người dùng.

- **Model:** Quản lý dữ liệu của hệ thống, như thông tin đơn hàng, món ăn, danh mục... được định nghĩa trong **postgreSQL** để thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên cơ sở dữ liệu.

#### **Client-Server Architecture:**

- **Client:** Phía client, ứng dụng frontend sẽ gửi các yêu cầu HTTP đến backend.
- **Server:** Backend của hệ thống được xây dựng bằng **Node.js** và **Express.js**, xử lý các yêu cầu HTTP từ client.
- Tất cả các trang được render bằng SSR.

## **2. Mẫu Design Pattern**

#### **Module Pattern:**

- Chia mã nguồn thành các module riêng biệt để quản lý các trách nhiệm khác nhau, ví dụ như `customer.controller.js`, `customer.services.js`, và `router.js`,.... Mỗi module chịu trách nhiệm cho một phần riêng biệt của ứng dụng, giúp mã dễ bảo trì và mở rộng.

#### **Model-View-Controller (MVC) Pattern:**

- Theo **kiến trúc MVC**, tách logic ứng dụng thành ba phần chính:
  - **Model:** Các dịch vụ (service) tương tác với cơ sở dữ liệu.
  - **View:** Các file `.ejs` để hiển thị giao diện người dùng.
  - **Controller:** Các router và các controller xử lý các yêu cầu HTTP và chuyển hướng đến các view phù hợp.
- Giúp dễ dàng quản lý và bảo trì mã nguồn.

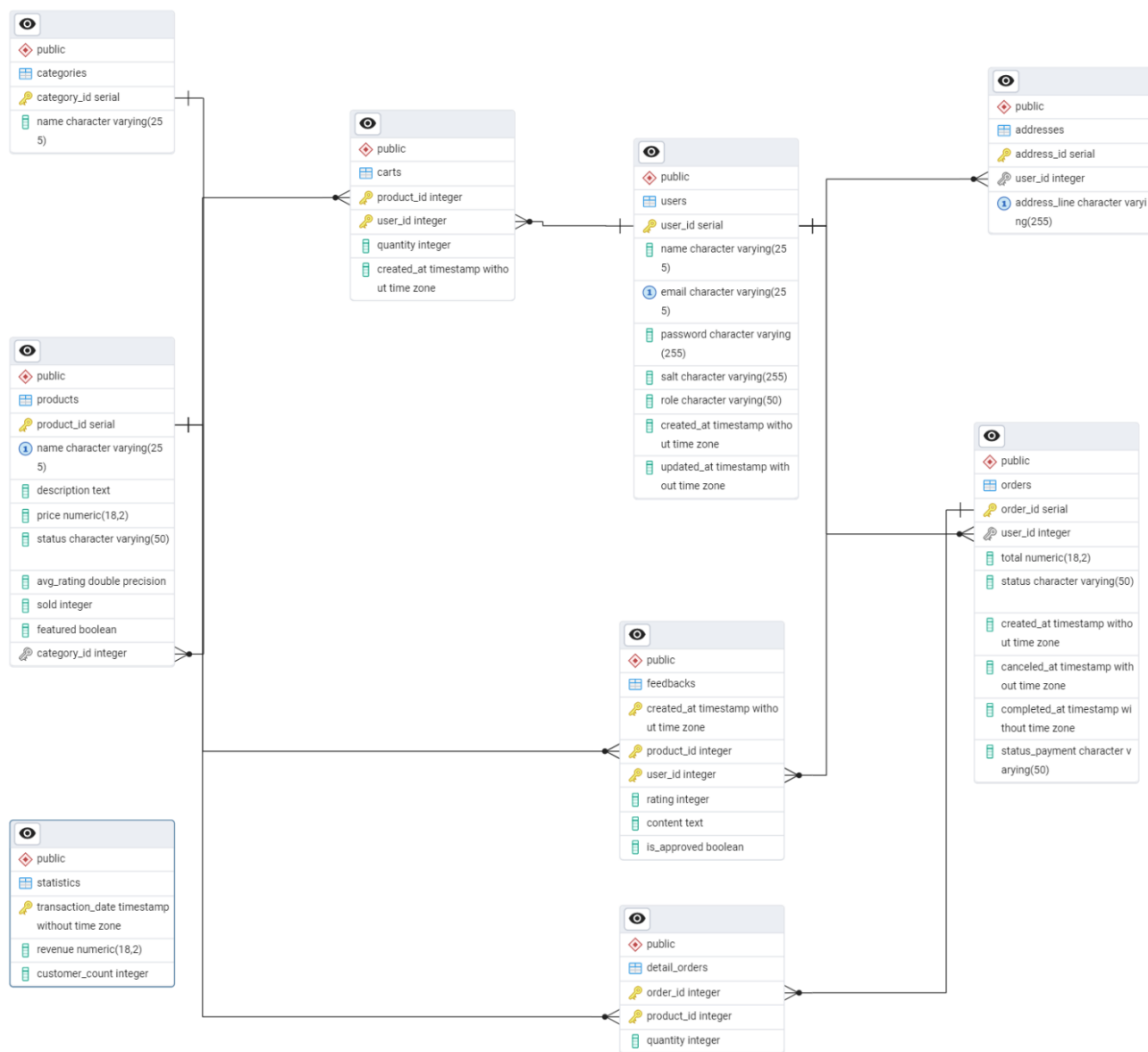
#### **Singleton Pattern:**

- Sử dụng kết nối cơ sở dữ liệu thông qua `knex` và `ConnectSessionKnexStore` để đảm bảo tính nhất quán và hiệu quả khi quản lý kết nối cơ sở dữ liệu và phiên làm việc của người dùng.



# 4 Thiết kế dữ liệu

## 4.1 Sơ đồ dữ liệu



## 4.2 Đặc tả dữ liệu

*Bảng Users*

Thuộc tính	Mô tả
<b>User_Id</b>	ID tự tăng, định danh người dùng Kiểu dữ liệu: SERIAL Ràng buộc: PRIMARY KEY
<b>name</b>	Tên người dùng Kiểu dữ liệu: VARCHAR(255) Ràng buộc: NOT NULL
<b>email</b>	Email của người dùng là duy nhất Kiểu dữ liệu: VARCHAR(255) Ràng buộc: UNIQUE, NOT NULL
<b>password</b>	Mật khẩu của người dùng Kiểu dữ liệu: VARCHAR(255) Ràng buộc: NOT NULL
<b>role</b>	Vai trò người dùng (admin, khách hàng) Kiểu dữ liệu: VARCHAR(50) Ràng buộc: NOT NULL
<b>salt</b>	Giá trị ngẫu nhiên được thêm vào chuỗi trước khi băm. Kiểu dữ liệu: VARCHAR(255) Ràng buộc: NOT NULL
<b>create_at</b>	Thời điểm tạo tài khoản Kiểu dữ liệu: TIMESTAMP Ràng buộc: DEFAULT CURRENT_TIMESTAMP

<b>update_at</b>	Thời điểm cập nhật thông tin Kiểu dữ liệu: TIMESTAMP Ràng buộc: DEFAULT CURRENT_TIMESTAMP
------------------	---

*Bảng Addresses*

Thuộc tính	Mô tả
<b>address_id</b>	ID tự tăng dùng để định danh một địa chỉ. Kiểu dữ liệu: SERIAL Ràng buộc: PRIMARY KEY
<b>user_id</b>	ID người dùng liên kết với địa chỉ. Kiểu dữ liệu: INT Ràng buộc: NOT NULL, FOREIGN KEY tham chiếu đến user_id trong bảng Users.
<b>address_line</b>	Địa chỉ cụ thể của người dùng. Kiểu dữ liệu: VARCHAR(255) Ràng buộc: NOT NULL, UNIQUE trong mỗi người dùng (kết hợp với user_id).
<b>unique_user_address</b>	Ràng buộc đảm bảo mỗi người dùng không thể có hai địa chỉ giống nhau (user_id, address_line).

*Bảng Categories*

Thuộc tính	Mô tả
<b>Categories_Id</b>	ID tự tăng dùng để định danh một danh mục Kiểu dữ liệu: SERIAL Ràng buộc: PRIMARY KEY
<b>name</b>	Tên danh mục Kiểu dữ liệu: VARCHAR(255) Ràng buộc: NOT NULL

*Bảng Products*

Thuộc tính	Mô tả
<b>Product_Id</b>	ID tự tăng dùng để định danh sản phẩm Kiểu dữ liệu: SERIAL Ràng buộc: PRIMARY KEY
<b>name</b>	Tên sản phẩm là duy nhất Kiểu dữ liệu: VARCHAR(255) Ràng buộc: UNIQUE, NOT NULL
<b>description</b>	Mô tả sản phẩm Kiểu dữ liệu: TEXT
<b>price</b>	Giá sản phẩm Kiểu dữ liệu: NUMERIC(18, 2) Ràng buộc: NOT NULL
<b>status</b>	Trạng thái sản phẩm (ví dụ: còn hàng, hết hàng) Kiểu dữ liệu: VARCHAR(50)
<b>avg_rating</b>	Điểm đánh giá trung bình Kiểu dữ liệu: FLOAT Ràng buộc: NOT NULL

<b>sold</b>	Số lượng sản phẩm đã bán Kiểu dữ liệu: INT Ràng buộc: NOT NULL
<b>featured</b>	Đánh dấu sản phẩm nổi bật Kiểu dữ liệu: BOOLEAN
<b>Categories_Id</b>	ID danh mục sản phẩm Kiểu dữ liệu: INT Ràng buộc: FOREIGN KEY tham chiếu đến <b>Categories(Categories_Id)</b>

*Bảng Feedbacks*

Thuộc tính	Mô tả
<b>create_at</b>	Thời điểm tạo đánh giá Kiểu dữ liệu: TIMESTAMP Ràng buộc: DEFAULT CURRENT_TIMESTAMP
<b>Product_Id</b>	Mã sản phẩm được đánh giá Kiểu dữ liệu: INT Ràng buộc: FOREIGN KEY tham chiếu đến bảng <b>Products(Product_Id)</b>
<b>User_Id</b>	Mã người dùng dùng để đánh giá Kiểu dữ liệu: INT Ràng buộc: FOREIGN KEY tham chiếu <b>Users(User_Id)</b>
<b>rating</b>	Điểm đánh giá (1 - 5) Kiểu dữ liệu: INT Ràng buộc: CHECK (rating >= 1 AND rating <= 5)

<b>content</b>	Nội dung đánh giá Kiểu dữ liệu: TEXT
<b>is_approved</b>	Trạng thái phê duyệt Kiểu dữ liệu: BOOLEAN
<b>Khóa chính</b>	Kết hợp của create_at, Product_Id, User_Id

**Bảng Cart**

Thuộc tính	Mô tả
<b>Product_Id</b>	Mã sản phẩm trong giỏ hàng Kiểu dữ liệu: INT Ràng buộc: FOREIGN KEY tham chiếu đến bảng <b>Products(Product_Id)</b>
<b>User_Id</b>	Mã người dùng sở hữu giỏ hàng Kiểu dữ liệu: INT Ràng buộc: FOREIGN KEY tham chiếu <b>Users(User_Id)</b>
<b>quantity</b>	Số lượng sản phẩm Kiểu dữ liệu: INT Ràng buộc: NOT NULL
<b>create_at</b>	Thời điểm thêm sản phẩm vào giỏ Kiểu dữ liệu: TIMESTAMP Ràng buộc: DEFAULT CURRENT_TIMESTAMP
<b>Khóa chính</b>	Kết hợp của Product_Id, User_Id

**Bảng Orders**

Thuộc tính	Mô tả
------------	-------

<b>id</b>	Mã đơn hàng dùng để định danh đơn hàng Kiểu dữ liệu: SERIAL Ràng buộc: PRIMARY KEY
<b>user_id</b>	Mã người dùng đặt hàng Kiểu dữ liệu: INT Ràng buộc: FOREIGN KEY tham chiếu <b>Users(User_Id)</b>
<b>total</b>	Tổng giá trị đơn hàng Kiểu dữ liệu: NUMERIC(18, 2) Ràng buộc: NOT NULL
<b>status</b>	Trạng thái đơn hàng Kiểu dữ liệu: VARCHAR(50)
<b>create_at</b>	Thời điểm tạo đơn hàng Kiểu dữ liệu: TIMESTAMP Ràng buộc: DEFAULT CURRENT_TIMESTAMP
<b>canceled_at</b>	Thời điểm hủy đơn hàng Kiểu dữ liệu: TIMESTAMP
<b>completed_at</b>	Thời điểm hoàn thành đơn hàng Kiểu dữ liệu: TIMESTAMP
<b>status_payment</b>	Trạng thái thanh toán Kiểu dữ liệu: VARCHAR(50)

**Bảng Detail\_Orders**

Thuộc tính	Mô tả
<b>Order_id</b>	ID đơn hàng Kiểu dữ liệu: INT Ràng buộc: FOREIGN KEY tham chiếu <b>Orders(id)</b>

<b>Product_Id</b>	ID sản phẩm trong đơn hàng Kiểu dữ liệu: INT Ràng buộc: FOREIGN KEY tham chiếu <b>Products(Product_Id)</b>
<b>quantity</b>	Số lượng sản phẩm Kiểu dữ liệu: INT Ràng buộc: NOT NULL
<b>Khóa chính</b>	Kết hợp Order_id, Product_Id

### *Bảng Statistics*

Thuộc tính	Mô tả
<b>Transaction_Date</b>	Ngày giao dịch Kiểu dữ liệu: TIMESTAMP Ràng buộc: PRIMARY KEY
<b>Revenue</b>	Doanh thu Kiểu dữ liệu: NUMERIC(18, 2) Ràng buộc: NOT NULL
<b>Customer_Count</b>	Số lượng khách hàng Kiểu dữ liệu: INT Ràng buộc: NOT NULL