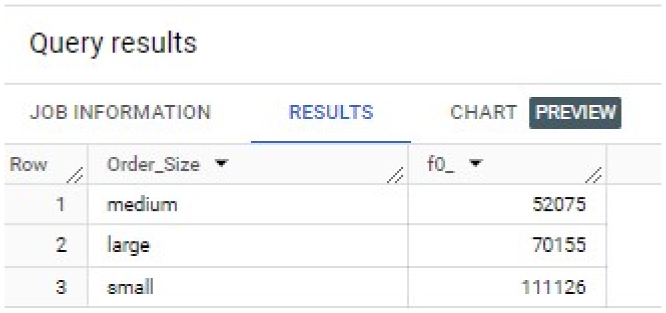
- Find the number of orders that have small, medium or large order value (small:0-10 dollars, medium:10-20 dollars, large:20+)

```sql
with cte_1 as(
SELECT
basket_id,
round(sum(sales_value),0) as Sale_value_order
FROM `future-oasis-409217.Bigquery_project_marsh122523.transaction_data`
group by basket_id
),cte_2 as(
  select
  basket_id,
  Sale_value_order,
  case when sale_value_order <10 then "small"
       when 10<=sale_value_order and sale_value_order <20 then "medium"
       when 20<=sale_value_order then "large"
       end as Order_Size
  from cte_1
)
select
distinct Order_Size,
count(basket_id)over(partition by Order_Size)
from cte_2
```

Result



Query results

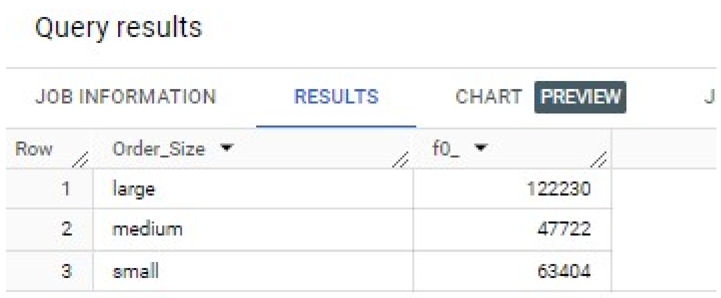
Row	Order_Size	f0_
1	medium	52075
2	large	70155
3	small	111126

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

```sql
with cte_1 as(
SELECT
basket_id,
round(sum(sales_value),0) as Sale_value_order
FROM `future-oasis-409217.Bigquery_project_marsh122523.transaction_data`
group by basket_id
),cte_2 as(
  select
  basket_id,
  Sale_value_order,
  case when sale_value_order <5 then "small"
       when 5<=sale_value_order and sale_value_order <10 then "medium"
       when 10<=sale_value_order then "large"
       end as Order_Size
  from cte_1
)
select
distinct Order_Size,
count(basket_id)over(partition by Order_Size)
```

```
from cte_2
```

Result


Query results

JOB INFORMATION RESULTS CHART PREVIEW

Row	Order_Size	f0_
1	large	122230
2	medium	47722
3	small	63404

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

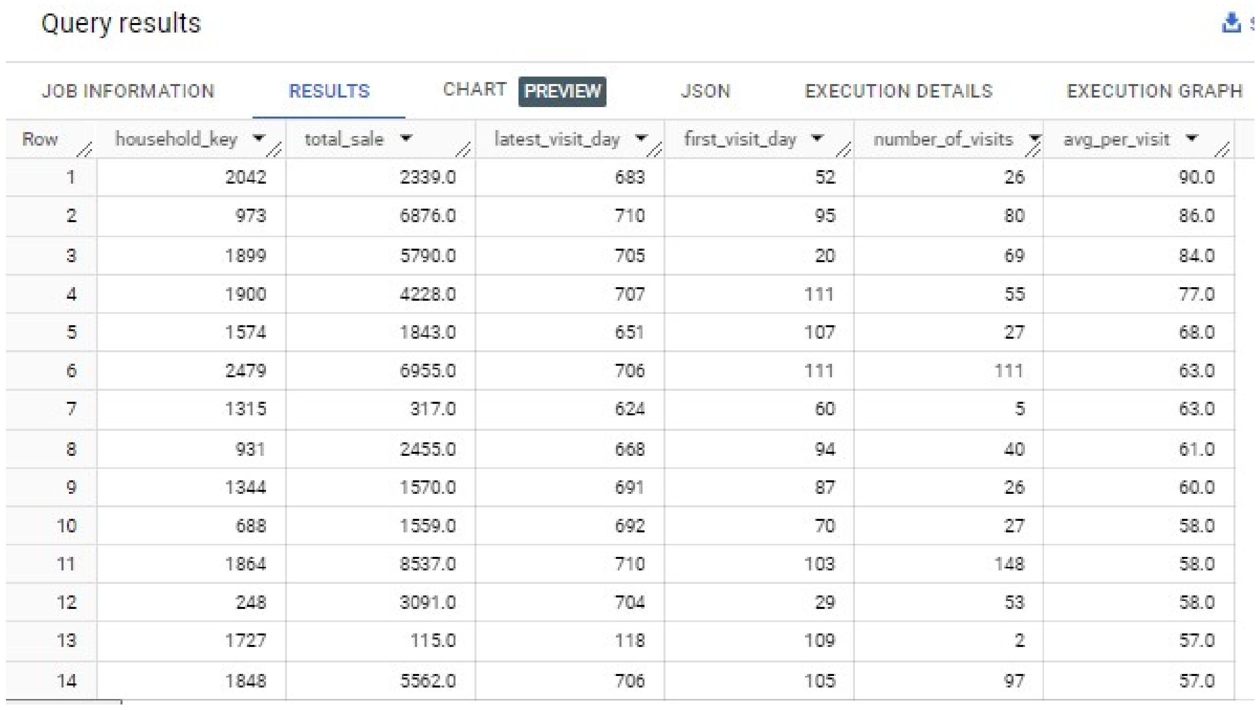
```
with cte_1 as (
select
household_key,
store_id,
week_no
FROM `future-oasis-409217.Bigquery_project_marsh122523.transaction_data`
group by 1,2,3
), cte_2 as(
select
week_no,
store_id,
count(household_key) over(partition by week_no,store_id ) as customer_footfall
from cte_1
order by customer_footfall desc
), cte_3 as (
select
week_no,
store_id,
customer_footfall,
dense_rank()over(partition by week_no order by customer_footfall desc) as Customer_champions
from cte_2
group by 1,2,3
)
select
*
from cte_3
where Customer_champions<4
order by week_no
```

- 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

```
with cte_1 as(
select
household_key,
basket_id,
sum(sales_value)  as order_value,
day,
WEEK_NO
from `future-oasis-409217.Bigquery_project_marsh122523.transaction_data`
group by household_key,basket_id,day,week_no
)
```

```
select
distinct household_key,
round(sum(order_value) over( partition by household_key ),0) as total_sale,
max(day)over (partition by household_key) as latest_visit_day,
min(day)over (partition by household_key) as first_visit_day,
count(basket_id)over(partition by household_key) as number_of_visits,
round(avg(cte_1.order_value)over(partition by household_key),0) as avg_per_visit,
from cte_1
order by avg_per_visit desc
```

Result

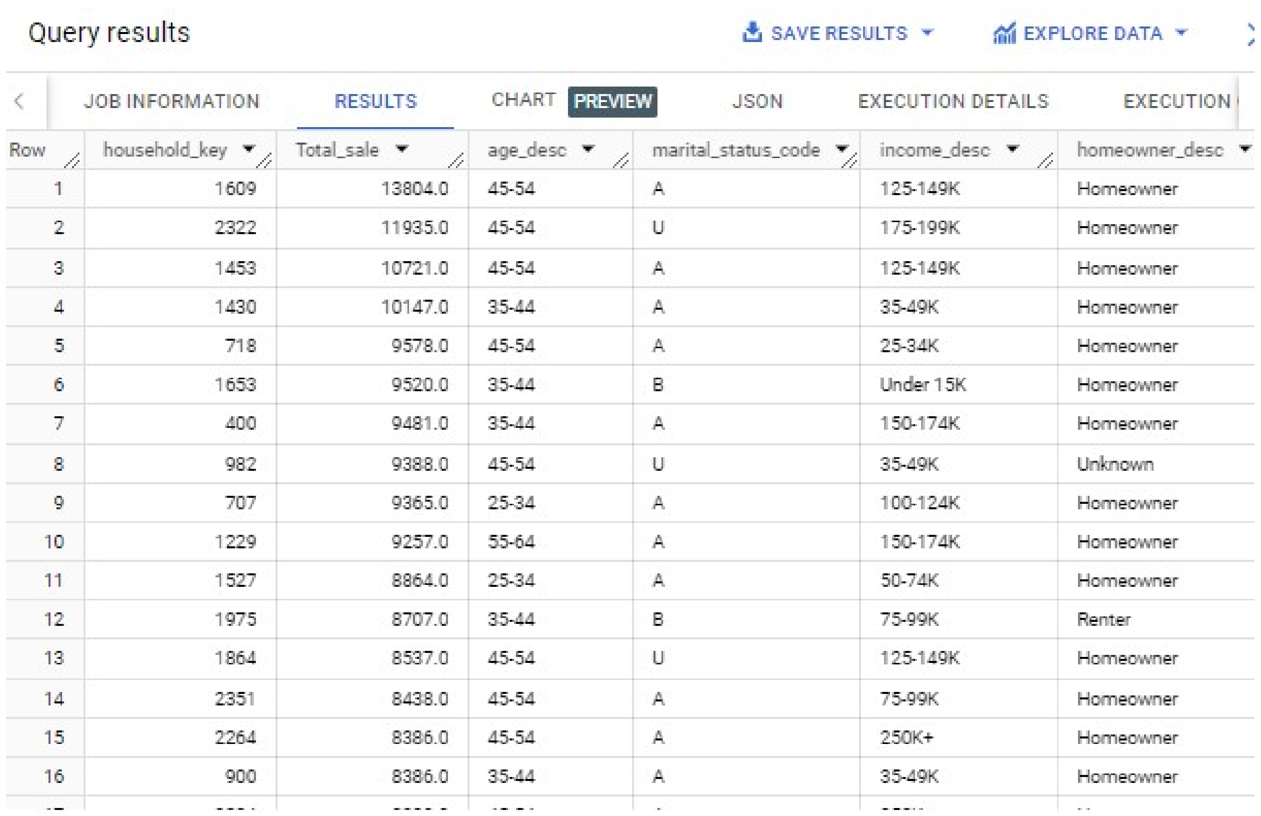


Row	household_key	total_sale	latest_visit_day	first_visit_day	number_of_visits	avg_per_visit
1	2042	2339.0	683	52	26	90.0
2	973	6876.0	710	95	80	86.0
3	1899	5790.0	705	20	69	84.0
4	1900	4228.0	707	111	55	77.0
5	1574	1843.0	651	107	27	68.0
6	2479	6955.0	706	111	111	63.0
7	1315	317.0	624	60	5	63.0
8	931	2455.0	668	94	40	61.0
9	1344	1570.0	691	87	26	60.0
10	688	1559.0	692	70	27	58.0
11	1864	8537.0	710	103	148	58.0
12	248	3091.0	704	29	53	58.0
13	1727	115.0	118	109	2	57.0
14	1848	5562.0	706	105	97	57.0

- 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 cte_1 as(
SELECT
household_key,
round(sum(sales_value),0) as Total_sale
FROM  `future-oasis-409217.Bigquery_project_marsh122523.transaction_data`
group by 1
)
select
a.household_key,
a.Total_sale,
b.age_desc,
b.marital_status_code,
b.income_desc,
b.homeowner_desc,
b.hh_comp_desc,
b.household_size_desc,
b.kid_category_desc
FROM  cte_1 a inner join `future-oasis-409217.Bigquery_project_marsh122523.demographic` b on
a.household_key = b.household_key
order by a.Total_sale desc
```

Result



Row	household_key	Total_sale	age_desc	marital_status_code	income_desc	homeowner_desc
1	1609	13804.0	45-54	A	125-149K	Homeowner
2	2322	11935.0	45-54	U	175-199K	Homeowner
3	1453	10721.0	45-54	A	125-149K	Homeowner
4	1430	10147.0	35-44	A	35-49K	Homeowner
5	718	9578.0	45-54	A	25-34K	Homeowner
6	1653	9520.0	35-44	B	Under 15K	Homeowner
7	400	9481.0	35-44	A	150-174K	Homeowner
8	982	9388.0	45-54	U	35-49K	Unknown
9	707	9365.0	25-34	A	100-124K	Homeowner
10	1229	9257.0	55-64	A	150-174K	Homeowner
11	1527	8864.0	25-34	A	50-74K	Homeowner
12	1975	8707.0	35-44	B	75-99K	Renter
13	1864	8537.0	45-54	U	125-149K	Homeowner
14	2351	8438.0	45-54	A	75-99K	Homeowner
15	2264	8386.0	45-54	A	250K+	Homeowner
16	900	8386.0	35-44	A	35-49K	Homeowner

- 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)

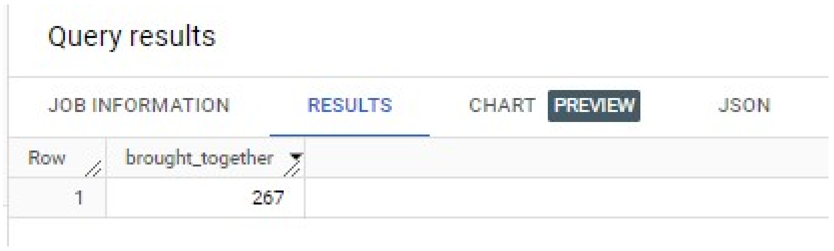
```
with cte_1 as (
SELECT
#a.household_key,
a.basket_id,
a.product_id,
b.SUB_COMMODITY_DESC,
count(a.basket_id)over(partition by a.product_id) as count_pro
FROM `future-oasis-409217.Bigquery_project_marsh122523.transaction_data`a  inner join  `future-oasis-409217.Bigquery_project_marsh122523.products` b
     on a.product_id=b.PRODUCT_ID

),cte_2 as(

 select
 product_id,
  basket_id,
  SUB_COMMODITY_DESC,
  count_pro,
  dense_rank() over (order by count_pro desc) as rank_pro
  from cte_1
  order by rank_pro
)

Select
count(distinct basket_id) as brought_together
from cte_2 where rank_pro=6 and basket_id in (select basket_id from cte_2 where rank_pro=4)
```

Result



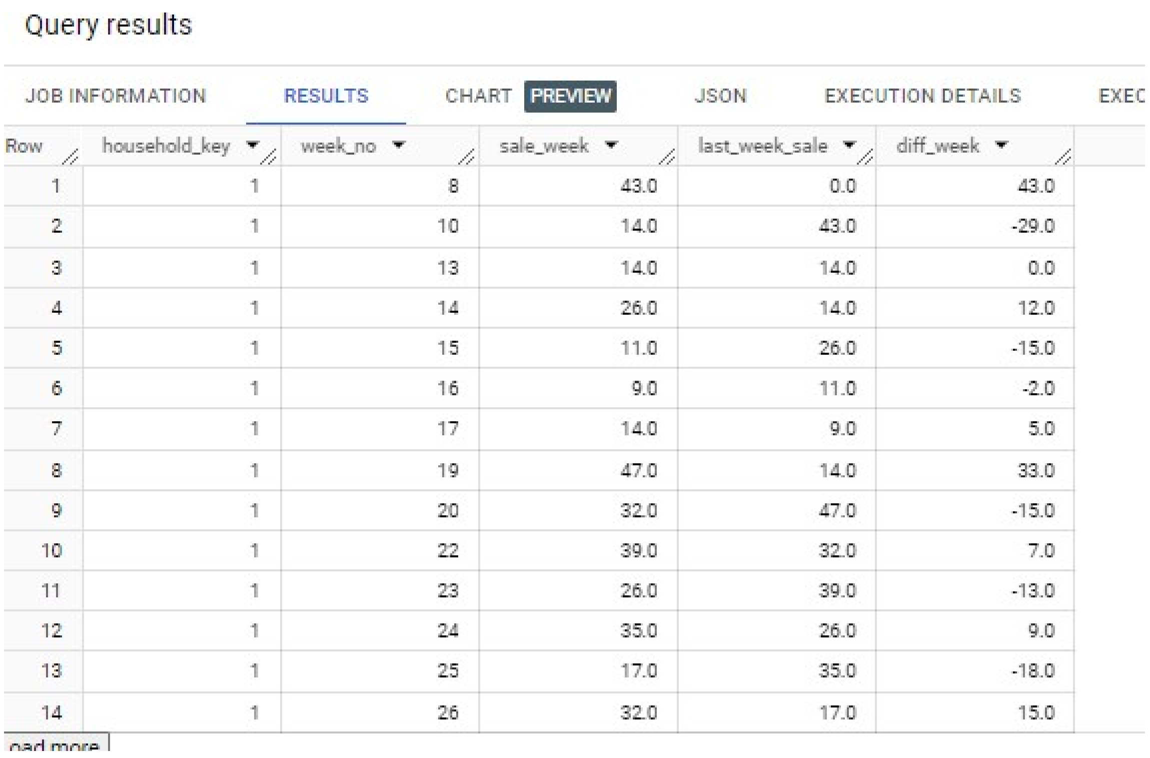
Row	brought_together
1	267

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

```
with cte_1 as(
select
distinct household_key,
round(sum(sales_value) over(partition by household_key,week_no),0) as sale_week,
week_no
FROM `future-oasis-409217.Bigquery_project_marsh122523.transaction_data`
order by 1,3 desc
)

select
household_key,
week_no,
sale_week,
ifnull(lag(sale_week)over(partition by household_key order by week_no),0) as last_week_sale,
sale_week-ifnull((lag(sale_week)over(partition by household_key order by week_no)),0) as diff_week
from cte_1
order by household_key
```

Result


Query results

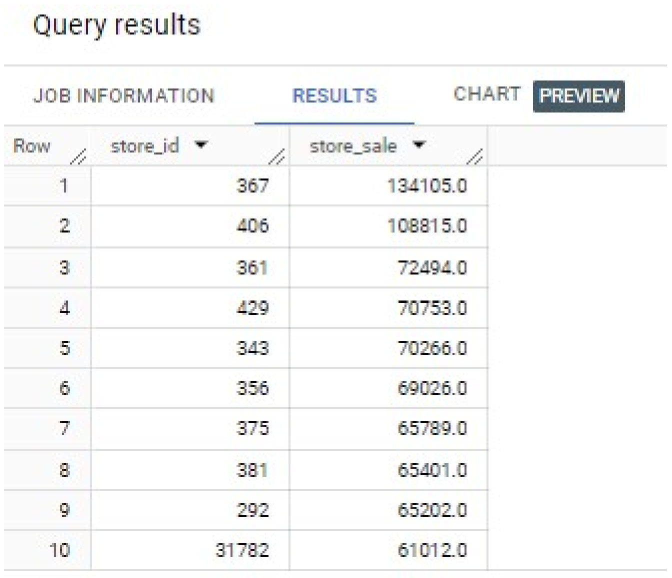
JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXEC

Row	household_key	week_no	sale_week	last_week_sale	diff_week
1	1	8	43.0	0.0	43.0
2	1	10	14.0	43.0	-29.0
3	1	13	14.0	14.0	0.0
4	1	14	26.0	14.0	12.0
5	1	15	11.0	26.0	-15.0
6	1	16	9.0	11.0	-2.0
7	1	17	14.0	9.0	5.0
8	1	19	47.0	14.0	33.0
9	1	20	32.0	47.0	-15.0
10	1	22	39.0	32.0	7.0
11	1	23	26.0	39.0	-13.0
12	1	24	35.0	26.0	9.0
13	1	25	17.0	35.0	-18.0
14	1	26	32.0	17.0	15.0

oad more

- Top 10 stores by sale_value

```
select
distinct store_id,
round(sum(sales_value) over(partition by store_id),0) as store_sale
FROM `future-oasis-409217.Bigquery_project_marsh122523.transaction_data`
order by 2 desc
limit 10
```
result


Query results

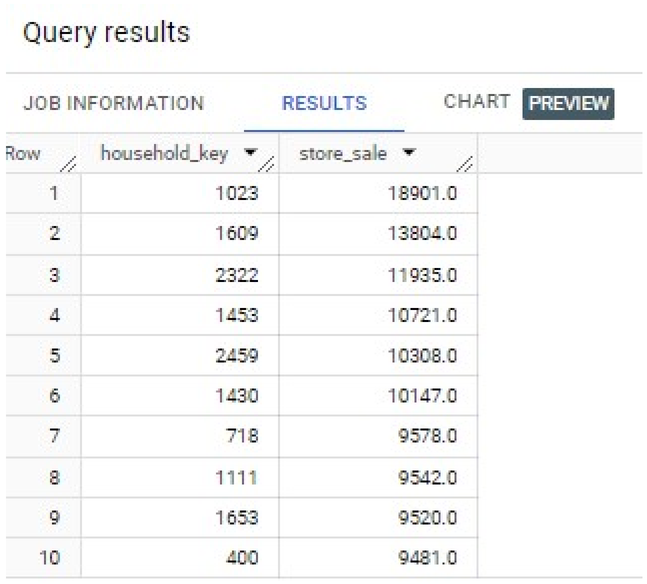
JOB INFORMATION | RESULTS | CHART PREVIEW

Row	store_id	store_sale
1	367	134105.0
2	406	108815.0
3	361	72494.0
4	429	70753.0
5	343	70266.0
6	356	69026.0
7	375	65789.0
8	381	65401.0
9	292	65202.0
10	31782	61012.0

- Top 10 customers

```
select
distinct household_key,
round(sum(sales_value) over(partition by household_key),0) as store_sale
FROM `future-oasis-409217.Bigquery_project_marsh122523.transaction_data`
order by 2 desc
limit 10
```

Result



Row	household_key	store_sale
1	1023	18901.0
2	1609	13804.0
3	2322	11935.0
4	1453	10721.0
5	2459	10308.0
6	1430	10147.0
7	718	9578.0
8	1111	9542.0
9	1653	9520.0
10	400	9481.0

Insights
1. Small basket value are more
2. Need to plan the discounts based on the customer sale pattern
3. Top 10 stores need to be recognized
4. Combinations which are giving more sales need to be sold as a combo
5. Need to capture demographic details of all household ids