Creating a dataset for SQL practice that simulates a real-world business scenario can be highly beneficial for learning and applying SQL skills. Here, I'll create a sample dataset for an e-commerce company, which includes typical tables such as Customers, Products, Orders, OrderDetails, and Reviews.

keyboard\_arrow\_down

Business Scenario

Imagine an e-commerce company that sells various products. The company tracks customers, products, orders, order details, and customer reviews. You want to analyze sales, customer behavior, and product performance using SQL queries.

Tables and Columns

1. Customers

* CustomerID: Unique identifier for each customer
* FirstName: Customer's first name
* LastName: Customer's last name
* Email: Customer's email address
* PhoneNumber: Customer's phone number
* Address: Customer's address
* City: Customer's city
* State: Customer's state
* ZipCode: Customer's zip code
* Country: Customer's country
* SignUpDate: Date the customer signed up

2. Products

* ProductID: Unique identifier for each product
* ProductName: Name of the product
* Category: Product category
* Price: Price of the product
* StockQuantity: Quantity of the product in stock
* SupplierID: ID of the supplier

3. Orders

* OrderID: Unique identifier for each order
* CustomerID: ID of the customer who placed the order
* OrderDate: Date the order was placed
* TotalAmount: Total amount of the order

4. OrderDetails

* OrderDetailID: Unique identifier for each order detail
* OrderID: ID of the order
* ProductID: ID of the product
* Quantity: Quantity of the product ordered
* UnitPrice: Price per unit of the product
* TotalPrice: Total price for the quantity ordered

5. Reviews

* ReviewID: Unique identifier for each review
* CustomerID: ID of the customer who wrote the review
* ProductID: ID of the product being reviewed
* Rating: Rating given by the customer (1 to 5)
* ReviewDate: Date the review was written
* Comments: Review comments

Sample Data

Create database salesdb;

Use salesdb;

Customers

CREATE TABLE Customers (  
    CustomerID INT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    Email VARCHAR(100),  
    PhoneNumber VARCHAR(20),  
    Address VARCHAR(100),  
    City VARCHAR(50),  
    State VARCHAR(50),  
    ZipCode VARCHAR(10),  
    Country VARCHAR(50),  
    SignUpDate DATE  
);  
  
INSERT INTO Customers VALUES  
(1, 'John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Elm St', 'Springfield', 'IL', '62701', 'USA', '2023-01-15'),  
(2, 'Jane', 'Smith', 'jane.smith@example.com', '234-567-8901', '456 Oak St', 'Chicago', 'IL', '60601', 'USA', '2023-02-20'),  
(3, 'Alice', 'Johnson', 'alice.johnson@example.com', '345-678-9012', '789 Pine St', 'Houston', 'TX', '77001', 'USA', '2023-03-10');

Products

CREATE TABLE Products (  
    ProductID INT PRIMARY KEY,  
    ProductName VARCHAR(100),  
    Category VARCHAR(50),  
    Price DECIMAL(10, 2),  
    StockQuantity INT,  
    SupplierID INT  
);  
  
INSERT INTO Products VALUES  
(1, 'Laptop', 'Electronics', 999.99, 50, 1),  
(2, 'Smartphone', 'Electronics', 599.99, 200, 2),  
(3, 'Tablet', 'Electronics', 399.99, 150, 1);

Orders

CREATE TABLE Orders (  
    OrderID INT PRIMARY KEY,  
    CustomerID INT,  
    OrderDate DATE,  
    TotalAmount DECIMAL(10, 2),  
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);  
  
INSERT INTO Orders VALUES  
(1, 1, '2023-03-15', 1599.98),  
(2, 2, '2023-04-10', 599.99),  
(3, 3, '2023-05-05', 999.99);

OrderDetails

CREATE TABLE OrderDetails (  
    OrderDetailID INT PRIMARY KEY,  
    OrderID INT,  
    ProductID INT,  
    Quantity INT,  
    UnitPrice DECIMAL(10, 2),  
    TotalPrice DECIMAL(10, 2),  
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),  
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  
);  
  
INSERT INTO OrderDetails VALUES  
(1, 1, 1, 1, 999.99, 999.99),  
(2, 1, 2, 1, 599.99, 599.99),  
(3, 2, 2, 1, 599.99, 599.99),  
(4, 3, 1, 1, 999.99, 999.99);

Reviews

CREATE TABLE Reviews (  
    ReviewID INT PRIMARY KEY,  
    CustomerID INT,  
    ProductID INT,  
    Rating INT,  
    ReviewDate DATE,  
    Comments TEXT,  
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),  
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  
);  
  
INSERT INTO Reviews VALUES  
(1, 1, 1, 5, '2023-03-20', 'Great laptop!'),  
(2, 2, 2, 4, '2023-04-15', 'Good smartphone, but battery life could be better.'),  
(3, 3, 1, 5, '2023-05-10', 'Excellent performance!');

Business Questions for SQL Practice

1. **Customer Analysis**:
   * How many customers signed up each month?
   * Find customers who have placed more than one order.
2. **Product Analysis**:
   * Which products have the highest sales?
   * Find the total stock quantity for each product category.
3. **Order Analysis**:
   * Calculate the total sales amount for each month.
   * Find the average order value.
4. **Review Analysis**:
   * Calculate the average rating for each product.
   * List all reviews for products in the 'Electronics' category.

Example Queries

1. How many customers signed up each month?

SELECT   
    DATE\_FORMAT(SignUpDate, '%Y-%m') AS Month,  
    COUNT(CustomerID) AS SignUpCount  
FROM   
    Customers  
GROUP BY   
    Month;

2. Which products have the highest sales?

SELECT   
    P.ProductName,  
    SUM(OD.Quantity) AS TotalSold  
FROM   
    OrderDetails OD  
JOIN   
    Products P ON OD.ProductID = P.ProductID  
GROUP BY   
    P.ProductName  
ORDER BY   
    TotalSold DESC  
LIMIT 5;

3. Calculate the total sales amount for each month.

SELECT   
    DATE\_FORMAT(OrderDate, '%Y-%m') AS Month,  
    SUM(TotalAmount) AS TotalSales  
FROM   
    Orders  
GROUP BY   
    Month;

4. Find the average rating for each product.

SELECT   
    P.ProductName,  
    AVG(R.Rating) AS AverageRating  
FROM   
    Reviews R  
JOIN   
    Products P ON R.ProductID = P.ProductID  
GROUP BY   
    P.ProductName;