**BÀI LÀM**

1. **TẠO CƠ SỞ DỮ LIỆU**
2. **Mô tả cơ sở dữ liệu**

Trang web sàn thương mại điện tử cung cấp một nền tảng cho cả người mua và người bán tham gia hoạt động mua bán trực tuyến. Các tài khoản được chia thành hai loại chính: tài khoản người mua và tài khoản người bán. Người mua có khả năng mua nhiều mặt hàng khác nhau và từ đó tạo ra nhiều đơn hàng khác nhau. Mỗi đơn hàng được liệt kê trong phần "đơn hàng" (orders), với các trạng thái khác nhau như: đang xử lý, đang giao, đã giao và bị hủy. Khi một đơn hàng đã được giao thành công, người mua có thể đánh giá đơn hàng đó dựa trên thang điểm từ 1 đến 5 và cung cấp thông tin chi tiết hơn về sản phẩm đã mua. Trong hệ thống mua sắm trực tuyến này, mỗi người mua có thể tạo ra nhiều đơn hàng khác nhau, phản ánh sự đa dạng trong sở thích mua sắm của họ. Mỗi đơn hàng được thể hiện trong bảng "orders", nơi chứa thông tin chi tiết về đơn hàng như số đơn hàng, danh sách các mặt hàng được mua, và trạng thái của đơn hàng. Trạng thái của đơn hàng được biểu thị bằng các số từ 1 đến 4, tương ứng với các trạng thái: đang xử lý, đang giao, đã giao và bị hủy. Khi một đơn hàng đã được giao thành công, người mua có thể đánh giá đơn hàng đó dựa trên thang điểm từ 1 đến 5. Đánh giá này cung cấp thông tin quan trọng cho cả người bán và các người mua tiềm năng về chất lượng của sản phẩm và dịch vụ. Đánh giá này thường bao gồm đánh giá về sản phẩm, dịch vụ giao hàng, và trải nghiệm tổng thể với cửa hàng. Thông tin chi tiết về đánh giá có thể bao gồm các yếu tố như chất lượng của sản phẩm, tính chất đúng hạn của giao hàng, và mức độ hài lòng với dịch vụ khách hàng. Đối với mỗi đánh giá, người mua có thể cung cấp thông tin chi tiết hơn về sản phẩm, bao gồm các thành phần có trong bảng "orderdetails". Các thành phần này có thể bao gồm mô tả chi tiết về sản phẩm, giá cả, số lượng, và bất kỳ thông tin khác liên quan đến mặt hàng được mua.

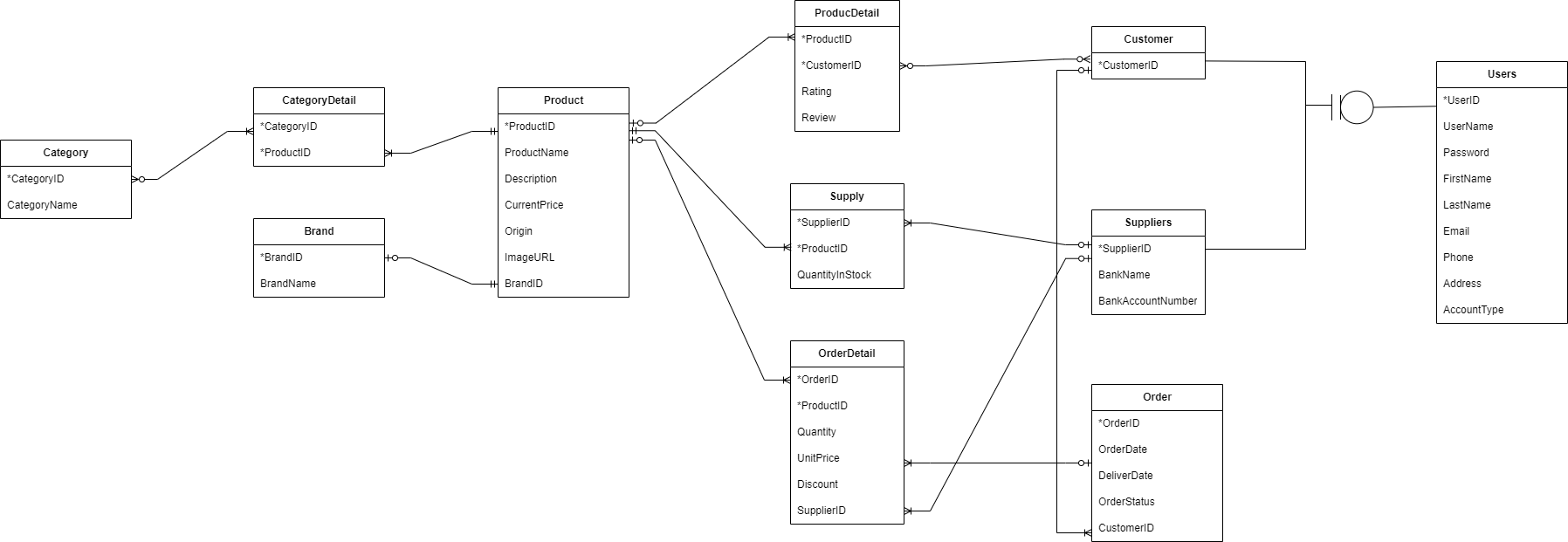
Tài khoản người mua được quản lý trong bảng "users", chứa thông tin cá nhân của người mua như tên, địa chỉ, thông tin liên lạc, và các thông tin khác liên quan đến tài khoản và lịch sử mua hàng của họ. Điều này giúp hệ thống theo dõi và quản lý thông tin của mỗi người mua một cách hiệu quả, từ đó tạo ra trải nghiệm mua sắm tốt nhất cho họ. Thông tin về tài khoản người mua bao gồm các thành phần có trong bảng "người mua" (buyers). Trong khi đó, người bán là các nhà sách có thể bán nhiều sách khác nhau và mỗi sản phẩm chỉ được bán bởi một nhà sách duy nhất. Thông tin về người bán được hiển thị trên trang web và bao gồm các thành phần trong bảng "người bán" (sellers).

Trang web sàn thương mại điện tử là một nền tảng đa dạng, kết nối giữa người mua và người bán trong quá trình mua bán hàng hóa trực tuyến. Tại đây, người mua và người bán đều có tài khoản riêng, mỗi tài khoản mang đến những chức năng và tính năng đặc biệt.

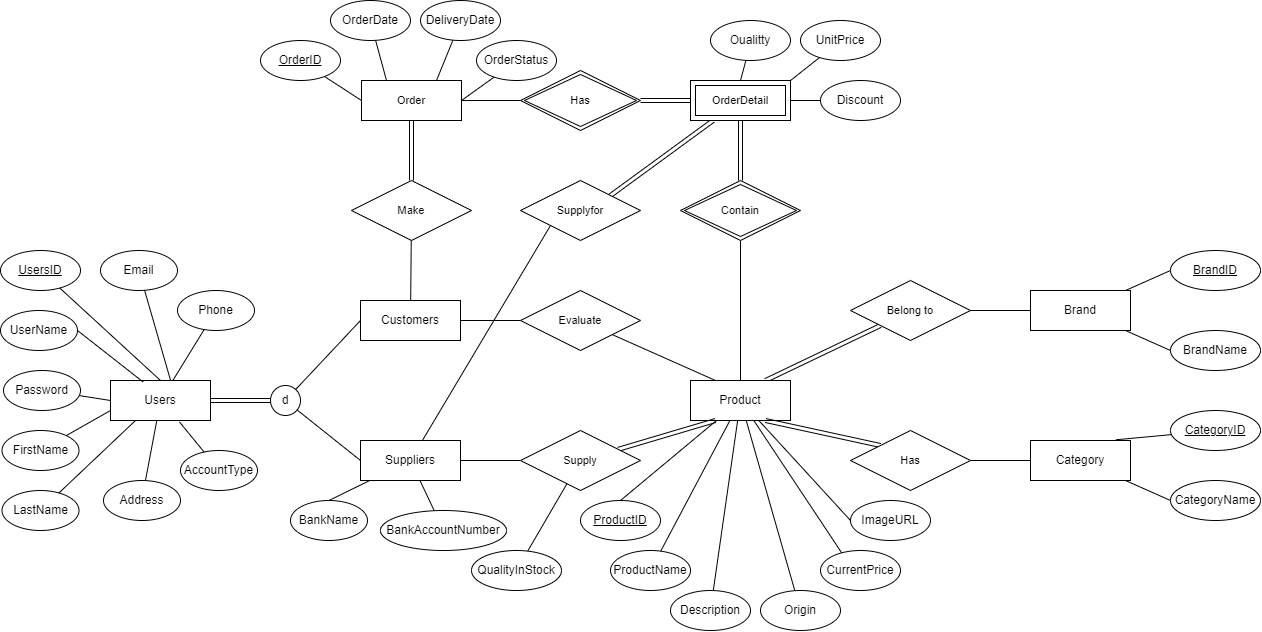
Với tài khoản người mua, họ có thể thực hiện các giao dịch mua sắm, tạo ra đơn hàng với các sản phẩm khác nhau. Mỗi đơn hàng được theo dõi thông qua trạng thái từ quá trình xử lý, giao hàng cho đến khi đơn hàng được giao thành công hoặc hủy bỏ. Khi một đơn hàng được hoàn tất, người mua có cơ hội đánh giá sản phẩm và dịch vụ, cung cấp ý kiến phản hồi cho cộng đồng. Thông tin của người mua được quản lý trong bảng "người mua", bao gồm các chi tiết về tài khoản và lịch sử giao dịch. Trong khi đó, người bán, thường là các nhà sách, cũng có tài khoản riêng và có thể quản lý các sản phẩm của mình trên trang web. Mỗi cuốn sách được phân loại theo độ tuổi và thể loại sách, giúp người mua dễ dàng tìm kiếm và lựa chọn sản phẩm phù hợp. Thông tin về người bán được hiển thị trong bảng "người bán", cung cấp các thông tin cần thiết như địa chỉ, thông tin liên hệ và danh sách sản phẩm mà họ đang cung cấp.

Từ sự kết nối chặt chẽ giữa người mua và người bán trên trang web này, người dùng không chỉ trải nghiệm một quá trình mua sắm trực tuyến thuận tiện và linh hoạt, mà còn tạo ra một cộng đồng mua bán đáng tin cậy và phát triển. Qua việc chia sẻ trải nghiệm, đánh giá sản phẩm và tương tác trực tiếp, cả người mua và người bán đều được hưởng lợi từ sự tin cậy và sự phát triển của cộng đồng này. Sự kết nối này không chỉ là một nền tảng thương mại điện tử, mà còn là một môi trường nơi mọi người có thể giao lưu, chia sẻ và tạo ra những mối quan hệ kinh doanh lâu dài. Chính nhờ vào sự kết nối này, thị trường sách trực tuyến ngày càng trở nên sôi động và phát triển, mang lại lợi ích to lớn cho cả người mua và người bán.

1. **Lược đồ cơ sở dữ liệu**
2. **Mô hình quan hệ - thực thể (Entity - Relationship Model)**

****

**b. Lược đồ cơ sở dữ liệu**

****

1. **Script tạo bảng và chèn dữ liệu**

CREATE TABLE Brand (

BrandID INT AUTO\_INCREMENT PRIMARY KEY,

BrandName VARCHAR (255) NOT NULL

);

CREATE TABLE Category (

CategoryID INT AUTO\_INCREMENT PRIMARY KEY,

CategoryName VARCHAR (255) NOT NULL

);

CREATE TABLE Accounts (

UserID INT AUTO\_INCREMENT PRIMARY KEY,

UserName VARCHAR (255) NOT NULL,

PassWordd VARCHAR (255) NOT NULL,

FirstName VARCHAR (25) NOT NULL,

LastName VARCHAR (255) NOT NULL,

Email VARCHAR (255) NOT NULL,

Phone VARCHAR (10) NOT NULL,

Address VARCHAR (255) NOT NULL,

AccountType ENUM('Buyer', 'Seller') NOT NULL

);

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

FOREIGN KEY (CustomerID) REFERENCES Accounts(UserID) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Suppliers (

SupplierID INT PRIMARY KEY,

BankName VARCHAR (255) NOT NULL,

BankAccountNumber DECIMAL(10,0) NOT NULL,

FOREIGN KEY (SupplierID) REFERENCES Accounts(UserID) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Products (

ProductID INT AUTO\_INCREMENT PRIMARY KEY,

ProductName VARCHAR (255) NOT NULL,

ProductDescription VARCHAR (255) NOT NULL,

BrandID INT NOT NULL,

CurrentPrice INT NOT NULL,

Origin VARCHAR (255) NOT NULL,

ImageURL VARCHAR (255) NOT NULL,

FOREIGN KEY (BrandID) REFERENCES Brand(BrandID) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Supply (

SupplierID INT ,

ProductID INT,

QuantityInStock INT NOT NULL,

PRIMARY KEY (SupplierID, ProductID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Orders (

OrderID INT AUTO\_INCREMENT PRIMARY KEY,

CustomerID INT NOT NULL,

OrderDate DATE NOT NULL,

DeliverDate DATE ,

OrderStatus tinyint NOT NULL,

-- Processing==1,Shipping==2,Delivered==3, Cancelled==4

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE CategoryDetail (

CategoryID INT,

ProductID INT,

PRIMARY KEY (CategoryID, ProductID),

FOREIGN KEY (CategoryID) REFERENCES Category (CategoryID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (ProductID) REFERENCES products (ProductID) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Ordersdetail (

OrderID INT ,

ProductID INT ,

SupplierID INT NOT NULL ,

Quantity INT NOT NULL,

UnitPrice INT NOT NULL,

Discount DECIMAL (4, 2) NOT NULL DEFAULT 0,

PRIMARY KEY (OrderID, ProductID),

FOREIGN KEY (OrderID) REFERENCES Orders (OrderID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (ProductID) REFERENCES products (ProductID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (SupplierID) REFERENCES Suppliers (SupplierID) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE Productsdetail (

CustomerID INT ,

ProductID INT ,

Rating DECIMAL (4, 1),

Review VARCHAR (255),

PRIMARY KEY (CustomerID, ProductID),

FOREIGN KEY (CustomerID) REFERENCES Customers (CustomerID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (ProductID) REFERENCES Products (ProductID) ON DELETE CASCADE ON UPDATE CASCADE

);

**II. Truy vấn SQL**

1. **Thống kê các sản phẩm chưa được mua lần nào**

**Script:**

SELECT a.ProductName AS `Tên sản phẩm`,

d.BrandName AS `Tên thương hiệu`,

c.CategoryName AS `Tên danh mục`,

g.FirstName AS `Họ nhà cung cấp`,

g.LastName AS `Tên nhà cung cấp`

FROM Products AS a

JOIN CategoryDetail AS b ON a.ProductID = b.ProductID

JOIN Category AS c ON c.CategoryID = b.CategoryID

JOIN Brand AS d ON d.BrandID = a.BrandID

JOIN Supply AS e ON e.ProductId = a.ProductID

JOIN Suppliers AS f ON f.SupplierID = e.SupplierID

JOIN Accounts AS g ON g.UserID = f.SupplierID

WHERE a.ProductID NOT IN (

SELECT a.ProductID

FROM Ordersdetail AS a

JOIN Orders AS b ON b.OrderID = a.OrderID

WHERE b.OrderStatus = '3');

**Mô tả:**  Bằng cách đặt lại các tên bảng, ta sử dụng phép kết tự nhiên (JOIN) giữa các bảng để kết hợp thông tin từ các bảng khác nhau. Điều kiện WHERE được sử dụng để lọc ra các sản phẩm chưa từng được mua, trái ngược với đã đặt và hoàn thành giao hàng (OrderStatus = '3') thông qua một câu truy vấn con trên bảng "Ordersdetail" và "Orders". Cuối cùng, kết quả trả về sẽ bao gồm tên sản phẩm, tên thương hiệu, tên danh mục, họ và tên của nhà cung cấp.

**Kết quả:**

****

1. **Thống kê số lượng sản phẩm theo từng thương hiệu**

**Script:**

SELECT b.BrandName AS `Thương hiệu`,

SUM(e.QuantityInstock) AS `Số lượng sản phẩm tồn kho của mỗi thương hiệu`

FROM Products AS a

JOIN Brand AS b ON b.BrandID = a.BrandID

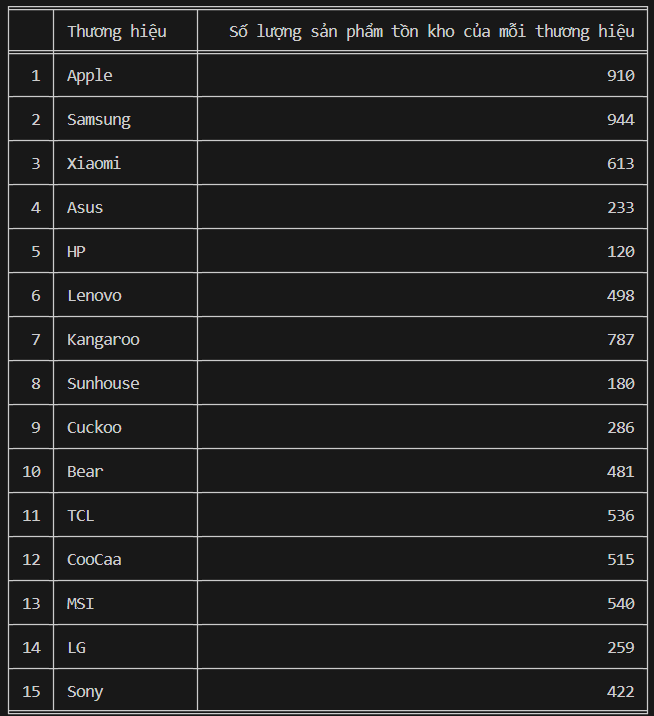
JOIN Supply AS e ON e.ProductId = a.ProductID

JOIN Suppliers AS f ON f.SupplierID = e.SupplierID

GROUP BY b.BrandID;

**Mô tả:** Câu truy vấn sử dụng các phép kết tự nhiên(JOIN) giữa các bảng để kết hợp thông tin từ các bảng khác nhau. Sau đó, câu truy vấn nhóm các kết quả theo BrandID để tính tổng số lượng sản phẩm tồn kho của mỗi thương hiệu. Cuối cùng, kết quả trả về sẽ bao gồm tên thương hiệu và số lượng sản phẩm tồn kho tương ứng của mỗi thương hiệu.

**Kết quả:**

****

1. **Lấy thông tin sản phẩm có điểm đánh giá trung bình lớn nhất theo từng thương hiệu**

**Script:**

SELECT l.BrandName AS `Tên thương hiệu`,

m.ProductName AS `Tên sản phẩm`,

temp.AvgRating2 AS `Điểm đánh giá trung bình`

FROM Products AS m

JOIN Brand AS l ON l.BrandID = m.BrandID

JOIN (

SELECT x.ProductID, x.BrandID, AVG(y.Rating) AS AvgRating2

FROM Products AS x

JOIN ProductsDetail AS y ON x.ProductID = y.ProductID

JOIN Brand AS z ON z.BrandID = x.BrandID

GROUP BY x.ProductID, x.BrandID

HAVING (x.BrandID, AvgRating2) IN (

SELECT max\_ratings.BrandID, MAX(AvgRating)

FROM (

SELECT a.ProductID, a.BrandID, AVG(b.Rating) AS AvgRating

FROM Products AS a

JOIN ProductsDetail AS b ON a.ProductID = b.ProductID

JOIN Brand AS c ON c.BrandID = a.BrandID

GROUP BY a.ProductID, a.BrandID

ORDER BY AvgRating DESC

) AS max\_ratings

GROUP BY max\_ratings.BrandID

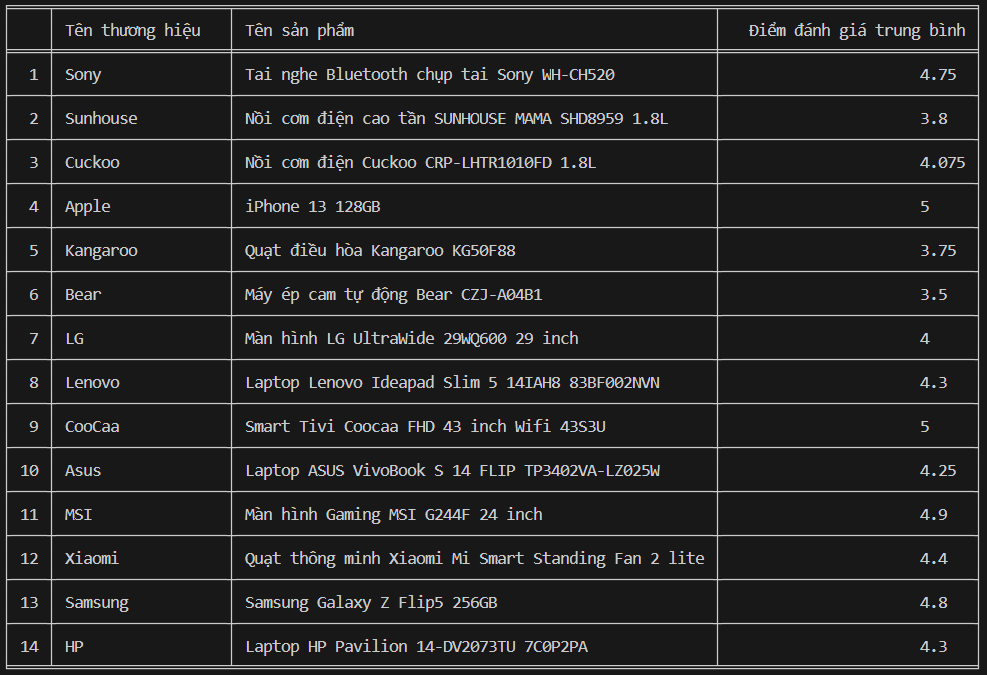
)

) AS temp ON temp.BrandID = m.BrandID AND temp.ProductID = m.ProductID;

**Mô tả:** Đầu tiên, thực hiện truy vấn dữ liệu từ các bảng "Products" và "Brand" để hiển thị thông tin về tên thương hiệu, tên sản phẩm và điểm đánh giá trung bình của mỗi sản phẩm. Để tính toán điểm đánh giá trung bình, một bảng tạm thời được tạo ra bằng cách kết hợp dữ liệu từ các bảng "Products" và "ProductsDetail". Bảng tạm thời này chứa các cột ProductID, BrandID và AvgRating2, trong đó AvgRating2 là điểm đánh giá trung bình của từng sản phẩm.

Sau đó, bảng tạm thời này được kết hợp với bảng "Products" thông qua các cột BrandID và ProductID để thu thập thông tin về tên thương hiệu và tên sản phẩm tương ứng. Điều này được thực hiện thông qua một câu lệnh JOIN. Cuối cùng, câu lệnh HAVING được sử dụng để chọn ra sản phẩm có điểm đánh giá trung bình cao nhất cho mỗi thương hiệu. Điều này đảm bảo rằng chỉ có một sản phẩm được chọn cho mỗi thương hiệu, với điểm đánh giá cao nhất.

**Kết quả:**

****

1. **Lấy thông tin các sản phẩm có số lượng bán được ít hơn 10 trong tháng 1 năm 2024 của tài khoản (người bán) "XYZ"**

**Script:**

SELECT temp.ProductName AS `Tên sản phẩm`, temp.soluongbanduoc AS `Số lượng bán được`

FROM (

SELECT b.ProductID, c.ProductName, SUM(b.Quantity) AS soluongbanduoc

FROM orders AS a

JOIN ordersdetail AS b ON a.orderid = b.orderid

JOIN products AS c ON c.productID = b.ProductID

WHERE a.orderstatus = 3 AND MONTH(a.OrderDate) = 1 AND YEAR(a.OrderDate) = 2024

AND b.SupplierID = (SELECT n.UserID FROM Accounts AS n WHERE n.UserName = 'XYZ')

GROUP BY b.ProductID

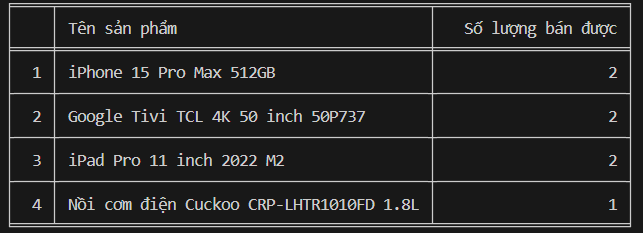
HAVING soluongbanduoc < 10

) AS temp;

**Mô tả:** Từ bảng "orders", dữ liệu về đơn hàng được kết hợp với thông tin chi tiết từ bảng "ordersdetail" và thông tin về sản phẩm từ bảng "products". Các đơn hàng được lọc để chỉ chọn những đơn có trạng thái là 3 (tức là đã hoàn thành), được đặt trong tháng 1 và năm 2024, và được cung cấp bởi người dùng có tên người dùng là 'XYZ'.

Sau đó, dữ liệu sau khi lọc được nhóm lại theo ProductID để tính tổng số lượng bán được của mỗi sản phẩm. Các sản phẩm có tổng số lượng bán được ít hơn 10 được chọn ra bằng câu lệnh HAVING.

**Kết quả:**

****

1. **Lấy thông tin các sản phẩm có số lượng bán được nhiều nhất trong từng tháng của năm 2023**

**Script:**

SELECT MONTH(m.OrderDate) AS `Thang (2023)`, k.ProductName AS `Ten san pham`, SUM(n.quantity) AS `So luong ban duoc`

FROM orders AS m

JOIN ordersdetail AS n ON m.orderid = n.orderid

JOIN products AS l ON l.productID = n.ProductID

JOIN Products AS k ON k.ProductID = l.ProductID

WHERE (

MONTH(m.OrderDate),

(

SELECT SUM(b.Quantity)

FROM orders AS a

JOIN ordersdetail AS b ON a.orderid = b.orderid

WHERE a.orderstatus = 3

AND YEAR(a.OrderDate) = 2023

AND MONTH(a.OrderDate) = MONTH(m.OrderDate)

AND b.ProductID = n.ProductID

)

) IN (

SELECT MaxTotal.Month, MAX(MaxTotal.TotalQuantity)

FROM (

SELECT c.ProductID, MONTH(a.OrderDate) AS Month, SUM(b.Quantity) AS TotalQuantity

FROM orders AS a

JOIN ordersdetail AS b ON a.orderid = b.orderid

JOIN products AS c ON c.productID = b.ProductID

WHERE a.orderstatus = 3

AND YEAR(a.OrderDate) = 2023

GROUP BY c.ProductID, MONTH(a.OrderDate)

ORDER BY TotalQuantity DESC

) AS MaxTotal

GROUP BY MaxTotal.Month

HAVING MAX(MaxTotal.TotalQuantity)

)

GROUP BY n.ProductID, MONTH(m.OrderDate), k.ProductName

ORDER BY MONTH(m.OrderDate) ASC;

**Mô tả:** Dữ liệu từ bảng "orders" được kết hợp với thông tin chi tiết từ bảng "ordersdetail" và thông tin về sản phẩm từ bảng "products". Đồng thời, sản phẩm cũng được kết hợp với chính nó thông qua bảng "Products", để có thể truy xuất thông tin về tên sản phẩm. Sai đó, thông qua câu lệnh WHERE, chỉ các đơn hàng có trạng thái là 3 (đã hoàn thành) và được đặt trong năm 2023 mới được lựa chọn.

Dữ liệu sau khi lọc được nhóm lại theo ProductID và tháng (tính từ OrderDate) để tính tổng số lượng bán được của mỗi sản phẩm trong từng tháng rồi lưu trữ trong bảng MaxTotal. Câu lệnh HAVING được sử dụng để chọn ra cặp tháng và số lượng bán được có tổng số lượng bán được lớn nhất cho mỗi tháng trong năm 2023 Điều này đảm bảo chỉ có một sản phẩm có tổng số lượng bán được lớn nhất cho mỗi tháng.

**Kết quả:**

****

1. **Thống kê số lượng sản phẩm theo từng thương hiệu**

**Script:**

SELECT months.month AS `Thang`, IFNULL(monthly\_data.TotalPrice, 0) AS `Tong doanh thu thang`

FROM (

SELECT 1 AS month UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5 UNION SELECT 6 UNION

SELECT 7 UNION SELECT 8 UNION SELECT 9 UNION SELECT 10 UNION SELECT 11 UNION SELECT 12

) AS months

LEFT JOIN (

SELECT MONTH(a.OrderDate) AS OrderMonth, SUM(b.UnitPrice\*b.Quantity) AS TotalPrice

FROM orders AS a

JOIN ordersdetail AS b ON a.orderid = b.orderid

JOIN products AS c ON c.productID = b.ProductID

WHERE a.orderstatus = 3

AND YEAR(a.OrderDate) = 2023

AND b.SupplierID = (

SELECT n.UserID

FROM Accounts AS n

WHERE n.UserName = 'XYZ'

)

GROUP BY MONTH(a.OrderDate), b.SupplierID

) AS monthly\_data ON months.month = monthly\_data.OrderMonth

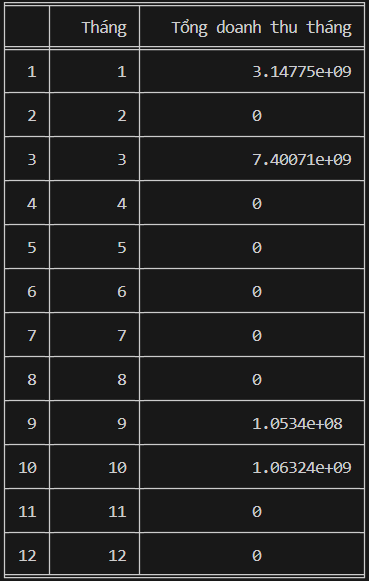
GROUP BY months.month, monthly\_data.TotalPrice;

**Mô tả:** Đầu tiên, một bảng tạm thời được tạo ra chứa các số từ 1 đến 12, đại diện cho các tháng trong năm, bằng cách sử dụng câu lệnh UNION. Sau đó, bảng tạm thời này được kết hợp với dữ liệu về tổng doanh thu hàng tháng (nếu có) thông qua LEFT JOIN. Việc sử dụng LEFT JOIN đảm bảo rằng các tháng không có doanh thu cũng được bao gồm trong kết quả cuối cùng.

Dữ liệu về tổng doanh thu hàng tháng được tính toán thông qua việc kết hợp dữ liệu từ các bảng "orders", "ordersdetail" và "products". Các đơn hàng chỉ được chọn nếu có trạng thái là 3 (đã hoàn thành), được đặt trong năm 2023 và được cung cấp bởi người dùng có tên người dùng là 'XYZ'.

Sau đó, dữ liệu được nhóm lại theo tháng và ID của nhà cung cấp để tính tổng doanh thu hàng tháng. Cuối cùng, thông tin về tháng và tổng doanh thu hàng tháng được chọn để hiển thị, và được nhóm lại theo tháng và tổng doanh thu để đảm bảo kết quả duy nhất cho mỗi tháng. Nếu một tháng không có doanh thu, giá trị của nó sẽ được thay thế bằng 0 để đảm bảo tính toán chính xác và đồng nhất.

**Kết quả:**

****