

**Question 1: Find the number of orders that have small, medium or large order value (small:0-10 dollars, medium:10-20 dollars, large:20+)**

Query :**SELECT**

**CASE**

**WHEN** SALES\_VALUE **BETWEEN** 0 **AND** 10 **THEN** 'Small'

**WHEN** SALES\_VALUE **BETWEEN** 10 **AND** 20 **THEN** 'Medium'

**ELSE** 'Large'

**END AS** Order\_Value\_Category,

**COUNT**(\*) **AS** Number\_of\_Orders

**FROM** `PC\_Dataset\_2024.transaction`

**GROUP BY** Order\_Value\_Category;

Result:

Order_Value_Category ▾	Number_of_Orders
Small	1259081
Large	12536
Medium	26869

Insight:Understanding the distribution of order values can help in pricing strategy and promotions.Here largest number of orders are coming from small category

**Question 2: Find the number of orders that are small, medium, or large order value (small: 0-5 dollars, medium: 5-10 dollars, large: 10+).**

Query:**SELECT**

**CASE**

**WHEN** SALES\_VALUE **BETWEEN** 0 **AND** 5 **THEN** 'Small'

**WHEN** SALES\_VALUE **BETWEEN** 5 **AND** 10 **THEN** 'Medium'

**ELSE** 'Large'

**END AS** Order\_Value\_Category,

**COUNT**(\*) **AS** Number\_of\_Orders

**FROM** PC\_Dataset\_2024.transaction

**GROUP BY** Order\_Value\_Category;

Result:

Row	Order_Value_Category ▾	Number_of_Orders
1	Small	1145982
2	Medium	113099
3	Large	39405

Similar to Q1, but with a different segmentation to understand how smaller transactions contribute to overall sales.

**Question3: Find top 3 stores with the highest foot traffic for each week (Foot traffic: number of customers transacting).**

```
Query: WITH Store_Traffic AS (
  SELECT
    STORE_ID,
    WEEK_NO,
    COUNT(DISTINCT household_key) AS Foot_Traffic
  FROM PC_Dataset_2024.transaction
  GROUP BY STORE_ID, WEEK_NO
)
SELECT
  STORE_ID,
  WEEK_NO,
  Foot_Traffic
FROM (
  SELECT
    STORE_ID,
    WEEK_NO,
    Foot_Traffic,
    ROW_NUMBER() OVER (PARTITION BY WEEK_NO ORDER BY Foot_Traffic DESC) AS Rank
  FROM Store_Traffic
) AS RankedTraffic
WHERE Rank <= 3;
```

Result :

Row	STORE_ID	WEEK_NO	Foot_Traffic
1	367	58	40
2	343	58	28
3	381	58	28
4	367	100	40
5	381	100	33
6	424	100	29
7	367	40	38
8	32004	40	28
9	343	40	28
10	367	36	42

**Insights :** Based on the data downloaded , 367,343,381 are the most busy stores for the brand and we can plan before for the inventory for the stores and staff so that the customer experience doesn't impact

**Question: 4 Create a basic customer profiling with the first, last visit, number of visits, average money spent per visit, and total money spent, ordered by the highest average money.**

Query: `SELECT  
household_key,  
MIN(DAY) AS First_Visit,  
MAX(DAY) AS Last_Visit,  
COUNT(BASKET_ID) AS Number_of_Visits,  
AVG(SALES_VALUE) AS Avg_Money_Spent_Per_Visit,  
SUM(SALES_VALUE) AS Total_Money_Spent  
FROM PC_Dataset_2024.transaction  
GROUP BY household_key  
ORDER BY Avg_Money_Spent_Per_Visit DESC;`

Result:

Row	household_key	First_Visit	Last_Visit	Number_of_Visits	Avg_Money_Spent_P	Total_Money_Sp
1	1730	34	707	99	16.73494949494...	1656.760000000
2	1727	109	118	9	12.72333333333...	114.
3	2163	51	674	21	10.53904761904...	221.
4	1339	52	701	18	10.41833333333...	187.
5	991	44	665	44	10.26363636363...	451
6	2219	80	702	32	10.05187499999...	321.6600000000
7	2428	67	702	18	10.0	180.0000000000
8	755	36	709	576	9.481840277777...	5461.539999999
9	1023	107	710	2202	8.583601271571...	18901.09000000

Insights: Customer profiling helps in targeted marketing and customer retention strategies.

>Household key 1730 has the highest Average per visit spent

>Household key 1023 has the highest total money spent

**Question 5 : Single customer analysis selecting the most spending customer for whom we have demographic information (because not all customers in the transaction data are present in the demographic table).**

Query: `WITH Total_Spending AS (  
SELECT  
household_key,  
SUM(SALES_VALUE) AS Total_Spent  
FROM `PC_Dataset_2024.transaction`  
GROUP BY household_key  
)  
Top_Spenders AS (  
SELECT  
household_key,  
Total_Spent`

```

FROM Total_Spending
ORDER BY Total_Spent DESC
LIMIT 10
)
SELECT
    d.*,
    ts.Total_Spent
FROM `PC_Dataset_2024.demographics` AS d
JOIN Top_Spenders AS ts
ON d.household_key = ts.household_key;

```

Result:

Row	AGE_DESC	MARITAL_STATUS_CODE	INCOME_DESC	HOMEOWNER_DESC
1	45-54	A	25-34K	Homeowner
2	35-44	A	35-49K	Homeowner
3	45-54	A	125-149K	Homeowner
4	45-54	A	125-149K	Homeowner
5	35-44	A	150-174K	Homeowner
6	45-54	U	175-199K	Homeowner
7	35-44	B	Under 15K	Homeowner

Row	HOUSEHOLD_SIZE_DESC	KID_CATEGORY_DESC	household_key	Total_Spent
1	5+	3+	718	9577.629999999...
2	3	1	1430	10147.209999999...
3	3	1	1453	10720.719999999...
4	5+	3+	1609	13804.379999999...
5	3	1	400	9481.189999999...
6	1	None/Unknown	2322	11934.659999999...
7	1	None/Unknown	1653	9519.929999999...

Insight: Household key 1609 is the highest spender in the list

	AGE_DESC	MARITAL_STATUS_CODE	INCOME_DESC	HOMEOWNER_DESC	HH_COMP_DESC	HOUSEHOLD_SIZE_DESC	KID_CATEGORY_DESC	household_key	Total_Spent
2	45-54	A	125-149K	Homeowner	2 Adults Kids	5+	3+	1609	13804.38

1. Kids category for the guy is 3+
2. Recommendation is to make this guy keep engaging on the stores by giving loyalty offers

**Question 6: Find products (SUB\_COMMODITY\_DESC) that are most frequently bought together and the count of each combination bought together. Do not print a combination twice (A-B / B-A).**

Query:

```

WITH ProductPairs AS (
    SELECT
        p1.SUB_COMMODITY_DESC AS product_A,
        p2.SUB_COMMODITY_DESC AS product_B,
        COUNT(*) AS pair_count
    FROM
        `PC_Dataset_2024.transaction` t1
    JOIN
        `PC_Dataset_2024.transaction` t2
        ON t1.BASKET_ID = t2.BASKET_ID -- Ensure they are in the same basket
        AND t1.PRODUCT_ID < t2.PRODUCT_ID -- Prevent counting A-B and B-A separately
    JOIN
        `PC_Dataset_2024.product` p1
        ON t1.PRODUCT_ID = p1.PRODUCT_ID
    JOIN
        `PC_Dataset_2024.product` p2
        ON t2.PRODUCT_ID = p2.PRODUCT_ID
    GROUP BY
        p1.SUB_COMMODITY_DESC, p2.SUB_COMMODITY_DESC
)
SELECT
    product_A,
    product_B,
    pair_count
FROM
    ProductPairs
ORDER BY
    pair_count DESC;

```

Result:-

Row	product_A	product_B	pair_count
1	YOGURT NOT MULTI-PACKS	YOGURT NOT MULTI-PACKS	15947
2	BABY FOOD - BEGINNER	BABY FOOD - BEGINNER	10080
3	SS ECONOMY ENTREES/DINN...	SS ECONOMY ENTREES/DINN...	6633
4	SOFT DRINK POWDER POUCHES	SOFT DRINK POWDER POUCHES	6375

Insights : Yogurt and Baby Foods is the most paired item based on the data.

**Question 7:Find the weekly change in Revenue Per Account (RPA) (difference in spending by each customer compared to last week) using the LAG function.**

```
Query : WITH Weekly_Spending AS (
SELECT
    household_key,
    WEEK_NO,
    SUM(SALES_VALUE) AS Weekly_Spent
FROM `PC_Dataset_2024.transaction`
GROUP BY household_key, WEEK_NO
),
RPA_Change AS (
SELECT
    household_key,
    WEEK_NO,
    Weekly_Spent,
    LAG(Weekly_Spent, 1, 0) OVER (PARTITION BY household_key ORDER BY WEEK_NO) AS
Prev_Week_Spent,
    Weekly_Spent - LAG(Weekly_Spent, 1, 0) OVER (PARTITION BY household_key ORDER BY
WEEK_NO) AS RPA_Change
FROM Weekly_Spending
)
SELECT
    household_key,
    WEEK_NO,
    RPA_Change
FROM RPA_Change;
```

Result:

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS		EXECUTION GRAPH
Row	WEEK_NO	household_key	R... WEEK_NO	Weekly_Spent	Prev_Week_Spent	FRPA_Change	
1	13	63	13	27.64	0.0	27.64	
2	14	63	14	3.0	27.64	-24.64	
3	16	63	16	32.69	3.0	29.6899999999...	
4	17	63	17	7.68999999999...	32.69	-25.0	
5	23	63	23	2.14	7.68999999999...	-5.5499999999...	
6	25	63	25	14.52	2.14	12.3799999999...	
7	26	63	26	11.73	14.52	-2.7899999999...	
8	27	63	27	9.26	11.73	-2.4700000000...	
9	28	63	28	8.66	9.26	-0.5999999999...	
10	31	63	31	2.98	8.66	-5.68	

Insight: Tracking changes in revenue per account over time can help identify trends in customer behavior and inform retention strategies.

Recommendations:

**Pricing Strategy:** Adjust prices based on the distribution of order sizes.

**Targeted Marketing:** Focus marketing efforts on high-value customers and stores with the highest traffic.

**Inventory Management:** Ensure that frequently bought-together products are well-stocked.

**Retention Strategies:** Track RPA changes to identify at-risk customers and offer them incentives.

### Additional Question 1: Finding Most Selling Products

Query: `SELECT`

`p.SUB_COMMODITY_DESC AS product_name,`

`SUM(t.SALES_VALUE) AS total_revenue`

`FROM`

``PC_Dataset_2024.transaction` t`

`JOIN`

``PC_Dataset_2024.product` p`

`ON t.PRODUCT_ID = p.PRODUCT_ID`

`GROUP BY`

`p.SUB_COMMODITY_DESC`

`ORDER BY`

`total_revenue DESC`

`LIMIT 10;`

Result

Row	product_name	total_revenue
1	GASOLINE-REG UNLEADED	315997.0900000...
2	FLUID MILK WHITE ONLY	80754.44000000...
3	SOFT DRINKS 12/18&15PK CA...	79214.43999999...
4	BEERALEMALT LIQUORS	75036.18000000...
5	CIGARETTES	48179.15000000...
6	CHOICE BEEF	38382.08999999...
7	SHREDDED CHEESE	34252.77999999...
8	PRIMAL	32829.33000000...
9	PREMIUM	32537.04000000...
10	BABY DIAPERS	30421.63000000...

Knowing these will help in promotions and other marketing strategies.

**Additional Question 2:** Find the Time where max orders were transacted:-

```

Query: SELECT
CASE
    WHEN TRANS_TIME BETWEEN 0 AND 599 THEN 'Midnight to 6 AM'
    WHEN TRANS_TIME BETWEEN 600 AND 1159 THEN '6 AM to Noon'
    WHEN TRANS_TIME BETWEEN 1200 AND 1759 THEN 'Noon to 6 PM'
    WHEN TRANS_TIME BETWEEN 1800 AND 2359 THEN '6 PM to Midnight'
END AS time_of_day,
COUNT(BASKET_ID) AS order_count
FROM `PC_Dataset_2024.transaction`
GROUP BY time_of_day
ORDER BY time_of_day;

```

Result:-

Row	time_of_day	order_count
1	6 AM to Noon	197897
2	6 PM to Midnight	418605
3	Midnight to 6 AM	20660
4	Noon to 6 PM	661324

**Recommendation:** We should Focus on increasing the share for Midnight to 6AM and max orders come in Noon to 6pm.

Additional Questions:How does the presence of children in a household affect spending habits?

```

Query: SELECT d.KID_CATEGORY_DESC, AVG(t.SALES_VALUE) AS avg_spending
FROM `PC_Dataset_2024.transaction` t
JOIN `PC_Dataset_2024.demographics` d ON t.household_key = d.household_key
GROUP BY d.KID_CATEGORY_DESC
ORDER BY avg_spending DESC;

```

Row	KID_CATEGORY_DESC	avg_spending
1	3+	3.255037104325...
2	None/Unknown	3.151318410560...
3	1	3.149138794892...
4	2	3.050615581098...

**Insight:** 3+ Category has higher spending in comparison to others as expected