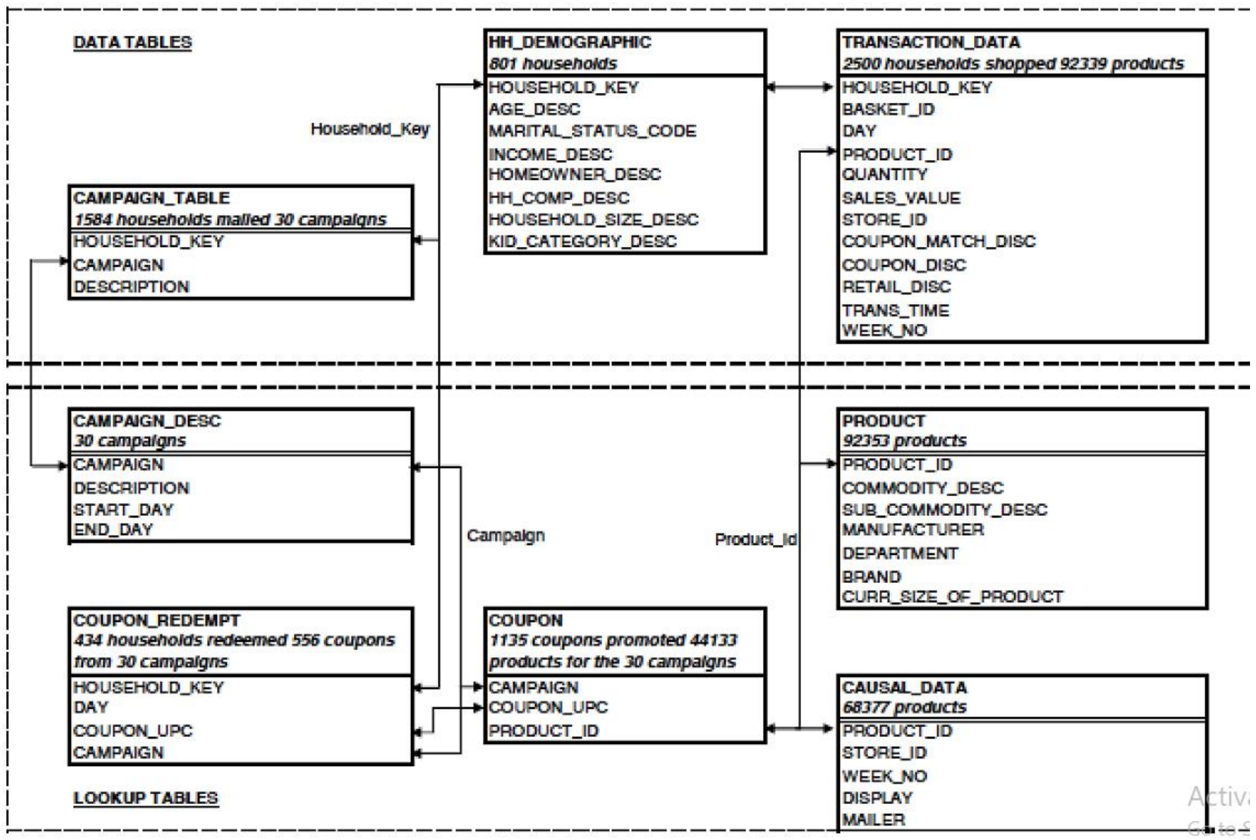
Dunnhumby - The Complete Journey

Context

Global leader in Customer data science and analytics, dunnhumby has experts in working with brands, grocery retail, retail pharmacy, and retailer financial services. With deep heritage and expertise in retail — one of the world's most competitive markets, with a deluge of multi-dimensional data — dunnhumby today enables businesses all over the world, across industries, to be Customer First.

This business case has

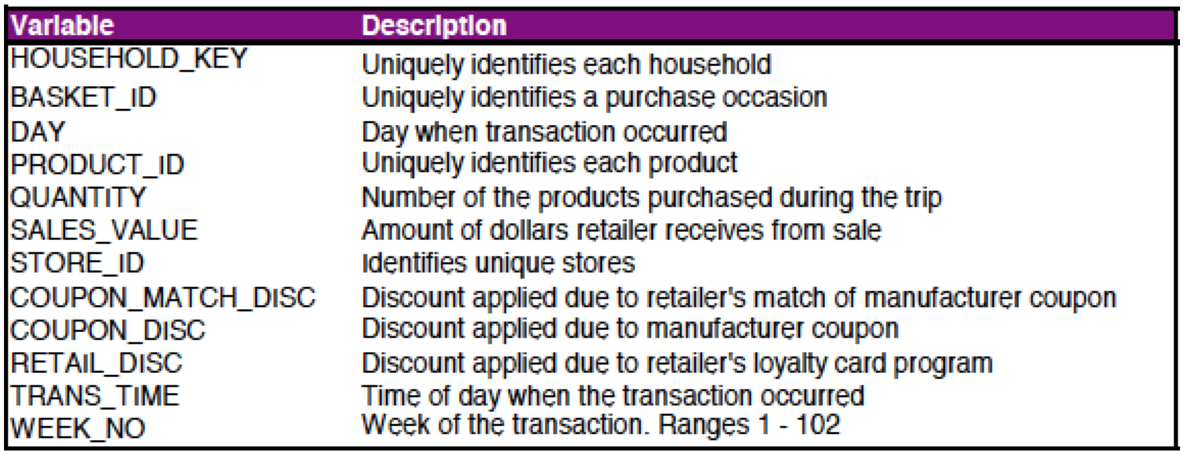
- Household level transactions over two years from a group of 2,500 households who are frequent shoppers at a retailer
- All of a household’s purchases within the store, not just those from a limited number of categories
- Demographics and direct marketing contact history for select households



- **hh_demographic**

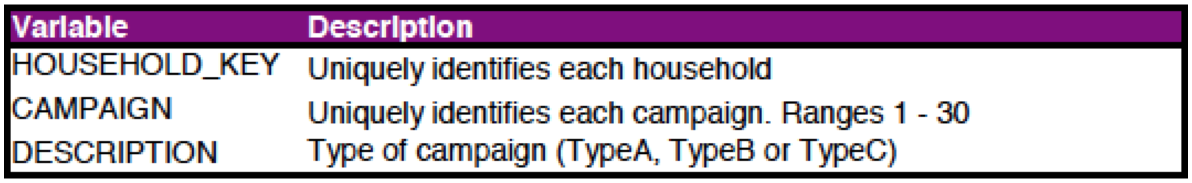
Variable	Description
HOUSEHOLD_KEY	Uniquely identifies each household
AGE_DESC	Estimated age range
MARITAL_STATUS_CODE	Marital Status (A - Married, B- Single, U - Unknown)
INCOME_DESC	Household income
HOMEOWNER_DESC	Homeowner, renter, etc.
HH_COMP_DESC	Household composition
HOUSEHOLD_SIZE_DESC	Size of household up to 5+
KID_CATEGORY_DESC	Number of children present up to 3+

- **transaction_data**



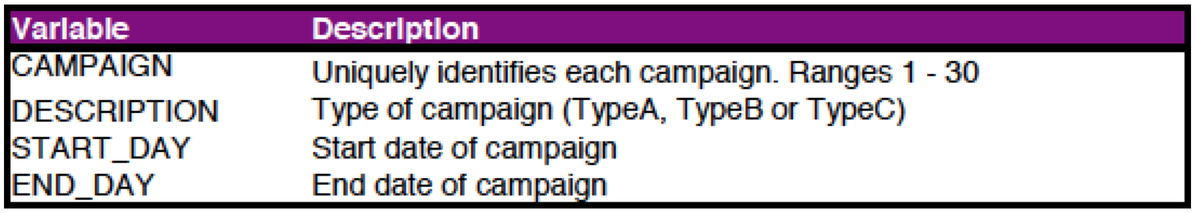
Variable	Description
HOUSEHOLD_KEY	Uniquely identifies each household
BASKET_ID	Uniquely identifies a purchase occasion
DAY	Day when transaction occurred
PRODUCT_ID	Uniquely identifies each product
QUANTITY	Number of the products purchased during the trip
SALES_VALUE	Amount of dollars retailer receives from sale
STORE_ID	Identifies unique stores
COUPON_MATCH_DISC	Discount applied due to retailer's match of manufacturer coupon
COUPON_DISC	Discount applied due to manufacturer coupon
RETAIL_DISC	Discount applied due to retailer's loyalty card program
TRANS_TIME	Time of day when the transaction occurred
WEEK_NO	Week of the transaction. Ranges 1 - 102

- **campaign_table**



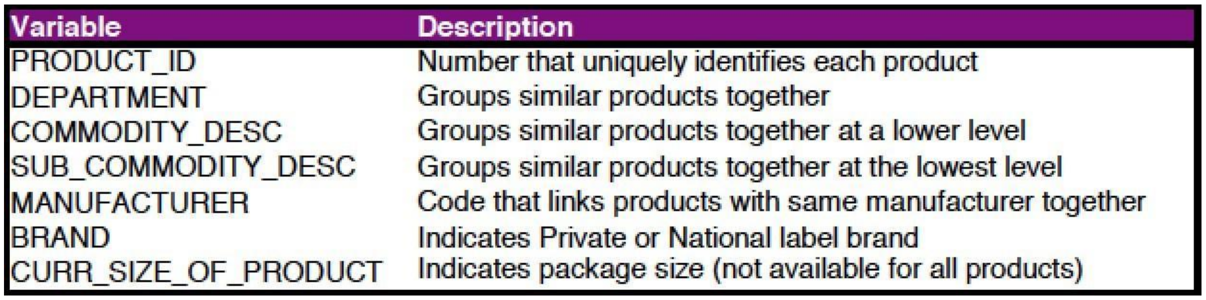
Variable	Description
HOUSEHOLD_KEY	Uniquely identifies each household
CAMPAIGN	Uniquely identifies each campaign. Ranges 1 - 30
DESCRIPTION	Type of campaign (TypeA, TypeB or TypeC)

- **capaign_desc**



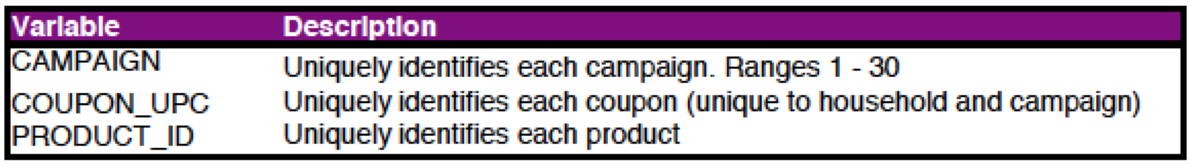
Variable	Description
CAMPAIGN	Uniquely identifies each campaign. Ranges 1 - 30
DESCRIPTION	Type of campaign (TypeA, TypeB or TypeC)
START_DAY	Start date of campaign
END_DAY	End date of campaign

- **product**



Variable	Description
PRODUCT_ID	Number that uniquely identifies each product
DEPARTMENT	Groups similar products together
COMMODITY_DESC	Groups similar products together at a lower level
SUB_COMMODITY_DESC	Groups similar products together at the lowest level
MANUFACTURER	Code that links products with same manufacturer together
BRAND	Indicates Private or National label brand
CURR_SIZE_OF_PRODUCT	Indicates package size (not available for all products)

- **coupon**



Variable	Description
CAMPAIGN	Uniquely identifies each campaign. Ranges 1 - 30
COUPON_UPC	Uniquely identifies each coupon (unique to household and campaign)
PRODUCT_ID	Uniquely identifies each product

- **coupon_redempt**

Variable	Description
HOUSEHOLD_KEY	Uniquely identifies each household
DAY	Day when transaction occurred
COUPON_UPC	Uniquely identifies each coupon (unique to household and campaign)
CAMPAIGN	Uniquely identifies each campaign

- **Causal_data - event info and some other details**

Variable	Description
PRODUCT_ID	Uniquely identifies each product
STORE_ID	Identifies unique stores
WEEK_NO	Week of the transaction
DISPLAY	Display location (see below)
MAILER	Mailer location (see below)

Field	Contents
DISPLAY	0 - Not on Display
	1 - Store Front
	2 - Store Rear
	3 - Front End Cap
	4 - Mid-Aisle End Cap
	5 - Rear End Cap
	6 - Side-Aisle End Cap
	7 - In-Aisle
	9 - Secondary Location Display
	A - In-Shelf
MAILER	0 - Not on ad
	A - Interior page feature
	C - Interior page line item
	D - Front page feature
	F - Back page feature
	H - Wrap front feature
	J - Wrap interior coupon
	L - Wrap back feature
	P - Interior page coupon
	X - Free on interior page
	Z - Free on front page, back page or wrap

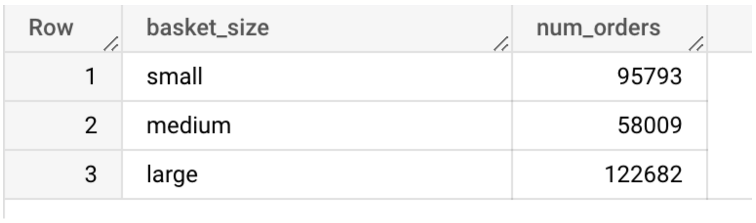
Top questions:

1. Find the number of orders that are small, medium or large order value(small:0-5$, medium:5-10$, large:10+)
2. Find top 3 stores with highest foot traffic for each week (Foot traffic: number of customers transacting)
3. 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
4. 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 profiling data)
5. Find products(product table :SUB_COMMODITY_DESC) which are most frequently bought together
6. **Find out on which weeks does each household shop and find their cumulative spending over time(sum of all previous) (uses sum over partition)**
7. **Find the weekly change in Revenue Per Account (RPA) (spending by each customer compared to last week)(use lag function)**
8. **Find number of returning customers and percent of returning customers for all week**
9. **Quarterly analysis: sales comparison: total sale amount (create a new quarter column using case where,12 weeks(3 months)=1 quarter)**
   (Use cte tables)
10. **How are the sales for individual stores changing over the quarters**
11. **Customer churn analysis for each quarter (churned customers : that never shop after that particular quarter)**
12. **Find the retained customers for each quarter(retained :Households who were there in previous quarters and are there in the current quarter)**
13. Calculate Customer lifetime value(CLV) for different age group
    Average purchase value — the value of all customer purchases over a particular time frame , divided by the number of purchases in that period
    Average purchase frequency — divide the number of purchases in that same time period by the number of individual customers who made a transaction over the same period
    Customer value — the average purchase frequency multiplied by the average purchase value
    Average customer lifespan — the average length of time a customer continues buying from you
    CLV = customer value X average customer lifespan

Answers:

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

```
select basket_size , count(*) as num_orders
from (
 select case
   when sales_value between 0 and 10 then 'small'
   when sales_value between 10 and 20 then 'medium'
   when sales_value >20 then 'large'
    end as basket_size
 from(
   select sum(SALES_VALUE) as sales_value from
`dunnhumbysql.complete.transaction_data`
group by BASKET_ID))
group by basket_size
```

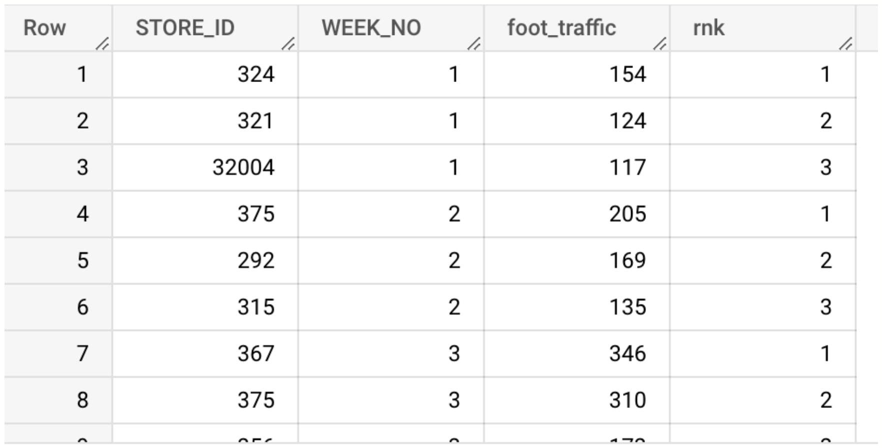


Row	basket_size	num_orders
1	small	95793
2	medium	58009
3	large	122682

2. Find week over week top 3 stores with highest foot traffic (Foot traffic: number of households transacting)

```
select * from(
select STORE_ID, WEEK_NO, count(household_key) as foot_traffic,
RANK() OVER(PARTITION BY WEEK_NO order by count(household_key) desc ) as rnk
from `dunnhumbysql.complete.transaction_data`
group by STORE_ID,WEEK_NO)
where rnk<4
```

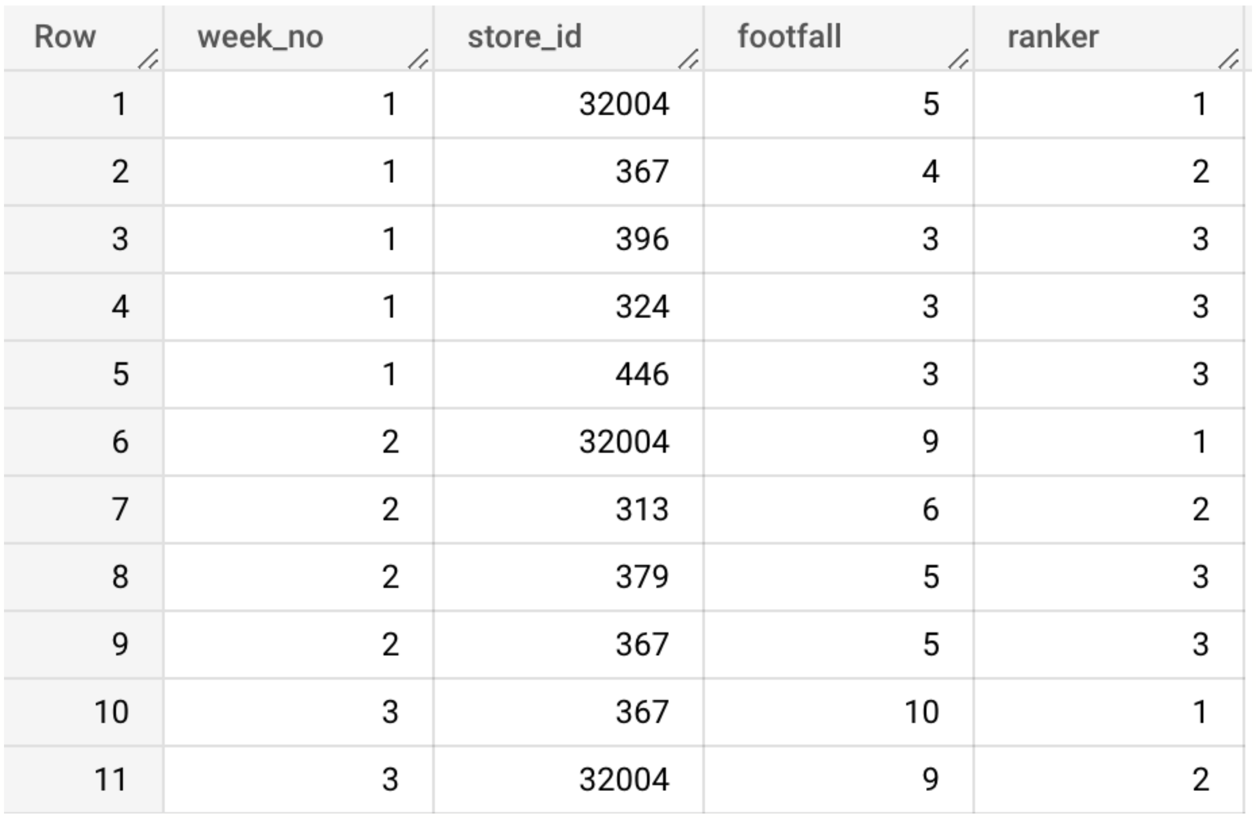
```
order by WEEK_NO, foot_traffic desc
```

- 

Row	STORE_ID	WEEK_NO	foot_traffic	rnk
1	324	1	154	1
2	321	1	124	2
3	32004	1	117	3
4	375	2	205	1
5	292	2	169	2
6	315	2	135	3
7	367	3	346	1
8	375	3	310	2

Variation:
Find week over week top 3 stores with most number of distinct households transacting

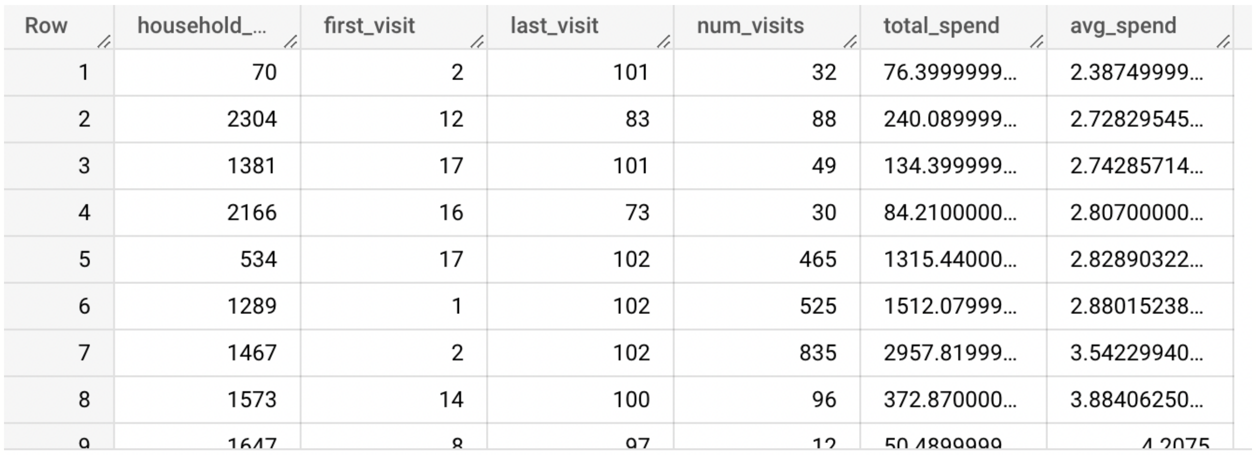
```
with base as (select
STORE_ID, WEEK_NO, count(distinct household_key) as footfall
from `dunnhumby.transaction_data`
group by 1,2
),
base_2 as (
select *, dense_rank() over (partition by base.week_no order by footfall desc) as
ranker
from base
order by week_no asc
)
select week_no, store_id, footfall, ranker from base_2
where ranker <=3
order by week_no, ranker asc
```



Row	week_no	store_id	footfall	ranker
1	1	32004	5	1
2	1	367	4	2
3	1	396	3	3
4	1	324	3	3
5	1	446	3	3
6	2	32004	9	1
7	2	313	6	2
8	2	379	5	3
9	2	367	5	3
10	3	367	10	1
11	3	32004	9	2

3. 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

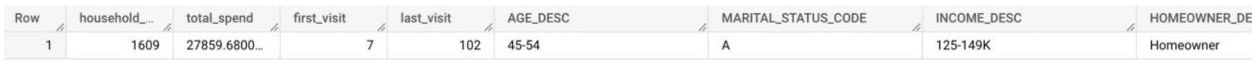
```
select household_key, min(WEEK_NO) as first_visit,
max(WEEK_NO) last_visit, count(distinct(BASKET_ID)) as num_visits,
sum(SALES_VALUE) as total_spend, (sum(SALES_VALUE)/count(distinct(BASKET_ID)))
as avg_spend
from `dunnhumbysql.complete.transaction_data`
group by household_key
order by avg_spend
```



Row	household_...	first_visit	last_visit	num_visits	total_spend	avg_spend
1	70	2	101	32	76.3999999...	2.38749999...
2	2304	12	83	88	240.089999...	2.72829545...
3	1381	17	101	49	134.399999...	2.74285714...
4	2166	16	73	30	84.2100000...	2.80700000...
5	534	17	102	465	1315.44000...	2.82890322...
6	1289	1	102	525	1512.07999...	2.88015238...
7	1467	2	102	835	2957.81999...	3.54229940...
8	1573	14	100	96	372.870000...	3.88406250...
9	1647	8	97	12	50.4899999	4.2075

4. **Do a customer analysis for the most spending customer for whom we have demographic information**

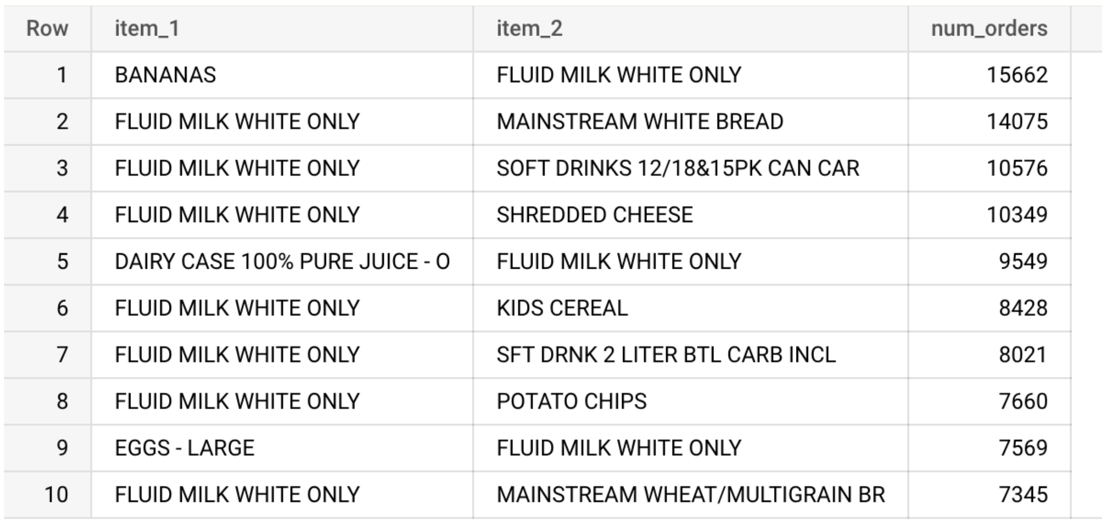
```
with cte as(
select t.household_key, sum(SALES_VALUE) as total_spendfrom
`dunnhumbysql.complete.transaction_data` t
inner join `dunnhumbysql.complete.hh_demographic` d
on d.household_key=t.household_key
group by t.household_key
order by total_spend desc
limit 1
)
select cte.*, d.* from cte
inner join `dunnhumbysql.complete.hh_demographic` d
on cte.household_key=d.household_key
```



Row	household_...	total_spend	first_visit	last_visit	AGE_DESC	MARITAL_STATUS_CODE	INCOME_DESC	HOMEOWNER_DE
1	1609	27859.6800...	7	102	45-54	A	125-149K	Homeowner

5. **Find products(product table:SUB_COMMODITY_DESC) which are most frequently bought together**

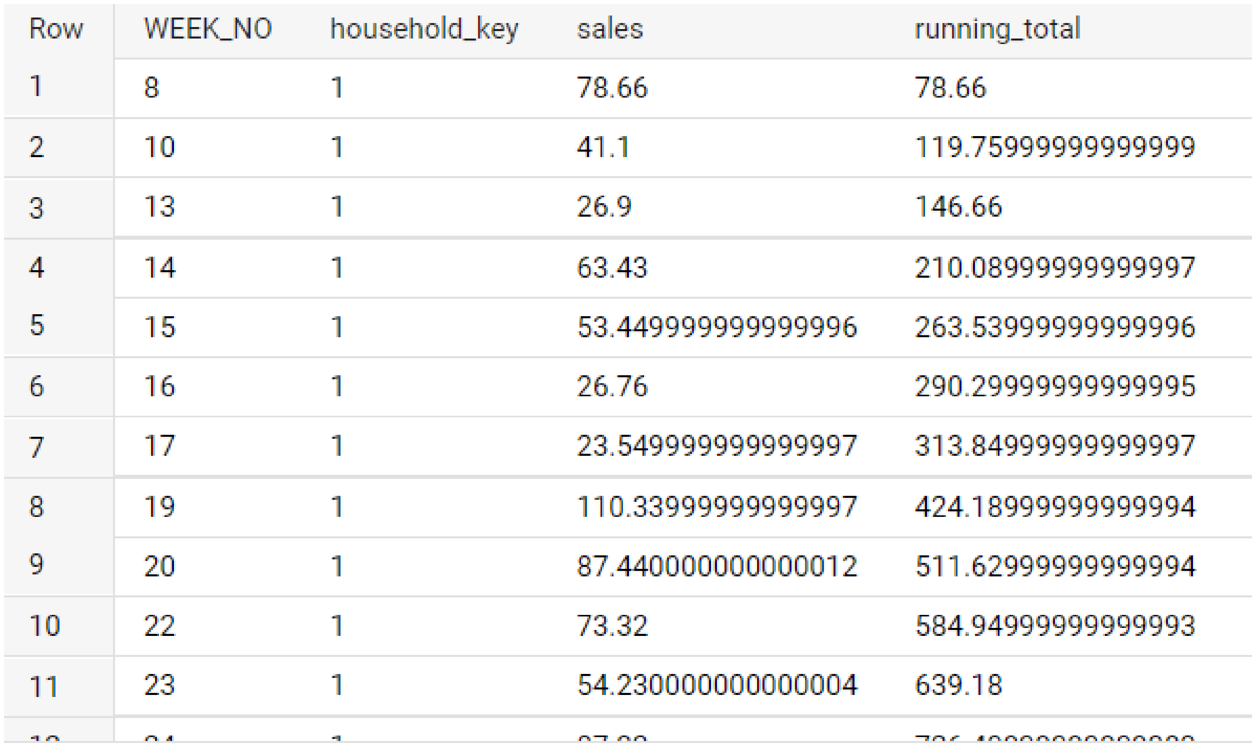
```sql
with cte as (
SELECT *
FROM `dunnhumbysql.complete.product` p
join  `dunnhumbysql.complete.transaction_data` t
on p.PRODUCT_ID=t.PRODUCT_ID
)
select t.SUB_COMMODITY_DESC as item_1, t2.SUB_COMMODITY_DESC as item_2,
count(distinct t.BASKET_ID) as num_orders
from cte t
inner join cte t2
on t.BASKET_ID=t2.BASKET_ID
and t.SUB_COMMODITY_DESC<t2.SUB_COMMODITY_DESC
group by t.SUB_COMMODITY_DESC, t2.SUB_COMMODITY_DESC
order by num_orders desc
limit 10
```



Row	item_1	item_2	num_orders
1	BANANAS	FLUID MILK WHITE ONLY	15662
2	FLUID MILK WHITE ONLY	MAINSTREAM WHITE BREAD	14075
3	FLUID MILK WHITE ONLY	SOFT DRINKS 12/18&15PK CAN CAR	10576
4	FLUID MILK WHITE ONLY	SHREDDED CHEESE	10349
5	DAIRY CASE 100% PURE JUICE - O	FLUID MILK WHITE ONLY	9549
6	FLUID MILK WHITE ONLY	KIDS CEREAL	8428
7	FLUID MILK WHITE ONLY	SFT DRNK 2 LITER BTL CARB INCL	8021
8	FLUID MILK WHITE ONLY	POTATO CHIPS	7660
9	EGGS - LARGE	FLUID MILK WHITE ONLY	7569
10	FLUID MILK WHITE ONLY	MAINSTREAM WHEAT/MULTIGRAIN BR	7345

6. Find out on which weeks does each household shop and find their cumulative spending over time

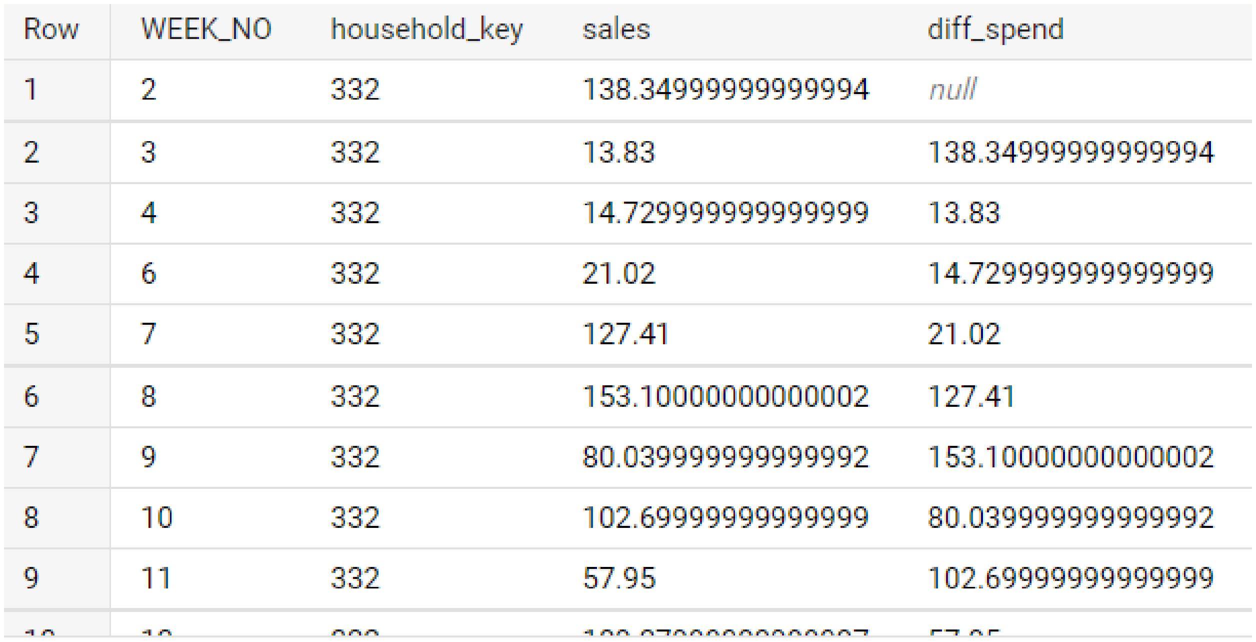
```sql
with cte as(
select WEEK_NO , household_key , sum(SALES_VALUE) as sales
FROM `dunnhumbysql.complete.transaction_data`
group by WEEK_NO, household_key
)

SELECT
   *,
  SUM(sales) OVER (PARTITION BY household_key ORDER BY week_no) AS
running_total
  from cte
```



Row	WEEK_NO	household_key	sales	running_total
1	8	1	78.66	78.66
2	10	1	41.1	119.75999999999999
3	13	1	26.9	146.66
4	14	1	63.43	210.08999999999997
5	15	1	53.449999999999996	263.53999999999996
6	16	1	26.76	290.29999999999995
7	17	1	23.549999999999997	313.84999999999997
8	19	1	110.33999999999997	424.18999999999994
9	20	1	87.440000000000012	511.62999999999994
10	22	1	73.32	584.94999999999993
11	23	1	54.230000000000004	639.18

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

```sql
with cte as(
select WEEK_NO , household_key , sum(SALES_VALUE) as sales
FROM `dunnhumbysql.complete.transaction_data`
group by WEEK_NO, household_key
)

SELECT
   *,
  lag(sales)OVER (PARTITION BY household_key ORDER BY week_no) as
diff_spend,
  from cte
```

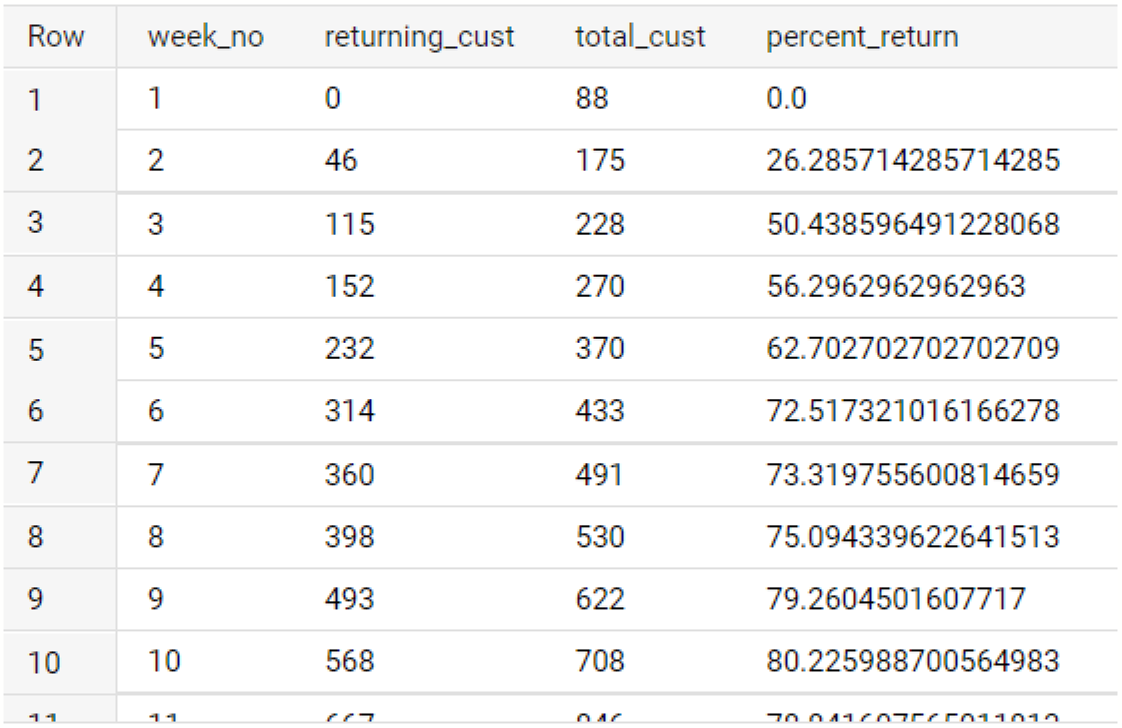


Row	WEEK_NO	household_key	sales	diff_spend
1	2	332	138.34999999999994	*null*
2	3	332	13.83	138.34999999999994
3	4	332	14.729999999999999	13.83
4	6	332	21.02	14.729999999999999
5	7	332	127.41	21.02
6	8	332	153.10000000000002	127.41
7	9	332	80.039999999999992	153.10000000000002
8	10	332	102.69999999999999	80.039999999999992
9	11	332	57.95	102.69999999999999

8. Find number of returning customers and percent of returning customers for all week

```
with cte as(
select b.week_no,  a.household_key,
CASE when min(a.week_no)<b.week_no then 1 else 0
end as decider
from `dunnhumbysql.complete.transaction_data` a
left join  `dunnhumbysql.complete.transaction_data`   b
on a.household_key=b.household_key
group by b.week_no, a.household_key
) select week_no,sum(decider) as returning_cust, count(decider) as
total_cust, (sum(decider)/count(decider))*100 as percent_return
from cte
```

```
group by 1
order by 1
```



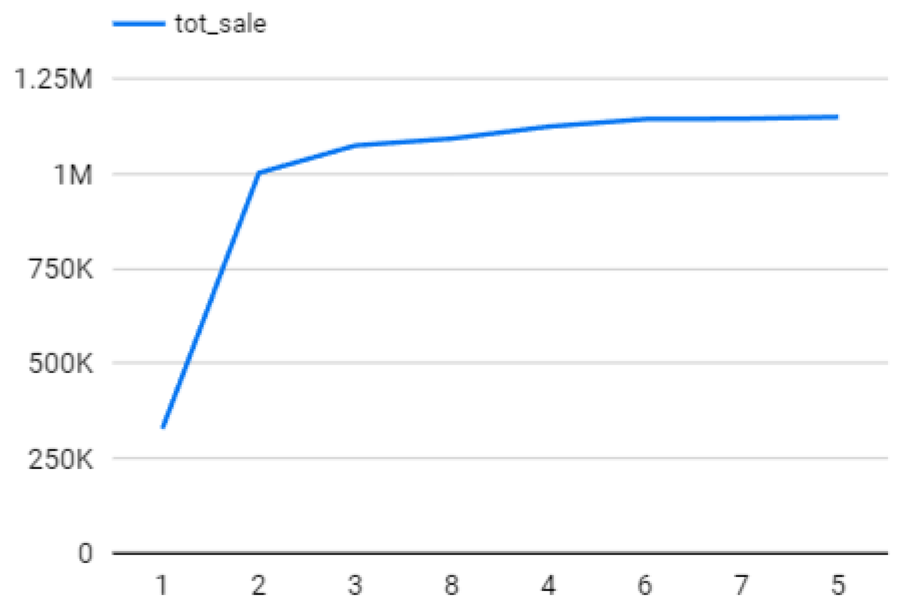
Row	week_no	returning_cust	total_cust	percent_return
1	1	0	88	0.0
2	2	46	175	26.285714285714285
3	3	115	228	50.438596491228068
4	4	152	270	56.2962962962963
5	5	232	370	62.702702702702709
6	6	314	433	72.517321016166278
7	7	360	491	73.319755600814659
8	8	398	530	75.094339622641513
9	9	493	622	79.2604501607717
10	10	568	708	80.225988700564983
11	11	667	846	[illegible]

9. Quarterly analysis: sales comparison over quarters

```
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 `dunnhumbysql.complete.transaction_data`)
select quarter, sum(sales_value) as tot_sale
from cte
```

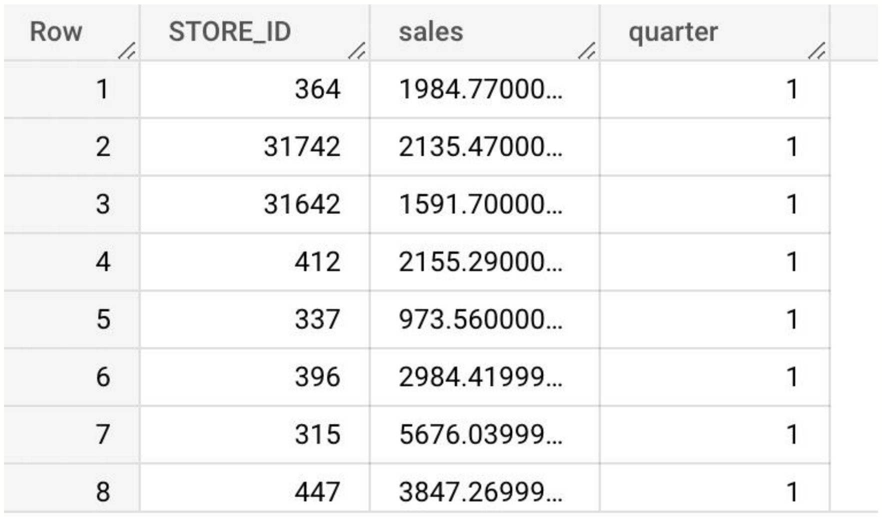
```
group by quarter
```

Row	quarter	tot_sale
1	1	328865.3099999472
2	2	1001743.239999364
3	3	1073977.4699991948
4	4	1123719.6399991605
5	5	1148910.6099991191
6	6	1143552.5799991363
7	7	1144484.6599992837
8	8	1092209.5699993679



10. How are the sales for individual stores changing over the quarters

```
select STORE_ID ,sum(sales) as sales, 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( select week_no, STORE_ID, sum(SALES_VALUE) as sales,
from  `dunnhumbysql.complete.transaction_data` group by STORE_ID, WEEK_NO)
group by  quarter, store_id
```



Row	STORE_ID	sales	quarter
1	364	1984.77000...	1
2	31742	2135.47000...	1
3	31642	1591.70000...	1
4	412	2155.29000...	1
5	337	973.560000...	1
6	396	2984.41999...	1
7	315	5676.03999...	1
8	447	3847.26999...	1

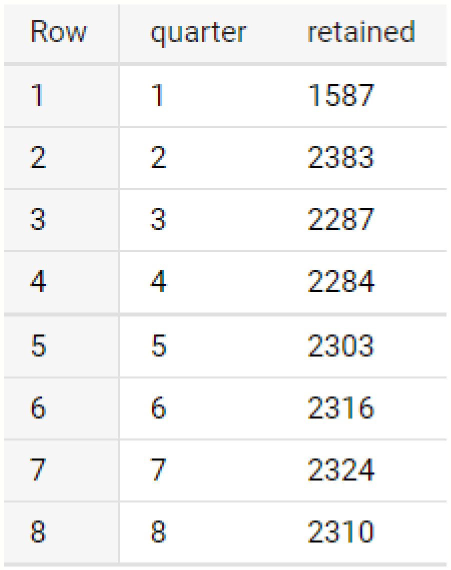
11. Customer churn analysis for each quarter

```
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 `dunnhumbysql.complete.transaction_data`)
select  a.quarter, count(distinct(a.household_key)) as chrned
from cte a
FULL OUTER JOIN cte b
on a.household_key=b.household_key
and a.quarter<b.quarter
where b.household_key is NUll
group by a.quarter
order by quarter
```

Row	quarter	chrned
1	1	5
2	2	3
3	3	5
4	4	8
5	5	16
6	6	34
7	7	119
8	8	2310

12. Find the retained customers for each quarter(Households who were there last quarters and are there in the current quarter

```
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 `dunnhumbysql.complete.transaction_data`)
select  a.quarter, count(distinct(a.household_key)) as retained
from cte a
left join cte b
on a.household_key=b.household_key and a.quarter>b.quarter
group by a.quarter
```



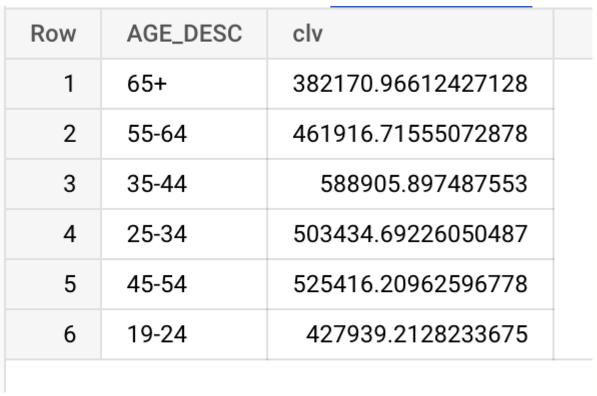
Row	quarter	retained
1	1	1587
2	2	2383
3	3	2287
4	4	2284
5	5	2303
6	6	2316
7	7	2324
8	8	2310

13. Calculate Customer lifetime value(CLV) for different age group

- **Average purchase value –** the value of all customer purchases over a particular time frame , divided by the number of purchases in that period
- **Average purchase frequency –** divide the number of purchases in that same time period by the number of individual customers who made a transaction over the same period
- **Customer value –** the average purchase frequency multiplied by the average purchase value

- **Average customer lifespan** – the average length of time a customer continues buying from you
- **CLV = customer value X average customer lifespan**

```
select AGE_DESC, (avg_purch_val*avg_purch_freq*avg_cust_lifespan) as clv
from(
with cte as (
select household_key, (max(WEEK_NO)- min (WEEK_NO)) as cust_duration
from `dunnhumbysql.complete.transaction_data`
group by household_key

)
select AGE_DESC,  sum(SALES_VALUE)/count(distinct(BASKET_ID)) as
avg_purch_val,
count(distinct(BASKET_ID))/count(distinct(d.household_key)) as avg_purch_freq,
(sum(cte.cust_duration)/count(1)) as avg_cust_lifespan,
from `dunnhumbysql.complete.transaction_data` t
inner join `dunnhumbysql.complete.hh_demographic` d
on t.household_key=d.household_key
join cte
on cte.household_key=d.household_key
group by AGE_DESC
)
```



Row	AGE_DESC	clv
1	65+	382170.96612427128
2	55-64	461916.71555072878
3	35-44	588905.897487553
4	25-34	503434.69226050487
5	45-54	525416.20962596778
6	19-24	427939.2128233675

Questions

1. **Exploratory queries**
    a. Find out which age group is the most active shopper(join hh_demographic and transaction_data)
    b. Which week had the best sales
    c. What is the average basket size for shoppers (Divide it in small, medium, large)
    d. Find foot traffic for each store per week. (Foot traffic: number of customers transacting)
    e. Top5 spending customers (households) with sales value in integer
2. **Customer profiling**

    a. Create a basic customer profiling with first last visit and total money spent for all customers
    b. 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)
    c. What products does the customer buy most
    d. Which promotional campaigns were they a part of(campaign_table)
3. **Product analysis**
    a. Find the most selling product
    b. When did the product sell the most and where
    c. Where was the product placed in store and featured in ad for that particular store and week
    d. Was it a part of some campaigns
    e. How many household did actually redeem coupons for this product in each campaign
    f. Which products were the best seller(top 3) for each week and what quantity did they sell
4. **Advance analysis and queries**
    a. Find out on which weeks does each household shop and find their cumulative spending over time(sum of all previous) (uses sum over partition)

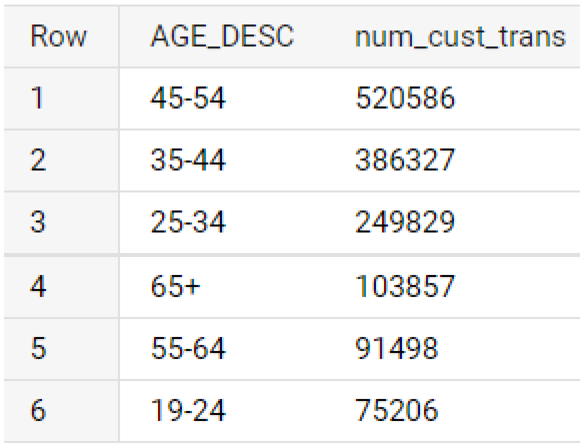
b. Find the trend in spending for each customer (spending compared to last purchase)(use lag function)
c. Find number of returning customers and percent of returning customers for all week
d. Quarterly analysis: sales comparison(create a new quarter column using case when,12 weeks(3 months)=1 quarter)
(Use cte tables)
e. Are the customers spending more or less over time (group in 25 week segments)
f. Customer churn analysis for each quarter
g. Find the retained customers for each quarter
h.

Question and queries

1. Exploratory queries

- **1. A. Find out which age group is the most active shopper**

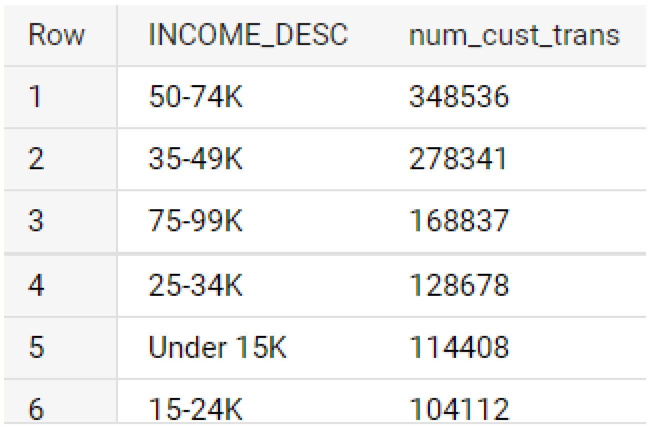
```
SELECT distinct(AGE_DESC), count(1) as num_cust_trans
FROM `dunnhumbysql.complete.hh_demographic`h
join `dunnhumbysql.complete.transaction_data`t
on t.household_key=h.household_key
group by AGE_DESC
order by num_cust_trans desc
```



Row	AGE_DESC	num_cust_trans
1	45-54	520586
2	35-44	386327
3	25-34	249829
4	65+	103857
5	55-64	91498
6	19-24	75206

- **1.b. Find out which income group shops the most**

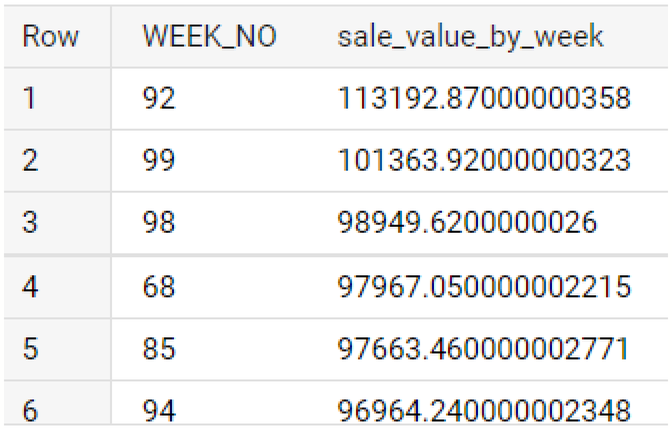
```
SELECT distinct(INCOME_DESC), count(1) as num_cust_trans
FROM `dunnhumbysql.complete.hh_demographic`h
join `dunnhumbysql.complete.transaction_data`t
on t.household_key=h.household_key
group by INCOME_DESC
order by num_cust_trans desc
```



Row	INCOME_DESC	num_cust_trans
1	50-74K	348536
2	35-49K	278341
3	75-99K	168837
4	25-34K	128678
5	Under 15K	114408
6	15-24K	104112

- **1.c. Which week had the best sales**

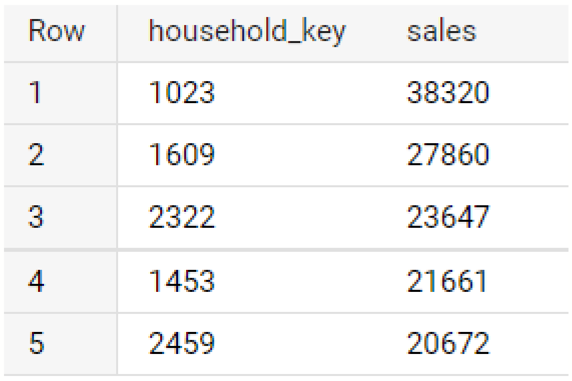
```
SELECT WEEK_NO, sum(SALES_VALUE) as sale_value_by_week
from `dunnhumbysql.complete.transaction_data`
group by WEEK_NO
order by sale_value_by_week desc
```



Row	WEEK_NO	sale_value_by_week
1	92	113192.87000000358
2	99	101363.92000000323
3	98	98949.6200000026
4	68	97967.050000002215
5	85	97663.460000002771
6	94	96964.240000002348

- **1.f. Top5 spending households with sales value in integer**

```
select   household_key,cast( sum(SALES_VALUE)  as int) as sales
from `dunnhumbysql.complete.transaction_data`
group by  household_key
order by sales desc
Limit 5
```

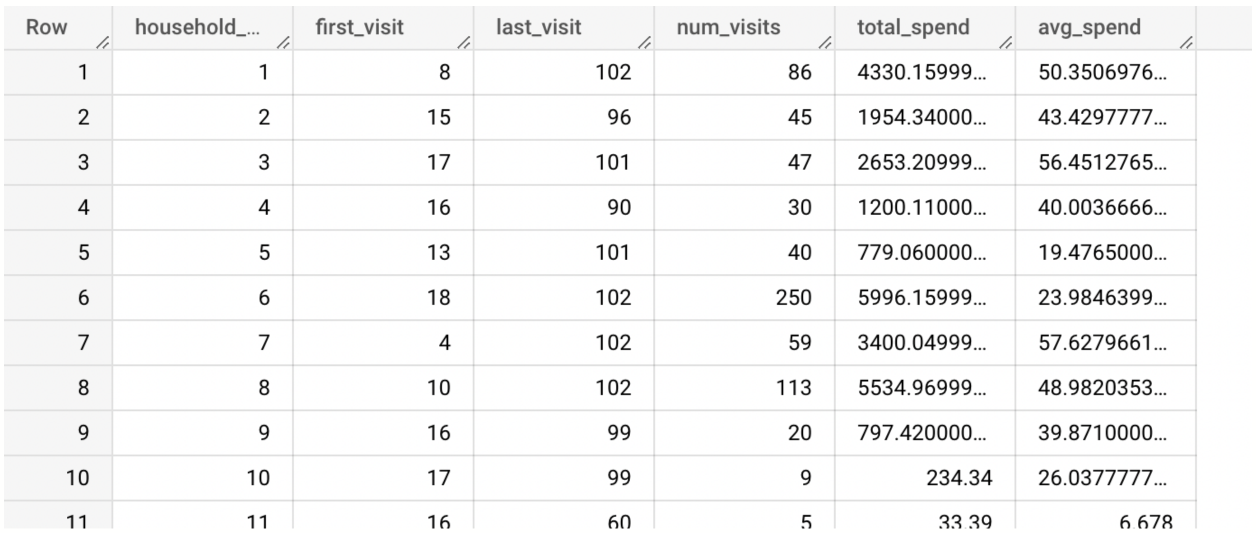


Row	household_key	sales
1	1023	38320
2	1609	27860
3	2322	23647
4	1453	21661
5	2459	20672

2. Customer analysis

- **2.a Create a basic customer profiling with first, last visit, number of visits, average money spent per visit and total money spent**

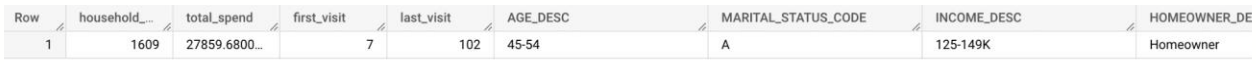
```
select household_key, min(WEEK_NO) as first_visit,
max(WEEK_NO) last_visit, count(distinct(BASKET_ID)) as num_visits,
sum(SALES_VALUE) as total_spend, (sum(SALES_VALUE)/count(distinct(BASKET_ID)))
as avg_spend
from `dunnhumbysql.complete.transaction_data`
group by household_key
order by household_key
```



Row	household_...	first_visit	last_visit	num_visits	total_spend	avg_spend
1	1	8	102	86	4330.15999...	50.3506976...
2	2	15	96	45	1954.34000...	43.4297777...
3	3	17	101	47	2653.20999...	56.4512765...
4	4	16	90	30	1200.11000...	40.0036666...
5	5	13	101	40	779.060000...	19.4765000...
6	6	18	102	250	5996.15999...	23.9846399...
7	7	4	102	59	3400.04999...	57.6279661...
8	8	10	102	113	5534.96999...	48.9820353...
9	9	16	99	20	797.420000...	39.8710000...
10	10	17	99	9	234.34	26.0377777...
11	11	16	60	5	33.39	6.678

- **2.b. Do a customer analysis for the most spending customer for whom we have demographic information**

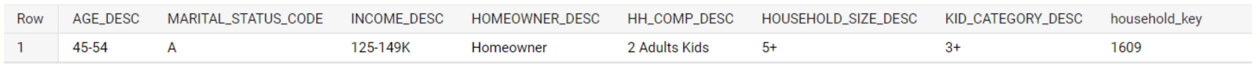
```
with cte as(
select t.household_key, sum(SALES_VALUE) as total_spend,
min(WEEK_NO) as first_visit,max(WEEK_NO) last_visit
from `dunnhumbysql.complete.transaction_data` t
inner join `dunnhumbysql.complete.hh_demographic` d
on d.household_key=t.household_key
group by t.household_key
order by total_spend desc
limit 1
)
select cte.*, d.* from cte
inner join `dunnhumbysql.complete.hh_demographic` d
on cte.household_key=d.household_key
```



Row	household_...	total_spend	first_visit	last_visit	AGE_DESC	MARITAL_STATUS_CODE	INCOME_DESC	HOMEOWNER_DE
1	1609	27859.6800...	7	102	45-54	A	125-149K	Homeowner

- **2.c. Get the demographic information for that**

```
select*
from `dunnhumbysql.complete.hh_demographic`
where household_key =(
select d.household_key
from `dunnhumbysql.complete.transaction_data` t
inner join `dunnhumbysql.complete.hh_demographic` d
on d.household_key=t.household_key
group by household_key
order by sum(SALES_VALUE) desc
limit 1
)
```



Row	AGE_DESC	MARITAL_STATUS_CODE	INCOME_DESC	HOMEOWNER_DESC	HH_COMP_DESC	HOUSEHOLD_SIZE_DESC	KID_CATEGORY_DESC	household_key
1	45-54	A	125-149K	Homeowner	2 Adults Kids	5+	3+	1609

- **2.d. What products does the top spender buys the most?**

```
select household_key, PRODUCT_ID, count(QUANTITY) as quant
from `dunnhumbysql.complete.transaction_data`
where household_key in(
select d.household_key,
from `dunnhumbysql.complete.transaction_data` t
inner join `dunnhumbysql.complete.hh_demographic` d
on t.household_key=d.household_key
group by d.household_key
order by sum(SALES_VALUE) desc
limit 1)
group by household_key, PRODUCT_ID
order by quant desc
limit 3
```

Row	household_key	PRODUCT_ID	quant
1	1609	1082185	160
2	1609	6632283	141
3	1609	951590	125

- **2.e. How many campaigns were they a part of?**

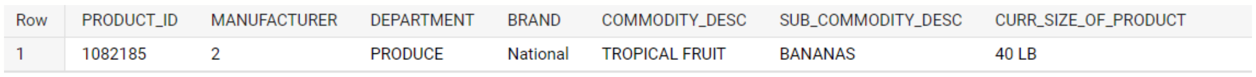
```
select*
from `dunnhumbysql.complete.campaign_table`
where household_key=1609
```

Row	DESCRIPTION	household_key	CAMPAIGN
1	TypeA	1609	13
2	TypeA	1609	18
3	TypeB	1609	11

3. Product analysis

- **3.a. For this we'll choose the best selling product from the transaction data ie :product_id:1082185**

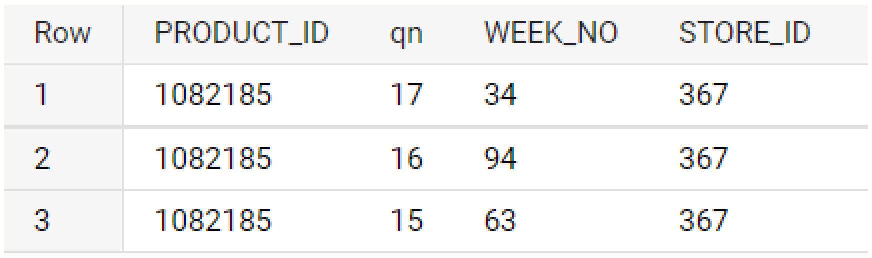
```
select * from `dunnhumbysql.complete.product`
where PRODUCT_ID = (
SELECT PRODUCT_ID,
FROM `dunnhumbysql.complete.transaction_data`
group by PRODUCT_ID
order by count(1)
limit 1
)
```



Row	PRODUCT_ID	MANUFACTURER	DEPARTMENT	BRAND	COMMODITY_DESC	SUB_COMMODITY_DESC	CURR_SIZE_OF_PRODUCT
1	1082185	2	PRODUCE	National	TROPICAL FRUIT	BANANAS	40 LB

- **3.b. When did the product sell the most and where (top 3)**

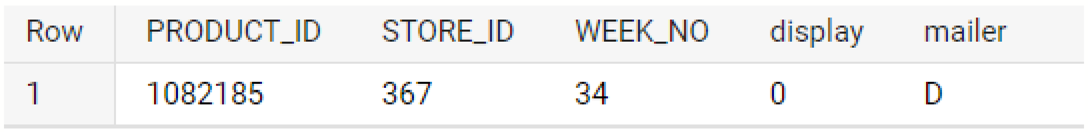
```
select PRODUCT_ID, count(QUANTITY) as qn, WEEK_NO, STORE_ID
from `dunnhumbysql.complete.transaction_data`
where PRODUCT_ID=(
SELECT PRODUCT_ID,
FROM `dunnhumbysql.complete.transaction_data`
group by PRODUCT_ID
order by count(1) desc
limit 1
)
group by PRODUCT_ID, WEEK_NO, STORE_ID
order by qn desc
limit 3
```



Row	PRODUCT_ID	qn	WEEK_NO	STORE_ID
1	1082185	17	34	367
2	1082185	16	94	367
3	1082185	15	63	367

- **3.c. Where was the product placed and featured for that particular store and week**

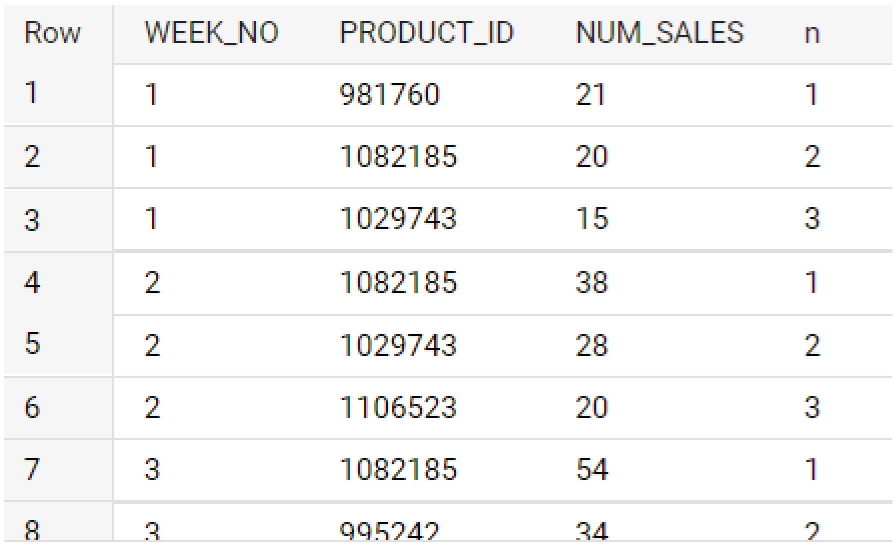
```
with cte as(
select PRODUCT_ID,  WEEK_NO, STORE_ID
from `dunnhumbysql.complete.transaction_data`
where PRODUCT_ID=(
SELECT PRODUCT_ID,
FROM `dunnhumbysql.complete.transaction_data`
group by PRODUCT_ID
order by count(1) desc
limit 1
)
group by PRODUCT_ID,STORE_ID, WEEK_NO
order by count(1) desc
limit 3
)

select c.*
from `dunnhumbysql.complete.causal_data` c
right join cte t
on c.PRODUCT_ID=t.PRODUCT_ID
where c.WEEK_NO in (t.WEEK_NO )
and c.STORE_ID in (t.STORE_ID)
```



Row	PRODUCT_ID	STORE_ID	WEEK_NO	display	mailer
1	1082185	367	34	0	D

- **3.f. Which products were the best seller(top 3) for each week and what quantity did they sell**

```
SELECT *
FROM (
    SELECT WEEK_NO, PRODUCT_ID,COUNT(PRODUCT_ID)AS NUM_SALES, ROW_NUMBER() OVER
(PARTITION BY WEEK_NO  ORDER BY COUNT(PRODUCT_ID) DESC) AS n
    FROM `dunnhumbysql.complete.transaction_data`
    GROUP BY WEEK_NO  , PRODUCT_ID
    ORDER BY WEEK_NO
) AS x
WHERE n <= 3
```

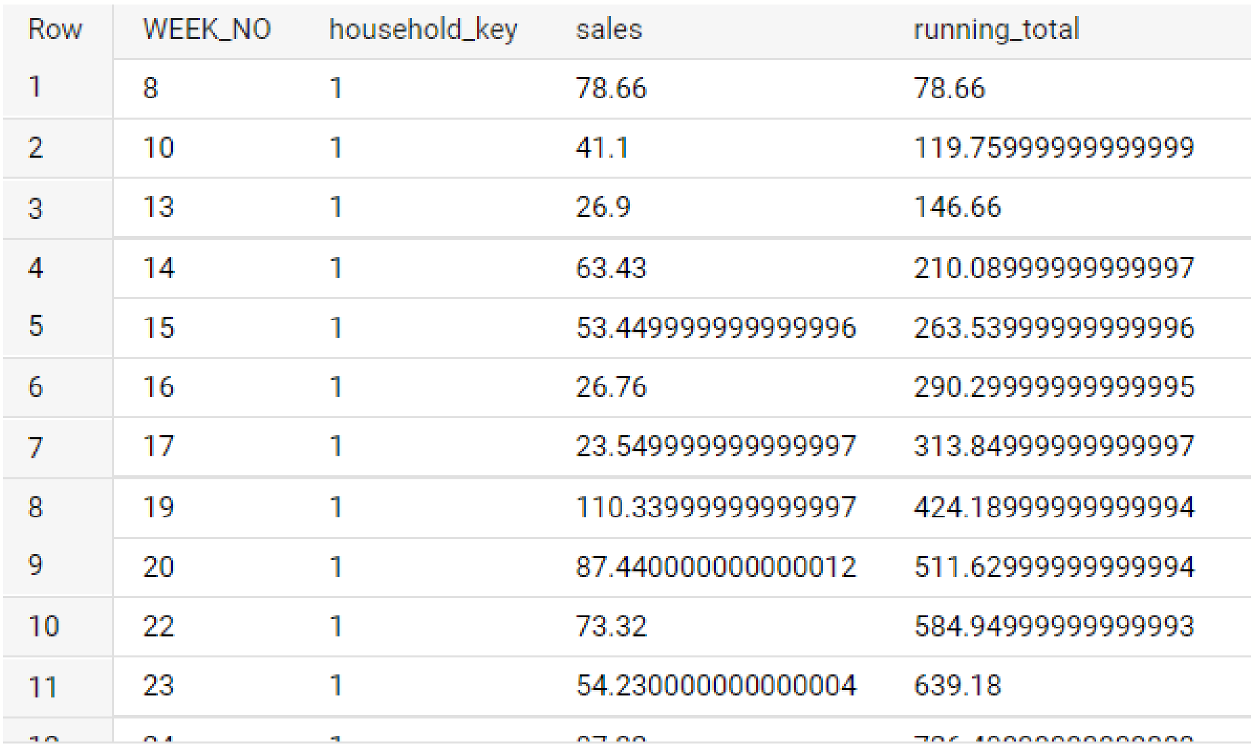


Row	WEEK_NO	PRODUCT_ID	NUM_SALES	n
1	1	981760	21	1
2	1	1082185	20	2
3	1	1029743	15	3
4	2	1082185	38	1
5	2	1029743	28	2
6	2	1106523	20	3
7	3	1082185	54	1
8	3	995242	34	2

4. Advance analysis and queries

- **4.a. Find out on which weeks does each household shop and find their cumulative spending over time**

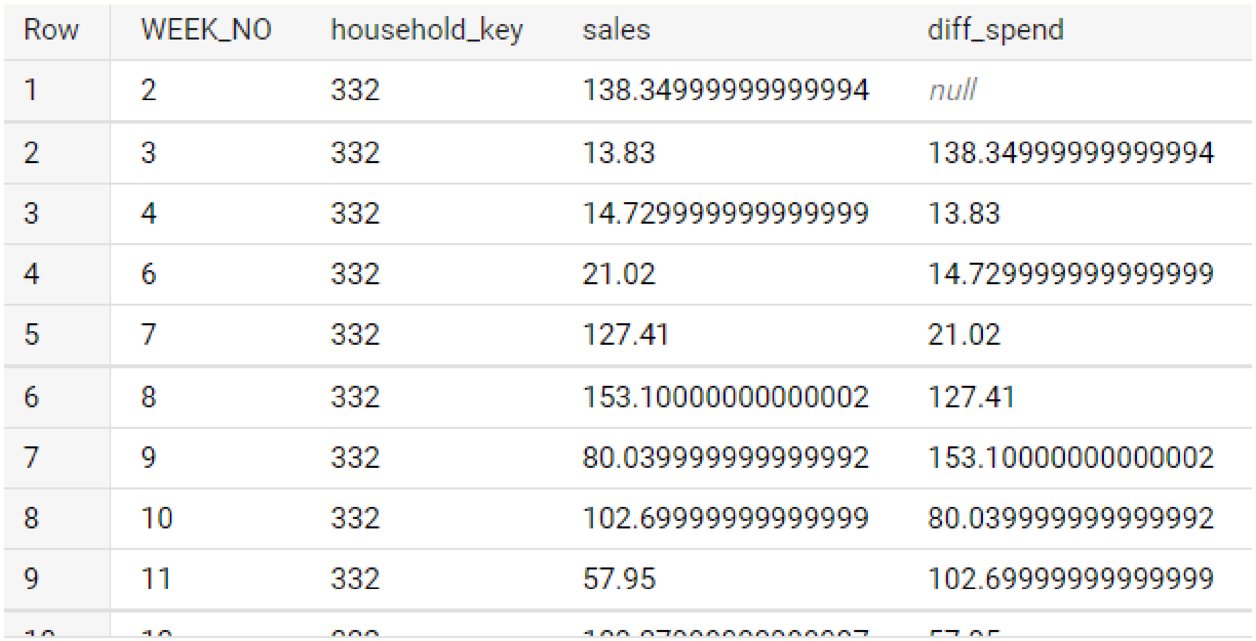
```
with cte as(
select WEEK_NO , household_key , sum(SALES_VALUE) as sales
FROM `dunnhumbysql.complete.transaction_data`
group by WEEK_NO, household_key
)

SELECT
   *,
  SUM(sales) OVER (PARTITION BY household_key ORDER BY week_no) AS
running_total
  from cte
```



Row	WEEK_NO	household_key	sales	running_total
1	8	1	78.66	78.66
2	10	1	41.1	119.75999999999999
3	13	1	26.9	146.66
4	14	1	63.43	210.08999999999997
5	15	1	53.449999999999996	263.53999999999996
6	16	1	26.76	290.29999999999995
7	17	1	23.549999999999997	313.84999999999997
8	19	1	110.33999999999997	424.18999999999994
9	20	1	87.440000000000012	511.62999999999994
10	22	1	73.32	584.94999999999993
11	23	1	54.230000000000004	639.18

- **4.b. Find the trend in spending for each customer**

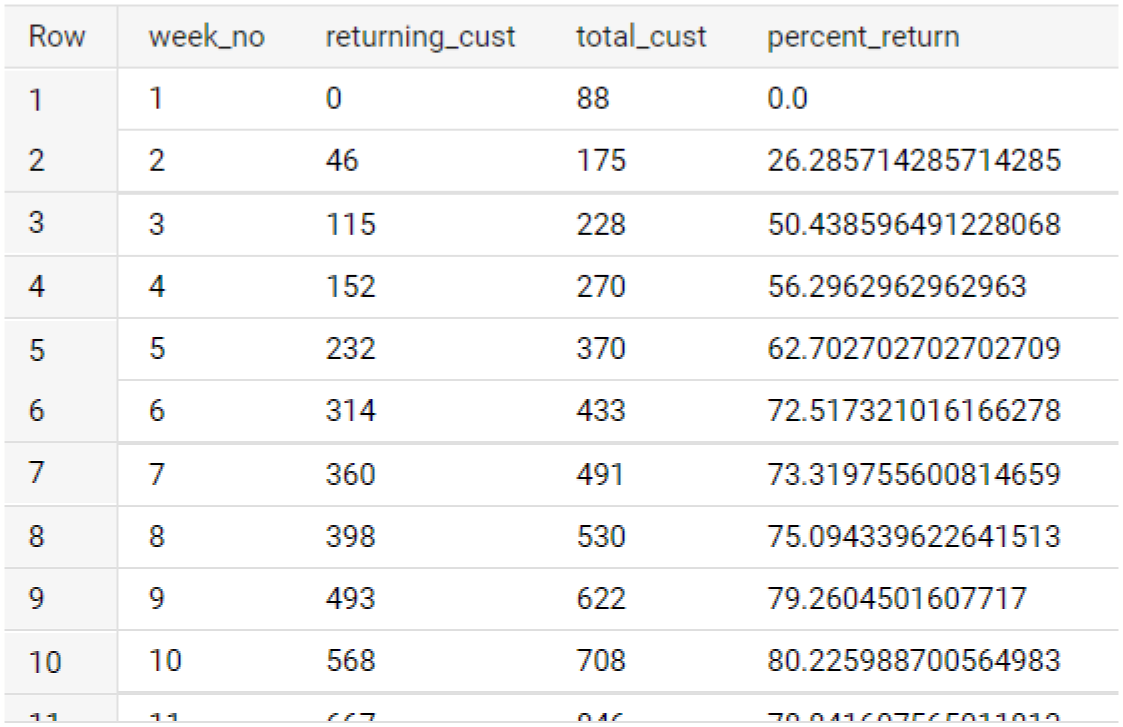
```
with cte as(
select WEEK_NO , household_key , sum(SALES_VALUE) as sales
FROM `dunnhumbysql.complete.transaction_data`
group by WEEK_NO, household_key
)

SELECT
   *,
  lag(sales)OVER (PARTITION BY household_key ORDER BY week_no) as
diff_spend,
  from cte
```



Row	WEEK_NO	household_key	sales	diff_spend
1	2	332	138.34999999999994	*null*
2	3	332	13.83	138.34999999999994
3	4	332	14.729999999999999	13.83
4	6	332	21.02	14.729999999999999
5	7	332	127.41	21.02
6	8	332	153.10000000000002	127.41
7	9	332	80.039999999999992	153.10000000000002
8	10	332	102.69999999999999	80.039999999999992
9	11	332	57.95	102.69999999999999

- **4.c. Find number of returning customers and percent of returning customers for all week**

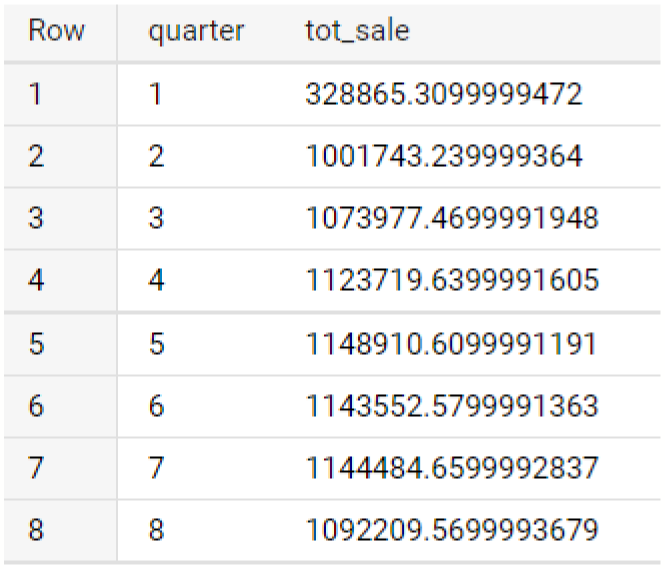
```
with cte as(
select b.week_no,  a.household_key, CASE
when min(a.week_no)<b.week_no then 1 else 0
end as decider
from `dunnhumbysql.complete.transaction_data` a
left join  `dunnhumbysql.complete.transaction_data`   b
on a.household_key=b.household_key
group by b.week_no, a.household_key
) select week_no,sum(decider) as returning_cust, count(decider) as
total_cust, (sum(decider)/count(decider))*100 as percent_return
from cte
group by 1
order by 1
```



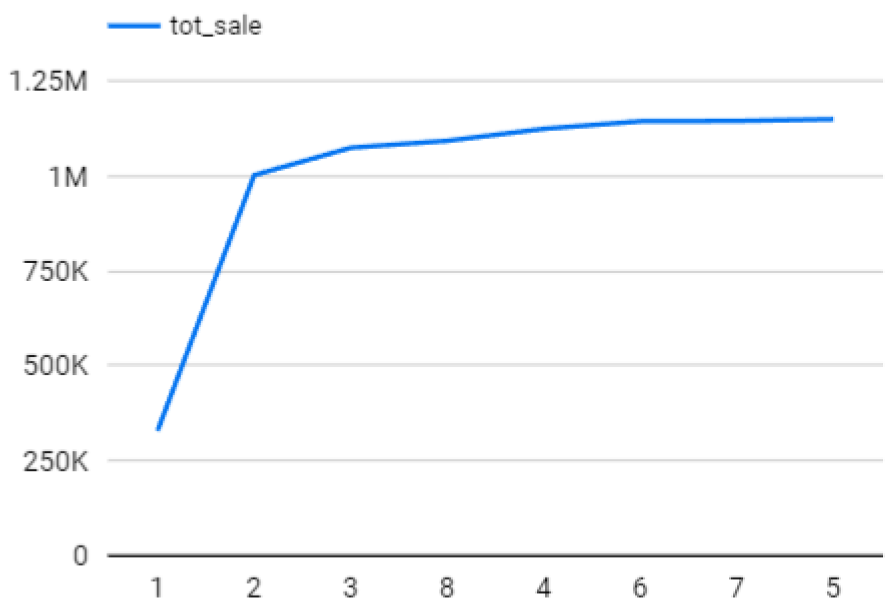
Row	week_no	returning_cust	total_cust	percent_return
1	1	0	88	0.0
2	2	46	175	26.285714285714285
3	3	115	228	50.438596491228068
4	4	152	270	56.2962962962963
5	5	232	370	62.702702702702709
6	6	314	433	72.517321016166278
7	7	360	491	73.319755600814659
8	8	398	530	75.094339622641513
9	9	493	622	79.2604501607717
10	10	568	708	80.225988700564983

- **4.d. Quarterly analysis: sales comparison over quarters**

```
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 `dunnhumbysql.complete.transaction_data`)
select quarter, sum(sales_value) as tot_sale
from cte
group by quarter
```

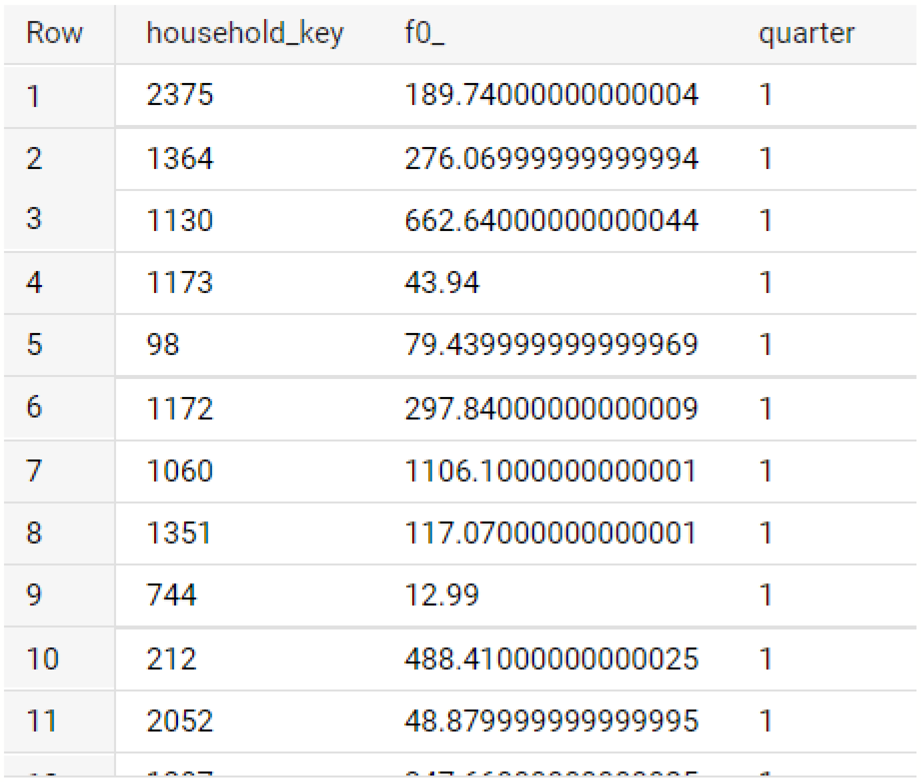


Row	quarter	tot_sale
1	1	328865.3099999472
2	2	1001743.239999364
3	3	1073977.4699991948
4	4	1123719.6399991605
5	5	1148910.6099991191
6	6	1143552.5799991363
7	7	1144484.6599992837
8	8	1092209.5699993679



- **4.e. Are the customers spending more or less over time**

```
select household_key,sum(sales), 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( select week_no, household_key, sum(SALES_VALUE) as sales,
from  `dunnhumbysql.complete.transaction_data` group by household_key,
WEEK_NO)
  group by  household_key, quarter
```



Row	household_key	f0_	quarter
1	2375	189.74000000000004	1
2	1364	276.06999999999994	1
3	1130	662.64000000000044	1
4	1173	43.94	1
5	98	79.439999999999969	1
6	1172	297.84000000000009	1
7	1060	1106.1000000000001	1
8	1351	117.07000000000001	1
9	744	12.99	1
10	212	488.41000000000025	1
11	2052	48.879999999999995	1

For better understanding selecting a single customer

```
select household_key,sum(sales), 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( select week_no, household_key, sum(SALES_VALUE) as sales,
from  `dunnhumbysql.complete.transaction_data` group by household_key,
WEEK_NO)
where household_key=2375
  group by  household_key, quarter
```

Row	household_key	f0_	quarter
1	2375	189.74000000000004	1
2	2375	55.760000000000005	3
3	2375	538.78000000000031	4
4	2375	437.61000000000007	5
5	2375	532.57000000000028	6
6	2375	624.73000000000025	7
7	2375	462.21000000000009	8

- **4.f. Customer churn analysis for each quarter**

```
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 `dunnhumbysql.complete.transaction_data`)
```

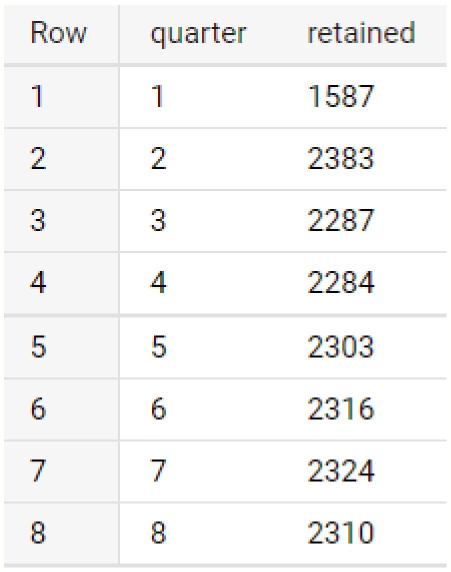
```
select  a.quarter, count(distinct(a.household_key)) as chrned
from cte a
FULL OUTER JOIN cte b
on a.household_key=b.household_key
and a.quarter<b.quarter
where b.household_key is NUll
group by a.quarter
order by quarter
```

Row	quarter	chrned
1	1	5
2	2	3
3	3	5
4	4	8
5	5	16
6	6	34
7	7	119
8	8	2310

- **4.g. Find the retained customers for each quarter**

```
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 `dunnhumbysql.complete.transaction_data`)
select  a.quarter, count(distinct(a.household_key)) as retained
```

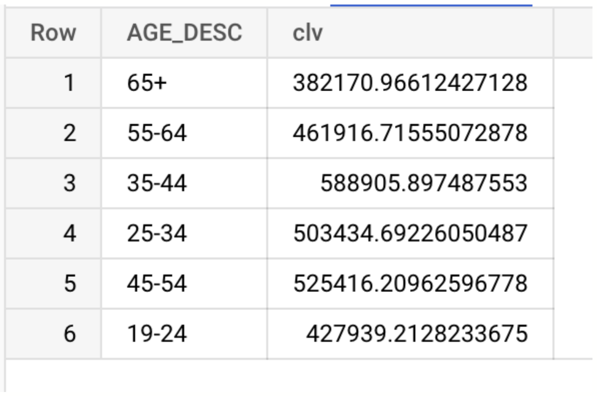
```
from cte a
left join cte b
on a.household_key=b.household_key and a.quarter>b.quarter
group by a.quarter
```



Row	quarter	retained
1	1	1587
2	2	2383
3	3	2287
4	4	2284
5	5	2303
6	6	2316
7	7	2324
8	8	2310

- **4.h. Calculate Customer lifetime value(CLV) for different age group**
    - **Average purchase value –** the value of all customer purchases over a particular time frame , divided by the number of purchases in that period
    - **Average purchase frequency –** divide the number of purchases in that same time period by the number of individual customers who made a transaction over the same period
    - **Customer value –** the average purchase frequency multiplied by the average purchase value
    - **Average customer lifespan –** the average length of time a customer continues buying from you
    - **CLV = customer value X average customer lifespan**

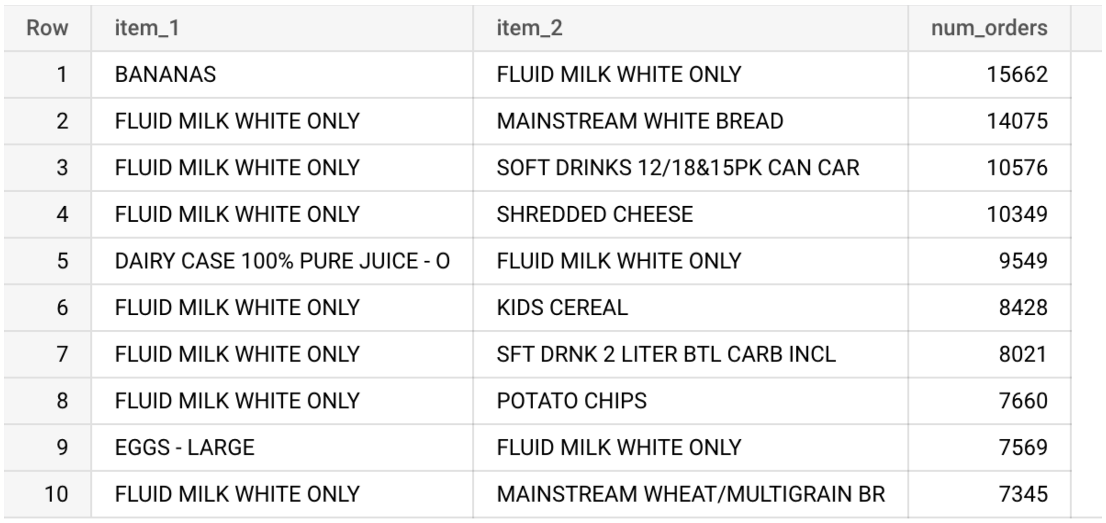
```
select AGE_DESC, (avg_purch_val*avg_purch_freq*avg_cust_lifespan) as clv
from(
with cte as (
select household_key, (max(WEEK_NO)- min (WEEK_NO)) as cust_duration
from `dunnhumbysql.complete.transaction_data`
group by household_key

)
select AGE_DESC,  sum(SALES_VALUE)/count(distinct(BASKET_ID)) as
avg_purch_val,
count(distinct(BASKET_ID))/count(distinct(d.household_key)) as avg_purch_freq,
(sum(cte.cust_duration)/count(1)) as avg_cust_lifespan,
from `dunnhumbysql.complete.transaction_data` t
inner join `dunnhumbysql.complete.hh_demographic` d
on t.household_key=d.household_key
join cte
on cte.household_key=d.household_key
group by AGE_DESC
)
```



Row	AGE_DESC	clv
1	65+	382170.96612427128
2	55-64	461916.71555072878
3	35-44	588905.897487553
4	25-34	503434.69226050487
5	45-54	525416.20962596778
6	19-24	427939.2128233675

- **4.i. Find products(product:SUB_COMMODITY_DESC) which are most frequently bought together**

```
with cte as (
SELECT *
FROM `dunnhumbysql.complete.product` p
join  `dunnhumbysql.complete.transaction_data` t
on p.PRODUCT_ID=t.PRODUCT_ID
)
select t.SUB_COMMODITY_DESC as item_1, t2.SUB_COMMODITY_DESC as item_2,
count(distinct t.BASKET_ID) as num_orders
from cte t
inner join cte t2
on t.BASKET_ID=t2.BASKET_ID
and t.SUB_COMMODITY_DESC<t2.SUB_COMMODITY_DESC
group by t.SUB_COMMODITY_DESC, t2.SUB_COMMODITY_DESC
order by num_orders desc
limit 10
```



Row	item_1	item_2	num_orders
1	BANANAS	FLUID MILK WHITE ONLY	15662
2	FLUID MILK WHITE ONLY	MAINSTREAM WHITE BREAD	14075
3	FLUID MILK WHITE ONLY	SOFT DRINKS 12/18&15PK CAN CAR	10576
4	FLUID MILK WHITE ONLY	SHREDDED CHEESE	10349
5	DAIRY CASE 100% PURE JUICE - O	FLUID MILK WHITE ONLY	9549
6	FLUID MILK WHITE ONLY	KIDS CEREAL	8428
7	FLUID MILK WHITE ONLY	SFT DRNK 2 LITER BTL CARB INCL	8021
8	FLUID MILK WHITE ONLY	POTATO CHIPS	7660
9	EGGS - LARGE	FLUID MILK WHITE ONLY	7569
10	FLUID MILK WHITE ONLY	MAINSTREAM WHEAT/MULTIGRAIN BR	7345