Project Theme: Retail Business Analysis

This project analyses sales, customer behaviour, and product performance for a fictitious e-commerce company.

Database Name: RetailAnalytics

Tables

1. **Customers**
    - Customer details and demographics.

Column Name	Data Type	Description
CustomerID	INT	Primary Key
FirstName	VARCHAR(50)	Customer's first name
LastName	VARCHAR(50)	Customer's last name
Gender	VARCHAR(10)	Gender (Male/Female/Other)
DateOfBirth	DATE	Customer's date of birth
City	VARCHAR(50)	City of residence
State	VARCHAR(50)	State of residence
SignupDate	DATE	Date of customer signup

```
CREATE TABLE Customers (
    CustomerID INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Gender VARCHAR(10),
    DateOfBirth DATE,
    City VARCHAR(50),
    State VARCHAR(50),
    SignupDate DATE NOT NULL
);
```

---

2. **Products**
    - Product catalog with pricing and category details.

Column Name	Data Type	Description
ProductID	INT	Primary Key
ProductName	VARCHAR(100)	Name of the product
Category	VARCHAR(50)	Product category (e.g., Electronics, Apparel)
Price	DECIMAL(10,2)	Price of the product
StockQuantity	INT	Number of items in stock

```
CREATE TABLE Products (
    ProductID INT AUTO_INCREMENT PRIMARY KEY,
    ProductName VARCHAR(100) NOT NULL,
    Category VARCHAR(50),
    Price DECIMAL(10, 2) NOT NULL,
    StockQuantity INT NOT NULL
);
```

---

3. **Orders**
    - Order transactions.

Column Name	Data Type	Description
OrderID	INT	Primary Key
CustomerID	INT	Foreign Key (Customers)
OrderDate	DATE	Date of the order

Column Name	Data Type	Description
OrderStatus	VARCHAR(20)	Status of the order (Completed, Pending, Canceled)

```
CREATE TABLE Orders (
    OrderID INT AUTO_INCREMENT PRIMARY KEY,
    CustomerID INT NOT NULL,
    OrderDate DATE NOT NULL,
    OrderStatus VARCHAR(20),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

---

4. **OrderDetails**
    - Details of products purchased in each order.

Column Name	Data Type	Description
OrderDetailID	INT	Primary Key
OrderID	INT	Foreign Key (Orders)
ProductID	INT	Foreign Key (Products)
Quantity	INT	Number of units purchased
TotalPrice	DECIMAL(10,2)	Total price for the line item

```
CREATE TABLE OrderDetails (
    OrderDetailID INT AUTO_INCREMENT PRIMARY KEY,
    OrderID INT NOT NULL,
    ProductID INT NOT NULL,
    Quantity INT NOT NULL,
    TotalPrice DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
```

```
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```

Sample Data

Customers

```
INSERT INTO Customers (CustomerID, FirstName, LastName, Gender, DateOfBirth, City, State, SignupDate)
VALUES
(1, 'Alice', 'Johnson', 'Female', '1990-05-15', 'New York', 'NY', '2021-01-10'),
(2, 'Bob', 'Smith', 'Male', '1985-07-20', 'Los Angeles', 'CA', '2021-02-12'),
(3, 'Charlie', 'Brown', 'Male', '1992-03-25', 'Chicago', 'IL', '2021-03-15'),
(4, 'Diana', 'Prince', 'Female', '1988-11-11', 'Miami', 'FL', '2021-04-18'),
(5, 'Edward', 'Elric', 'Male', '1995-01-01', 'Austin', 'TX', '2021-05-20');
```

Products

```
INSERT INTO Products (ProductID, ProductName, Category, Price, StockQuantity)
VALUES
(1, 'Smartphone', 'Electronics', 699.99, 50),
(2, 'Laptop', 'Electronics', 1299.99, 30),
(3, 'Jeans', 'Apparel', 49.99, 100),
(4, 'Headphones', 'Electronics', 199.99, 80),
(5, 'Shoes', 'Apparel', 79.99, 60);
```

Orders

```
INSERT INTO Orders (OrderID, CustomerID, OrderDate, OrderStatus)
VALUES
(1, 1, '2022-01-15', 'Completed'),
(2, 2, '2022-02-10', 'Completed'),
(3, 3, '2022-03-05', 'Pending'),
(4, 4, '2022-03-15', 'Completed'),
```

(5, 5, '2022-04-01', 'Canceled');

OrderDetails

INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity, TotalPrice)

VALUES

(1, 1, 1, 2, 1399.98),

(2, 1, 3, 1, 49.99),

(3, 2, 2, 1, 1299.99),

(4, 4, 4, 3, 599.97),

(5, 5, 5, 2, 159.98);

Analysis Questions

1. Customer Insights

- Which state has the highest number of customers?
- What is the average age of customers?

2. Sales Performance

- What is the total revenue generated?
- Which product category generates the highest revenue?

3. Product Analysis

- Which product has the highest number of units sold?
- What is the average order value?

4. Order Analysis

- What percentage of orders are completed?
- How many customers placed more than one order?

Solutions (SQL Queries)

1. Which state has the highest number of customers?

SELECT

State, COUNT(*) AS CustomerCount

FROM

```
    Customers
GROUP BY State
ORDER BY CustomerCount DESC
LIMIT 1;
```

2. Total revenue generated.

```
SELECT
    SUM(TotalPrice) AS TotalRevenue
FROM
    OrderDetails;
```

3. Product category with the highest revenue.

```
SELECT
    p.Category, SUM(od.TotalPrice) AS CategoryRevenue
FROM
    OrderDetails od
        JOIN
    Products p ON od.ProductID = p.ProductID
GROUP BY p.Category
ORDER BY CategoryRevenue DESC
LIMIT 1;
```

4. Percentage of completed orders.

```
SELECT
    (COUNT(CASE
        WHEN OrderStatus = 'Completed' THEN 1
    END) * 100.0 / COUNT(*)) AS CompletedPercentage
FROM
    Orders;
```

5. Average order value.

```
SELECT
    AVG(TotalPrice) AS AverageOrderValue
FROM
    OrderDetails;
```

Expected Outputs

- **State with highest customers:** NY.
- **Total revenue generated:** $2,749.96.
- **Category with highest revenue:** Electronics.
- **Percentage of completed orders:** 66.67%.
- **Average order value:** Varies based on additional data.

Advanced Analysis Questions

1. Lifetime Value (LTV) of Each Customer

```
SELECT
    c.CustomerID,
    CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,
    SUM(od.TotalPrice) AS LifetimeValue
FROM
    Customers c
        JOIN
    Orders o ON c.CustomerID = o.CustomerID
        JOIN
    OrderDetails od ON o.OrderID = od.OrderID
WHERE
    o.OrderStatus = 'Completed'
GROUP BY c.CustomerID , c.FirstName , c.LastName
ORDER BY LifetimeValue DESC
```

```
LIMIT 5;
```

---

2. Customers with No Purchases in the Last 6 Months

```
SELECT
    c.CustomerID,
    CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,
    SUM(od.TotalPrice) AS LifetimeValue
FROM
    Customers c
        JOIN
    Orders o ON c.CustomerID = o.CustomerID
        JOIN
    OrderDetails od ON o.OrderID = od.OrderID
WHERE
    o.OrderStatus = 'Completed'
GROUP BY c.CustomerID , c.FirstName , c.LastName
ORDER BY LifetimeValue DESC
LIMIT 5;
```

---

3. Lowest Inventory Turnover

```
SELECT
    p.ProductID,
    p.ProductName,
    SUM(od.Quantity) AS UnitsSold,
    p.StockQuantity AS CurrentStock,
    (SUM(od.Quantity) / NULLIF(p.StockQuantity + SUM(od.Quantity), 0)) AS
InventoryTurnover
FROM
    Products p
        LEFT JOIN
```

```
    OrderDetails od ON p.ProductID = od.ProductID
GROUP BY p.ProductID , p.ProductName , p.StockQuantity
ORDER BY InventoryTurnover ASC
LIMIT 1;
```

How This Enhances the Portfolio

- Demonstrates proficiency in **SQL joins**, **window functions**, and **complex aggregations**.
- Shows the ability to derive actionable business insights from raw data.
- Adds a real-world element with advanced metrics like **LTV**, **cohort analysis**, and **inventory turnover**.