Retail Sales & Customer Performance Insights

This project is an end-to-end data analysis solution designed to help a retail business understand its sales performance and customer demographics.

Mr. Harshraj V Suryawanshi

12/18/25

Data Analyst

***Project Overview**: Starting with a raw dataset containing data quality issues (inconsistent text,and missing values), I used **MS SQL Server** to clean and standardize the data. I then performed Exploratory Data Analysis (EDA) using SQL to uncover trends. Finally, I built an interactive **Power BI** dashboard to visualize KPIs and provide actionable business recommendations.*

❓ Business Problem

The stakeholders wanted to answer four key questions to improve profitability:

1. **Sales Trends:** How is revenue performing month-over-month?
2. **Customer Value:** Who are the high-value customers, and does the Loyalty Program actually drive revenue?
3. **Product Performance:** Which categories are driving the business, and which are lagging?
4. **Operational Insights:** How can inventory and marketing be optimized based on purchasing behaviour?

--Data Cleaning--

--Fix Inconsistent 'gender' values

```
UPDATE customer_info
    SET gender = CASE
        WHEN LOWER(gender) LIKE 'fem%' THEN 'Female'
        WHEN LOWER(gender) LIKE 'mal%' THEN 'Male'
    ELSE 'Other'
        END;
```

---Fix Inconsistent 'loyalty_tier' casing

```
UPDATE customer_info
    SET loyalty_tier = UPPER(TRIM(loyalty_tier));
```

---Set NULL regions to 'Unknown'

```
UPDATE customer_info
    SET region = 'Unknown'
    WHERE region IS NULL;
```

--Handle Missing Financial Data

```
UPDATE sales_data
        SET unit_price = 0.00
        WHERE unit_price IS NULL;
UPDATE sales_data
        SET quantity = 1
    WHERE quantity IS NULL;
```

--VERIFICATION

```
SELECT DISTINCT gender FROM customer_info;
SELECT DISTINCT loyalty_tier FROM customer_info;
```

--EDA (Exploratory Data Analysis)--

--Sales Performance Analysis.

1. Total Revenue: What is the total revenue generated across all transactions?

```
SELECT
    SUM(UNIT_PRICE) AS TOTAL_REVENUE
FROM SALES_DATA;
```

2. Monthly Sales Trend: Calculate the total revenue generated per month to see seasonal trends.

```
SELECT
        MONTH(ORDER_DATE)AS MONTH_NO,
        DATENAME(MONTH, ORDER_DATE) AS MONTH,
        SUM(UNIT_PRICE) AS MONTHLY_REVENUE
    FROM SALES_DATA
        GROUP BY MONTH(ORDER_DATE) , DATENAME(MONTH, ORDER_DATE)
        ORDER BY  MONTH(ORDER_DATE);
```

3. Quarterly Revenue Growth: What is the total revenue per quarter

```
SELECT
    DATEPART(QUARTER,order_date)
                AS QUARTER,
```

```
    SUM(UNIT_PRICE)
FROM SALES_DATA
                GROUP BY  DATEPART(QUARTER,order_date)
               ORDER BY  DATEPART(QUARTER,order_date);
```

4. **Sales by Region: Which region has generated the highest total sales revenue?**

```
    SELECT
        REGION,
       SUM(UNIT_PRICE) AS Regional_Revenue
FROM SALES_DATA
                GROUP BY REGION
                ORDER BY SUM(UNIT_PRICE)
                        DESC ;
```

-- Customer Insights

5. Customer Distribution by Region: How many unique customers are there in each region?

```
    SELECT
        REGION,
       COUNT(DISTINCT CUSTOMER_ID)
                AS No_of_Customer
FROM CUSTOMER_INFO
                GROUP BY REGION
               ORDER BY COUNT(DISTINCT CUSTOMER_ID)
                        DESC ;
```

6. Top 5 Customers by Revenue: Who are the top 5 customers who have spent the most money?

```
SELECT TOP 5
        C.customer_id,
         C.EMAIL,
```

```
        C.GENDER,
        C.REGION,
       C.LOYALTY_TIER,
     SUM(S.UNIT_PRICE)  AS TOTAL_SPEND
FROM customer_info C
        JOIN
   SALES_DATA S
          ON C.CUSTOMER_ID = S.CUSTOMER_ID
            GROUP BY C.EMAIL,C.GENDER,C.REGION,C.LOYALTY_TIER,C.customer_id
             ORDER BY SUM(S.UNIT_PRICE) DESC  ;
```

7. Purchase Frequency: What is the average number of transactions per customer?

```
 WITH  CUSTOMER_TRANSACTION_COUNT
 AS (
      SELECT CUSTOMER_ID, COUNT(DISTINCT ORDER_ID) AS TRANSACTIONS
FROM SALES_DATA
     GROUP BY CUSTOMER_ID
          )
SELECT
   CUSTOMER_ID,
   AVG(TRANSACTIONS) AS Avg_Transactions_Per_Customer
FROM CUSTOMER_TRANSACTION_COUNT
  GROUP BY CUSTOMER_ID;
```

8. Revenue by Loyalty Tier: Which loyalty tier contributes the most to total revenue?

```
 SELECT
      C.LOYALTY_TIER,
    SUM(S.UNIT_PRICE) AS REVENUE
```

```
FROM customer_info C
    JOIN
SALES_DATA S
        ON
    C.CUSTOMER_ID = S.CUSTOMER_ID
            GROUP BY C.LOYALTY_TIER
            ORDER BY SUM(S.UNIT_PRICE)
    DESC ;
```

9. Gender-Based Spending: What is the average spend per transaction for each gender group?

```
SELECT
    C.GENDER,
    ROUND(AVG(S.unit_price),2)
            AS AVG_TRANSACTION
FROM CUSTOMER_INFO C
            JOIN
        SALES_DATA S
ON  C.customer_id = S.customer_id
                GROUP BY C.GENDER
                ORDER BY ROUND(AVG(S.unit_price),2)
            DESC ;
```

-- Product Analysis

10. Top Selling Products (Quantity): Which are the top 10 products based on the total quantity sold?

```
select TOP 10
```

```
        p.product_id,
        p.product_name,
        sum(s.quantity) AS QUANTITY_SOLD
from product_info p
        join
    sales_data s
on  p.product_id = s.product_id
        group by p.product_id,p.product_name
        ORDER BY sum(s.quantity) DESC;
```

11. Least Popular Products: Which products have been sold less than least times in total?

```
    select top 1
        p.product_id,
        p.product_name,
        sum(s.quantitY)  AS QUANTITY_SOLD
from product_info p
        join
    sales_data s
        on  p.product_id = s.product_id
            group by  p.product_id, p.product_name
            order by sum(s.quantity);
```

12. Revenue by Product Category: Which product category generates the highest revenue?

```
select
    p.category,
    sum(s.unit_price) AS Revenue
from product_info p
    join
sales_data s
on  p.product_id = s.product_id
```

```
group by p.category
order by sum(s.unit_price) desc;
```

13. Product Pricing Strategy: What is the average price of products in each category?

```
SELECT
    category,
    ROUND(AVG(base_price),2) AS Avg_Price
FROM product_info
    GROUP BY category
    ORDER BY Avg_Price DESC;
```

14. Cross-Selling Potential: Which products are most frequently bought together in the same transaction?

```
SELECT TOP 10
    s1.product_id AS Product_A,
    s2.product_id AS Product_B,
    COUNT(*) AS Times_Bought_Together
FROM sales_data s1
    JOIN
    sales_data s2
        ON s1.customer_id = s2.customer_id      -- Same Customer
        AND s1.order_date = s2.order_date -- Same Day
        AND s1.product_id <> s2.product_id      -- Different Products
            GROUP BY s1.product_id, s2.product_id
            ORDER BY Times_Bought_Together DESC;
```