Sales Data Analysis – SQL Queries

q1

select count(distinct customer\_unique\_id) from `targetData.customers`;

q2

select count(distinct order\_id) from `targetData.orders`;

q3

select min(order\_purchase\_timestamp), max(order\_purchase\_timestamp)

from `targetData.orders`;

q4

select count(distinct customer\_city) as city, count(distinct customer\_state) as state from `targetData.customers`;

q5

select count(distinct seller\_city) as city, count(distinct seller\_state) as state, count(distinct seller\_id) as sellers from `targetData.sellers`;

q6

select count(distinct product\_id) as prod\_id, count(distinct product\_category) as prod\_cat from `targetData.products`;

q7

select payment\_type, count(distinct order\_id)

from `targetData.payments`

group by payment\_type

order by 2 desc;

q8

select count(distinct order\_id), max(review\_score), min(review\_score) from `targetData.order\_reviews`;

q9

select distinct ord\_year, ord\_month, ord\_quarter,

count(distinct order\_id) over(partition by ord\_year,ord\_month) as month\_total,

count(distinct order\_id) over(partition by ord\_year,ord\_quarter) as quarter\_total,

from(

select order\_id, customer\_id, order\_status, order\_purchase\_timestamp,

extract(year from order\_purchase\_timestamp) as ord\_year,

extract(month from order\_purchase\_timestamp) as ord\_month,

case

  when extract(month from order\_purchase\_timestamp) between 1 and 3 then "Q1"

  when extract(month from order\_purchase\_timestamp) between 4 and 6 then "Q2"

  when extract(month from order\_purchase\_timestamp) between 7 and 9 then "Q3"

  when extract(month from order\_purchase\_timestamp) between 10 and 12 then "Q4"

end as ord\_quarter,

from `targetData.orders`)

order by 1 desc,2 desc

q10

select distinct ord\_year, ord\_month, ord\_quarter,

count(distinct order\_id) over(partition by ord\_year,ord\_month) as month\_total,

count(distinct order\_id) over(partition by ord\_year,ord\_quarter) as quarter\_total,

from(

select order\_id, customer\_id, order\_status, order\_purchase\_timestamp,

extract(year from order\_purchase\_timestamp) as ord\_year,

extract(month from order\_purchase\_timestamp) as ord\_month,

case

  when extract(month from order\_purchase\_timestamp) between 1 and 3 then "Q1"

  when extract(month from order\_purchase\_timestamp) between 4 and 6 then "Q2"

  when extract(month from order\_purchase\_timestamp) between 7 and 9 then "Q3"

  when extract(month from order\_purchase\_timestamp) between 10 and 12 then "Q4"

end as ord\_quarter,

from `targetData.orders`

where order\_status = "canceled")

order by 1 desc,2 desc

q11

select distinct ord\_timing,

count(order\_id) over(partition by ord\_timing) as count\_timing

from

(select \*, case

  when extract(hour from order\_purchase\_timestamp) between 2 and 6 then "1Dawn"

  when extract(hour from order\_purchase\_timestamp) between 7 and 11 then "2Morning"

  when extract(hour from order\_purchase\_timestamp) between 12 and 16 then "3Afternoon"

  when extract(hour from order\_purchase\_timestamp) between 17 and 23 then "4Night"

  when extract(hour from order\_purchase\_timestamp) between 0 and 1 then "4Night"

end as ord\_timing

from `targetData.orders`)

order by 1;

q12

select \* from

(select extract(hour from order\_purchase\_timestamp) as ord\_timing, count(order\_id) as count\_hour

from `targetData.orders`

group by extract(hour from order\_purchase\_timestamp))

order by 2 desc;

q13

select distinct customer\_state, ord\_year, ord\_month,

count(order\_id) over(partition by customer\_state, ord\_year, ord\_month) as ord\_count

from

(select \*,

extract(year from order\_purchase\_timestamp) as ord\_year,

extract(month from order\_purchase\_timestamp) as ord\_month

from `targetData.orders` o join `targetData.customers` c using (customer\_id))

q14

select \* from

(select distinct ord\_year, ord\_month,

first\_value(customer\_state) over(partition by ord\_year, ord\_month order by count\_order desc) as highest\_order\_state,

nth\_value(customer\_state,2) over(partition by ord\_year, ord\_month order by count\_order desc) as second\_highest\_order\_state

from

(select distinct

extract(year from order\_purchase\_timestamp) as ord\_year,

extract(month from order\_purchase\_timestamp) as ord\_month,

customer\_state, count(order\_id) over(partition by extract(year from order\_purchase\_timestamp), extract(month from order\_purchase\_timestamp), customer\_state) as count\_order

from `targetData.orders` o join `targetData.customers` c using(customer\_id)))

where second\_highest\_order\_state is not null

order by 1,2

q15

select distinct customer\_state, count(distinct customer\_city) over(partition by customer\_state) as no\_of\_city, no\_of\_customers, percentage\_cust

from

(select customer\_state, customer\_city,

count(customer\_unique\_id) over(partition by customer\_state) as no\_of\_customers, (count(customer\_unique\_id) over(partition by customer\_state)\*100/(select count(customer\_unique\_id) from `targetData.customers`)) as percentage\_cust

from `targetData.customers`)

order by 3 desc

q16

select distinct date\_tag, sum(payment\_value) over(partition by date\_tag) as total\_order\_value

from

(select \*,

case

  when (extract(year from order\_purchase\_timestamp) = 2017 and extract(month from order\_purchase\_timestamp) between 1 and 8) then '2017JanAug'

  when (extract(year from order\_purchase\_timestamp) = 2018 and extract(month from order\_purchase\_timestamp) between 1 and 8) then '2018JanAug'

  else 'NA'

end as date\_tag

from `targetData.orders` inner join `targetData.payments` using(order\_id))

where date\_tag <> 'NA';

q17

select (lead(total\_order\_value) over(order by total\_order\_value) - total\_order\_value)\*100/total\_order\_value as prcnt\_incr

from

(

select distinct date\_tag, sum(payment\_value) over(partition by date\_tag) as total\_order\_value

from

(select \*,

case

  when (extract(year from order\_purchase\_timestamp) = 2017 and extract(month from order\_purchase\_timestamp) between 1 and 8) then '2017JanAug'

  when (extract(year from order\_purchase\_timestamp) = 2018 and extract(month from order\_purchase\_timestamp) between 1 and 8) then '2018JanAug'

  else 'NA'

end as date\_tag

from `targetData.orders` inner join `targetData.payments` using(order\_id))

where date\_tag <> 'NA'

)

q18

SELECT

  customer\_state,

  COUNT(DISTINCT order\_id) AS orders,

  AVG(price) AS avg\_price,

  SUM(price) AS total\_price,

  AVG(freight\_value) AS avg\_freight,

  SUM(freight\_value) AS sum\_freight

FROM

  `targetData.orders` o

LEFT JOIN

  `targetData.customers` c

USING

  (customer\_id)

LEFT JOIN

  `targetData.order\_items` oi

USING

  (order\_id)

GROUP BY

  customer\_state

ORDER BY

  orders DESC;

q19

SELECT

  customer\_state,

  ROUND(AVG(freight\_value)) AS avg\_freight\_val,

  ROUND(AVG(time\_to\_delivery)) AS avg\_delivery\_time,

  ROUND(AVG(diff\_estimated\_delivery)) AS avg\_delay,

  COUNT(order\_id) AS order\_count

FROM (

  SELECT

    \*,

    DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, day) AS time\_to\_delivery,

    DATE\_DIFF(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, day) AS diff\_estimated\_delivery

  FROM

    `targetData.orders` o

  LEFT JOIN

    `targetData.customers` c

  USING

    (customer\_id)

  LEFT JOIN

    `targetData.order\_items` oi

  USING

    (order\_id))

GROUP BY

  customer\_state

q20

SELECT

  DISTINCT EXTRACT(year

  FROM

    order\_purchase\_timestamp) AS ord\_year,

  EXTRACT(month

  FROM

    order\_purchase\_timestamp) AS ord\_month,

  payment\_type,

  COUNT(order\_id) OVER(PARTITION BY EXTRACT(year FROM order\_purchase\_timestamp),

    EXTRACT(month

    FROM

      order\_purchase\_timestamp),

    payment\_type) AS order\_count

FROM

  `targetData.orders` o

JOIN

  `targetData.payments` p

USING

  (order\_id)

ORDER BY

  4 desc

q21

SELECT

  ord\_month,

  payment\_type,

  order\_count,

  FIRST\_VALUE(payment\_type) OVER(PARTITION BY ord\_month ORDER BY order\_count DESC) AS preferred\_mode

FROM (

  SELECT

    DISTINCT EXTRACT(month

    FROM

      order\_purchase\_timestamp) AS ord\_month,

    payment\_type,

    COUNT(order\_id) OVER(PARTITION BY EXTRACT(month FROM order\_purchase\_timestamp),

      payment\_type) AS order\_count

  FROM

    `targetData.orders` o

  JOIN

    `targetData.payments` p

  USING

    (order\_id))

ORDER BY

  1,

  3 desc

q22

SELECT

  DISTINCT payment\_installments,

  COUNT(order\_id) OVER(PARTITION BY payment\_installments) AS count\_of\_orders

FROM

  `targetData.orders` o

JOIN

  `targetData.payments` p

USING

  (order\_id)

ORDER BY

  2 DESC;

q23

select review\_score, count(order\_id) as order\_count

from `targetData.order\_reviews`

group by review\_score

q24

SELECT

  DISTINCT product\_category,

  COUNT(product\_id) OVER(PARTITION BY product\_category) AS count\_product

FROM (

  SELECT

    DISTINCT product\_id,

    AVG(review\_score) OVER(PARTITION BY product\_id) AS avg\_rating,

    CASE

      WHEN AVG(review\_score) OVER (PARTITION BY product\_id) > 4 THEN "Excellent"

      WHEN AVG(review\_score) OVER (PARTITION BY product\_id) BETWEEN 3

    AND 4 THEN "Good"

      WHEN AVG(review\_score) OVER (PARTITION BY product\_id) BETWEEN 2 AND 3 THEN "Average"

      WHEN AVG(review\_score) OVER (PARTITION BY product\_id) < 2 THEN "Bad"

  END

    AS product\_category

  FROM

    `targetData.order\_items` o

  JOIN

    `targetData.products` p

  USING

    (product\_id)

  JOIN

    `targetData.order\_reviews` orr

  USING

    (order\_id))

q25

WITH

  payments1 AS (

  SELECT

    DISTINCT order\_id,

    SUM(payment\_value) OVER(PARTITION BY order\_id) AS total\_amount

  FROM

    `targetData.payments`),

  rfm AS (

  SELECT

    customer\_unique\_id,

    DATE\_DIFF('2018-11-01', MAX(EXTRACT(date

        FROM

          order\_purchase\_timestamp)), day) AS recency,

    COUNT(order\_id) AS frequency,

    SUM(total\_amount) AS monetary

  FROM

    `targetData.orders` o

  LEFT JOIN

    `targetData.customers` c

  USING

    (customer\_id)

  LEFT JOIN

    payments1

  USING

    (order\_id)

  GROUP BY

    customer\_unique\_id)

SELECT

  customer\_unique\_id,

  NTILE(4) OVER(ORDER BY recency) AS r\_score,

  NTILE(4) OVER(ORDER BY frequency DESC) AS f\_score,

  NTILE(4) OVER(ORDER BY monetary DESC) AS m\_score,

  CONCAT(NTILE(4) OVER(ORDER BY recency), "-", NTILE(4) OVER(ORDER BY frequency DESC), "-", NTILE(4) OVER(ORDER BY monetary DESC)) AS rfm\_group

FROM

  rfm

ORDER BY

  5

q26

WITH

  payments1 AS (

  SELECT

    DISTINCT order\_id,

    SUM(payment\_value) OVER(PARTITION BY order\_id) AS total\_amount

  FROM

    `targetData.payments`),

  rfm AS (

  SELECT

    customer\_unique\_id,

    DATE\_DIFF('2018-11-01', MAX(EXTRACT(date

        FROM

          order\_purchase\_timestamp)), day) AS recency,

    COUNT(order\_id) AS frequency,

    SUM(total\_amount) AS monetary

  FROM

    `targetData.orders` o

  LEFT JOIN

    `targetData.customers` c

  USING

    (customer\_id)

  LEFT JOIN

    payments1

  USING

    (order\_id)

  GROUP BY

    customer\_unique\_id),

  rfmgroup AS (

  SELECT

    customer\_unique\_id,

    NTILE(4) OVER(ORDER BY recency) AS r\_score,

    NTILE(4) OVER(ORDER BY frequency DESC) AS f\_score,

    NTILE(4) OVER(ORDER BY monetary DESC) AS m\_score,

    CONCAT(NTILE(4) OVER(ORDER BY recency), "-", NTILE(4) OVER(ORDER BY frequency DESC), "-", NTILE(4) OVER(ORDER BY monetary DESC)) AS rfm\_group

  FROM

    rfm)

SELECT

  rfm\_group,

  COUNT(\*) AS no\_of\_customers

FROM

  rfmgroup

GROUP BY

  rfm\_group

ORDER BY

  1