E-commerce SQL Analysis

Problem Statement

Analysing the sales, product and customer data for an e-commerce company. Getting various insights and calculating various KPI and data with SQL in Big Query.

Expectations:

This project aims to leverage the power of e-commerce data (sales, product, and demographic) analysed through SQL to unlock actionable insights driving profitable growth. By delving into customer behaviour, product trends, and sales patterns, we will uncover hidden value that can inform key business decisions. You need to find these patterns and calculate various metrics and KPIs that suit the data and the goal.

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

```
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_category,
    COUNT(*) AS order_count
FROM myproj2-408807.trans.trans
GROUP BY order_category
ORDER BY order_category;
```

Row	order_category	order_count
1	Large	12536
2	Medium	26869
3	Small	1259081

.....

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

```
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_category,
    COUNT(*) AS order_count
FROM myproj2-408807.trans.trans
GROUP BY order_category
ORDER BY order_category;
```

Row	order_category	order_count
1	Large	39405
2	Medium	113099
3	Small	1145982

……

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

```
WITH RankedStores AS (
    SELECT
        store_id,
        week_no,
        COUNT(DISTINCT household_key) AS customer_count,
        ROW_NUMBER() OVER (PARTITION BY week_no ORDER BY COUNT(DISTINCT
household_key) DESC) AS store_rank
    FROM myproj2-408807.trans.trans
    GROUP BY week_no, store_id
)

SELECT
    week_no,
    store_id,
    customer_count
FROM RankedStores
WHERE store_rank <= 3
ORDER BY week_no, store_rank;
```

week_no	store_id	customer_count
1	32004	5
1	324	3
1	367	3
2	32004	7
2	313	6
2	367	5
3	367	10
3	32004	9
3	356	8
4	367	17
4	32004	11
4	446	8
5	367	15
5	32004	13
5	292	8
6	32004	21
6	367	17
6	356	15
7	367	19
7	356	13
7	446	12

……

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

```
SELECT
    household_key,
    MIN(trans_time) AS first_visit,
    MAX(trans_time) AS last_visit,
    COUNT(DISTINCT basket_id) AS number_of_visits,
    ROUND(AVG(sales_value), 2) AS avg_money_spent_per_visit,
    SUM(sales_value) AS total_money_spent
FROM myproj2-408807.trans.trans
GROUP BY household_key
ORDER BY avg_money_spent_per_visit DESC;
```

household_key	first_visit	last_visit	number_of_visits	avg_money_spent_per_visit	total_money_spent
1730	722	1802	83	16.73	1656.76
1727	2129	2139	2	12.72	114.51
2163	1058	1455	13	10.54	221.32
1339	1012	2302	6	10.42	187.53
991	946	1958	18	10.26	451.6
2219	1013	1926	12	10.05	321.66
2428	708	1842	14	10	180
755	532	1935	201	9.48	5461.54
1023	738	2252	422	8.58	18901.09
120	838	1739	13	8.18	130.92
821	857	2055	13	7.8	226.22
1471	722	2305	22	7.61	304.34
37	26	2353	56	7.55	1388.34
1607	707	2244	105	7.22	1841.2
1100	1002	2214	17	7.14	171.28
930	1007	1856	34	7.02	470.33
2149	818	2232	87	6.92	1834.82
1895	39	2209	43	6.81	605.69
435	610	2304	72	6.77	899.77

……

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)

```
SELECT
    D.*,
    T.total_money_spent
FROM myproj2-408807.demographic.demog D
JOIN (
    SELECT
        household_key,
        SUM(sales_value) AS total_money_spent
    FROM myproj2-408807.trans.trans
    GROUP BY household_key
```

```
    ORDER BY total_money_spent DESC
    LIMIT 2
) T ON D.household_key = T.household_key;
```

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

……

6. Find products(product table : SUB_COMMODITY_DESC) which are most frequently bought together and the count of each combination bought together.

```
SELECT
    p1.sub_commodity_desc AS product_1,
    p2.sub_commodity_desc AS product_2,
    COUNT(*) AS combination_count
FROM myproj2-408807.trans.trans t1
JOIN myproj2-408807.sales.product p1 ON t1.product_id = p1.product_id
JOIN myproj2-408807.trans.trans t2 ON t1.basket_id = t2.basket_id
JOIN myproj2-408807.sales.product p2 ON t2.product_id = p2.product_id
    AND p1.sub_commodity_desc < p2.sub_commodity_desc
GROUP BY p1.sub_commodity_desc, p2.sub_commodity_desc
ORDER BY combination_count DESC;
```

product_1	product_2	combination_count
FLUID MILK WHITE ONLY	YOGURT NOT MULTI-PACKS	5953
BANANAS	FLUID MILK WHITE ONLY	4365
FLUID MILK WHITE ONLY	SOFT DRINKS 12/18&15PK CAN CAR	4326
FLUID MILK WHITE ONLY	MAINSTREAM WHITE BREAD	3934
BANANAS	YOGURT NOT MULTI-PACKS	3847
FLUID MILK WHITE ONLY	SHREDDED CHEESE	3840
FLUID MILK WHITE ONLY	SFT DRNK 2 LITER BTL CARB INCL	3494

……

7. Find the weekly change in Revenue Per Account (RPA) (difference in spending by each customer compared to last week)

```
SELECT
    household_key,
    week_no,
    sales_value,
```

```
    LAG(sales_value) OVER (PARTITION BY household_key ORDER BY week_no) AS
previous_week_sales,
    COALESCE(sales_value - LAG(sales_value) OVER (PARTITION BY household_key
ORDER BY week_no), 0) AS weekly_change_in_sales
FROM myproj2-408807.trans.trans
ORDER BY household_key, week_no;
```

household_key	week_no	sales_value	previous_week_sales	weekly_change_in_sales
1	8	1.5		0
1	8	0.77	1.5	-0.73
1	8	2.5	0.77	1.73
1	8	3.99	2.5	1.49
1	8	2.5	3.99	-1.49
1	8	6	2.5	3.5
1	8	3.99	6	-2.01
1	8	2.69	3.99	-1.3
1	8	1.49	2.69	-1.2
1	8	2.19	1.49	0.7
1	8	0.79	2.19	-1.4
1	8	2.99	0.79	2.2
1	8	3.99	2.99	1
1	8	7.19	3.99	3.2
1	10	2.99	7.19	-4.2
1	10	1.88	2.99	-1.11
1	10	0.92	1.88	-0.96

……

8. This query helps visualize the trend in monthly revenue over time.

```
SELECT EXTRACT(YEAR FROM TIMESTAMP_SECONDS(trans_time)) AS year,
EXTRACT(MONTH FROM TIMESTAMP_SECONDS(trans_time)) AS month, SUM(sales_value)
AS monthly_revenue FROM myproj2-408807.trans.trans GROUP BY year, month ORDER
BY year, month;
```

Row	year	month	monthly_revenue
1	1970	1	4029338.410022...

……

9. Understand the AOV based on customer segments (e.g., marital status, income).

```
SELECT d.marital_status_code, ROUND(AVG(t.sales_value), 2) AS avg_order_value
FROM myproj2-408807.trans.trans t JOIN myproj2-408807.demographic.demog d ON
t.household_key = d.household_key GROUP BY d.marital_status_code ORDER BY
avg_order_value DESC;
```

Row	marital_status_code	avg_order_value
1	A	3.26
2	B	3.1
3	U	3.06

.....

10. Identify the manufacturers generating the most revenue.

```
SELECT p.manufacturer, SUM(t.sales_value) AS total_revenue FROM myproj2-408807.trans.trans t JOIN myproj2-408807.sales.product p ON t.product_id = p.product_id GROUP BY p.manufacturer ORDER BY total_revenue DESC;
```

manufacturer	total_revenue
69	1090508.12
2	174160.8
764	83073.14
103	59227.85
1208	57409.78
317	51754.18
544	51087.33
1251	42872.18
673	41710.46
194	36736.85
1046	35361.51
239	33484.83
794	32348.44
539	31432.6
397	30391.19
151	26079.82
2224	23325.63
751	22033.66

.....

11. Understand how often customers make purchases.

```
SELECT CASE WHEN visit_count = 1 THEN '1' WHEN visit_count BETWEEN 2 AND 5 THEN '2-5' WHEN visit_count BETWEEN 6 AND 10 THEN '6-10' ELSE 'More than 10' END AS visit_frequency, COUNT(DISTINCT household_key) AS customer_count FROM ( SELECT household_key, COUNT(DISTINCT basket_id) AS visit_count FROM myproj2-408807.trans.trans GROUP BY household_key ) AS subquery GROUP BY visit_frequency;
```

visit_frequency	customer_count
More than 10	2358
6-10	95
2-5	43
1	4

Insights :

- Implement targeted marketing campaigns based on customer segments to enhance engagement and sales.
- Improve user experience on the website or app to increase customer satisfaction and retention.
- Invest in data-driven decision-making by regularly analysing and acting upon insights obtained from data.

1. Order Value 0-10 Dollars:

Recommendation: Encourage upselling or offering bundled deals to increase the average order value. Promotions for products in slightly higher price ranges might boost overall sales.

2. Store ID 32004 with Highest Foot Traffic:

Recommendation: Investigate why this specific store has high foot traffic. Assess the store's layout, marketing strategies, or location advantages to replicate successful elements in other stores.

3. Average Money Spent per Visit:

Recommendation: Target customers spending around 17 dollars per visit with personalized offers or loyalty programs to increase their spending further. Providing incentives might encourage higher-value purchases.

4. Demographic Details of Top Spender:

Recommendation: Develop targeted marketing strategies catering to customers in the age range of 45-54, married, with a household income of 125-149k and adult kids. Tailor promotions or loyalty rewards to retain such high-value customers.

5. Frequently Bought Together Products:

Recommendation: Consider marketing campaigns bundling "Yogurt not multi-packs" with "Fluid Milk White Only" to encourage more combined purchases. Create promotions or deals leveraging this buying behavior.

6. Average Order Value for Married and Single Customers:

Recommendation: Explore ways to enhance the average order value for both customer segments. Create personalized offers or product recommendations to increase spending across different demographics.

7. Manufacturer with Highest Revenue:

Recommendation: Explore partnerships or promotional opportunities with the manufacturer generating the highest revenue. Strengthening ties with successful manufacturers can boost overall sales.

8. Visit Frequency More Than 10:

Recommendation: Reward and retain customers with high visit frequencies through exclusive perks or loyalty programs. Recognize their loyalty with incentives to maintain engagement.