{c}odeBetter
Rise above the rest

Internship Task - RDBMS and SQL Task #3

The case will be based on a hypothetical business scenario involving a retail store's database. The database contains tables for Customers, Orders, Products, OrderDetails, and Payments.

Database Schema

Customers Table:

Column Name	Data Type	Description
customer_id	INT	Primary key
first_name	VARCHAR(100)	Customer's first name
last_name	VARCHAR(100)	Customer's last name
email	VARCHAR(100)	Customer's email address
phone	VARCHAR(20)	Customer's phone number
address	TEXT	Customer's address
join_date	DATE	Date customer joined

Products Table:

Column Name	Data Type	Description
product_id	INT	Primary key
product_name	VARCHAR(100)	Product name
category	VARCHAR(50)	Product category
price	DECIMAL(10,2)	Product price
stock_quantity	INT	Quantity available in stock

Orders Table:

Column Name	Data Type	Description
order_id	INT	Primary key
customer_id	INT	Foreign key (references Customers)
order_date	DATE	Date order was placed
total_amount	DECIMAL(10,2)	Total order amount
order_status	VARCHAR(20)	Order status (e.g., 'Pending', 'Shipped')

OrderDetails Table:

Column Name	Data Type	Description
order_detail_id	INT	Primary key
order_id	INT	Foreign key (references Orders)
product_id	INT	Foreign key (references Products)
quantity	INT	Quantity of product in the order
unit_price	DECIMAL(10,2)	Price per product

Payments Table:

Column Name	Data Type	Description
payment_id	INT	Primary key
order_id	INT	Foreign key (references Orders)
payment_date	DATE	Date the payment was made
payment_amount	DECIMAL(10,2)	Amount paid
payment_method	VARCHAR(20)	Payment method (e.g., 'Credit Card', 'PayPal')

Case Study: Retail Store Analysis

Background:

You are working as a database analyst for a retail store. The store has multiple customers who place orders for various products. The store wants to generate several reports to analyze customer behavior, sales trends, and payment methods.

SQL Queries for the Case Study

1. Find the Total Number of Orders for Each Customer

2. Find the Total Sales Amount for Each Product (Revenue per Product)

3. Find the Most Expensive Product Sold

4. Get the List of Customers Who Have Placed Orders in the Last 30 Days

5. Calculate the Total Amount Paid by Each Customer

6. Get the Number of Products Sold by Category

7. List All Orders That Are Pending (i.e., Orders that haven't been shipped yet)

8. Find the Average Order Value (Total Order Amount / Number of Orders)

9. List the Top 5 Customers Who Have Spent the Most Money

10. Find the Products That Have Never Been Sold

Task Summary:

This SQL case study covers a variety of business analysis queries, including customer insights, product sales, revenue, order status, and payment methods. The queries leverage basic SQL concepts like `JOIN`, `GROUP BY`, aggregate functions (`SUM()`, `AVG()`), and filtering with `WHERE`. The goal is to help analyse the store's performance, customer purchasing behaviour, and sales trends.

Each query is designed to answer a specific business question and can be adapted to meet additional reporting needs.