**Que – 1.** Find the number of orders that are small, medium or large order values (small : 0-5$, medium : 5-10$, large : 10+)

**Ans – 1.**

with t as

(select basket\_id,sum(quantity\*sales\_value) as total\_price,

case when sum(quantity\*sales\_value) <= 5

then "Small"

when sum(quantity\*sales\_value) <= 10

then "Medium"

else "Large"

end as size\_of\_order

from fluid-arc-394607.dunnhumby.transaction\_data

group by 1)

select size\_of\_order,count(\*) from t

group by 1

**Que – 2.** Find top 3 stores with the highest foot traffic for each week (Foot traffic : number of customer transacting)

**Ans – 2.**

with t1 as

(select store\_id,week\_no,count(household\_key)as footfall from fluid-arc-394607.dunnhumby.transaction\_data

group by 1,2),

t as

(select \*,dense\_rank()over(partition by week\_no order by footfall desc) as ranker

from t1)

select week\_no,store\_id,footfall,ranker from t

where ranker<=3

order by week\_no

**Que – 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.

**Ans – 3.**

select household\_key,

min(week\_no) as first\_visit,

max(week\_no) as last\_visit,

count(distinct basket\_id) as total\_visit,

round(sum(quantity\*sales\_value),2) as total\_amount,

round(sum(quantity\*sales\_value)/count(basket\_id),2) as avg\_money

from fluid-arc-394607.dunnhumby.transaction\_data

group by 1

-------------------------------------------------------------------------

**OR**

with base as

------------------------------------------------------------------------------------

(select household\_key,

min(week\_no) as first\_visit,

max(week\_no) as last\_visit,

count(distinct week\_no) as total\_visit,

round(sum(quantity\*sales\_value),2) as total\_amount

from fluid-arc-394607.dunnhumby.transaction\_data

group by 1),

------------------------------------------------------------------------------------

base1 as

(select household\_key,round(sum(total\_spent)/count(household\_key),2)as avg\_spent from

------------------------------------------------------------------------------------

(select household\_key,basket\_id,sum(quantity\*sales\_value) as total\_spent

from fluid-arc-394607.dunnhumby.transaction\_data

group by 1,2)

------------------------------------------------------------------------------------

group by 1)

------------------------------------------------------------------------------------

select b1.\*,b2.avg\_spent

from base as b1

left join base1 as b2

on b1.household\_key=b2.household\_key

order by b1.household\_key

**Que – 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)

**Ans – 4.**

with base as

(select household\_key,

min(week\_no) as first\_visit,

max(week\_no) as last\_visit,

count(distinct basket\_id) as total\_visit,

round(sum(quantity\*sales\_value),2) as total\_amount,

round(sum(quantity\*sales\_value)/count(basket\_id),2) as avg\_money

from fluid-arc-394607.dunnhumby.transaction\_data

group by 1)

select \* from fluid-arc-394607.dunnhumby.hh\_demographic as a

join base as b

on a.household\_key=b.household\_key

order by b.total\_amount desc

limit 1

**Que – 5.** Find products(product table : SUB\_COMMODITY\_DESC) which are most frequently bought together.

**Ans – 5.**

with base as

(select basket\_id,p.sub\_commodity\_desc as product

from fluid-arc-394607.dunnhumby.transaction\_data as t

inner join fluid-arc-394607.dunnhumby.product as p

on t.product\_id=p.product\_id)

select a.product,b.product,count(distinct a.basket\_id)as popularity

from base a

inner join base b

on a.basket\_id=b.basket\_id and a.product<b.product

group by 1,2

order by count(distinct a.basket\_id)desc

**Que – 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)

**Ans – 6.**

with base as

(select household\_key,week\_no,round(sum(quantity\*sales\_value),2) as spending

from fluid-arc-394607.dunnhumby.transaction\_data as t

group by 1,2)

select \*,round(sum(spending)over(partition by household\_key order by week\_no),2) as running\_sum

from base

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

**Ans – 7.**

with base as

(select household\_key,week\_no,round(sum(quantity\*sales\_value),2) as spending

from fluid-arc-394607.dunnhumby.transaction\_data as t

group by 1,2),

base2 as

(select \*,lag(spending,1)over(partition by household\_key order by week\_no) as prev\_week\_rev,

case when lag(spending,1)over(partition by household\_key order by week\_no)=0 then spending

else lag(spending,1)over(partition by household\_key order by week\_no)

end as final\_prev\_week\_rev

from base)

select \*,round((spending-final\_prev\_week\_rev)/final\_prev\_week\_rev\*100,2) as per\_change

from base2

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

**Ans – 8.**

with base as

(select household\_key, min(week\_no)over(partition by household\_key)as min\_week,week\_no

from fluid-arc-394607.dunnhumby.transaction\_data),

base\_2 as

(select \*, case when week\_no>min\_week then "returning" else "new\_customer" end as label

from base),

base\_3 as

(select week\_no,label,count(distinct household\_key) as count\_cust from base\_2

group by 1,2

order by 1 asc),

base\_4 as

(select \*,sum(count\_cust)over(partition by week\_no)as total\_cust

from base\_3)

select week\_no,round((count\_cust/total\_cust)\*100,2) as returning\_percentage

from base\_4

where label="returning"

order by week\_no

OR

-----------------------------------------------------------------------------------with base as

(select household\_key,week\_no

from fluid-arc-394607.dunnhumby.transaction\_data as t

group by 1,2),

base2 as

(select \*,lag(week\_no,1)over(partition by household\_key order by week\_no) as prev\_week\_no

from base),

base3 as

(select count(\*)over(partition by week\_no)

from base2

order by week\_no)

select week\_no,count(diff)/count(prev\_week\_no)\*100 as percentage\_ret\_customer

from base3

where prev\_week\_no is not null

group by 1

-- as total\_cutomer\_returning

--where diff=1

**Que – 9.** Quarterly analysis: sales comparison: total sale amount (create a new quarter column case where 12 weeks/3 months)-1 quarter)

**Ans – 9.**

with base as

(select \*,ceil(week\_no/12)as quarter

from fluid-arc-394607.dunnhumby.transaction\_data)

select quarter,round(sum(sales\_value\*quantity),2)as total\_sales

from base

group by 1

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

**Ans – 10.**

with base as

(select \*,ceil(week\_no/12)as quarter

from fluid-arc-394607.dunnhumby.transaction\_data)

select store\_id,quarter,round(sum(sales\_value\*quantity),2)as total\_sales

from base

group by 1,2

order by store\_id

**Que – 11.** Customer churn analysis for each quarter (churned customers : that never shop after that particular quarter)

**Ans – 11.**

with base as

(select household\_key,ceil(week\_no/12)as quarter

from fluid-arc-394607.dunnhumby.transaction\_data

group by 1,2)

select x.quarter,count(x.household\_key)as customer\_churned

from base as x

left join base as y

on x.household\_key=y.household\_key and x.quarter<y.quarter

where y.quarter is null

group by 1

order by x.quarter

**OR**

with base as

(select ceil(week\_no/12)as quarter,household\_key

from fluid-arc-394607.dunnhumby.transaction\_data

group by 1,2)

select x.quarter,count(distinct x.household\_key)as customer\_churned

from base as x

full outer join base as y

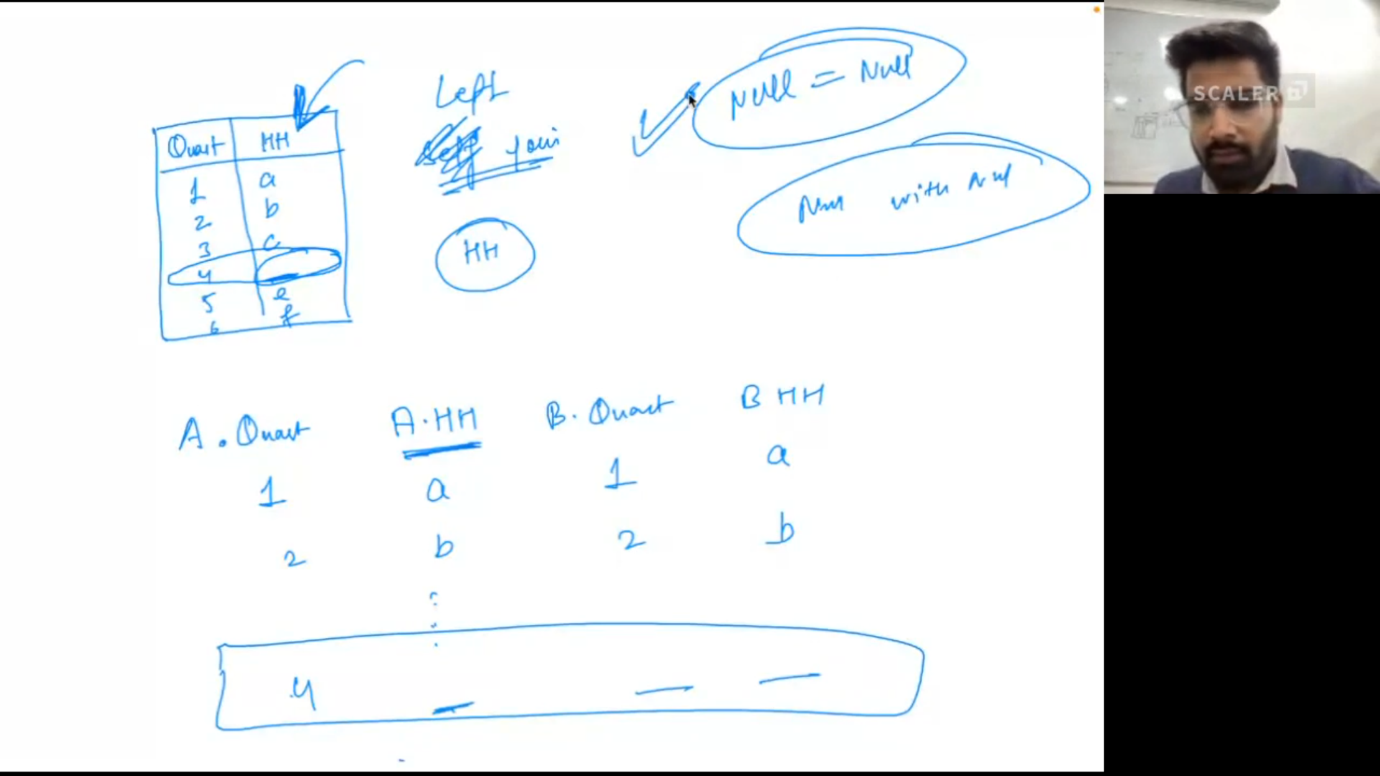
on x.household\_key=y.household\_key and x.quarter<y.quarter

where y.household\_key is null

group by 1

order by 1 asc

Here, we can’t use self join because Null will not going to connect with Null.



**Que – 12.** Find the retained customers for each quarter (retained : Households who were there in previous quarters and are there in the current quarter)

**Ans – 12.**

with base0 as

(select \*,ceil(week\_no/12)as quarter

from fluid-arc-394607.dunnhumby.transaction\_data),

base as

(select household\_key, min(quarter)over(partition by household\_key)as min\_quarter,quarter

from base0),

base\_2 as

(select \*, case when quarter>min\_quarter then "returning" else "new\_customer" end as label

from base),

base\_3 as

(select quarter,label,count(distinct household\_key) as count\_cust from base\_2

group by 1,2

order by 1 asc),

base\_4 as

(select \*,sum(count\_cust)over(partition by cast(quarter as int))as total\_cust

from base\_3)

select quarter,round((count\_cust/total\_cust)\*100,2) as returning\_percentage

from base\_4

where label="returning"

order by quarter

**Que – 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

**Ans – 13.**

with base as

(select household\_key, max(week\_no)-min(week\_no) as cust\_duration

from fluid-arc-394607.dunnhumby.transaction\_data

group by 1),

base\_2 as

(select AGE\_DESC, sum(a.cust\_duration)/count(\*)as avg\_cust\_lifespan,count(distinct b.basket\_id)/count(distinct a.household\_key)as aver\_pur\_freq,sum(b.quantity\*b.sales\_value)/count(distinct b.basket\_id)as avg\_pur\_value

from base a

right join fluid-arc-394607.dunnhumby.transaction\_data b

on a.household\_key=b.household\_key

left join fluid-arc-394607.dunnhumby.hh\_demographic c

on a.household\_key=c.household\_key

group by 1)

select \*,aver\_pur\_freq\*avg\_pur\_value\*avg\_cust\_lifespan as CLV from base\_2