SQL scripts

Order transaction - https://console.cloud.google.com/bigquery?sq=444080637399:aed5163ceb5c4f818b4ff02adb78b9f6

select order\_id, user\_id, product\_id, status, created\_at, avg(sale\_price) from bigquery-public-data.thelook\_ecommerce.order\_items

group by order\_id, user\_id, product\_id, status, created\_at;

Products - <https://console.cloud.google.com/bigquery?sq=444080637399:796b4bbf881c4578828e234372e63b0a>

-- Group by product\_id, year-month to get total sale\_price for completed orders, with product names, after removing duplicates

WITH duplicates AS (

SELECT

order\_id,

product\_id,

sale\_price,

created\_at,

status,

ROW\_NUMBER() OVER (PARTITION BY order\_id, product\_id ORDER BY created\_at) AS row\_num

FROM

bigquery-public-data.thelook\_ecommerce.order\_items

WHERE

status = 'Complete' -- Filter only completed orders

)

-- Remove duplicates, join with products table to get product name, and group by product\_id and year-month

SELECT

product\_id,

p.name AS product\_name, -- Pull in the product name

FORMAT\_DATE('%Y-%m', DATE(created\_at)) AS sale\_year\_month,

SUM(sale\_price) AS total\_sale\_price

FROM

(SELECT oi.\*

FROM bigquery-public-data.thelook\_ecommerce.order\_items oi

JOIN duplicates d

ON oi.order\_id = d.order\_id

AND oi.product\_id = d.product\_id

WHERE d.row\_num = 1

) clean\_data

JOIN

bigquery-public-data.thelook\_ecommerce.products p

ON clean\_data.product\_id = p.id -- Join on product\_id = id in products table

GROUP BY

product\_id,

p.name,

sale\_year\_month

ORDER BY

sale\_year\_month,

product\_id;

Customer Segmentation - <https://console.cloud.google.com/bigquery?sq=444080637399:0384271bacc84b198a98b47807b0116b>

WITH customer\_order\_summary AS (

-- Summarize each customer's order history and spend

SELECT

u.id,

u.first\_name,

u.last\_name,

u.country,

u.city,

COUNT(DISTINCT oi.order\_id) AS order\_count,

SUM(oi.sale\_price) AS total\_spend,

MIN(oi.created\_at) AS first\_order\_date,

MAX(oi.created\_at) AS last\_order\_date

FROM

`bigquery-public-data.thelook\_ecommerce.order\_items` AS oi

JOIN

`bigquery-public-data.thelook\_ecommerce.users` AS u

ON

oi.user\_id = u.id

GROUP BY

u.id, u.first\_name, u.last\_name,u.country,u.city

),

customer\_segments AS (

-- Apply rules to categorize customers into segments

SELECT

id,

first\_name,

last\_name,

order\_count,

total\_spend,

first\_order\_date,

last\_order\_date,

country,

city,

CASE

WHEN order\_count = 1 THEN 'New'

WHEN order\_count > 1 AND total\_spend > 400 THEN 'VIP'

ELSE 'Returning'

END AS customer\_segment

FROM

customer\_order\_summary

)

-- Final query to display segmented customers

SELECT

id,

first\_name,

last\_name,

order\_count,

total\_spend,

customer\_segment,

country,

city,

FROM

customer\_segments

ORDER BY

customer\_segment, total\_spend DESC;

CLV and Churn – https://console.cloud.google.com/bigquery?sq=444080637399:f444110048c448368dd55ceff607aafb

WITH duplicates AS (

SELECT

order\_id,

user\_id,

product\_id,

sale\_price,

created\_at,

status,

ROW\_NUMBER() OVER (PARTITION BY order\_id, product\_id ORDER BY created\_at) AS row\_num

FROM

bigquery-public-data.thelook\_ecommerce.order\_items

WHERE

status = 'Complete' -- Filter only completed orders

),

clean\_data AS (

SELECT oi.\*

FROM bigquery-public-data.thelook\_ecommerce.order\_items oi

JOIN duplicates d

ON oi.order\_id = d.order\_id

AND oi.product\_id = d.product\_id

WHERE d.row\_num = 1

)

-- Calculate CLV per customer and churn status

SELECT

user\_id,

SUM(sale\_price) AS clv, -- Total sales per customer, representing CLV

COUNT(DISTINCT FORMAT\_DATE('%Y-%m', DATE(oi.created\_at))) AS total\_months, -- Number of months the customer has been active

MIN(DATE(created\_at)) AS first\_purchase\_date, -- First purchase date

MAX(DATE(created\_at)) AS last\_purchase\_date, -- Last purchase date

CASE

WHEN DATE\_DIFF(CURRENT\_DATE(), MAX(DATE(created\_at)), MONTH) > 6 THEN 'Churned' -- Consider churn if no purchase in the last 6 months

ELSE 'Active'

END AS churn\_status

FROM

clean\_data oi

JOIN

bigquery-public-data.thelook\_ecommerce.products p

ON oi.product\_id = p.id -- Join to pull in product names (optional for CLV)

GROUP BY

user\_id

ORDER BY

clv DESC;