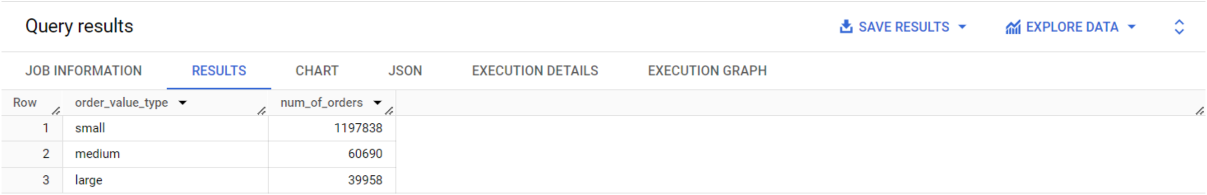
E-commerce SQL Analysis

1. **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:**

```
with cte as
(
  select round(sales_value*quantity,2) as order_value from
`e_commerce_data_analysis.transactions`
),
cte1 as
(
  select
    cte.order_value,
  case
    when cte.order_value between 0 and 10 then 'small'
    when cte.order_value between 10 and 20 then 'medium'
    else 'large'
  end as order_value_type
  from cte
)
select
  cte1.order_value_type,
  count(*) as num_of_orders
from cte1
group by cte1.order_value_type
order by num_of_orders desc;
```

- **Output:**



Row	order_value_type	num_of_orders
1	small	1197838
2	medium	60690
3	large	39958

- **Insights:**
    - Small Orders (0-10 dollars) are the most common, but customers also make medium (10-20 dollars) and large orders (20+ dollars).

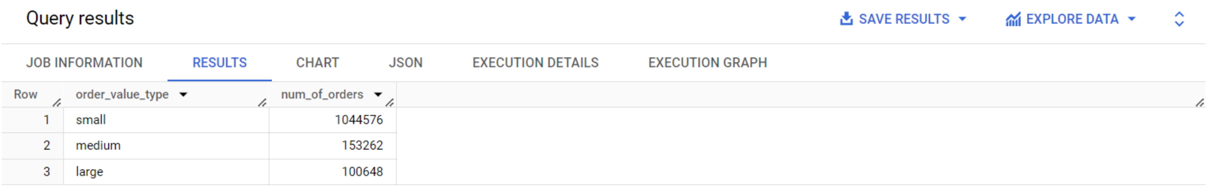
- **Recommendations:**
    - Offer discounts or bundles to small-order customers to increase their basket size, and reward high-value customers to encourage repeat purchases.

2. **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:**

```
with cte as
(
 select round(sales_value*quantity,2) as order_value from
`e_commerce_data_analysis.transactions`
),
cte1 as
(
 select
 cte.order_value,
 case
  when cte.order_value between 0 and 5 then 'small'
  when cte.order_value between 5 and 10 then 'medium'
  else 'large'
 end as order_value_type
 from cte
)
select
cte1.order_value_type,
count(*) as num_of_orders
from cte1
group by cte1.order_value_type
order by num_of_orders desc
```

- **Output:**



Row	order_value_type	num_of_orders
1	small	1044576
2	medium	153262
3	large	100648

- **Insights:**
  - Small Orders (0-5 dollars) are the most common, but customers also make medium (5-10 dollars) and large orders (10+ dollars).

- **Recommendations:**
  - Offer discounts or bundles to small-order customers to increase their basket size, and reward high-value customers to encourage repeat purchases.

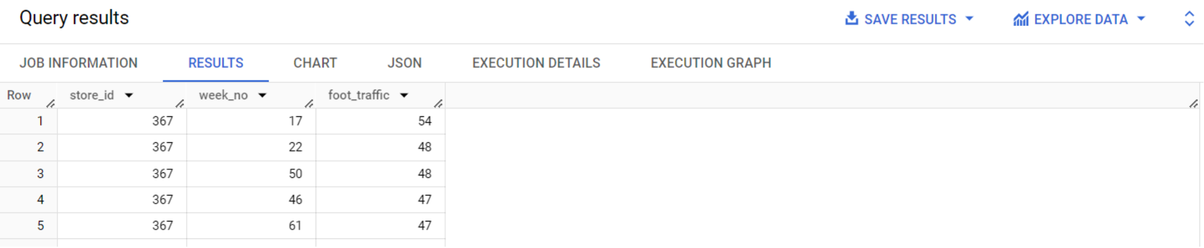
3. **Question 3:** Find top 3 stores with highest foot traffic for each week (Foot traffic: number of customers transacting)

- **Query:**

```
with cte as
(
 select
       store_id,
       week_no,
       count(distinct household_key) as foot_traffic
 from `e_commerce_data_analysis.transactions`
 group by store_id, week_no
),
cte2 as
(
 select
     *,
     dense_rank() over(partition by week_no order by cte.foot_traffic desc) as
     rnk from cte
)

select
     cte2.store_id,
     cte2.week_no,
     cte2.foot_traffic
from cte2
where rnk <= 3
order by foot_traffic desc, week_no
```

- **Output**



Row	store_id	week_no	foot_traffic
1	367	17	54
2	367	22	48
3	367	50	48
4	367	46	47
5	367	61	47

- **Insights:**
  - Some stores consistently have more customers visiting, while others have fewer.

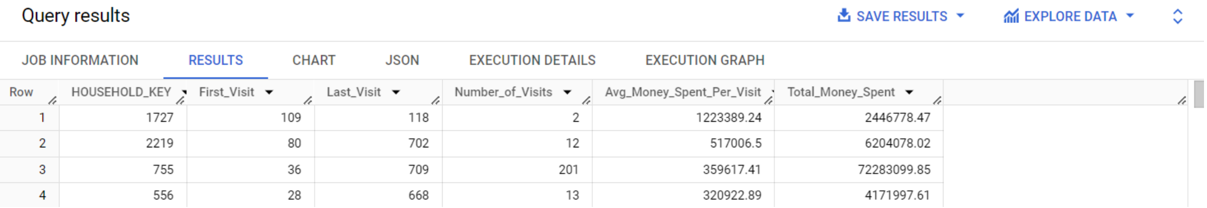
- **Recommendations**:
  - Learn from the successful stores and apply their strategies (e.g., promotions) to underperforming stores.

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

    - **Query**

```
SELECT
   HOUSEHOLD_KEY,
   MIN(DAY) AS First_Visit,
   MAX(DAY) AS Last_Visit,
   COUNT(DISTINCT BASKET_ID) AS Number_of_Visits,
   ROUND(SUM(SALES_VALUE*QUANTITY) / COUNT(DISTINCT
BASKET_ID), 2) AS Avg_Money_Spent_Per_Visit,
   ROUND(SUM(SALES_VALUE*QUANTITY), 2) AS Total_Money_Spent
FROM `e_commerce_data_analysis.transactions`
GROUP BY HOUSEHOLD_KEY
ORDER BY Avg_Money_Spent_Per_Visit DESC;
```

    - **Output**



Row	HOUSEHOLD_KEY	First_Visit	Last_Visit	Number_of_Visits	Avg_Money_Spent_Per_Visit	Total_Money_Spent
1	1727	109	118	2	1223389.24	2446778.47
2	2219	80	702	12	517006.5	6204078.02
3	755	36	709	201	359617.41	72283099.85
4	556	28	668	13	320922.89	4171997.61

    - **Insights:**
        - Some customers visit frequently but spend less, while others visit less often but spend more.

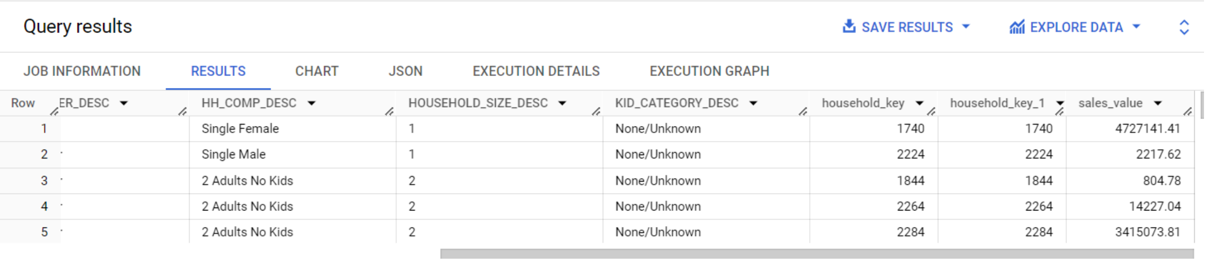
    - **Recommendations**:
        - Reward high-spending customers with loyalty perks and encourage frequent visitors to spend more through targeted offers.

5. **Question 5:** Do a single customer analysis selecting most spending customer for whom we have demographic information(because not all customers in transaction data are present in demographic table)(show the demographic as well as total spent)

- **Query**

```
with cte1 as
(
 select
 household_key,
 round(sum(quantity*sales_value),2) as sales_value
 from `e_commerce_data_analysis.transactions`
 group by household_key
 order by sales_value desc
)
select * from `e_commerce_data_analysis.hh_demographic` as demographic join
cte1
on demographic.household_key = cte1.household_key
```

- Output



Query results

Row	ER_DESC	HH_COMP_DESC	HOUSEHOLD_SIZE_DESC	KID_CATEGORY_DESC	household_key	household_key_1	sales_value
1		Single Female	1	None/Unknown	1740	1740	4727141.41
2		Single Male	1	None/Unknown	2224	2224	2217.62
3		2 Adults No Kids	2	None/Unknown	1844	1844	804.78
4		2 Adults No Kids	2	None/Unknown	2264	2264	14227.04
5		2 Adults No Kids	2	None/Unknown	2284	2284	3415073.81

- **Insights:**
    - The highest-spending customers tend to come from specific demographic groups (age, income, household size).

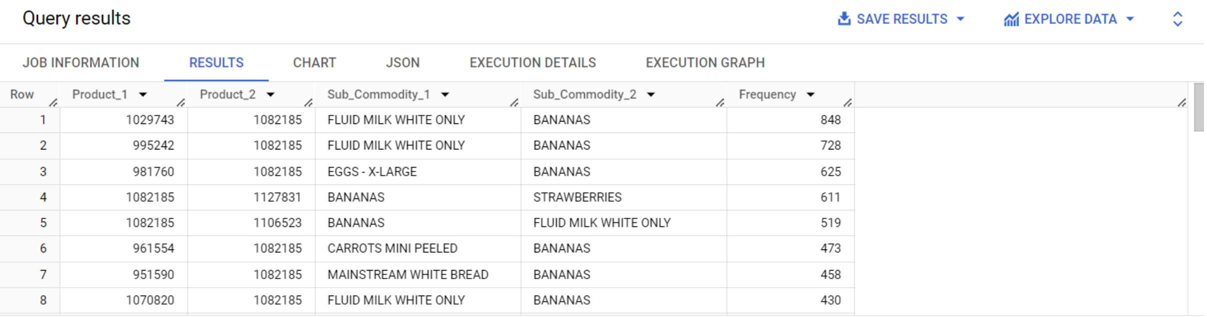
- **Recommendations**:
    - Focus marketing on similar demographic groups to attract more high-spending customers.

6. **Question 6:** Find products(product table : SUB_COMMODITY_DESC) which 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 Product_Combinations AS (
   SELECT
      LEAST(p1.PRODUCT_ID, p2.PRODUCT_ID) AS Product_1,
      GREATEST(p1.PRODUCT_ID, p2.PRODUCT_ID) AS Product_2,
      COUNT(*) AS Frequency
   FROM `e_commerce_data_analysis.transactions` p1
   JOIN `e_commerce_data_analysis.transactions` p2
      ON p1.BASKET_ID = p2.BASKET_ID
      AND p1.PRODUCT_ID < p2.PRODUCT_ID
   GROUP BY LEAST(p1.PRODUCT_ID, p2.PRODUCT_ID),
GREATEST(p1.PRODUCT_ID, p2.PRODUCT_ID)
)
SELECT
   pc.Product_1,
   pc.Product_2,
   p1.SUB_COMMODITY_DESC AS Sub_Commodity_1,
   p2.SUB_COMMODITY_DESC AS Sub_Commodity_2,
   pc.Frequency
FROM Product_Combinations pc
JOIN `e_commerce_data_analysis.product` p1 ON pc.Product_1 =
p1.PRODUCT_ID
JOIN `e_commerce_data_analysis.product` p2 ON pc.Product_2 =
p2.PRODUCT_ID
ORDER BY Frequency DESC;
```

- **Output:**



Row	Product_1	Product_2	Sub_Commodity_1	Sub_Commodity_2	Frequency
1	1029743	1082185	FLUID MILK WHITE ONLY	BANANAS	848
2	995242	1082185	FLUID MILK WHITE ONLY	BANANAS	728
3	981760	1082185	EGGS - X-LARGE	BANANAS	625
4	1082185	1127831	BANANAS	STRAWBERRIES	611
5	1082185	1106523	BANANAS	FLUID MILK WHITE ONLY	519
6	961554	1082185	CARROTS MINI PEELED	BANANAS	473
7	951590	1082185	MAINSTREAM WHITE BREAD	BANANAS	458
8	1070820	1082185	FLUID MILK WHITE ONLY	BANANAS	430

- **Insights:**
    - Certain products are often bought together.

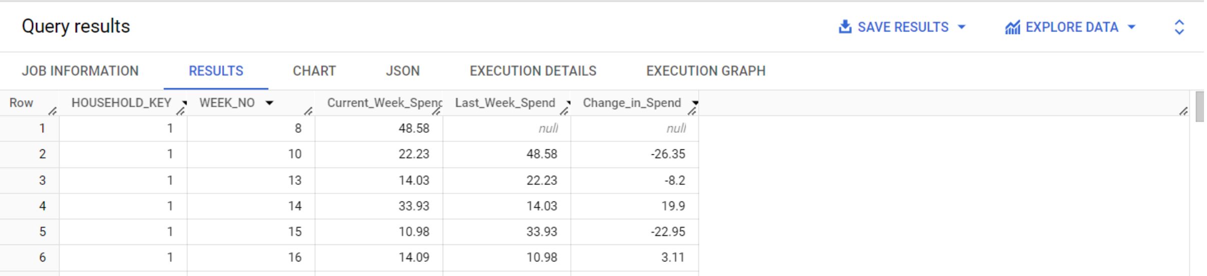
- **Recommendation**:
    - Highlight these product combinations in-store and online, and offer combo deals to increase sales.

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

- **Query**

```
WITH Weekly_Spending AS (
   SELECT
      HOUSEHOLD_KEY,
      WEEK_NO,
      SUM(SALES_VALUE*QUANTITY) AS Total_Weekly_Spend
   FROM `e_commerce_data_analysis.transactions`
   GROUP BY HOUSEHOLD_KEY, WEEK_NO
),
Weekly_Change AS (
   SELECT
      HOUSEHOLD_KEY,
      WEEK_NO,
      Total_Weekly_Spend,
      LAG(Total_Weekly_Spend, 1) OVER (PARTITION BY
HOUSEHOLD_KEY ORDER BY WEEK_NO) AS Last_Week_Spend,
      (Total_Weekly_Spend - LAG(Total_Weekly_Spend, 1) OVER
(PARTITION BY HOUSEHOLD_KEY ORDER BY WEEK_NO)) AS
Change_in_Spend
   FROM Weekly_Spending
)
SELECT
   HOUSEHOLD_KEY,
   WEEK_NO,
   round(Total_Weekly_Spend,2) AS Current_Week_Spend,
   round(Last_Week_Spend, 2) as Last_Week_Spend,
   round(Change_in_Spend, 2) as Change_in_Spend
FROM Weekly_Change
ORDER BY HOUSEHOLD_KEY, WEEK_NO;
```

- **Output**



Query results

Row	HOUSEHOLD_KEY	WEEK_NO	Current_Week_Spenc	Last_Week_Spend	Change_in_Spend
1	1	8	48.58	*null*	*null*
2	1	10	22.23	48.58	-26.35
3	1	13	14.03	22.23	-8.2
4	1	14	33.93	14.03	19.9
5	1	15	10.98	33.93	-22.95
6	1	16	14.09	10.98	3.11

- **Insights:**
  - Customer spending changes week by week, with some customers spending more or less over time.

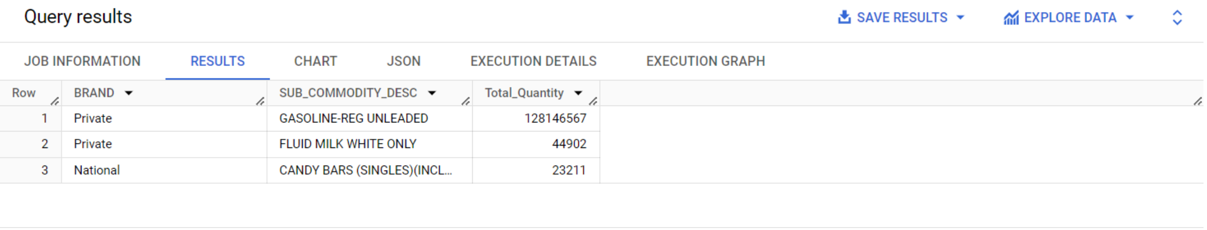
- **Recommendations**:
  - Track changes and offer discounts to win back customers who spend less, while rewarding those who spend more.

8. **Question 8:** Find the most popular brand and SUB_COMMODITY_DESC (in terms of quantity purchased) and whether it's a private or national label.

   - **Query**

```
SELECT
   BRAND,
   SUB_COMMODITY_DESC,
   SUM(QUANTITY) AS Total_Quantity
FROM `e_commerce_data_analysis.transactions` t
JOIN `e_commerce_data_analysis.product` p ON t.PRODUCT_ID =
p.PRODUCT_ID
GROUP BY BRAND, SUB_COMMODITY_DESC
ORDER BY Total_Quantity DESC
LIMIT 3;
```

   - **Output**



Row	BRAND	SUB_COMMODITY_DESC	Total_Quantity
1	Private	GASOLINE-REG UNLEADED	128146567
2	Private	FLUID MILK WHITE ONLY	44902
3	National	CANDY BARS (SINGLES)(INCL...	23211

   - **Insights**
     - Certain brands and products are more popular, especially private labels or national brands.

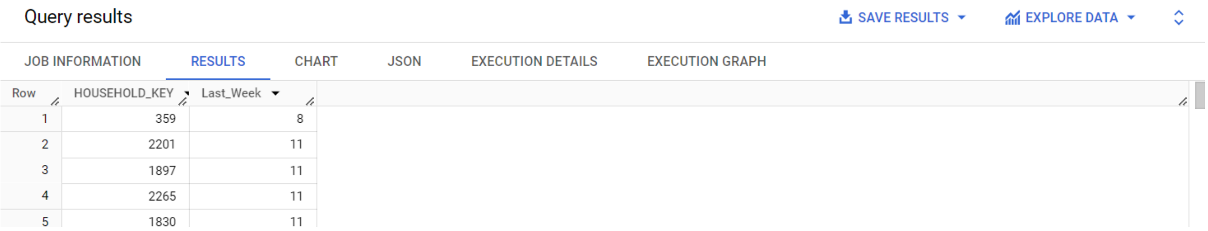
   - **Recommendations**:
     - Promote private-label products to price-conscious customers and use national brands for premium marketing.

9. **Question 9:** Find customers who haven't shopped for more than 8 weeks. List the last week they made a purchase.

- **Query**

```
WITH Last_Visit AS (
   SELECT
      HOUSEHOLD_KEY,
      MAX(WEEK_NO) AS Last_Week
   FROM `e_commerce_data_analysis.transactions`
   GROUP BY HOUSEHOLD_KEY
)
SELECT
   HOUSEHOLD_KEY,
   Last_Week
FROM Last_Visit
WHERE Last_Week < (SELECT MAX(WEEK_NO) FROM
`e_commerce_data_analysis.transactions`) - 8
ORDER BY Last_Week;
```

- **Output**



Row	HOUSEHOLD_KEY	Last_Week
1	359	8
2	2201	11
3	1897	11
4	2265	11
5	1830	11

- **Insights**
  - Some customers haven’t shopped in 8 weeks.

- **Recommendations**:
  - Send them special offers to encourage them to shop again.

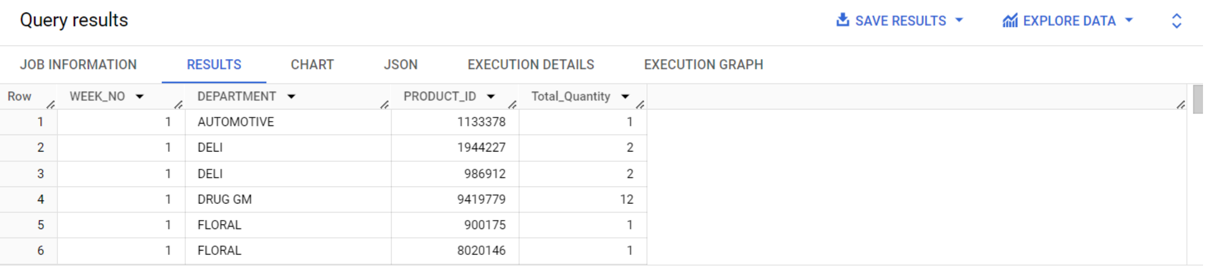
10. **Question 10:** Identify the most popular product (in terms of quantity purchased) within each department for each week.

- Query

```
WITH Weekly_Product_Purchases AS (
  SELECT
    WEEK_NO,
    DEPARTMENT,
    p.PRODUCT_ID,
    SUM(QUANTITY) AS Total_Quantity
  FROM `e_commerce_data_analysis.transactions` t
  JOIN `e_commerce_data_analysis.product` p ON t.PRODUCT_ID =
p.PRODUCT_ID
  GROUP BY WEEK_NO, DEPARTMENT, PRODUCT_ID
  HAVING SUM(QUANTITY) > 0
)

SELECT
  WEEK_NO,
  DEPARTMENT,
  PRODUCT_ID,
  Total_Quantity
FROM (
  SELECT
    WEEK_NO,
    DEPARTMENT,
    PRODUCT_ID,
    Total_Quantity,
    DENSE_RANK() OVER (PARTITION BY WEEK_NO, DEPARTMENT
ORDER BY Total_Quantity DESC) AS rnk
  FROM Weekly_Product_Purchases
) AS Ranked_Products
WHERE rnk = 1
ORDER BY WEEK_NO, DEPARTMENT;
```

- **Output**



Row	WEEK_NO	DEPARTMENT	PRODUCT_ID	Total_Quantity
1	1	AUTOMOTIVE	1133378	1
2	1	DELI	1944227	2
3	1	DELI	986912	2
4	1	DRUG GM	9419779	12
5	1	FLORAL	900175	1
6	1	FLORAL	8020146	1

- **Insights:**
  - Each department has best-selling products.

- **Recommendations**:
  - Focus on promoting these top-selling items to increase overall department sales.