**Project Title: Online Grocery Sales Dashboard (Blinkit Style)**

**Objective:**

To analyze sales, products, and customer orders using SQL queries and generate insights for business performance.

**Dataset:**

* Table 1: Products (ProductID, ProductName, Category, Price)
* Table 2: Orders(OrderID, OrderDate, CustomerID, ProductID, Quantity, TotalAmount)

**🔍 Key SQL Tasks:**

**-- 1. Total sales and number of orders by month**

select monthname(orderdate) as Sales\_month,

count(orderid) as Num\_of\_orders,

sum(totalamount) as Total\_sales

from orders

group by Sales\_month order by Sales\_month ;

**-- 2. Top 5 best-selling products**

Select p.productname,sum(o.quantity) as total\_quantity from orders o

join products p on

p.productid = o.productid

Group by productname order by total\_quantity Desc limit 5;

**-- 3. Revenue by product category**

select p.category, sum(o.totalamount) as Revenue from orders o

join products p on

p.productid = o.productid

Group by p.category;

**-- 4. Identify repeat customer’s vs one-time buyers**

with cte as (

select customerID, count(orderid) as order\_count from orders

group by customerid )

select customerid,order\_count,

case

when order\_count > 1 then 'Repeat customer'

else 'one time buyer'

end as buyer\_type

from cte ;

**-- 5. Find the most frequently ordered products**

select p.productname,count(o.orderid) as Numof\_times\_orders from orders o

join products p on

p.productid= o.productid

group by p.productname

order by Numof\_times\_orders Desc

limit 1;

## **INTERMEDIATE SQL QUESTIONS**

### 1. ****Top N Products by Sales****

Write a query to find the **top 5 products** with the highest total sales (quantity × price).

SELECT

p.productName,

SUM(o.quantity \* p.price) AS total\_sales

FROM

products p

JOIN

orders o ON p.ProductID = o.ProductID

GROUP BY

p.productName

ORDER BY

total\_sales DESC

LIMIT 5;

### 2. ****Monthly Revenue Trend****

Show the **total revenue for each month** over the last 6 months.

SELECT

DATE\_FORMAT(o.orderdate, '%Y-%m') AS order\_month,

SUM(o.quantity \* p.price) AS total\_revenue

FROM

orders o

JOIN

products p ON o.ProductID = p.ProductID

WHERE

o.orderdate >= DATE\_SUB(CURDATE(), INTERVAL 6 MONTH)

GROUP BY

order\_month

ORDER BY

order\_month DESC;

### 3. ****Category-wise Revenue****

Find the **total revenue per product category**. Include only categories with revenue > ₹10,000.

SELECT

p.category,

SUM(o.quantity \* p.price) AS total\_sales

FROM

products p

JOIN

orders o ON p.ProductID = o.ProductID

GROUP BY

p.category

Having total\_sales > 10000

order by total\_sales Desc ;

### 4. ****Customer with Most Orders****

Find the **customer ID** with the highest number of orders and their total spending.

SELECT

CustomerID,

COUNT(OrderID) AS num\_orders,

SUM(TotalAmount) AS total\_spending

FROM orders

GROUP BY CustomerID

ORDER BY num\_orders DESC

LIMIT 1;

### 5. ****Average Order Value****

Calculate the **average total amount** per order. Round the result to 2 decimal places.

SELECT

SUM(TotalAmount) AS total\_revenue,

COUNT(OrderID) AS total\_orders,

Round( SUM(TotalAmount) / COUNT(OrderID) , 2) AS Average\_total

From orders;

## **ADVANCED SQL QUESTIONS**

### 6. ****Top Selling Product per Category****

For each category, find the **product with the highest total quantity sold**.

with sales as (

SELECT

p.Category,

p.ProductName,

SUM(o.Quantity) AS total\_quantity

FROM orders o

join products p on

p.productID = o.productID

GROUP BY Category, ProductName

),

ranked\_products AS (

SELECT \*,

ROW\_NUMBER() OVER (PARTITION BY Category ORDER BY total\_quantity DESC) AS rn

FROM sales

)

SELECT Category, ProductName, total\_quantity

FROM ranked\_products

WHERE rn = 1;

### 7. ****Orders Without Matching Products****

Write a query to find orders that **do not have a matching product** in the product table (simulate data integrity check using LEFT JOIN).

SELECT o.\*

FROM orders o

LEFT JOIN products p ON o.ProductID = p.ProductID

WHERE p.ProductID IS NULL;

### 8. ****Running Total of Revenue by Date****

Show **daily revenue** and a **running total** of revenue ordered by date.

with cte as (

select

orderdate,

Sum(Totalamount) as Daily\_revenue

From orders

Group by orderdate

)

select

orderdate,

Daily\_revenue,

SUM(Daily\_revenue) OVER (ORDER BY OrderDate) AS running\_total From cte

order by orderdate;

### 9. ****Most Popular Day of the Week for Orders****

Determine the **day of the week** (e.g., Monday, Tuesday…) with the **most orders placed**.

select dayname(orderdate) as Day\_Week,count(\*) as order\_count from orders

group by Day\_week

order by order\_count Desc limit 1;

### 10. ****% Contribution of Each Category to Total Revenue****

Show each product category’s **percentage contribution** to the total revenue.

with cte as (

select p.category, sum(o.totalamount) as category\_revenue from orders o

join products p on

p.productid = o.productid

Group by p.category

)

select category , category\_revenue ,

Round (100 \* category\_revenue / sum(category\_revenue) over (), 2) as percentage\_total\_revenue

from cte

order by percentage\_total\_revenue Desc;