**Top questions:**

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

select market\_basket, count(distinct basket\_id) as orders from (

select \*,

case when sales between 0 and 5 then 'small'

when sales > 5 and sales <= 10 then 'medium'

else 'large'

end as market\_basket

from

(

select basket\_id, sum(quantity \* sales\_value) as sales from `dunnhumby.transaction\_data`

group by 1

) base

)

group by 1

1. Find **week over week** **top 3 stores** with highest foot traffic

with base as (

select store\_id, week\_no, count(household\_key) as footfall

from `dunnhumby.transaction\_data`

group by 1,2

),

base\_2 as (

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

from base

)

select week\_no, store\_id, footfall, ranker from base\_2

where ranker<=3

order by week\_no asc

1. Could you provide insights into the temporal behavior of households by presenting the household\_key, the initial week of visitation, the concluding week of visitation, the total count of distinct basket\_id visits, the cumulative expenditure, and the average expenditure per visit?

select household\_key, min(WEEK\_NO) as first\_visit,

max(week\_no) as last\_visit,

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

sum(quantity\* sales\_value) as total\_money,

sum(quantity\* sales\_value)\*1.0/count(basket\_id) as avg\_money

from `dunnhumby.transaction\_data`

group by 1

1. 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)

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,

SUM(quantity\* sales\_value) AS total\_money,

SUM(quantity\* sales\_value)\*1.0/COUNT(basket\_id) AS avg\_money

FROM

`dunnhumby.transaction\_data`

GROUP BY

1),

base\_2 AS (

SELECT

\*

FROM

base b

INNER JOIN

`dunnhumby.hh\_demographic` d

ON

b.household\_key = d.household\_key)

SELECT

\*

FROM

base\_2

ORDER BY

total\_money desc

1. Find products which are most frequently bought together

with base as (

select basket\_id, p.SUB\_COMMODITY\_DESC as product

from `dunnhumby.transaction\_data` t

inner join

`dunnhumby.product` 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

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

with base as (

select household\_key, week\_no, sum(QUANTITY\* SALES\_VALUE) as spending

from `dunnhumby.transaction\_data`

group by 1,2

)

select \*, sum (spending) over (partition by household\_key order by week\_no asc) as runnning\_sum

from base

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

with base as (

select household\_key, week\_no, sum(QUANTITY\* SALES\_VALUE) as revenue

from `dunnhumby.transaction\_data`

group by 1,2

),

base\_2 as (

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

case when

lag (revenue, 1) over (partition by household\_key order by week\_no asc) =0 then revenue

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

end as final\_prev\_week\_rev

from base

)

select \*, round((revenue-final\_prev\_week\_rev)/final\_prev\_week\_rev\*100,2) as per\_change from base\_2

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

with base as (

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

from `dunnhumby.transaction\_data`

),

base\_2 as (

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

from base

)

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

group by 1,2

order by 1 asc

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

WITH

base AS (

SELECT

STORE\_ID,

CEIL(week\_no/12) AS quarter,

SUM(SALES\_VALUE\*QUANTITY) AS total\_sales

FROM

`dunnhumby.transaction\_data`

GROUP BY

1,

2),

base\_2 AS (

SELECT

\*,

LAG(total\_sales,1) OVER (PARTITION BY store\_id ORDER BY quarter ASC) AS pre\_quarter\_sales,

CASE

WHEN lag (total\_sales, 1) OVER (PARTITION BY store\_id ORDER BY quarter ASC) =0 THEN base.total\_sales

ELSE

(lag (total\_sales,

1) OVER (PARTITION BY store\_id ORDER BY quarter ASC) )

END

AS final\_prev\_quarter\_rev

FROM

base )

SELECT

\*,

ROUND((total\_sales-final\_prev\_quarter\_rev)/final\_prev\_quarter\_rev\*100,2) AS per\_change

FROM

base\_2

1. Customer churn analysis for each quarter (churned customers : that never shop after that particular quarter)

with base as (

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

from `dunnhumby.transaction\_data`

group by 1,2

)

select a.quarter, count(distinct a.household\_key) as churned

from base a full outer join

base b

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

and b.quarter>a.quarter

where b.household\_key is null

group by 1

order by 1 asc

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

WITH base AS (

  SELECT

    CEIL(week\_no / 12) AS quarter,

    household\_key

  FROM

    `dunnhumby.transaction\_data`

  GROUP BY 1,2 )

SELECT a.quarter,

  COUNT(DISTINCT a.household\_key) AS total\_customers,

  COUNT(DISTINCT CASE WHEN b.household\_key IS NULL THEN a.household\_key

  END ) AS churned,

  COUNT(DISTINCT

    CASE

      WHEN b.quarter = a.quarter - 1 THEN a.household\_key

  END ) AS retained

FROM base a

LEFT JOIN base b

ON a.household\_key = b.household\_key

  AND b.quarter = a.quarter - 1

GROUP BY 1

ORDER BY 1 ASC;

1. 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

with base as (

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

from `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 `dunnhumby.transaction\_data` b on a.household\_key=b.household\_key

left join `dunnhumby.demographics` 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