

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.

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 category as (  
  select Total_Sales,   
  case when Total_Sales between 0 and 9 then 'Small $0-$9'  
  when Total_Sales between 10 and 19 then 'Medium $10-$19'  
  when Total_Sales > 19 then 'Large $20+'  
  end as Orders_Category  
  from `e_com.transactions`  
)  
  
select Orders_Category, count(Total_Sales) as total_order  
from category  
where Orders_Category is not null  
group by Orders_Category;
```

JOB INFORMATION		RESULTS	CHART
Row	Orders_Category	total_order	
1	Small \$0-\$9	991736	
2	Medium \$10-\$19	29944	
3	Large \$20+	15269	

- **Insight: -** A large proportion of orders fall under the "medium" range (\$10-\$20), indicating that customers frequently purchase items within this price bracket. This could reflect a sweet spot in pricing for most products.

➤ **Recommendation: -**

1. Consider running promotions or discounts on high-value items to shift some of the medium-order customers into the large-order category.
2. Create bundling offers for small-order items to encourage customers to increase their spending.

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 category as (  
  select Total_Sales,  
  case when Total_Sales between 0 and 4 then 'Small $0 - $4'  
  when Total_Sales between 5 and 9 then 'Medium $5 - $10'  
  when Total_Sales > 9 then 'Large $10+'  
  end as Orders_Category  
  from `e_com.transactions`  
)  
  
select Orders_Category, count(Total_Sales) as total_order  
from category  
where Orders_Category is not null  
group by Orders_Category;
```

JOB INFORMATION		RESULTS	CHART	JSON
Row	Orders_Category	total_order		
1	Small \$0 - \$4	803360		
2	Medium \$5 - \$10	122302		
3	Large \$10+	56832		

- **Insight: -** The high volume of medium and large orders highlights that customers are willing to invest more in products over \$5, possibly indicating a preference for value-added or premium offerings.
- **Recommendation: -** Medium and large orders dominate, so further optimize the product mix in these ranges by offering discounts, loyalty programs, or exclusive deals to retain and grow these customers.

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

```
with foot_traffic as (
  select Store_ID, Week_No, count(*) as customer_transacting,
  dense_rank() over(partition by Week_No order by count(*) desc) as rn
  from `e_com.transactions`
  where Sales_Value is not null or Sales_Value = 0
  group by Store_ID, Week_No
)
select Week_No, Store_ID
from foot_traffic
where rn <= 3
order by Week_No asc;
```

JOB INFORMATION		RESULTS	CHART
Row	Week_No	Store_ID	
1	1	324	
2	1	32004	
3	1	296	
4	2	315	
5	2	375	
6	2	403	
7	3	375	
8	3	367	
9	3	408	
10	4	367	
11	4	32004	

- **Insight: -** The greatest number of transaction customer done is on the Store with ID 367 followed by Store ID 361 and 357

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 customer_first_visit,
max(DAY) as customer_last_visit, count(distinct DAY) as customer_number_of_visit,
round(avg(Total_Sales),1) as customer_average_spent_per_visit,
round(sum(Total_Sales),1) as customer_total_spent
from 'e_com.transactions'
group by Household_Key
order by 1 asc;
```

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	Household_Key	customer_first_visit	customer_last_visit	customer_number_of	customer_average_s	customer_total_spen
1	1	2000-02-20	2001-12-06	72	2.9	2473.2
2	2	2000-04-12	2001-10-29	44	3.5	1235.7
3	3	2000-04-22	2001-12-03	45	3.9	1789.0
4	4	2000-04-13	2001-09-18	29	4.6	693.5
5	5	2000-03-25	2001-12-03	29	4.3	477.3
6	6	2000-04-27	2001-12-07	201	3.7	3358.3
7	7	2000-01-23	2001-12-09	56	3.1	1996.3
8	8	2000-03-05	2001-12-06	102	3.4	3380.3
9	9	2000-04-13	2001-11-19	18	4.5	496.5
10	10	2000-04-23	2001-11-15	5	3.6	125.9

- **Insight: -**

The query is extraction the customer ID which is Household Key and what was the first visit, last visit, how many times he/she visits, the average amount he/she spends whenever he/she visits the store and the total amount he/she spends

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 most_spending_table as (
select dem.household_key, dem.age_bucket, dem.marital_status, dem.income_bucket, dem.homeowner,
dem.household_comp, dem.household_size, dem.kid_category,
sum(Total_Sales) as most_spend
from e_com.demographics dem
left join e_com.transactions tra on dem.household_key = tra.Household_Key
group by dem.household_key, dem.age_bucket, dem.marital_status, dem.income_bucket, dem.homeowner,
dem.household_comp, dem.household_size, dem.kid_category
)
select household_key, age_bucket, marital_status, income_bucket, homeowner, household_comp,
household_size, kid_category
from most_spending_table
where most_spend = (
select max(most_spend)
from most_spending_table
);
```

JOB INFORMATION								
RESULTS								
Row	household_key	age_bucket	marital_status	income_bucket	homeowner	household_comp	household_size	kid_category
1	1609	45-54	Married	125-149K	Homeowner	2 Marriedduits Kids	5+	3+

➤ **Insight: -**

1. The customer who spent most amount is with the household key 1609 and we can see their demographics.
2. They are in the age bucket of 45-54, married.
3. Their income is between \$125 - \$149. They are a homeowner

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 p1.SUB_COMMODITY_DESC AS product_A,
       p2.SUB_COMMODITY_DESC AS product_B,
       COUNT(*) AS pair_count
FROM `e_com.transactions` t1
JOIN `e_com.products` p1 ON t1.product_id = p1.product_id
JOIN `e_com.transactions` t2 ON t1.Basket_ID = t2.Basket_ID
      AND t1.product_id < t2.product_id
JOIN `e_com.products` p2 ON t2.product_id = p2.product_id
GROUP BY p1.SUB_COMMODITY_DESC, p2.SUB_COMMODITY_DESC
ORDER BY pair_count DESC;
```

Query results

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	product_A	product_B	pair_count			
1	YOGURT NOT MULTI-PACKS	YOGURT NOT MULTI-PACKS	12773			
2	BABY FOOD - BEGINNER	BABY FOOD - BEGINNER	8171			
3	SS ECONOMY ENTREES/DINN...	SS ECONOMY ENTREES/DINN...	5413			
4	SOFT DRINK POWDER POUCHES	SOFT DRINK POWDER POUCHES	5192			
5	FRZN SS PREMIUM ENTREES/...	FRZN SS PREMIUM ENTREES/...	5139			
6	SFT DRINK 2 LITER BTL CARB I...	SFT DRINK 2 LITER BTL CARB I...	4474			
7	SOFT DRINKS 12/18&15PK CA...	SOFT DRINKS 12/18&15PK CA...	4391			
8	CANDY BARS (SINGLES)(INCL...	CANDY BARS (SINGLES)(INCL...	3442			
9	CANNED CAT FOOD (9 LIVES/F...	CANNED CAT FOOD (9 LIVES/F...	3076			
10	FLUID MILK WHITE ONLY	SOFT DRINKS 12/18&15PK CA...	2911			
11	FLUID MILK WHITE ONLY	YOGURT NOT MULTI-PACKS	2815			

➤ **Insights:**

1. The above query output shows which product is brought together frequently and how much time it was purchased by the customer.
2. These output we can use to increase the number of products in our inventory which is highest and medium selling

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

```

1 WITH weekly_revenue AS (
2   SELECT
3     Household_Key,
4     Week_No AS week_start,
5     round(SUM(Total_Sales)) AS total_spending
6   FROM `e-com.transactions`
7   GROUP BY Household_Key, Week_No
8 )
9
10 SELECT
11   Household_Key,
12   week_start,
13   total_spending,
14   round(LAG(total_spending) OVER (PARTITION BY Household_Key ORDER BY week_start)) AS
    previous_week_spending,
15   (total_spending - round(LAG(total_spending) OVER (PARTITION BY Household_Key ORDER BY week_start))) AS
    weekly_change_in_rpa
16 FROM weekly_revenue
17 ORDER BY Household_key, week_start;

```

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	Household_Key	week_start	total_spending	previous_week_spending	weekly_change_in_rpa	
1	1	8	49.0	null	null	
2	1	10	20.0	49.0	-29.0	
3	1	13	15.0	20.0	-5.0	
4	1	14	30.0	15.0	15.0	
5	1	15	15.0	30.0	-15.0	
6	1	16	15.0	15.0	0.0	
7	1	17	16.0	15.0	1.0	
8	1	19	50.0	16.0	34.0	
9	1	20	40.0	50.0	-10.0	
10	1	22	48.0	40.0	8.0	
11	1	23	31.0	48.0	-17.0	
12	1	24	36.0	31.0	5.0	

➤ **Insights:**

1. The above query shows the weekly change in revenue per account. For every customer which is household key we can see the revenue changing weekly which we can use for the inventory stocks.
2. How much revenue is generated each week, what percentage of revenue we incline or decline every week

➤ **Recommendation:**

1. Most of the products are sold within the category of small, so we need to focus of medium and large category of products as well.
2. We can combine the bundle of small with medium and small with large and then we can sell, so that the inventory will not have any expired products.
3. The top revenue we generate are from store Id 367, 361 and 357. we need to focus more on this store because they capture a large number of customers.
4. The average amount customer spent whenever they visit the store is between \$ 4 - \$ 5 and customer visit the store frequently.
5. The total amount a customer spent on an average is between \$1500 to \$2000.