

E-commerce SQL Analysis

Problem Statement

Analyzing 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.

1. Basic Structure and Data type of columns in the Data:

Extracting distinct Column names in all the tables and their Data types. It can be done for each table as well with filter for 'table_name' using 'Where'.

Query:

```
select distinct column_name, data_type
from `ecomm`.`INFORMATION_SCHEMA.COLUMNS
```

Result:

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS	EXEC
Row	column_name	data_type					
1	PRODUCT_ID	INT64					
2	MANUFACTURER	INT64					
3	DEPARTMENT	STRING					
4	BRAND	STRING					
5	COMMODITY_DESC	STRING					
6	SUB_COMMODITY_DESC	STRING					
7	CURR_SIZE_OF_PRODUCT	STRING					
8	int64_field_0	INT64					
9	household_key	INT64					
10	BASKET_ID	INT64					
11	DAY	INT64					
12	QUANTITY	INT64					
13	SALES_VALUE	FLOAT64					
14	STORE_ID	INT64					

Results per page: 50 ▼ 1 - 26 of 26

2. 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

    Order_Value_Category,

    COUNT(DISTINCT BASKET_ID) AS Order_Count

FROM (

    SELECT

        BASKET_ID,

        CASE

            WHEN SUM(SALES_VALUE) BETWEEN 0 AND 10 THEN 'Small`
            WHEN SUM(SALES_VALUE) BETWEEN 10 AND 20 THEN 'Medium`
            WHEN SUM(SALES_VALUE) BETWEEN 20 AND 100 THEN 'Large'
            WHEN SUM(SALES_VALUE) BETWEEN 100 AND 200 THEN 'High'
            ELSE 'Very High'

        END AS Order_Value_Category

    FROM `ecomm.transaction`

    GROUP BY BASKET_ID

)

GROUP BY Order_Value_Category
```

Result:

Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW
Row	Order_Value_Category	Order_Count		
1	Large	65381		
2	Small	116415		
3	Medium	49630		
4	High	1860		
5	Very High	70		

3. 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
  Order_Value_Category,
  COUNT(DISTINCT BASKET_ID) AS Order_Count
FROM (
  SELECT
    BASKET_ID,
    CASE
      WHEN SUM(SALES_VALUE) BETWEEN 0 AND 5 THEN 'Very Small'
      WHEN SUM(SALES_VALUE) BETWEEN 5 AND 10 THEN 'Small'
      WHEN SUM(SALES_VALUE) BETWEEN 10 AND 20 THEN 'Medium'
      WHEN SUM(SALES_VALUE) BETWEEN 20 AND 50 THEN 'Large'
      WHEN SUM(SALES_VALUE) BETWEEN 50 AND 100 THEN 'High'
      ELSE 'Very High'
    END AS Order_Value_Category
  FROM `ecomm.transaction`
  GROUP BY BASKET_ID
)
GROUP BY Order_Value_Category
```

Result:

JOB INFORMATION		RESULTS	CHART	PREVIEW
Row	Order_Value_Category	Order_Count		
1	Large	51262		
2	Very Small	70842		
3	Medium	49630		
4	Small	45573		
5	Very High	1930		
6	High	14119		

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

Query:

```
WITH StoreFootTraffic AS (
```

```

SELECT

    WEEK_NO,

    STORE_ID,

    COUNT(DISTINCT household_key) AS Foot_Traffic,

    ROW_NUMBER() OVER (PARTITION BY WEEK_NO ORDER BY COUNT(DISTINCT
household_key) DESC) AS Rank

FROM `ecomm.transaction`

GROUP BY WEEK_NO, STORE_I
)

```

```

SELECT

    WEEK_NO,
    STORE_ID,
    Foot_Traffic

FROM StoreFootTraffic

WHERE Rank <= 3

order by WEEK_NO

```

Result:

Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON
Row	WEEK_NO	STORE_ID	Foot_Traffic		
1	1	32004	5		
2	1	324	3		
3	1	367	3		
4	2	32004	7		
5	2	313	6		
6	2	367	5		
7	3	367	10		
8	3	32004	9		
9	3	356	8		
10	4	367	17		
11	4	32004	11		
12	4	446	8		
13	5	367	15		
14	5	32004	12		

#Extra- Frequency of Coming in Top 3

Query:

```
WITH StoreFootTraffic AS (  
  
    SELECT  
  
        WEEK_NO,  
  
        STORE_ID,  
  
        COUNT(DISTINCT household_key) AS Foot_Traffic,  
  
        ROW_NUMBER() OVER (PARTITION BY WEEK_NO ORDER BY COUNT(DISTINCT  
household_key) DESC) AS Rank  
  
    FROM `ecomm.transaction`  
  
    GROUP BY WEEK_NO, STORE_ID  
  
),  
  
Top3Stores AS (  
  
    SELECT  
  
        WEEK_NO,  
  
        STORE_ID,  
  
        Foot_Traffic,  
  
        Rank  
  
    FROM StoreFootTraffic  
  
    WHERE Rank <= 3  
  
)  
  
SELECT  
  
    STORE_ID,  
  
    COUNT(*) AS Times_In_Top3
```

```
FROM Top3Stores

GROUP BY STORE_ID

ORDER BY Times_In_Top3 DESC

LIMIT 3
```

Result:

JOB INFORMATION		RESULTS	CHAR
Row	STORE_ID	Times_In_Top3	
1	367	102	
2	343	55	
3	381	39	

5. 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 DISTINCT household_key ,  
  
NTH_VALUE(DAY,1) OVER(partition by household_key order by DAY,TRANS_TIME  
ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) min_day,  
  
NTH_VALUE(TRANS_TIME,1) OVER(partition by household_key order by  
DAY,TRANS_TIME ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)  
min_time,  
  
NTH_VALUE(DAY,1) OVER(partition by household_key order by DAY  
DESC,TRANS_TIME DESC ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED  
FOLLOWING) max_day,  
  
NTH_VALUE(TRANS_TIME,1) OVER(partition by household_key order by DAY  
DESC,TRANS_TIME DESC ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED  
FOLLOWING) max_time,  
  
COUNT(DISTINCT BASKET_ID) OVER(partition by household_key) num_visits ,  
  
Round(SUM(SALES_VALUE) OVER(partition by household_key) / COUNT(DISTINCT  
BASKET_ID) OVER(partition by household_key),2) avg_money_spend ,  
  
Round(SUM(SALES_VALUE) OVER(partition by household_key),2) as  
total_money_spend  
  
from `ecomm.transaction`  
  
order by avg_money_spend DESC;
```

Result:

Query results

[SAVE RESULTS](#)

[EXPLORE DATA](#)

JOB INFORMATION		RESULTS		CHART	PREVIEW	JSON	EXECUTION DETAILS		EXECUTION GRAPH	
Row	household_key	min_day	min_time	max_day	max_time	num_visits	avg_money_spend	total_money_spend		
1	2042	52	1842	683	1618	26	89.97	2339.21		
2	973	95	2128	710	2052	80	85.95	6875.89		
3	1899	20	1359	705	957	69	83.91	5789.59		
4	1900	111	1416	707	1318	55	76.87	4227.72		
5	1574	107	1137	651	1437	27	68.27	1843.3		
6	1315	60	2221	624	1636	5	63.48	317.39		
7	2479	111	922	706	1812	111	62.65	6954.64		
8	931	94	1245	668	1842	40	61.38	2455.29		
9	1344	87	1538	691	1722	26	60.4	1570.37		
10	248	29	1415	704	1634	53	58.32	3090.89		
11	688	70	1345	692	1434	27	57.74	1558.95		
12	1864	103	1358	710	1332	148	57.68	8537.28		
13	1848	105	1952	706	1502	97	57.34	5561.56		
14	1727	109	2129	118	2139	2	57.26	114.51		

- 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 cte as (

select household_key , ROUND(SUM(IFNULL(SALES_VALUE,0)),2)
total_spend_amount, SUM(IFNULL(QUANTITY,0)) total_qty

from `ecommm.transaction`

group by household_key

)

select a.* ,b.*




from cte a join `ecommm.demographic` b on a.household_key = b.household_key
```



```
order by a.total_spend_amount DESC
```

```
limit 1
```

Result:

Query results							 SAVE RESULTS ▾	 EXPLORE DATA ▾	
JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS	EXECUTION GRAPH		
Row	household_key ▾	total_spend_amount ▾	total_qty ▾	AGE_DESC ▾	MARITAL_STATUS_CODE ▾	INCOME_DESC ▾			
1	1609	13804.38	932787	45-54	A	125-149K			

- 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 ProductPairs AS (
```

```
SELECT
```

```
    a.PRODUCT_ID AS Product_1,
```

```
    b.PRODUCT_ID AS Product_2,
```

```
    COUNT(*) AS Frequency
```

```
FROM `ecomm.transaction` a
```

```
JOIN `ecomm.transaction` b ON a.BASKET_ID = b.BASKET_ID AND a.PRODUCT_ID <
b.PRODUCT_ID
```

```
GROUP BY a.PRODUCT_ID, b.PRODUCT_ID
```

```
)
```

```
SELECT
```

```
    p1.SUB_COMMODITY_DESC AS Product_1_Desc,
```

```
    p2.SUB_COMMODITY_DESC AS Product_2_Desc,
```

Frequency

FROM ProductPairs pp

JOIN `ecomm.product` p1 ON pp.Product_1 = p1.PRODUCT_ID

JOIN `ecomm.product` p2 ON pp.Product_2 = p2.PRODUCT_ID

ORDER BY Frequency DESC

Result:

Query results

 [SAVE](#)

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION
Row	Product_1_Desc	Product_2_Desc	Frequency			
1	FLUID MILK WHITE ONLY	BANANAS	848			
2	FLUID MILK WHITE ONLY	BANANAS	728			
3	EGGS - X-LARGE	BANANAS	625			
4	BANANAS	STRAWBERRIES	611			
5	BANANAS	FLUID MILK WHITE ONLY	519			
6	CARROTS MINI PEELED	BANANAS	473			
7	MAINSTREAM WHITE BREAD	BANANAS	458			
8	FLUID MILK WHITE ONLY	BANANAS	430			
9	BANANAS	FLUID MILK WHITE ONLY	427			
10	HAMBURGER BUNS	HOT DOG BUNS	421			

Results per page: 50 ▼

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

Query:

```
WITH WeeklySpending AS (  
  
    SELECT  
  
        household_key,  
  
        WEEK_NO,  
  
        SUM(SALES_VALUE) AS Weekly_Spending  
  
    FROM `ecomm.transaction`  
  
    GROUP BY household_key, WEEK_NO  
  
),  
  
LaggedSpending AS (  
  
    SELECT  
  
        household_key,  
  
        WEEK_NO,  
  
        Weekly_Spending,  
  
        LAG(Weekly_Spending) OVER(PARTITION BY household_key ORDER BY WEEK_NO)  
    AS    Lagged_Spending  
  
    FROM WeeklySpending  
  
)  
  
SELECT  
  
    household_key,  
  
    WEEK_NO,  
  
    round(Weekly_Spending - Lagged_Spending,2) AS Weekly_Change_RPA  
  
FROM LaggedSpending  
  
WHERE Lagged_Spending IS NOT NULL
```

order by Weekly_Change_RPA desc

Result:

Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW
Row	household_key	WEEK_NO	Weekly_Change_RPA	
1	1609	49	944.87	
2	1023	76	441.95	
3	2080	29	401.27	
4	2395	94	378.16	
5	1767	94	345.54	
6	1023	52	331.17	
7	1592	92	328.03	
8	1852	65	323.19	
9	788	37	320.52	
10	2266	62	310.89	

Res


9. Customer Retention Rate Analysis:

```
SELECT
    household_key,
    COUNT(DISTINCT WEEK_NO) AS Active_Weeks,
    MAX(WEEK_NO) - MIN(WEEK_NO) + 1 AS Total_Weeks,
    COUNT(DISTINCT WEEK_NO) / (MAX(WEEK_NO) - MIN(WEEK_NO) + 1) * 100 AS
Retention_Rate
FROM `ecommm.transaction`
```

GROUP BY household_key

ORDER BY household_key;

Result:

Query results						 SAVE RESULTS
JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS
Row	household_key	Active_Weeks	Total_Weeks	Retention_Rate		
1	1	66	95	69.47368421052...		
2	2	35	82	42.68292682926...		
3	3	37	85	43.52941176470...		
4	4	26	75	34.66666666666...		
5	5	25	89	28.08988764044...		
6	6	84	85	98.82352941176...		
7	7	47	99	47.47474747474...		
8	8	71	88	79.54545454545...		
Results per page:						50 ▼

10. Customer Churn Analysis:

```
WITH customer_last_purchase AS (  
  SELECT  
    household_key,  
    MAX(DAY) AS Last_Purchase_Day  
  FROM `ecomm.transaction`  
  GROUP BY household_key  
)  
  
SELECT  
  household_key,  
  Last_Purchase_Day,  
  DATE_DIFF(CURRENT_DATE(), Last_Purchase_Day, DAY) AS Days_Since_Last_Purchase  
FROM customer_last_purchase  
ORDER BY household_key;
```

11. Product Performance Over Time:

```
SELECT  
  WEEK_NO,  
  SUB_COMMODITY_DESC,
```

```
SUM(SALES_VALUE) AS Total_Sales
FROM `ecomm.transaction` AS t
JOIN `ecomm.product` AS p ON t.PRODUCT_ID = p.PRODUCT_ID
GROUP BY WEEK_NO, SUB_COMMODITY_DESC
ORDER BY WEEK_NO, Total_Sales DESC;
```

Query results

SAVE RESULTS

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS
Row	WEEK_NO	SUB_COMMODITY_DESC	Total_Sales			
1	1	SOFT DRINKS 12/18&15PK CA...	76.34999999999...			
2	1	FLUID MILK WHITE ONLY	51.890000000000...			
3	1	SNACKS/APPETIZERS	39.62			
4	1	BEERALEMALT LIQUORS	33.96			
5	1	CIGARETTES	32.38			
6	1	DAIRY CASE 100% PURE JUICE...	30.960000000000...			
7	1	PREMIUM	26.869999999999...			
8	1	SEAFOOD-FRE-NON RW-ALL	26.46			
9	1	MAINSTREAM WHITE BREAD	26.05			
10	1	FRZNR BBD STICK/DORTON	24.6			

Results per page: 50 1

12. Geographic Analysis:

```
SELECT

    d.HOMEOWNER_DESC,

    d.HOUSEHOLD_SIZE_DESC,

    COUNT(*) AS Customer_Count,

    SUM(t.SALES_VALUE) AS Total_Sales

FROM `ecomm.transaction` t

LEFT JOIN `ecomm.demographic` d ON t.household_key = d.household_key

GROUP BY d.HOMEOWNER_DESC, d.HOUSEHOLD_SIZE_DESC

ORDER BY Total_Sales DESC;
```

Query results

SAVE RESULTS

EXPLAIN

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS		EXECUTION
Row	HOMEOWNER_DESC	HOUSEHOLD_SIZE_DESC	Customer_Count	Total_Sales				
1	null	null	584629	1779261.559997...				
2	Homeowner	2	208394	688995.6199996...				
3	Unknown	1	110775	325434.9299999...				
4	Homeowner	1	77628	253355.9699999...				
5	Homeowner	3	77279	246607.5699999...				
6	Homeowner	5+	58335	195555.1100000...				
7	Unknown	2	46294	138747.2500000...				
8	Homeowner	4	41210	134652.3200000...				

Insights and Recommendations:

Insights & Recommendations:

Order Value Segmentation:

- Small and medium-sized orders represent the majority of transactions, indicating potential bundling or promotional opportunities for increasing the average order value.
- Recommend creating targeted marketing campaigns for encouraging larger basket sizes, possibly through loyalty programs or bundle discounts.

Store Foot Traffic Analysis:

- Identifying stores with high foot traffic helps understand customer preferences in different locations.
- Recommend optimizing inventory and promotional strategies in these high-traffic stores to capitalize on customer presence.

Customer Profiling:

- Customers with higher average spending might be ideal for loyalty programs or premium offerings.
- We can Plan targeted marketing strategies tailored to high-spending customers to increase retention and drive higher sales.

Product Association Analysis:

- Identifying products frequently bought together allows for strategic placement or bundling.
- Create product bundles or placing associated products nearby to encourage cross-selling and increase average order value.

Revenue Per Account (RPA) Analysis:

- Understanding weekly changes in spending per customer provides insights into customer behavior.
- Recommend targeted offers or personalized recommendations to retain customers during weeks when spending decreases.

Diversify Product Offerings Beyond Top Departments:

- Expand the product range across various departments beyond the top-selling ones to enhance product diversity and reduce dependency on a few major categories.

Early Sales Events and Promotions:

- Initiate early sales events (Weeks 1 to 10) to drive increased sales momentum and capture customer attention from the outset of each quarter or fiscal year.

Attracting High-Income Bracket Customers:

- Introduce premium or exclusive products appealing to customers with higher incomes (>100K) to cater to their preferences and boost sales among this segment.

Targeting Younger Demographics (Age Group 19-24):

- Attract younger customers by offering trendy and affordable products specifically designed for this age group's preferences.
- Provide student discounts, cashbacks, or special offers to entice the younger demographic segment to engage more with the platform.

Strategic Marketing Timing for Increased Orders:

- Increase order count during morning and night times by implementing limited-time discounts or flash sales tailored to these timeframes.
- Organize significant sale events with early morning or late-night starts to stimulate immediate purchases on a first-come, first-serve basis.

Enhance Order Value Segmentation Strategies:

- Further explore strategies to encourage larger basket sizes, such as implementing targeted bundle discounts, personalized promotions, or loyalty programs aimed at increasing the average order value.

Optimize Inventory and Promotions in High-Traffic Stores:

- Invest in high-traffic store locations by optimizing inventory, enhancing promotional strategies, and offering unique in-store experiences to capitalize on customer foot traffic.

Leverage Product Association Insights for Cross-Selling:

- Leverage product association analysis to create strategic product bundles or place associated products together to promote cross-selling opportunities and drive up the average order value.

Personalized Marketing for High-Spending Customers:

- Develop personalized marketing strategies tailored to high-spending customers to foster loyalty, increase retention, and drive higher sales through exclusive offers or premium services.

Monitor Revenue Per Account (RPA) Trends for Customer Retention:

- Continuously analyze weekly changes in spending per customer (RPA) to identify behavioral patterns and proactively offer targeted offers or personalized recommendations to retain customers during periods of reduced spending.

Extra Recomm:

Customer Retention: Implement targeted strategies to retain customers during inactive periods, such as personalized offers or loyalty programs.

Churn Prevention: Launch re-engagement campaigns for customers who haven't made recent purchases, offering exclusive promotions to win them back.

Product Strategy: Optimize inventory and marketing based on high-performing product categories, adjusting pricing or promotions for better results.

Geographic Focus: Tailor marketing messages and product offerings based on regional and demographic preferences to boost sales.

Segmented Marketing: Develop personalized marketing approaches using demographic and behavioral insights to effectively target customer segments.