CENTRALCART STORE SQL QUERIES

A. <mark>KPI's</mark>

1. **Total Revenue**

   SELECT SUM(Quantity * Price_After_Discount) AS Total_Revenue FROM Central_Store

	Total_Revenue
1	31538063.6663094

2. **Average Order Value**

   SELECT (SUM(Price) / COUNT(Order_ID)) AS Avg_order_Value
   FROM Central_store

	Avg_order_Value
1	1844.80174863522

3. **Revenue per Unit**

   SELECT
      ROUND(
         SUM(Quantity * Price_After_Discount) * 1.0
         / NULLIF(SUM(Quantity), 0),
         2
) AS Revenue_Per_Unit
   FROM Central_store;

	Revenue_Per_Unit
1	1670.09

4. **Total Orders**

   SELECT COUNT(DISTINCT Order_ID) AS Total_Orders FROM Central_store

	Total_Orders
1	5444

   Repeat Customer Rate

   With CustomerOrders As (

```
Select
    Customer_ID,
    Count(Distinct Order_ID) As Order_Count
From Central_store
Group by Customer_ID
)
Select
    Round(
        sum(Case When Order_Count > 1 THEN 1 ELSE 0 END) * 100.0
        / Count(*),
        2
    ) As Repeat_Customer_Rate_Percent
From CustomerOrders;
```

	Repeat_Customer_Rate_Percent
1	94.170000000000

Delay Rate

```
SELECT
    ROUND(
        SUM(CASE WHEN Delivery_Status = 'Delayed' THEN 1 ELSE 0 END) * 100.0
        / COUNT(*),
        2
    ) AS Delay_Rate_Percent
FROM Central_Store
```

	Delay_Rate_Percent
1	6.690000000000

B. Which products generated the highest revenue & which products are underperforming.

```
--Best Products--
SELECT Top 5
    Product_ID, Product_Name,
    SUM(Quantity * Price_After_Discount) AS Total_Revenue
FROM Central_store
GROUP BY Product_ID, Product_Name
```

```sql
ORDER BY Total_Revenue DESC;
```

	Product_ID	Product_Name	Total_Revenue
1	889231	Office File Folder	2990827.93823242
2	487193	Mixer Grinder	2534912.60302734
3	852600	USB Flash Drive 64GB	2092346.72387695
4	444931	Men Denim Jeans	2063315.50280762
5	759253	Extension Board	2041028.76904297

```sql
-- Worst products--
SELECT TOP 5
    Product_ID,
    Product_Name,
    SUM(Quantity *Price_After_Discount) AS Total_Revenue
FROM Central_store
GROUP BY Product_ID, Product_Name
ORDER BY Total_Revenue ASC;
```

	Product_ID	Product_Name	Total_Revenue
1	546906	Hand Sanitizer 500ml	12636.219997406
2	737483	Conditioner 340ml	23754.7199859619
3	637144	LED Bulb 12W	36200.7698364258
4	528646	Kids Casual Wear	63617.0696563721
5	349087	Dishwash Liquid 1L	80126.3598327637

c. Do discounts actually increase sales volume, or are they reducing profitability?

```sql
SELECT
    CASE
        WHEN Price_After_Discount < Price THEN 'Discounted'
        ELSE 'Non-Discounted'
    END AS Sale_Type,

    SUM(Quantity) AS Units_Sold,

    SUM(Quantity * Price_After_Discount) AS Net_Revenue,

    ROUND(
        SUM(Quantity * Price_After_Discount) * 1.0
        / NULLIF(SUM(Quantity), 0),
        2
    ) AS Revenue_Per_Unit
FROM Central_store
GROUP BY
    CASE
        WHEN Price_after_Discount < Price THEN 'Discounted'
```

```
        ELSE 'Non-Discounted'
    END;
```

	Sale_Type	Units_Sold	Net_Revenue	Revenue_Per_Unit
1	Discounted	15102	24614038.2889595	1629.85
2	Non-Discounted	3782	6924025.37734985	1830.78

D. <mark>**Analyze customer purchase frequency and evaluate the revenue contribution of repeat purchasers by gender.**</mark>

```sql
WITH CustomerStats AS (
  SELECT
    Customer_ID,
    Gender,
    COUNT(Product_ID) AS Purchase_Count,
    SUM(Quantity * Price_After_Discount) AS Revenue
  FROM Central_store
  GROUP BY Customer_ID, Gender
),
RepeatCustomers AS (
  SELECT *
  FROM CustomerStats
  WHERE Purchase_Count > 1
)
SELECT
  Gender,
  COUNT(Customer_ID) AS Repeat_Customers,
  SUM(Revenue) AS Repeat_Customer_Revenue,
  ROUND(
    SUM(Revenue) * 100.0 /
    (SELECT SUM(Revenue) FROM RepeatCustomers),
    2
  ) AS Revenue_Percentage
FROM RepeatCustomers
GROUP BY Gender
ORDER BY Repeat_Customer_Revenue DESC;
```

	Gender	Repeat_Customers	Repeat_Customer_Revenue	Revenue_Percentage
1	Female	875	16283825.5580063	57.51
2	Male	677	12033060.4888344	42.49

E. Which cities and shipping methods experience the highest proportion of delayed shipments, and how do these delays relate to return rates?

```sql
WITH CityDelay AS (
    SELECT
        Cities,
        COUNT(*) AS Total_Orders,
        SUM(CASE WHEN Delivery_Status = 'Delayed' THEN 1 ELSE 0 END) AS
Delayed_Orders,
        SUM(CASE WHEN Delivery_Status = 'Delayed' THEN 1 ELSE 0 END) * 100.0 /
COUNT(*) AS Delay_Rate
    FROM Central_store
    GROUP BY Cities
),
TopDelayedCities AS (
    SELECT TOP 10 Cities
    FROM CityDelay
    ORDER BY Delay_Rate DESC
)
SELECT
    c.Cities,
    c.Shipping_Method,

    COUNT(*) AS Total_Orders,
    SUM(CASE WHEN c.Delivery_Status = 'Delayed' THEN 1 ELSE 0 END) AS
Delayed_Orders,

    ROUND(
        SUM(CASE WHEN c.Delivery_Status = 'Delayed' THEN 1 ELSE 0 END) * 100.0
        / COUNT(*),
        2
    ) AS Delay_Rate_Percent
FROM Central_store c
JOIN TopDelayedCities t
    ON c.Cities = t.Cities
GROUP BY c.Cities, c.Shipping_Method
ORDER BY Delayed_orders DESC;
;
```

	Cities	Shipping_Method	Total_Orders	Delayed_Orders	Delay_Rate_Percent
1	Ahmedabad	Standard	306	29	9.480000000000
2	Kolkata	Standard	288	21	7.290000000000
3	Bengaluru	Standard	260	20	7.690000000000
4	Gurgaon	Standard	274	20	7.300000000000
5	Mumbai	Standard	310	19	6.130000000000
6	Chennai	Standard	313	19	6.070000000000
7	Hyderabad	Standard	314	18	5.730000000000
8	Jaipur	Standard	271	17	6.270000000000
9	Noida	Standard	263	15	5.700000000000
10	Pune	Standard	297	15	5.050000000000
11	Mumbai	Express	153	13	8.500000000000
12	Pune	Express	116	11	9.480000000000
13	Bengaluru	Express	126	10	7.940000000000
14	Hyderabad	Express	142	10	7.040000000000
15	Ahmedabad	Express	143	10	6.990000000000
16	Kolkata	Express	140	9	6.430000000000

F. Which months & quarters generate the highest sales and revenue.

```sql
WITH FixedDates AS (
    SELECT
        *,
        CONVERT(date, Order_Date, 105) AS Order_Date_Fixed
    FROM Central_store
)
    SELECT
        YEAR(Order_Date_Fixed) as Sales_Year,
        CONCAT('Q', DATEPART(QUARTER, Order_Date_Fixed)) AS Sales_Quarter,
SUM(Quantity) AS Total_Units_Sold,
        SUM(Quantity * Price_After_Discount) AS Total_Revenue
    FROM FixedDates
    GROUP BY
        YEAR(Order_Date_Fixed),
        DATEPART(QUARTER, Order_Date_Fixed)
ORDER BY
Sales_Year,
DATEPART(QUARTER, Order_Date_Fixed);
```

	Sales_Year	Sales_Quarter	Total_Units_Sold	Total_Revenue
1	2023	Q1	2921	4894174.34288406
2	2023	Q2	3068	4957479.33655167
3	2023	Q3	3066	5305116.69826126
4	2023	Q4	3215	5472677.37485886
5	2024	Q1	3131	5080439.90517426
6	2024	Q2	2963	4923996.61088943
7	2024	Q3	520	904179.397689819

G. Which cities generate the highest revenue & order volume.

```
SELECT
    Cities,
    COUNT(DISTINCT Order_ID) AS Total_Orders,
    SUM(Quantity) AS Total_Units_Sold,
    SUM(Quantity * Price_After_Discount) AS Total_Revenue
FROM Central_store
GROUP BY Cities
ORDER BY
    Total_Revenue DESC,
    Total_Orders DESC;
```

	Cities	Total_Orders	Total_Units_Sold	Total_Revenue
1	Hyderabad	527	1856	3185284.19956589
2	Mumbai	544	1865	3101701.17600632
3	Delhi	524	1783	2940223.97064972
4	Chennai	502	1754	2885092.83750916
5	Pune	475	1696	2883494.25236511
6	Noida	448	1625	2853045.34451294
7	Gurgaon	488	1688	2842231.62470245
8	Ahmedabad	509	1703	2780208.94720078
9	Jaipur	474	1631	2779740.15107727
10	Kolkata	499	1729	2767627.46075058
11	Bengaluru	454	1554	2519413.70196915

H. Which categories are driven by repeat purchases versus one-time buyers?

```
With CustomerCategoryOrders as (
  Select
        Customer_ID,
        Category,
        Count(Distinct Order_ID) as Order_Count
```

```sql
    From Central_Store
    Group By Customer_ID, Category
),
CategoryBuyerType as (
  Select
    Category,
    Case
      When Order_Count > 1 Then 'Repeat Buyer'
      Else 'One-time Buyer'
    End as Buyer_type
From CustomerCategoryOrders
)
Select
    Category,
    Buyer_Type,
    Count(*) as Customer_Count
From CategoryBuyerType
Group By Category, Buyer_Type
Order By Category, Customer_Count Desc;
```

	Category	Buyer_Type	Customer_Count
1	Bathroom items	One-time Buyer	322
2	Bathroom items	Repeat Buyer	82
3	Beverages & Drinks	One-time Buyer	116
4	Beverages & Drinks	Repeat Buyer	4
5	Clothes	One-time Buyer	438
6	Clothes	Repeat Buyer	187
7	Electronics	One-time Buyer	424
8	Electronics	Repeat Buyer	394
9	Hardware	One-time Buyer	6
10	Hygine & Daily use	One-time Buyer	232
11	Hygine & Daily use	Repeat Buyer	46
12	Kitchen Items	One-time Buyer	288
13	Kitchen Items	Repeat Buyer	57
14	Packaged Food	One-time Buyer	369
15	Packaged Food	Repeat Buyer	262