

Problem statement-1

Design a database schema for an e-commerce platform with tables for products, categories, and orders. Write a SQL query to fetch the total sales revenue for each category, ordering the results by revenue in descending order

Database Schema for an E-commerce Platform

The schema consists of three main tables: Products, Categories, and Orders.

1. Categories Table

Stores information about product categories.

```
CREATE TABLE Categories (  
    category_id INT PRIMARY KEY AUTO_INCREMENT,  
    category_name VARCHAR(100) UNIQUE NOT NULL  
);
```

2. Products Table

Stores product details, linked to categories.

```
CREATE TABLE Products (  
    product_id INT PRIMARY KEY AUTO_INCREMENT,  
    product_name VARCHAR(255) NOT NULL,  
    category_id INT NOT NULL,  
    price DECIMAL(10,2) NOT NULL CHECK (price >= 0),  
    FOREIGN KEY (category_id) REFERENCES Categories(category_id) ON DELETE CASCADE  
);
```

3. Orders Table

Stores order details, linking products to purchases.

```
CREATE TABLE Orders (  
    order_id INT PRIMARY KEY AUTO_INCREMENT,  
    product_id INT NOT NULL,  
    quantity INT NOT NULL CHECK (quantity > 0),  
    order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (product_id) REFERENCES Products(product_id) ON DELETE CASCADE  
);
```

SQL Query to Fetch Total Sales Revenue for Each Category

To calculate total revenue (price * quantity) per category and order results by revenue in descending order:

```
SELECT c.category_name, SUM(p.price * o.quantity) AS total_revenue
FROM Orders o
JOIN Products p ON o.product_id = p.product_id
JOIN Categories c ON p.category_id = c.category_id
GROUP BY c.category_name
ORDER BY total_revenue DESC;
```

Query Explanation

1. Joins the necessary tables:
 - Orders → Products (to get price).
 - Products → Categories (to get category_name).
2. Calculates total revenue:
 - SUM(p.price * o.quantity) for each category.
3. Groups by category_name to get revenue per category.
4. Orders the results by total_revenue DESC to show the highest revenue first.

This schema ensures data integrity (with FOREIGN KEY constraints) and optimized query performance (by indexing category_id and product_id).