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.

ASK:

Let’s start with By Asking what is required:

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+)

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+)

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

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

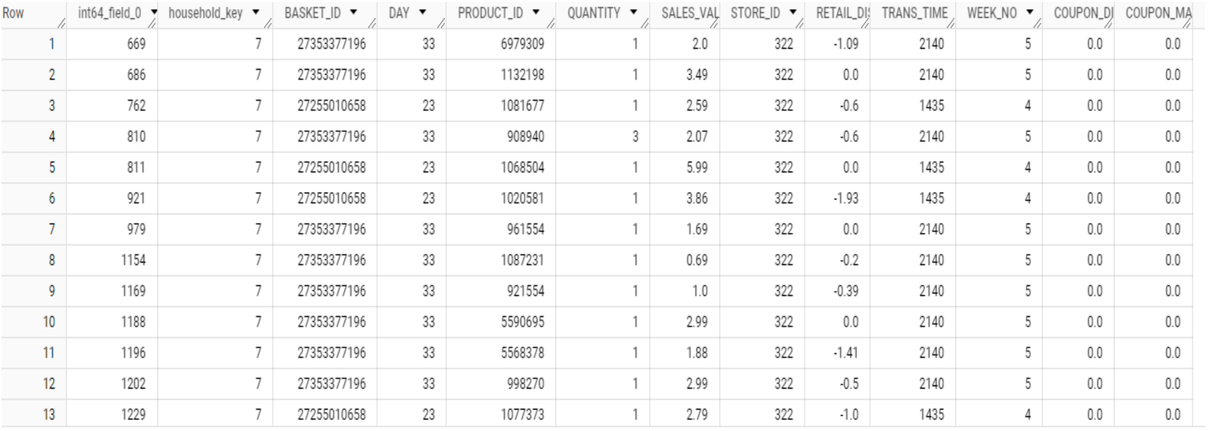
Question 5: Do a single customer analysis selecting most spending customer for whom we have demographic information(because not all customers in transaction data are present in demographic table)(show the demographic as well as total spent)

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)

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

Prepare: We check the structure of data by loading each table starting with transaction table

```
SELECT * FROM `Ecommerce. Transaction`;
```

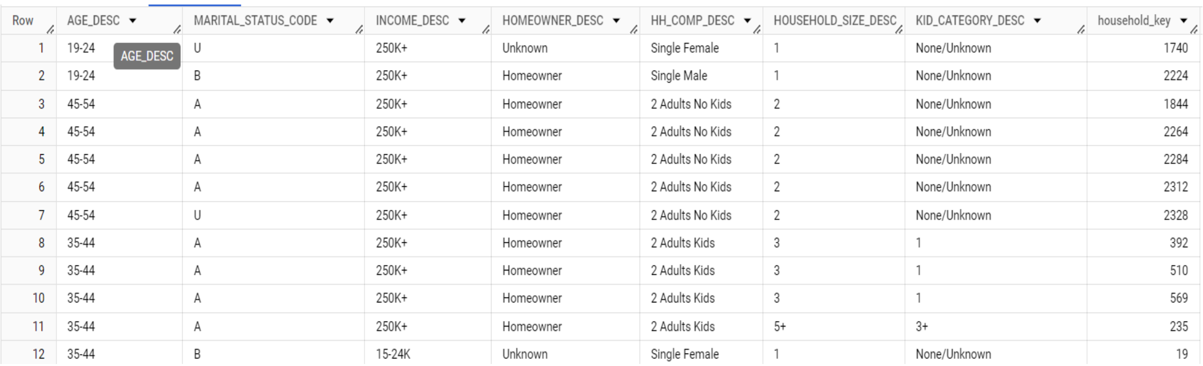


Row	int64_field_0 ▾	household_key ▾	BASKET_ID ▾	DAY ▾	PRODUCT_ID ▾	QUANTITY ▾	SALES_VAL	STORE_ID ▾	RETAIL_DI	TRANS_TIME	WEEK_NO ▾	COUPON_DI	COUPON_MA
1	669	7	27353377196	33	6979309	1	2.0	322	-1.09	2140	5	0.0	0.0
2	686	7	27353377196	33	1132198	1	3.49	322	0.0	2140	5	0.0	0.0
3	762	7	27255010658	23	1081677	1	2.59	322	-0.6	1435	4	0.0	0.0
4	810	7	27353377196	33	908940	3	2.07	322	-0.6	2140	5	0.0	0.0
5	811	7	27255010658	23	1068504	1	5.99	322	0.0	1435	4	0.0	0.0
6	921	7	27255010658	23	1020581	1	3.86	322	-1.93	1435	4	0.0	0.0
7	979	7	27353377196	33	961554	1	1.69	322	0.0	2140	5	0.0	0.0
8	1154	7	27353377196	33	1087231	1	0.69	322	-0.2	2140	5	0.0	0.0
9	1169	7	27353377196	33	921554	1	1.0	322	-0.39	2140	5	0.0	0.0
10	1188	7	27353377196	33	5590695	1	2.99	322	0.0	2140	5	0.0	0.0
11	1196	7	27353377196	33	5568378	1	1.88	322	-1.41	2140	5	0.0	0.0
12	1202	7	27353377196	33	998270	1	2.99	322	-0.5	2140	5	0.0	0.0
13	1229	7	27255010658	23	1077373	1	2.79	322	-1.0	1435	4	0.0	0.0

Transaction table has 1298486 rows and 13 columns

Loading Demographic Data

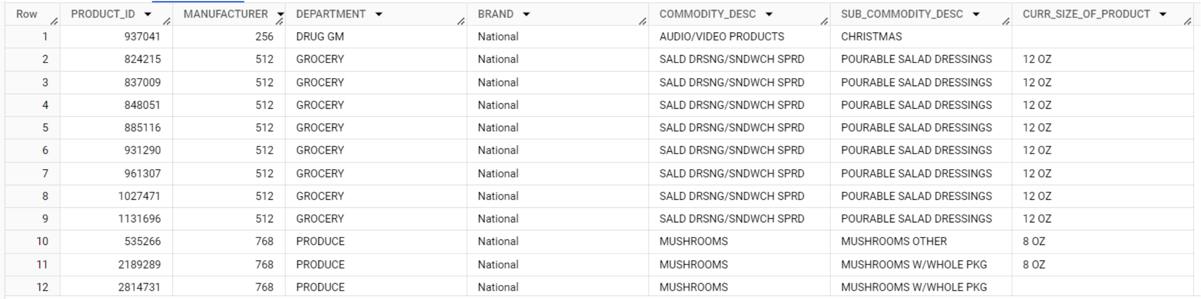
```
SELECT * FROM `Ecommerce.hh_demographic`;
```



Row	AGE_DESC	MARITAL_STATUS_CODE	INCOME_DESC	HOMEOWNER_DESC	HH_COMP_DESC	HOUSEHOLD_SIZE_DESC	KID_CATEGORY_DESC	household_key
1	19-24	U	250K+	Unknown	Single Female	1	None/Unknown	1740
2	19-24	B	250K+	Homeowner	Single Male	1	None/Unknown	2224
3	45-54	A	250K+	Homeowner	2 Adults No Kids	2	None/Unknown	1844
4	45-54	A	250K+	Homeowner	2 Adults No Kids	2	None/Unknown	2264
5	45-54	A	250K+	Homeowner	2 Adults No Kids	2	None/Unknown	2284
6	45-54	A	250K+	Homeowner	2 Adults No Kids	2	None/Unknown	2312
7	45-54	U	250K+	Homeowner	2 Adults No Kids	2	None/Unknown	2328
8	35-44	A	250K+	Homeowner	2 Adults Kids	3	1	392
9	35-44	A	250K+	Homeowner	2 Adults Kids	3	1	510
10	35-44	A	250K+	Homeowner	2 Adults Kids	3	1	569
11	35-44	A	250K+	Homeowner	2 Adults Kids	5+	3+	235
12	35-44	B	15-24K	Unknown	Single Female	1	None/Unknown	19

Loading Products Data

```
SELECT * FROM `Ecommerce.product`;
```

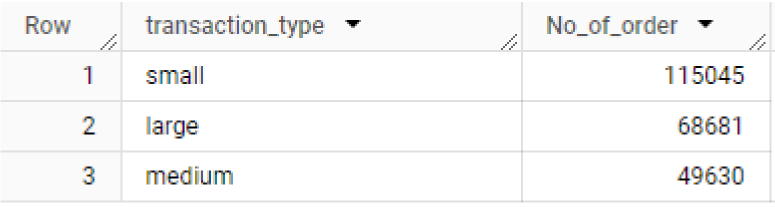


Row	PRODUCT_ID	MANUFACTURER	DEPARTMENT	BRAND	COMMODITY_DESC	SUB_COMMODITY_DESC	CURR_SIZE_OF_PRODUCT
1	937041	256	DRUG GM	National	AUDIO/VIDEO PRODUCTS	CHRISTMAS	
2	824215	512	GROCERY	National	SALD DRSNG/SNDWCH SPRD	POURABLE SALAD DRESSINGS	12 OZ
3	837009	512	GROCERY	National	SALD DRSNG/SNDWCH SPRD	POURABLE SALAD DRESSINGS	12 OZ
4	848051	512	GROCERY	National	SALD DRSNG/SNDWCH SPRD	POURABLE SALAD DRESSINGS	12 OZ
5	885116	512	GROCERY	National	SALD DRSNG/SNDWCH SPRD	POURABLE SALAD DRESSINGS	12 OZ
6	931290	512	GROCERY	National	SALD DRSNG/SNDWCH SPRD	POURABLE SALAD DRESSINGS	12 OZ
7	961307	512	GROCERY	National	SALD DRSNG/SNDWCH SPRD	POURABLE SALAD DRESSINGS	12 OZ
8	1027471	512	GROCERY	National	SALD DRSNG/SNDWCH SPRD	POURABLE SALAD DRESSINGS	12 OZ
9	1131696	512	GROCERY	National	SALD DRSNG/SNDWCH SPRD	POURABLE SALAD DRESSINGS	12 OZ
10	535266	768	PRODUCE	National	MUSHROOMS	MUSHROOMS OTHER	8 OZ
11	2189289	768	PRODUCE	National	MUSHROOMS	MUSHROOMS W/WHOLE PKG	8 OZ
12	2814731	768	PRODUCE	National	MUSHROOMS	MUSHROOMS W/WHOLE PKG	

Analyse:

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+)

```
WITH CTE1 as (
       SELECT BASKET_ID,
       SUM(SALES_VALUE) as Transaction_value,
       CASE
         WHEN SUM(SALES_VALUE) >= 0 AND SUM(SALES_VALUE) < 10 then "small"
         WHEN SUM(SALES_VALUE) > 10 AND SUM(SALES_VALUE) <= 20 then "medium"
         ELSE "large"
       END AS transaction_type
       FROM `Ecommerce.transaction`
       GROUP BY BASKET_ID)
SELECT
       transaction_type, COUNT(*) as No_of_order
FROM CTE1
GROUP BY transaction_type
ORDER BY No_of_order DESC;
```



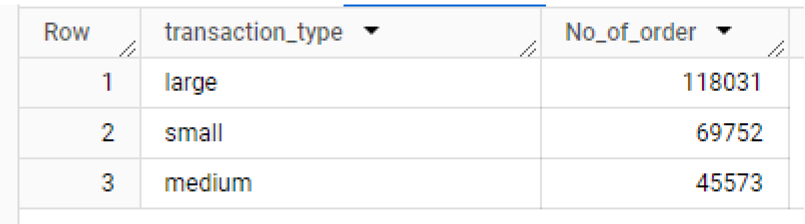
Row	transaction_type	No_of_order
1	small	115045
2	large	68681
3	medium	49630

Insight: We can observe small transactions bring more no of orders than the Large and Medium orders.

Recommendation: Medium order valve order are lowest we should work to increase this.

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+)

```
WITH CTE1 as (
       SELECT BASKET_ID,
       SUM(SALES_VALUE) as Transaction_value,
       CASE
         WHEN SUM(SALES_VALUE) >= 0 AND SUM(SALES_VALUE) < 5 then "small"
         WHEN SUM(SALES_VALUE) > 5 AND SUM(SALES_VALUE) <= 10 then "medium"
         ELSE "large"
       END AS transaction_type
       FROM `Ecommerce.transaction`
       GROUP BY BASKET_ID)
SELECT
       transaction_type,
       COUNT(*) as No_of_order
FROM CTE1
GROUP BY transaction_type
ORDER BY No_of_order DESC
```



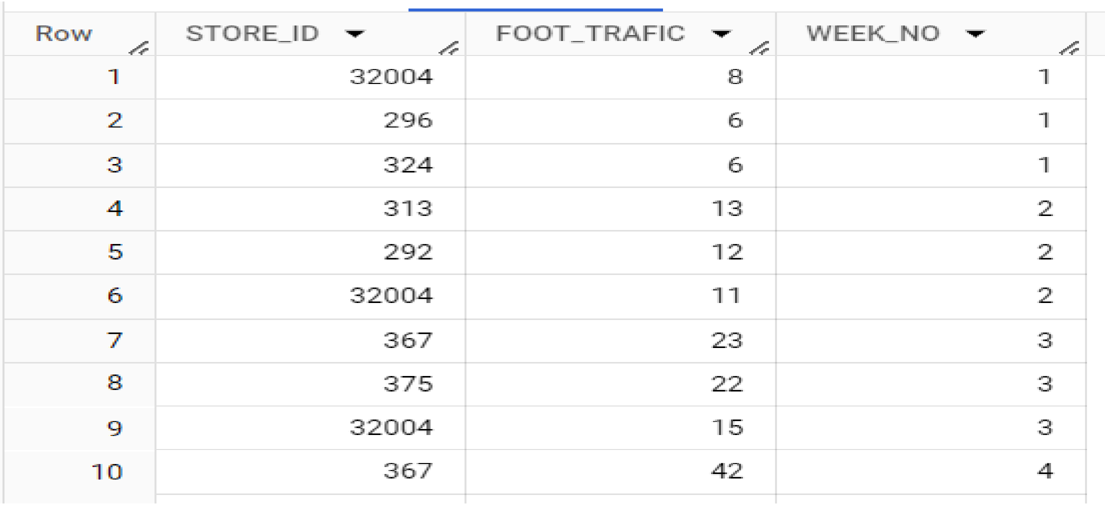
Row	transaction_type	No_of_order
1	large	118031
2	small	69752
3	medium	45573

Insight: We can observe small transactions bring more no of orders than the Large and Medium orders.

Recommendation: Medium order valve order is lowest we should work to increase this.

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

```
WITH CTE1 AS (
  SELECT STORE_ID,COUNT(DISTINCT BASKET_ID) AS FOOT_TRAFIC, WEEK_NO
  FROM `Ecommerce.transaction`
  GROUP BY STORE_ID, WEEK_NO),
CTE2 AS (
  SELECT STORE_ID, FOOT_TRAFIC,WEEK_NO,
  ROW_NUMBER() OVER(PARTITION BY WEEK_NO ORDER BY FOOT_TRAFIC DESC) as Ranks
  FROM CTE1 )
SELECT
  STORE_ID,
  FOOT_TRAFIC,
  WEEK_NO
FROM CTE2
WHERE Ranks<=3
ORDER BY WEEK_NO, Ranks;
```



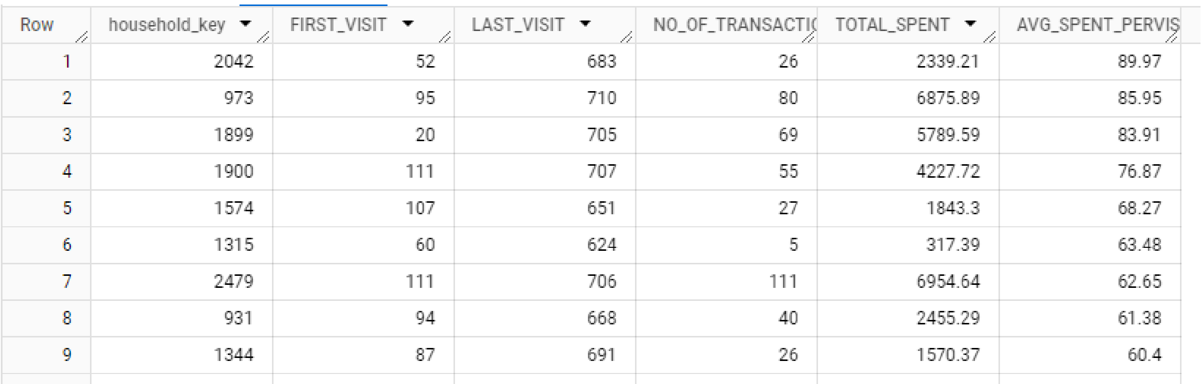
Row	STORE_ID	FOOT_TRAFIC	WEEK_NO
1	32004	8	1
2	296	6	1
3	324	6	1
4	313	13	2
5	292	12	2
6	32004	11	2
7	367	23	3
8	375	22	3
9	32004	15	3
10	367	42	4

Recommendation:

We should pay more attention on stores with highest foot traffic and keep it stocked to maximise the sales.

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

```
SELECT
    household_key, MIN(DAY) AS FIRST_VISIT, MAX(DAY) AS LAST_VISIT,
    COUNT( DISTINCT BASKET_ID) AS NO_OF_TRANSACTION, ROUND(SUM(SALES_VALUE),2) AS
TOTAL_SPENT,
    ROUND(SUM(SALES_VALUE)/COUNT(DISTINCT BASKET_ID),2) AS AVG_SPENT_PERVISIT
FROM `Ecommerce.transaction`
GROUP BY household_key
ORDER BY AVG_SPENT_PERVISIT DESC
```

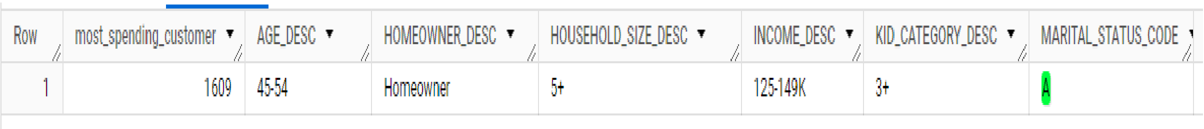


Row	household_key	FIRST_VISIT	LAST_VISIT	NO_OF_TRANSACTI	TOTAL_SPENT	AVG_SPENT_PERVIS
1	2042	52	683	26	2339.21	89.97
2	973	95	710	80	6875.89	85.95
3	1899	20	705	69	5789.59	83.91
4	1900	111	707	55	4227.72	76.87
5	1574	107	651	27	1843.3	68.27
6	1315	60	624	5	317.39	63.48
7	2479	111	706	111	6954.64	62.65
8	931	94	668	40	2455.29	61.38
9	1344	87	691	26	1570.37	60.4

Recommendation: We should give loyalty points to customers with highest no of transaction and highest Avg Spent per visit.
We should try to targeting customer with high Avg spent and low no of transaction.

#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)

```
WITH CTE1 as (
        SELECT
        T.household_key as most_spending_customer
        FROM `Ecommerce.transaction` T
        JOIN `Ecommerce.hh_demographic` D
        ON T.household_key=D.household_key
        GROUP BY T.household_key
        ORDER BY SUM(SALES_VALUE) DESC
        LIMIT 1)
SELECT
        most_spending_customer,
        *
FROM CTE1 C1
JOIN `Ecommerce.hh_demographic` D
on C1.most_spending_customer = D.household_key
```

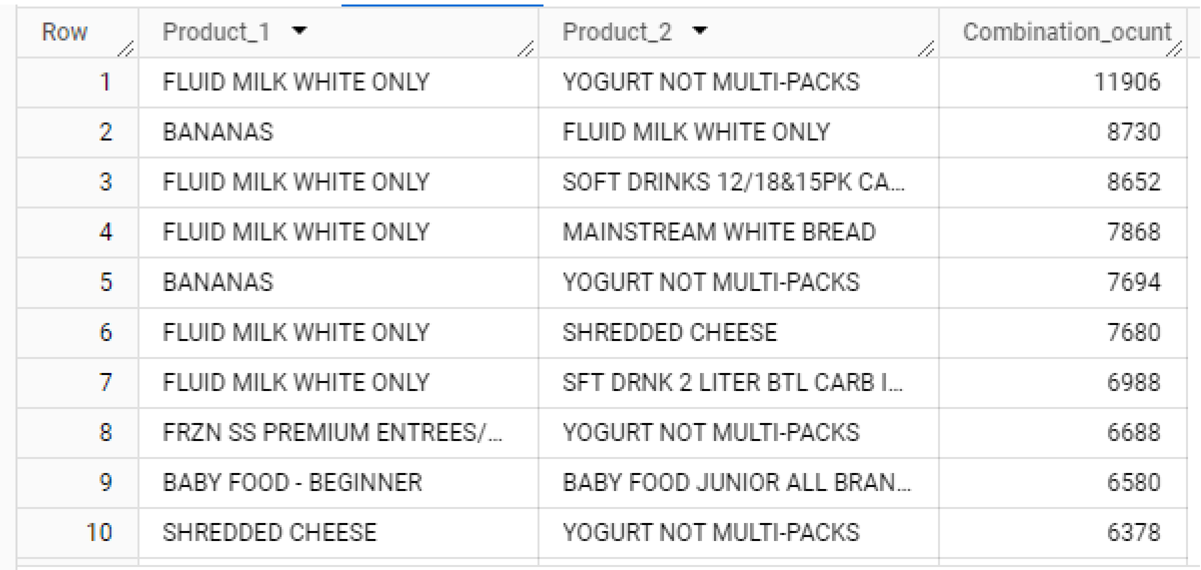


Row	most_spending_customer	AGE_DESC	HOMEOWNER_DESC	HOUSEHOLD_SIZE_DESC	INCOME_DESC	KID_CATEGORY_DESC	MARITAL_STATUS_CODE
1	1609	45-54	Homeowner	5+	125-149K	3+	A

Insight: The Customer 1609 is the highest spending customer who falls in age group 45-54, is homeowner with income between 125-149k and have 3+ kids.

Question 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)

```
SELECT
  LEAST(p1.SUB_COMMODITY_DESC, p2.SUB_COMMODITY_DESC) as Product_1,
  GREATEST(p1.SUB_COMMODITY_DESC, p2.SUB_COMMODITY_DESC) as Product_2,
  COUNT(*) as Combination_count
FROM `Ecommerce.transaction` as t1
join `Ecommerce.transaction` as t2
on t1.BASKET_ID=t2.BASKET_ID and t1.PRODUCT_ID<>t2.PRODUCT_ID
join `Ecommerce.product` as p1 on t1.PRODUCT_ID=p1.PRODUCT_ID
join `Ecommerce.product` as p2 on t2.PRODUCT_ID=p2.PRODUCT_ID
WHERE p1.SUB_COMMODITY_DESC <> p2.SUB_COMMODITY_DESC
GROUP BY
  LEAST (p1.SUB_COMMODITY_DESC, p2.SUB_COMMODITY_DESC),
  GREATEST (p1.SUB_COMMODITY_DESC, p2.SUB_COMMODITY_DESC)
ORDER BY
  Combination_count DESC
```

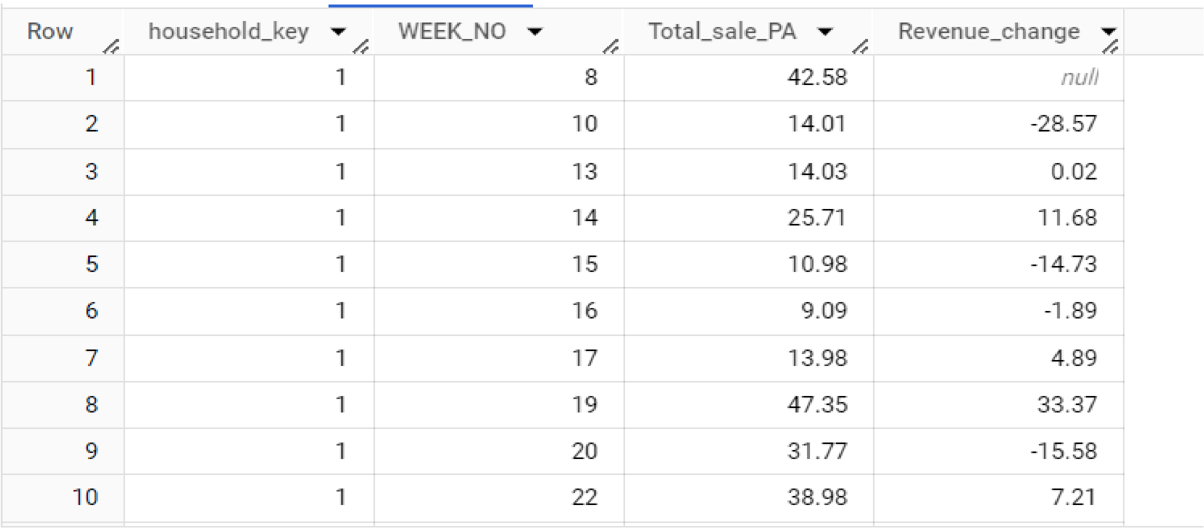


Row	Product_1	Product_2	Combination_ocunt
1	FLUID MILK WHITE ONLY	YOGURT NOT MULTI-PACKS	11906
2	BANANAS	FLUID MILK WHITE ONLY	8730
3	FLUID MILK WHITE ONLY	SOFT DRINKS 12/18&15PK CA...	8652
4	FLUID MILK WHITE ONLY	MAINSTREAM WHITE BREAD	7868
5	BANANAS	YOGURT NOT MULTI-PACKS	7694
6	FLUID MILK WHITE ONLY	SHREDDED CHEESE	7680
7	FLUID MILK WHITE ONLY	SFT DRNK 2 LITER BTL CARB I...	6988
8	FRZN SS PREMIUM ENTREES/...	YOGURT NOT MULTI-PACKS	6688
9	BABY FOOD - BEGINNER	BABY FOOD JUNIOR ALL BRAN...	6580
10	SHREDDED CHEESE	YOGURT NOT MULTI-PACKS	6378

Based on the above result we should recommend the products that are bought together more frequently when any one the products is being bought.

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

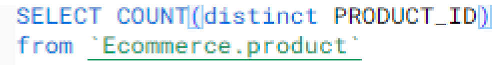
```
SELECT
 household_key, WEEK_NO,
 ROUND(SUM(SALES_VALUE),2) AS Total_sale_PA,
 ROUND(SUM(SALES_VALUE) - LAG(SUM(SALES_VALUE)) OVER (PARTITION BY household_key
ORDER BY WEEK_NO),2) AS Revenue_change
FROM `Ecommerce.transaction`
GROUP BY household_key, WEEK_NO
ORDER BY household_key, WEEK_NO
```



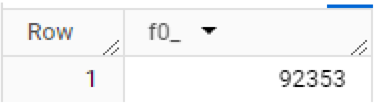
Row	household_key	WEEK_NO	Total_sale_PA	Revenue_change
1	1	8	42.58	null
2	1	10	14.01	-28.57
3	1	13	14.03	0.02
4	1	14	25.71	11.68
5	1	15	10.98	-14.73
6	1	16	9.09	-1.89
7	1	17	13.98	4.89
8	1	19	47.35	33.37
9	1	20	31.77	-15.58
10	1	22	38.98	7.21

Additional Question:

1. No of Distinct products being sold



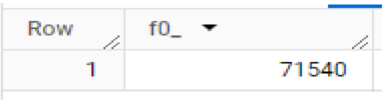
```
SELECT COUNT(distinct PRODUCT_ID)
from `Ecommerce.product`
```



Row	f0_
1	92353

Total of 92353 products are there for sale, now we will check count of products on which transaction happed

```
SELECT COUNT(DISTINCT T.PRODUCT_ID)
FROM `Ecommerce.product` as P
JOIN `Ecommerce.transaction` as T
ON P.PRODUCT_ID=T.PRODUCT_ID;
```

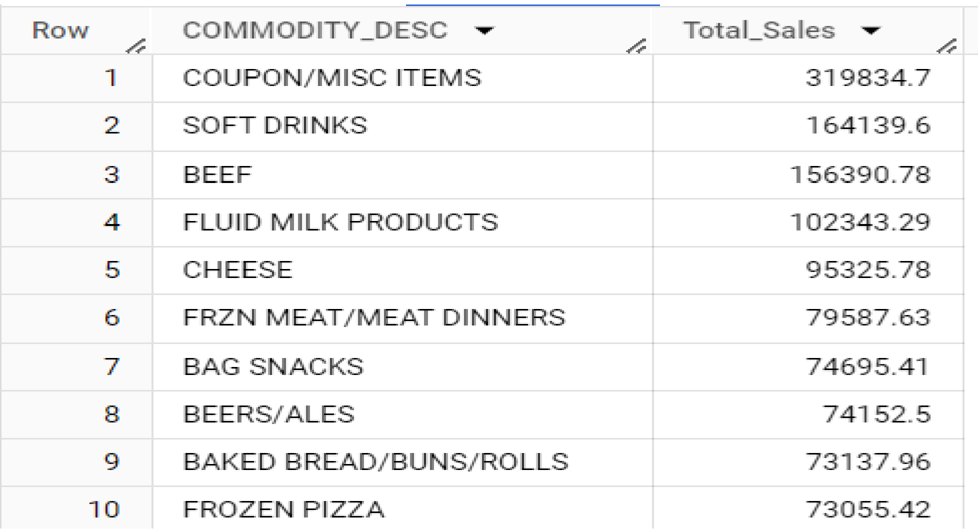


Row	f0_
1	71540

Of total 92353 products only on 71540 at least one transaction happened, we should think of discontinuing the remaining products with no transaction.

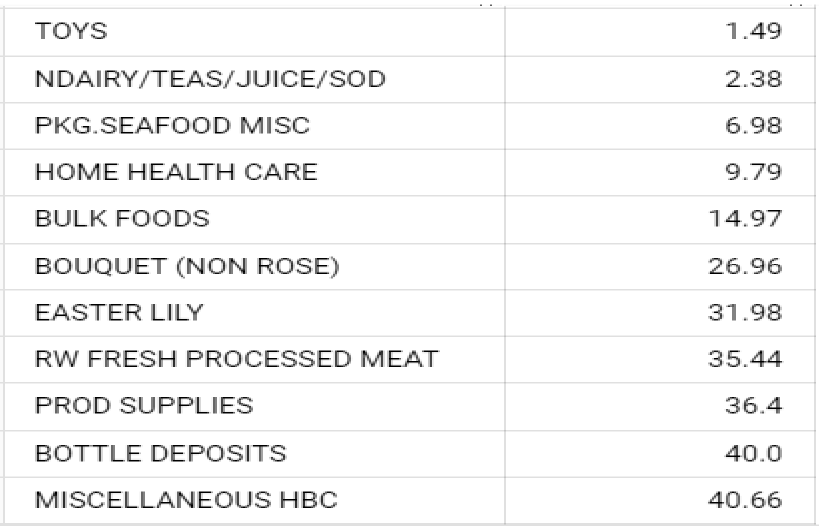
2. Top 10 and bottom 10 products by sales:

```
SELECT
COMMODITY_DESC, ROUND(SUM(SALES_VALUE),2) AS Total_Sales
FROM `Ecommerce.transaction` T
JOIN `Ecommerce.product` D
ON T.PRODUCT_ID = D.PRODUCT_ID
GROUP BY COMMODITY_DESC
ORDER BY Total_Sales DESC
```



Row	COMMODITY_DESC	Total_Sales
1	COUPON/MISC ITEMS	319834.7
2	SOFT DRINKS	164139.6
3	BEEF	156390.78
4	FLUID MILK PRODUCTS	102343.29
5	CHEESE	95325.78
6	FRZN MEAT/MEAT DINNERS	79587.63
7	BAG SNACKS	74695.41
8	BEERS/ALES	74152.5
9	BAKED BREAD/BUNS/ROLLS	73137.96
10	FROZEN PIZZA	73055.42

```
SELECT
COMMODITY_DESC, ROUND(SUM(SALES_VALUE),2) AS Total_Sales
FROM `Ecommerce.transaction` T
JOIN `Ecommerce.product` D
ON T.PRODUCT_ID = D.PRODUCT_ID
GROUP BY COMMODITY_DESC
ORDER BY Total_Sales
```

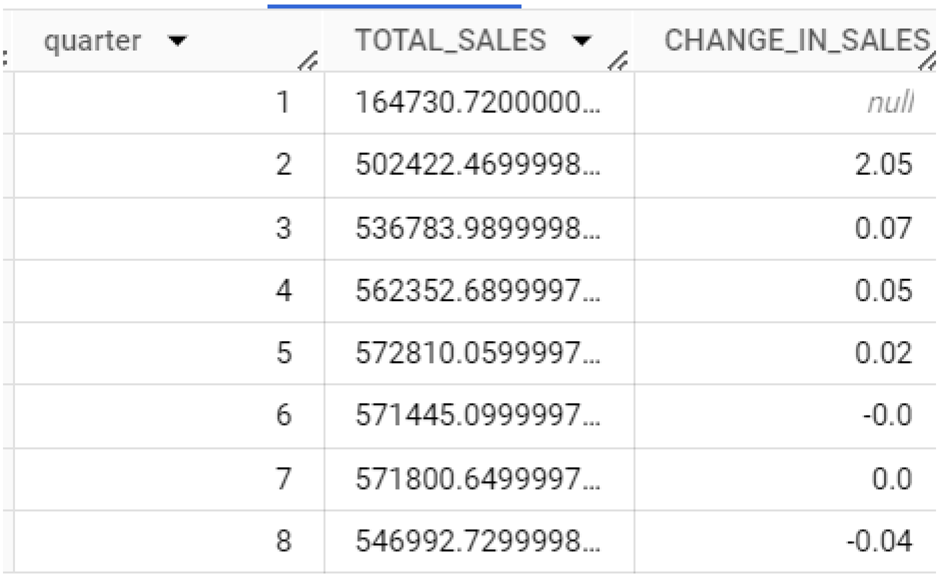


TOYS	1.49
NDAIRY/TEAS/JUICE/SOD	2.38
PKG.SEAFOOD MISC	6.98
HOME HEALTH CARE	9.79
BULK FOODS	14.97
BOUQUET (NON ROSE)	26.96
EASTER LILY	31.98
RW FRESH PROCESSED MEAT	35.44
PROD SUPPLIES	36.4
BOTTLE DEPOSITS	40.0
MISCELLANEOUS HBC	40.66

We should focus on these products, to see why sales on these products are low and should discontinue these products if required.

3. Quarterly sales analysis

```
WITH CTE AS(SELECT *,
CASE
  when WEEK_NO between 0 and 12 then 1
  when WEEK_NO between 13 and 25 then 2
  when WEEK_NO between 26 and 38 then 3
  when WEEK_NO between 39 and 51 then 4
  when WEEK_NO between 52 and 64 then 5
  when WEEK_NO between 65 and 77 then 6
  when WEEK_NO between 78 and 90 then 7
  when WEEK_NO between 91 and 102 then 8
END AS quarter
from `Ecommerce.transaction`)
SELECT quarter, SUM(SALES_VALUE) AS TOTAL_SALES,
ROUND((SUM(SALES_VALUE) - LAG(SUM(SALES_VALUE)) OVER(ORDER BY quarter))/
LAG(SUM(SALES_VALUE)) OVER(ORDER BY quarter),2) AS CHANGE_IN_SALES
FROM CTE
GROUP BY quarter
ORDER BY quarter
```



quarter	TOTAL_SALES	CHANGE_IN_SALES
1	164730.7200000...	*null*
2	502422.4699998...	2.05
3	536783.9899998...	0.07
4	562352.6899997...	0.05
5	572810.0599997...	0.02
6	571445.0999997...	-0.0
7	571800.6499997...	0.0
8	546992.7299998...	-0.04

After first quarter sales have been study but not increasing significantly which is somewhat concerning, we should look in it.