

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

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

$CLV = \text{customer value} \times \text{average customer lifespan}$

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

```

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

```

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

```

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_spend 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
```

5. Find products(product table: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

```

6. 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 `dunnhumysql.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

```

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

```

with cte as(

select WEEK_NO , household_key , sum(SALES_VALUE) as sales

FROM `dunnhumysql.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
```

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 `dunnhumysql.complete.transaction_data` a  
  
left join `dunnhumysql.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
```

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

```

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

```

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

```

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 `dunnhumby.sql.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

```

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 `dunnhumby.sql.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)
```