**MySQL Queries**

**Basic Queries:-**

**1. List all unique cities where customers are located.**

**->** select distinct customer\_city from ecommerce\_database.customers;

**2. Count the number of orders placed in 2021.**

**->** select count(\*) as 'Total\_orders\_placed\_in\_2021' from ecommerce\_database.orders where year(order\_approved\_at) = 2021;

**3. Find the total sales per category.**

**->** select products.product\_category as 'Product\_Category',round(sum(order\_items.price),2) as 'Total\_sales'

from ecommerce\_database.products join ecommerce\_database.order\_items

on ecommerce\_database.products.product\_id = ecommerce\_database.order\_items.product\_id

group by product\_category;

**4. Calculate the percentage of orders that were paid in installments.**

**->** select concat(round(( select count(\*) from

ecommerce\_database.orders o join ecommerce\_database.payments p on o.order\_id=p.order\_id

where p.payment\_installments > 0 )

/( select count(\*) from ecommerce\_database.orders )\*100,2),'%')

as 'Orders\_paid\_in\_installments';

**5. Count the number of customers from each state.**

**->** select customer\_state as 'Customer\_state',

count(customer\_id) as 'Total\_customers'

from ecommerce\_database.customers

group by customer\_state;

**Intermediate Queries:-**

**1. Calculate the number of orders per month in 2022.**

**->** select monthname(order\_purchase\_timestamp) as 'Month\_of\_2022',

count(order\_id) as 'Total\_orders'

from ecommerce\_database.orders

where year(order\_purchase\_timestamp)=2022

group by monthname(order\_purchase\_timestamp);

**2. Find the average number of products per order, grouped by customer city.**

**->** SELECT subquery.customer\_city AS 'Customer\_city',

round(avg(subquery.order\_product\_count),2) AS 'Avg\_no\_of\_products/order'

FROM (

SELECT x.customer\_city, y.order\_id, COUNT(z.order\_item\_id) AS order\_product\_count

FROM ecommerce\_database.customers x

JOIN ecommerce\_database.orders y ON x.customer\_id = y.customer\_id

JOIN ecommerce\_database.order\_items z ON y.order\_id = z.order\_id

GROUP BY x.customer\_city, y.order\_id

) AS subquery

GROUP BY subquery.customer\_city;

**3. Calculate the percentage of total revenue contributed by each product category.**

**->** SELECT UPPER(p.product\_category) AS category,

concat(ROUND((SUM(pa.payment\_value) / (SELECT SUM(payment\_value)

FROM ecommerce\_database.payments)) \* 100, 2),'%') AS sales\_percentage

FROM ecommerce\_database.products p

JOIN ecommerce\_database.order\_items i ON p.product\_id = i.product\_id

JOIN ecommerce\_database.payments pa ON pa.order\_id = i.order\_id

GROUP BY category

ORDER BY sales\_percentage DESC;

**4. Identify the correlation between product price and the number of times a product has been purchased.**

**->** SELECT p.product\_id AS Product\_id,

round(avg(i.price),2) AS AVG\_price,

COUNT(i.product\_id) AS Purchase\_count

FROM ecommerce\_database.products p

JOIN ecommerce\_database.order\_items i ON p.product\_id = i.product\_id

GROUP BY p.product\_id

ORDER BY AVG\_price desc;

**5. Calculate the total revenue generated by each seller, and rank them by revenue.**

**->**  SELECT s.seller\_id as Seller\_id,

concat(ROUND((SUM(pa.payment\_value) / (SELECT SUM(payment\_value)

FROM ecommerce\_database.payments)) \* 100, 2),'%') AS Sales\_percentage,

RANK() OVER (ORDER BY SUM(pa.payment\_value) DESC) AS 'Rank'

FROM ecommerce\_database.sellers s

JOIN ecommerce\_database.order\_items i ON s.seller\_id = i.seller\_id

JOIN ecommerce\_database.payments pa ON pa.order\_id = i.order\_id

GROUP BY Seller\_id

ORDER BY Sales\_percentage DESC;

**Advanced Queries:-**

**1. Calculate the moving average of order values for each customer over their order history.**

**->** select customer\_id,order\_purchase\_timestamp,round(payment,2) as Payment,

round((avg(payment) over(partition by customer\_id order by order\_purchase\_timestamp

rows between 2 preceding and current row)),2) as Moving\_avg

from

(select o.customer\_id,o.order\_purchase\_timestamp,

p.payment\_value as Payment

from ecommerce\_database.payments p join ecommerce\_database.orders o

on p.order\_id = o.order\_id) as subquery;

**2. Calculate the cumulative sales per month for each year.**

**->** SELECT YEAR(o.order\_purchase\_timestamp) AS Years,

MONTH(o.order\_purchase\_timestamp) AS Months,

ROUND(SUM(p.payment\_value), 2) AS Total\_Payment

FROM ecommerce\_database.payments p JOIN ecommerce\_database.orders o

ON p.order\_id = o.order\_id

GROUP BY Years, Months

ORDER BY Years, Months;

**3. Calculate the year-over-year growth rate of total sales.**

**->** WITH yearly\_payments AS (

SELECT YEAR(o.order\_purchase\_timestamp) AS Years,

ROUND(SUM(p.payment\_value), 2) AS Total\_Payment

FROM ecommerce\_database.payments p

JOIN ecommerce\_database.orders o

ON p.order\_id = o.order\_id

GROUP BY Years

ORDER BY Years

),

yearly\_growth AS (

SELECT P.Years,P.Total\_Payment,

(SELECT Total\_Payment FROM yearly\_payments WHERE Years = P.Years - 1) AS Previous\_Year\_Payment

FROM yearly\_payments P

)

SELECT YG.Years,

ROUND(

((YG.Total\_Payment - YG.Previous\_Year\_Payment) /

(YG.Previous\_Year\_Payment)) \* 100, 2

) AS Year\_Over\_Year\_Growth\_Rate

FROM yearly\_growth YG

WHERE YG.Years != 2020;

**4. Calculate the retention rate of customers, defined as the percentage of customers who make another purchase within 6 months of their first purchase.**

**->** WITH first\_orders AS (

SELECT c.customer\_id,

MIN(o.order\_purchase\_timestamp) AS first\_order

FROM ecommerce\_database.customers c

JOIN ecommerce\_database.orders o ON c.customer\_id = o.customer\_id

GROUP BY c.customer\_id

),

repeat\_orders AS (

SELECT fo.customer\_id,

COUNT(DISTINCT o.order\_id) AS next\_order\_count

FROM first\_orders fo

JOIN ecommerce\_database.orders o ON o.customer\_id = fo.customer\_id

WHERE o.order\_purchase\_timestamp > fo.first\_order

AND o.order\_purchase\_timestamp < DATE\_ADD(fo.first\_order, INTERVAL 6 MONTH)

GROUP BY fo.customer\_id

)

SELECT concat(round(100 \* (COUNT(DISTINCT fo.customer\_id) / NULLIF(COUNT(DISTINCT ro.customer\_id), 0)),2),'%')

AS Repeat\_Purchase\_Rate

FROM first\_orders fo LEFT JOIN repeat\_orders ro

ON fo.customer\_id = ro.customer\_id;

**5. Identify the top 3 customers who spent the most money in each year.**

**->** select years, customer\_id, payment,Rank\_of\_user

from

(select year(orders.order\_purchase\_timestamp) years,

orders.customer\_id,sum(payments.payment\_value) payment,

dense\_rank() over(partition by year(orders.order\_purchase\_timestamp)

order by sum(payments.payment\_value) desc) Rank\_of\_user

from ecommerce\_database.orders join ecommerce\_database.payments

on payments.order\_id = orders.order\_id

group by year(orders.order\_purchase\_timestamp),

orders.customer\_id) as a

where Rank\_of\_user <= 3 ;