

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+)

```
2
3 with category as (
4 select Total_Sales,
5 case when Total_Sales between 0 and 9 then 'Small $0-$9'
6 when Total_Sales between 10 and 19 then 'Medium $10-$19'
7 when Total_Sales > 19 then 'Large $20+'
8 end as Orders_Category
9 from `e_com.transactions`
10 )
11
12 select Orders_Category, count(Total_Sales) as total_order
13 from category
14 where Orders_Category is not null
15 group by Orders_Category;
16
```

### Query results

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

### Insights:

Most of the orders are from Small category which is between \$0 - \$9.

The order of Medium Category which is between \$10-\$19 are 29944.

The order of Large Category which is between \$20 - \$20+ are 15269.

-- 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+)

```
18
19 with category as (
20 select Total_Sales,
21 case when Total_Sales between 0 and 4 then 'Small $0 - $4'
22 when Total_Sales between 5 and 9 then 'Medium $5 - $10'
23 when Total_Sales > 9 then 'Large $10+'
24 end as Orders_Category
25 from `e_com.transactions`
26 )
27
28 select Orders_Category, count(Total_Sales) as total_order
29 from category
30 where Orders_Category is not null
31 group by Orders_Category;
32
```

### Query results

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

Insights:

Most of the orders are from Small category which is between \$0 - \$4.

The order of Medium Category which is between \$5-\$10 are 122302.

The order of Large Category which is between \$10 - \$10+ are 56832.

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

```

37
38 with foot_traffic as (
39   select Store_ID, Week_No, count(*) as customer_transacting,
40   dense_rank() over(partition by Week_No order by count(*) desc) as rn
41   from `e_com.transactions`
42   where Sales_Value is not null or Sales_Value = 0
43   group by Store_ID, Week_No
44 )
45
46 select Week_No, Store_ID
47 from foot_traffic
48 where rn <= 3
49 order by Week_No asc;
50

```

#### Query results

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
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				
12	4	320				
13	4	408				
14	5	32004				
15	5	324				
16	5	220				

Insights:

The most 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

```

50
51 --Question 4 Create a basic customer profiling with first, last visit, number of visits,
52 average money spent per visit and total money spent order by highest avg money
53
54 select Household_Key, min(DAY) as customer_first_visit,
55 max(DAY) as customer_last_visit, count(distinct DAY) as customer_number_of_visit,
56 round(avg(Total_Sales),1) as customer_average_spent_per_visit,
57 round(sum(Total_Sales),1) as customer_total_spent
58 from `e_com.transactions`
59 group by Household_Key
60 order by 1 asc;
61

```

#### Query results

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
11	11	2000-04-19	2001-02-15	5	2.8	25.4
12	12	2000-03-12	2001-05-18	5	3.1	138.4
13	13	2000-04-10	2001-12-09	166	6.3	7362.3
14	14	2000-01-04	2001-12-04	83	3.7	1719.1

#### Insights:

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 spend 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)

```
63
64 with most_spending_table as (
65   select dem.household_key, dem.age_bucket, dem.marital_status, dem.income_bucket, dem.homeowner,
66   dem.household_comp, dem.household_size, dem.kid_category,
67   sum(Total_Sales) as most_spend
68   from `e_com.demographics` dem
69   left join `e_com.transactions` tra on dem.household_key = tra.Household_Key
70   group by dem.household_key, dem.age_bucket, dem.marital_status, dem.income_bucket, dem.homeowner,
71   dem.household_comp, dem.household_size, dem.kid_category
72 )
73
74 select household_key, age_bucket, marital_status, income_bucket, homeowner, household_comp,
75 household_size, kid_category
76 from most_spending_table
77 where most_spend = (
78   select max(most_spend)
79   from most_spending_table
80 );
```

Query results

[SAVE RESULTS](#) [EXPLORE DATA](#)

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH			
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+	

Insights:

The customer who spent most amount is with the household key 1609 and we can see there demographics.

They are in the age bucket of 45-54, married.

There 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)

```

84
85 SELECT p1.SUB_COMMODITY_DESC AS product_A,
86        p2.SUB_COMMODITY_DESC AS product_B,
87        COUNT(*) AS pair_count
88 FROM `e_com.transactions` t1
89 JOIN `e_com.products` p1 ON t1.product_id = p1.product_id
90 JOIN `e_com.transactions` t2 ON t1.Basket_ID = t2.Basket_ID
91 AND t1.product_id < t2.product_id
92 JOIN `e_com.products` p2 ON t2.product_id = p2.product_id
93 GROUP BY p1.SUB_COMMODITY_DESC, p2.SUB_COMMODITY_DESC
94 ORDER BY pair_count DESC;
95
96
97
98

```

Query results

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:

The above query output shows which product is brought together frequently and how much time it was purchased by the customer.

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;

```

Query results

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
13	1	25	19.0	36.0	-17.0
14	1	26	39.0	19.0	20.0
15	1	28	48.0	39.0	9.0

Insights:

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.

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 stores 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.