Case Study - E-Commerce Company

0 - You are not clear / Repeat
1 - Done , Clear

9 - 11:15

11 onwards-
Open Mic

As a data analyst at our dynamic e-commerce company, you're tasked with leveraging our extensive databases to extract insights that drive our business strategies forward. Your analysis will inform various departments, from marketing to supply chain, providing them with actionable data to optimize our operations, enhance customer satisfaction, and boost our sales performance. This case study simulates real-world tasks you will encounter and requires you to apply your SQL skills to solve practical business problems.

Business Context
Your work will directly impact the following business verticals:
- Customer Insights: Understanding our customer base to tailor marketing strategies.
- Product Analysis: Evaluating product performance to inform stock and sales strategies.
- Sales Optimization: Analyzing sales data to identify trends, opportunities, and areas for improvement.
- Inventory Management: Managing stock levels to ensure product availability while minimizing excess inventory.

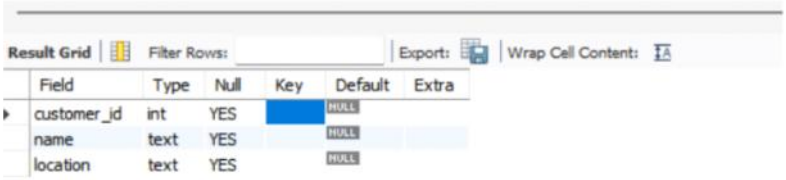
Dataset details:
- Customers Dataset: customer_id, name, and location
- Products Dataset: product_id, name, category, and price.
- Orders Dataset: order_id, order_date, customer_id, and total_amount.
- OrderDetails Dataset: order_id, product_id, quantity, and price_per_unit

Problem Statements

Customer Insights

1. Market Segmentation Analysis: Identify the top 3 cities with the highest number of customers to determine key markets for targeted marketing and logistic optimization.
2. Engagement Depth Analysis: Determine the distribution of customers by the number of orders placed. This insight will help in segmenting customers into one-time buyers, occasional shoppers, and regular customers for tailored marketing strategies.

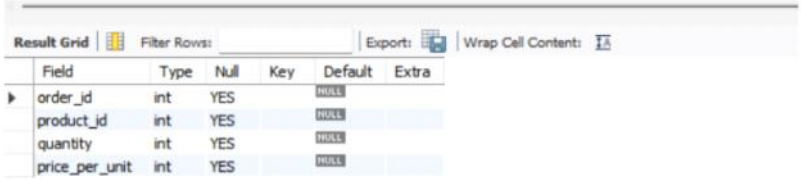
```
5 • DESC customers_india_adjusted;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
customer_id	int	YES		NULL	
name	text	YES		NULL	
location	text	YES		NULL	

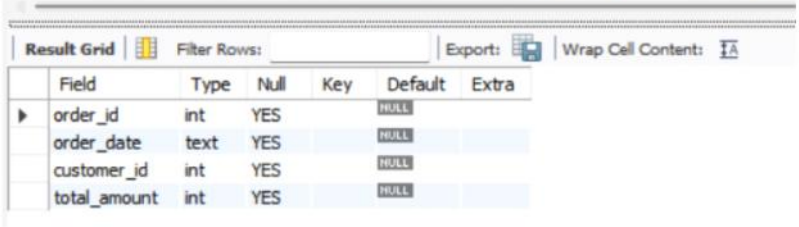
```
6 • DESC order_details_india_adjusted;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
order_id	int	YES		NULL	
product_id	int	YES		NULL	
quantity	int	YES		NULL	
price_per_unit	int	YES		NULL	

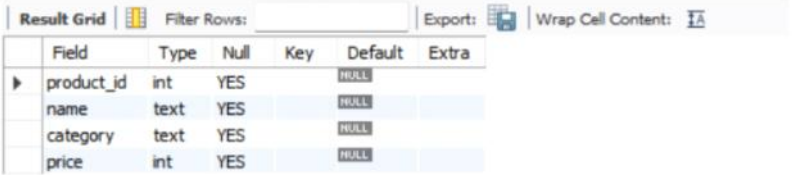
```
7 • DESC orders_india_adjusted;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
order_id	int	YES		NULL	
order_date	text	YES		NULL	
customer_id	int	YES		NULL	
total_amount	int	YES		NULL	

```
8 • DESC products_india_adjusted;
9
```

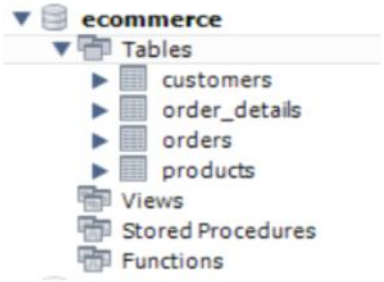


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

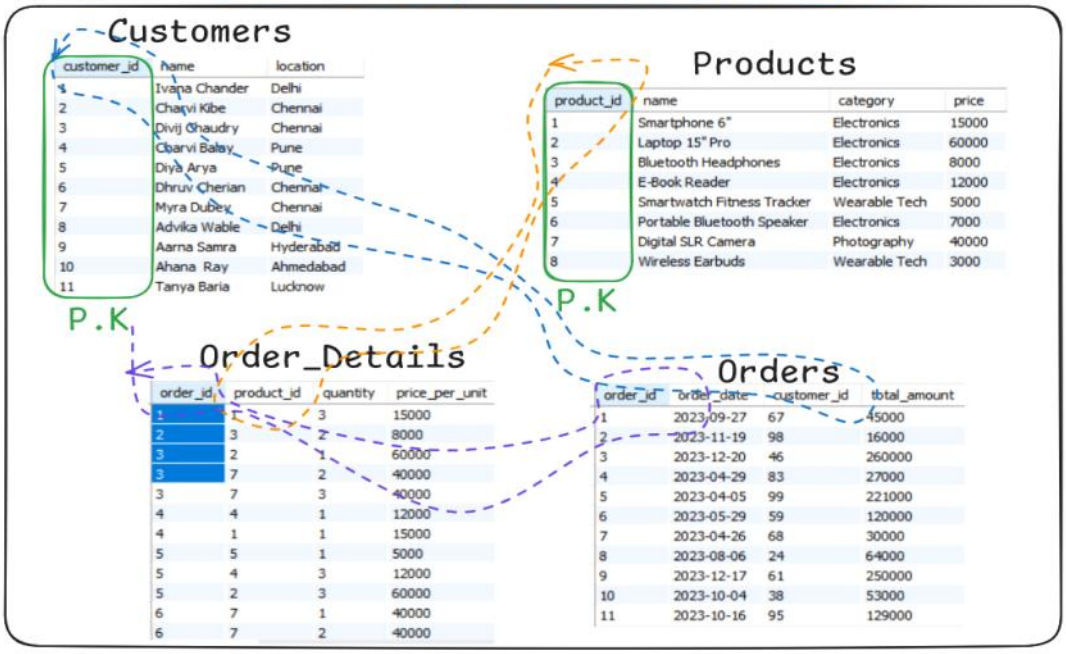
Field	Type	Null	Key	Default	Extra
product_id	int	YES		NULL	
name	text	YES		NULL	
category	text	YES		NULL	
price	int	YES		NULL	

```
DESC customers_india_adjusted;
DESC order_details_india_adjusted;
DESC orders_india_adjusted;
DESC products_india_adjusted;
```

```
-- DDL [Data Definition Language] -> Changes the Structure
ALTER TABLE customers_india_adjusted
RENAME TO Customers;

ALTER TABLE order_details_india_adjusted
RENAME TO Order_Details;

ALTER TABLE orders_india_adjusted
RENAME TO Orders;

ALTER TABLE products_india_adjusted
RENAME TO Products;
```



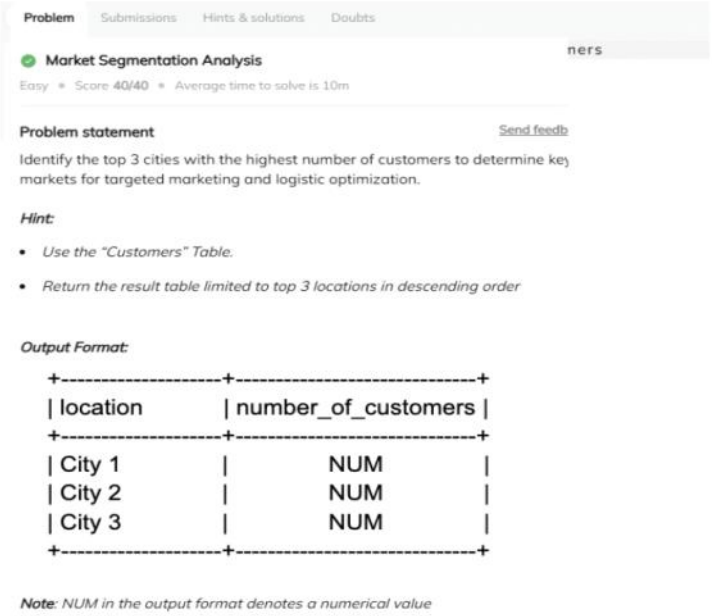
ERD : Entity Relationship Diagram



Customer Insights

1. Market Segmentation Analysis: Identify the top 3 cities with the highest number of customers to determine key markets for targeted marketing and logistic optimization.

Win + Shift + S -> [To take a ss on Windows]



Challenge 1

```sql
SELECT
    Location,
    COUNT(*) AS number_of_customers
FROM Customers
GROUP BY 1
ORDER BY 2 DESC
LIMIT 3;
```

Location	number_of_customers
Delhi	16
Chennai	15
Jaipur	11

2. Engagement Depth Analysis: Determine the distribution of customers by the number of orders placed. This insight will help in segmenting customers into one-time buyers, occasional shoppers, and regular customers for tailored marketing strategies.

Problem Submissions Hints & solutions Doubts

Challenge 2

Problem statement Send feedback

Determine the distribution of customers by the number of orders placed. This insight will help in segmenting customers into one-time buyers, occasional shoppers, and regular customers for tailored marketing strategies.

Hint:

- *Use the "Orders" table.*
- *Return the result table which helps you to segment customers on the basis of the number of orders in ascending order.*
- *Consider the following:*

NumberOfOrders	Terms
1	One-time buyer.
2-4	Occasional Shoppers.
>4	Regular customers.

Output Format:

```
+----------------------+----------------------+
| NumberOfOrders | CustomerCount |
+----------------------+----------------------+
```

Output Format:

```
+----------------------+----------------------+
| NumberOfOrders | CustomerCount |
+----------------------+----------------------+
|        NUM           |        NUM           |
|        NUM           |        NUM           |
|        NUM           |        NUM           |
|        NUM           |        NUM           |
|        NUM           |        NUM           |
+----------------------+----------------------+
```

Note: NUM in the output format denotes a numerical value

```sql
WITH CustomersOrders AS (
    SELECT
        Customer_id,
        COUNT(Order_id) AS NumberOfOrders
    FROM Orders
    GROUP BY 1
)
SELECT * FROM CustomersOrders;
```

Customer_id	NumberOfOrders
67	2
98	3
46	1
83	2
99	2
59	2
68	5
24	4
61	5
38	3
95	3
78	4
17	4
58	2
36	2
22	3
28	2

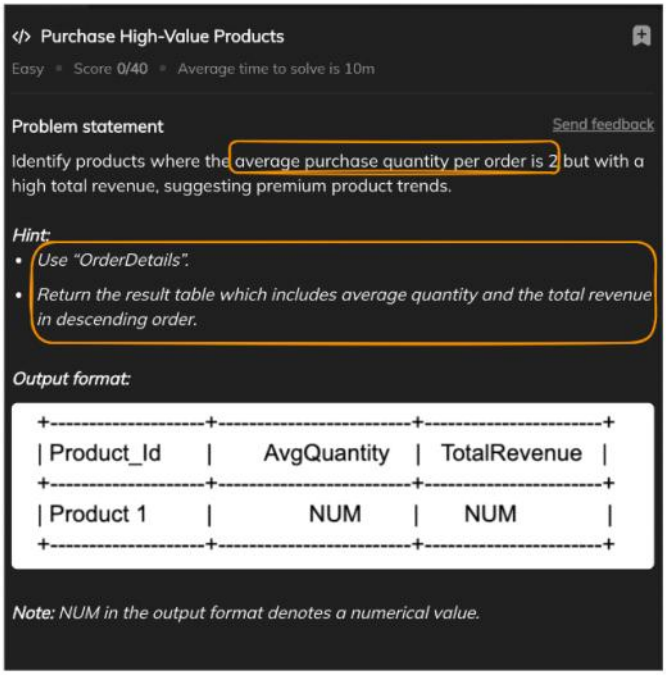
```sql
WITH CustomersOrders AS (
    SELECT
        Customer_id,
        COUNT(Order_id) AS NumberOfOrders
    FROM Orders
    GROUP BY 1
)
SELECT
    NumberOfOrders,
    COUNT(*) AS CustomerCount
FROM CustomersOrders
GROUP BY 1
ORDER BY 1;
```

NumberOfOrders	CustomerCount
1	26
2	26
3	18
4	6
5	6
6	1
8	1

Challenge 3

Product Insights

3. Single Purchase High-Value Products: Identify products where the average purchase quantity per order is one but with a high total revenue, suggesting premium product trends.



</> Purchase High-Value Products

Easy • Score 0/40 • Average time to solve is 10m

Problem statement Send feedback

Identify products where the average purchase quantity per order is 2 but with a high total revenue, suggesting premium product trends.

Hint:

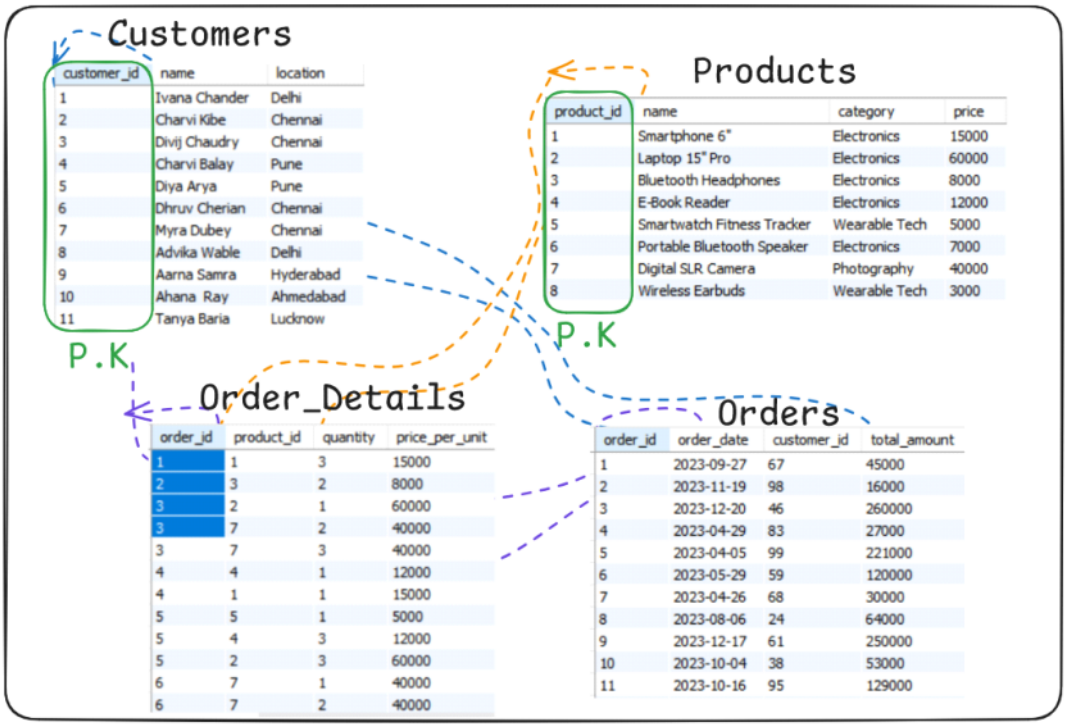
- *Use "OrderDetails".*
- *Return the result table which includes average quantity and the total revenue in descending order.*

Output format:

Product_Id	AvgQuantity	TotalRevenue
Product 1	NUM	NUM

Note: *NUM in the output format denotes a numerical value.*

ERD : Entity Relationship Diagram




```
-- Challenge 3 : Purchase High Value Products
SELECT * FROM Order_Details;

SELECT
    Product_id,
    AVG(quantity) AS AvgQuantity,
    SUM(quantity * price_per_unit) AS TotalRevenue
FROM Order_details
GROUP BY Product_id
HAVING AvgQuantity = 2
ORDER BY TotalRevenue DESC;
```

Product_id	AvgQuantity	TotalRevenue
1	2.0000	1620000
8	2.0000	390000

Challenge 4

4. Category-wise Customer Reach: For each product category, calculate the unique number of customers purchasing from it. This will help understand which categories have wider appeal across the customer base.

Problem statement Send feedback

For each product category, calculate the unique number of customers purchasing from it. This will help understand which categories have wider appeal across the customer base.

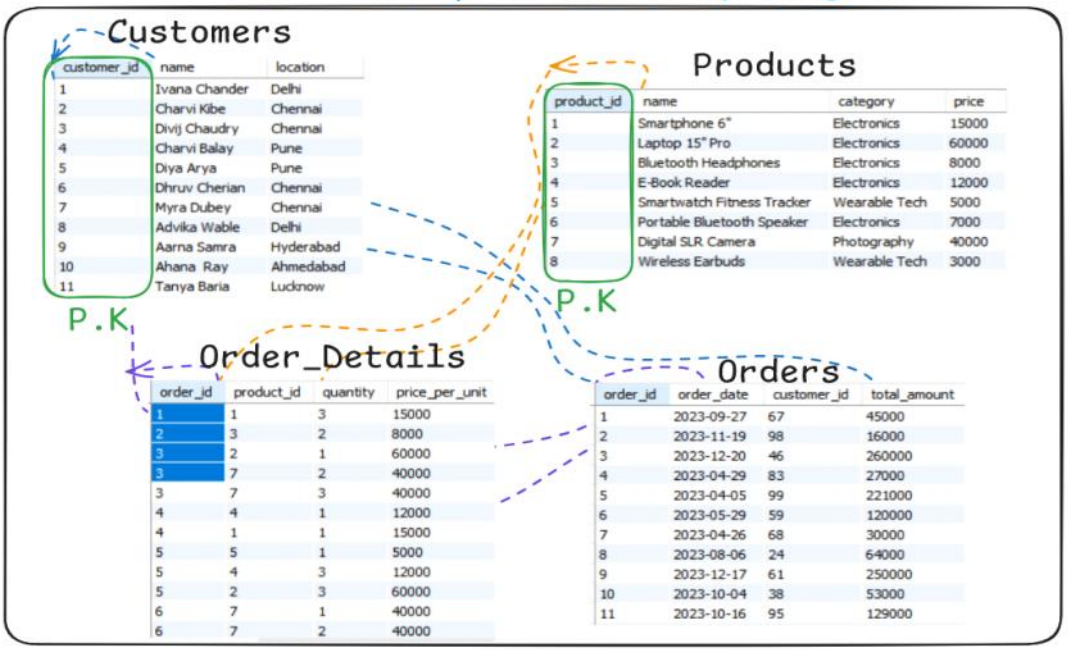
Hint:

- *Use the "Products", "OrderDetails" and "Orders" table.*
- *Return the result table which will help you count the unique number of customers in descending order.*

Output format:

```
+--------------------+------------------------+
| category           | unique_customers |
+--------------------+------------------------+
| Category 1         |          NUM           |
| Category 2         |          NUM           |
| Category 3         |          NUM           |
+--------------------+------------------------+
```

ERD : Entity Relationship Diagram




```sql
-- Challenge 4 : Category_Wise Customers Reach

SELECT
    p.Category,
    COUNT(DISTINCT o.customer_id) AS unique_customers
FROM Products p
JOIN Order_Details od
ON p.product_id = od.product_id
JOIN Orders o
ON o.order_id = od.order_id
GROUP BY 1
ORDER BY 2 DESC;
```

Category	unique_customers
Electronics	79
Wearable Tech	61
Photography	45