CERTIFIED DATA ANALYSTS

CAPSTONE PROJECT :
Customer Retention and Sales Optimization in Retail

NAZREEN AGOS BIN ABDUL LATIFF

SQL

To support the analysis for our capstone project, we designed a relational database named capstone, which includes four interrelated tables: Customers, Products, Campaigns, and Transactions. Each table captures essential information for understanding customer behavior, product performance, campaign effectiveness, and overall sales trends.

```
CREATE DATABASE capstone;

USE CAPSTONE;
```

1. Customers Table

```
CREATE TABLE Customers (
   CustomerID VARCHAR(255) PRIMARY KEY,
   Name_ VARCHAR(255),
   Gender VARCHAR(255),
   Age INT,
   Region VARCHAR(255),
   SignupDate DATE,
   LoyaltyScore INT
);
```

Purpose: This table stores demographic and behavioral information about each customer. It is crucial for understanding retention patterns, segmenting users, and evaluating loyalty.

2. Products Table

```
CREATE TABLE Products (
   ProductID VARCHAR(255) PRIMARY KEY,
   ProductName VARCHAR(255),
   Category VARCHAR(255),
   Price DECIMAL(10, 2),
   Supplier VARCHAR(255)
);
```

Purpose: Captures detailed information about each product available for sale. Essential for evaluating product-level sales performance and category trends.

3. Campaigns Table

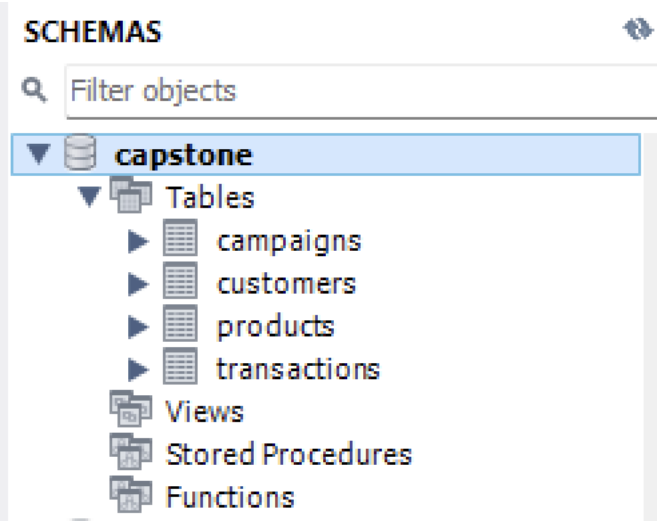
```
CREATE TABLE Campaigns (
   CampaignID VARCHAR(255) PRIMARY KEY,
   Channel_ VARCHAR(255),
   StartDate DATE,
   EndDate DATE,
   TargetAudience INT,
   Conversions INT
);
```

Purpose: Tracks details about marketing campaigns used to acquire or retain customers. Supports analysis of channel effectiveness and campaign ROI.

4. Transactions Table

```
CREATE TABLE Transactions (
   TransactionID VARCHAR(255) PRIMARY KEY,
   CustomerID VARCHAR(255),
   ProductID VARCHAR(255),
   Quantity INT,
   DiscountRate DECIMAL(10, 2),
   TotalPrice DECIMAL(10, 2),
   TransactionDate DATE
   );
```

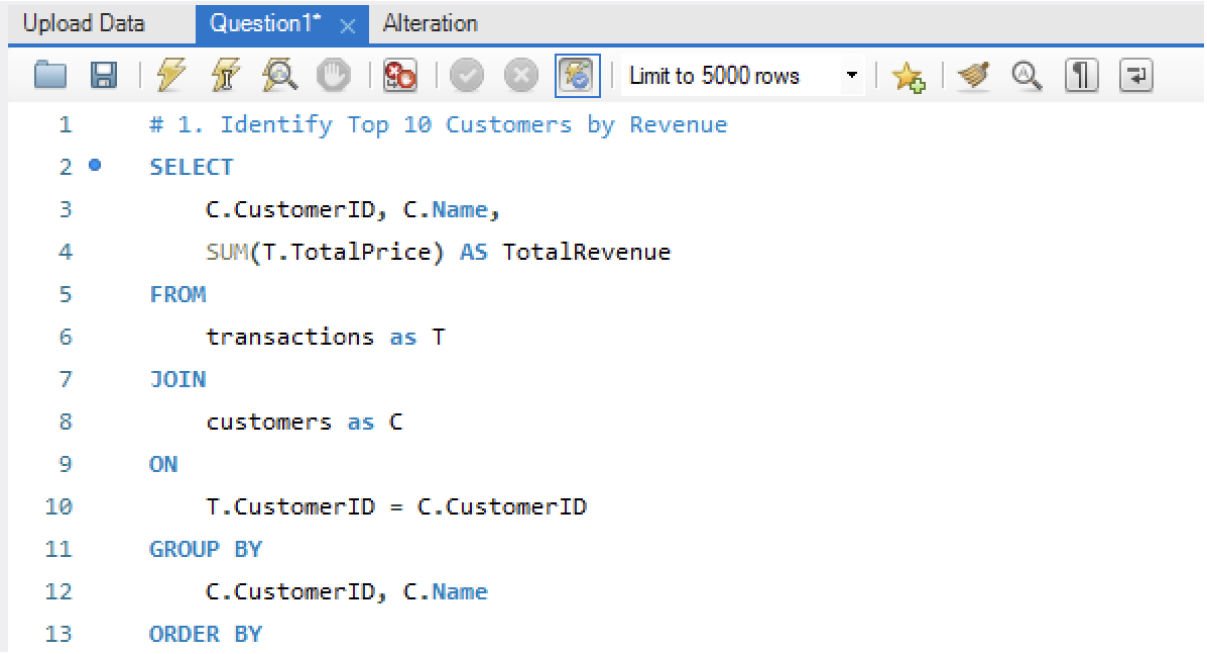
Purpose: Captures each purchase made by customers. It is the core table for sales analysis, customer lifetime value (CLV), and repeat purchase behavior.



Section: Data Analysis Using SQL

To derive meaningful insights from the capstone database, several SQL queries were written to answer key business questions related to **customer behavior**, **revenue trends**, and **customer value**. Below are the main queries used, along with their objectives and results interpretation strategies.

1. Identify Top 10 Customers by Revenue




```sql
# 1. Identify Top 10 Customers by Revenue
SELECT
    C.CustomerID, C.Name,
    SUM(T.TotalPrice) AS TotalRevenue
FROM
    transactions as T
JOIN
    customers as C
ON
    T.CustomerID = C.CustomerID
GROUP BY
    C.CustomerID, C.Name
ORDER BY
    TotalRevenue DESC
LIMIT 10;
```

CustomerID	Name	TotalRevenue
CUST0283	Laurie Franklin	13764.45
CUST0085	Aaron Black	13135.789999999999
CUST0055	Connie Soto	12836.38
CUST0154	Sandra Robinson	12612.08
CUST0046	Cody Conner	10680.16
CUST0070	John Nelson	10009.02
CUST0176	Wendy Carr	9160.16
CUST0232	Richard Fuller	9091.72
CUST0228	Christopher Gutierrez	8610.32
CUST0184	Jessica Warren DDS	8554.220000000001

Purpose: This query identifies the top 10 customers who have generated the highest revenue over the entire dataset. Understanding who your most valuable customers are can help in prioritizing loyalty programs, personalized marketing, or rewards.

Interpretation: Helps in forming a **High-Value Customer Segment**. Can be used to **monitor retention** of top-tier clients.

2. Find Repeat vs. One-Time Customers

```sql
SELECT
    CASE
        WHEN TransactionCount = 1 THEN 'One-Time Customer'
        ELSE 'Repeat Customer'
    END AS CustomerType,
    COUNT(*) AS NumberOfCustomers
FROM (
    SELECT
        CustomerID,
        COUNT(*) AS TransactionCount
    FROM
        Transactions
    GROUP BY
        CustomerID
) AS CustomerTransactionCounts
GROUP BY
    CASE
        WHEN TransactionCount = 1 THEN 'One-Time Customer'
        ELSE 'Repeat Customer'
    END;
```

CustomerType	NumberOfCustomers
Repeat Customer	290
One-Time Customer	7

Purpose: The queries categorize customers into **repeat** or **one-time buyers**. This is a core metric in customer retention studies, where the focus is on increasing the repeat buyer ratio.

Interpretation: A higher number of one-time customers may indicate **poor retention** or **weak follow-up marketing**.

3. Calculate Monthly Revenue Growth

Data Cleaning & Alteration: Fixing TransactionDate for Monthly Revenue Analysis.

During the monthly revenue trend analysis (Question 3), it was discovered that the TransactionDate column contained date values in string format (MM/DD/YYYY) rather than proper SQL DATE type. This caused issues when applying date functions like DATE_FORMAT() to group transactions by month and year. To resolve this, the date strings had to be **converted to true SQL DATE values**, which would allow accurate grouping, filtering, and sorting of transaction records based on time.

```
1     -- *Alteration due to question 3
2     USE capstone;
3
4     ALTER TABLE capstone.transactions ADD COLUMN CleanTransactionDate DATE;
5
6     SELECT* from transactions;
7
8     UPDATE capstone.transactions
9     SET CleanTransactionDate = STR_TO_DATE(TransactionDate, '%m/%d/%Y');
10
11    ALTER TABLE capstone.transactions DROP COLUMN TransactionDate;
12
13    ALTER TABLE capstone.transactions CHANGE COLUMN CleanTransactionDate TransactionDate DATE;
```

Summary, In preparation for monthly revenue analysis, the original TransactionDate column in the transactions table was found to be in string format (MM/DD/YYYY), which is incompatible with SQL's date functions. To resolve this, a multi-step transformation was applied: a new DATE-typed column was added, existing date strings were converted using STR_TO_DATE(), and the original column was safely dropped and replaced with the cleaned version. This data cleaning step was essential was essential for accurate time-series analysis and trend monitoring in our retail sales dataset.

```sql
SELECT
DATE_FORMAT(TransactionDate, '%Y-%m') AS month,
    SUM(TotalPrice) AS monthly_revenue
FROM
    transactions
GROUP BY
    month
ORDER BY
    month;
```

month	monthly_revenue
2024-04	41517.57
2024-05	94915.26999999997
2024-06	98410.12999999999
2024-07	101243.09000000001
2024-08	72949.84000000001
2024-09	81195.94999999997
2024-10	103340.52999999998
2024-11	85899.80999999997
2024-12	100367.87
2025-01	97029.82999999996
2025-02	85406.45000000001
2025-03	118256.16
2025-04	79069.59

Purpose: Tracks revenue performance over time. This time-series trend is useful for evaluating the **growth trajectory** of the business and detecting **seasonal fluctuations**.

Interpretation:

- Identify peak and off-peak months.
- Use as input for **forecasting** or campaign planning.
- Helps correlate revenue spikes with **campaign periods** (if cross-referenced).

4. Calculate Customer Lifetime Value (LTV)

```sql
# 4. Join customer and transaction data to get customer LTV.

SELECT
    c.CustomerID,
    c.Name,
    SUM(t.TotalPrice) AS lifetime_value
FROM
    customers c
JOIN
    transactions t ON c.CustomerID = t.CustomerID
GROUP BY
    c.CustomerID, c.Name
ORDER BY
    lifetime_value DESC;
```

month	monthly_revenue
2024-04	41517.57
2024-05	94915.26999999997
2024-06	98410.12999999999
2024-07	101243.09000000001
2024-08	72949.84000000001
2024-09	81195.94999999997
2024-10	103340.52999999998
2024-11	85899.80999999997
2024-12	100367.87
2025-01	97029.82999999996
2025-02	85406.45000000001
2025-03	118256.16
2025-04	79069.59

Purpose: This query calculates each customer's **Lifetime Value (LTV)**, the total amount spent by each customer across all purchases. It is a key metric in determining long-term business sustainability.

Interpretation:

- Customers with high LTV are more **profitable** over time.
- LTV helps in **budgeting customer acquisition costs** and justifying retention strategies.