

# SQL CASE STUDY

## DATA IN MOTION TINY SHOP SALES



DATA IN MOTION

Debapriya Saha

## **Overview:**

This case study uses SQL. To successfully answer all the questions, I have been exposed to the following areas of SQL:

- Basic aggregations
- CASE WHEN statements
- Window Functions
- Joins
- Date time functions
- CTEs

## **Ad hoc Requests:**

1. Which product has the highest price? Only return a single row.
2. Which customer has made the most orders?
3. What's the total revenue per product?
4. Find the day with the highest revenue.
5. Find the first order (by date) for each customer.
6. Find the top 3 customers who have ordered the most distinct products
7. Which product has been bought the least in terms of quantity?
8. What is the median order total?
9. For each order, determine if it was 'Expensive' (total over 300), 'Affordable' (total over 100), or 'Cheap'.
10. Find customers who have ordered the product with the highest price.

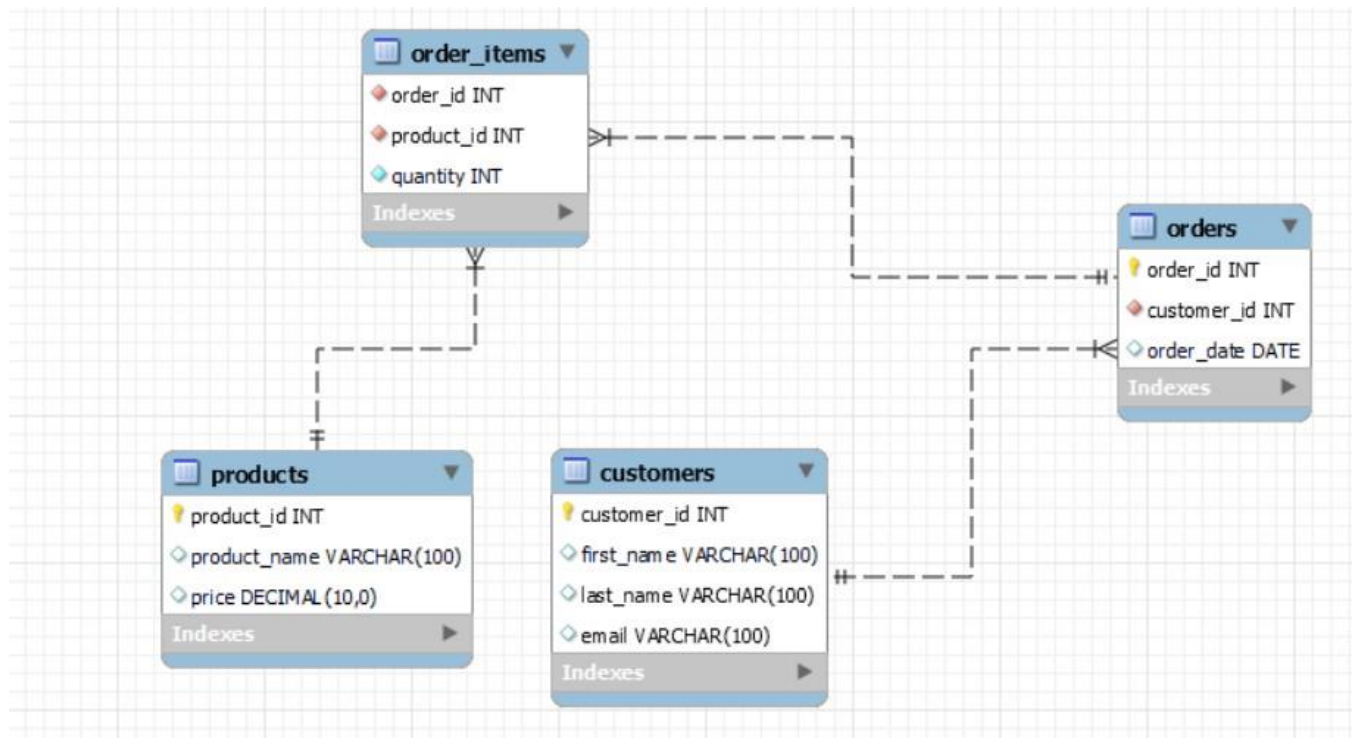
## Data Overview:

The data set contains the following 4 tables which you may refer to the relationship diagram below to understand the connection.

### TABLES:

- Customers
- Products
- Orders
- Order items

### Relationship Diagram:



## Solutions:

1. Which product has the highest price? Only return a single row.

```
WITH product_highest_price AS (  
  SELECT *,  
  DENSE_RANK() OVER (ORDER BY price DESC) as rnk  
  FROM products)  
SELECT product_name AS Product FROM product_highest_price  
WHERE rnk = 1;
```

Product
Product M

2. Which customer has made the most orders?

```
WITH Most_ordered_customer AS (  
  SELECT  
  CONCAT(c.first_name, " ", c.last_name) AS Customer_name,  
  COUNT(o.order_id) AS No_of_order,  
  RANK() OVER (ORDER BY COUNT(o.order_id) DESC) AS rnk  
  FROM orders AS o  
  LEFT JOIN customers AS c  
  ON o.customer_id = c.customer_id  
  GROUP BY o.customer_id)  
SELECT Customer_name, No_of_order  
FROM Most_ordered_customer WHERE rnk = 1;
```

Customer_name	No_of_order
John Doe	2
Jane Smith	2
Bob Johnson	2

### 3. What's the total revenue per product?

```
WITH Revenue AS (  
  SELECT  
    p.product_id AS Product_id,  
    p.product_name AS Product_name , p.price AS Price,  
    oi.quantity AS qty  
  FROM products AS p  
  LEFT JOIN order_items AS oi  
    ON oi.product_id = p.product_id)  
SELECT Product_name,  
SUM(Price * qty) AS Total_revenue  
FROM Revenue  
Group by Product_name;
```

Product_name	Total_revenue
Product A	100
Product B	270
Product C	320
Product D	150
Product E	180
Product F	420
Product G	240
Product H	270
Product I	300
Product J	660
Product K	360
Product L	390
Product M	840

### 4. Find the day with the highest revenue.

```
WITH Order_details AS (  
  SELECT  
    DAYNAME(o.order_date) AS Order_Day,  
    oi.order_id, oi.product_id, SUM(oi.quantity) AS Total_qty,  
    SUM(p.price * oi.quantity) AS Total_Revenue  
  FROM order_items AS oi  
  LEFT JOIN orders AS o  
    ON o.order_id = oi.order_id  
  LEFT JOIN products AS p  
    ON p.product_id = oi.product_id  
  GROUP BY Order_date),  
Most_revenue AS (  
  SELECT Order_Day, Total_Revenue,  
    ROW_NUMBER() OVER (ORDER BY Total_Revenue DESC) AS rn  
  FROM Order_details)  
SELECT Order_Day, Total_Revenue  
FROM Most_revenue  
WHERE rn = 1;
```

Order_Day	Total_Revenue
Tuesday	680

## 5. Find the first order (by date) for each customer.

```
WITH First_order_date AS (
SELECT
o.customer_id, o.order_date AS firsrt_order_date,
o.order_id,i.product_id,
ROW_NUMBER() OVER (PARTITION BY o.customer_id ORDER BY o.order_date) AS rn
FROM order_items AS i
LEFT JOIN orders AS o
ON i.order_id = o.order_id)
SELECT CONCAT(c.first_name," ",c.last_name) AS Customer_name,
p.product_name , firsrt_order_date
FROM First_order_date AS f
LEFT JOIN customers AS c
ON f.customer_id = c.customer_id
LEFT JOIN products AS p
ON f.product_id = p.product_id
WHERE rn = 1;
```

Customer_name	product_name	firsrt_order_date
John Doe	Product A	2023-05-01
Jane Smith	Product B	2023-05-02
Bob Johnson	Product A	2023-05-03
Alice Brown	Product D	2023-05-07
Charlie Davis	Product F	2023-05-08
Eva Fisher	Product H	2023-05-09
George Harris	Product J	2023-05-10
Ivy Jones	Product L	2023-05-11
Kevin Miller	Product D	2023-05-12
Lily Nelson	Product F	2023-05-13
Oliver Patterson	Product I	2023-05-14
Quinn Roberts	Product J	2023-05-15
Sophia Thomas	Product L	2023-05-16

## 6. Find the top 3 customers who have ordered the most distinct products.

```
WITH CTE1 AS (
SELECT o.customer_id,
oi.order_id, COUNT(DISTINCT oi.product_id) AS no_of_products
FROM order_items AS oi
LEFT JOIN orders AS o
ON o.order_id = oi.order_id
GROUP BY 1),CTE2 AS (
SELECT CTE1.customer_id,
CONCAT(C.first_name," ",C.last_name) AS Customer_Name,
no_of_products,
ROW_NUMBER() OVER (ORDER BY no_of_products DESC) AS rn
FROM CTE1
LEFT JOIN customers AS C
ON C.customer_id = CTE1.customer_id)
SELECT customer_id, Customer_Name, no_of_products
FROM CTE2 WHERE CTE2.rn < 4;
```

customer_id	Customer_Name	no_of_products
1	John Doe	3
2	Jane Smith	3
3	Bob Johnson	3

## 7. Which product has been bought the least in terms of quantity?

```
SELECT
DISTINCT p.product_name AS Product_Name,
SUM(i.quantity) OVER (PARTITION BY p.product_name) AS Qty
FROM order_items AS i
LEFT JOIN products AS p
ON i.product_id = p.product_id
ORDER BY Qty
LIMIT 1;
```

Product_Name	Qty
Product D	6

## 8. What is the median order total?

```
WITH Order_wise_amount AS (
SELECT i.order_id, SUM(i.quantity * p.price) AS Amount
FROM order_items AS i
LEFT JOIN products AS p
ON i.product_id = p.product_id
GROUP BY 1)
SELECT FLOOR(AVG(A.Amount)) AS median_of_order_total FROM (
SELECT d.Amount , @ROWNUM := @ROWNUM + 1 AS `row_number`,
@TOTAL_ROWS := @ROWNUM AS Total_row
FROM Order_wise_amount AS d , (SELECT @ROWNUM := 0) AS r
WHERE Amount IS NOT NULL
ORDER BY Amount) AS A
WHERE A.row_number IN (FLOOR((@TOTAL_ROWS+1)/2), FLOOR((@TOTAL_ROWS+2)/2));
```

median_of_order_total
225

9. For each order, determine if it was 'Expensive' (total over 300), 'Affordable' (total over 100), or 'Cheap'.

```
WITH Order_details AS (  
  SELECT  
    i.order_id AS Order_id,  
    SUM(i.quantity * p.price) AS Order_Amount  
  FROM order_items AS i  
  LEFT JOIN products AS p  
    ON i.product_id = p.product_id  
  GROUP BY 1)  
SELECT Order_id , Order_Amount,  
CASE  
  WHEN Order_Amount > 300 THEN 'Expensive'  
  WHEN Order_Amount > 100 THEN 'Affordable'  
  ELSE 'Cheap' END AS Type  
FROM Order_details;
```

Order_id	Order_Amount	Type
1	70	Cheap
2	150	Affordable
3	100	Cheap
4	160	Affordable
5	100	Cheap
6	110	Affordable
7	170	Affordable
8	290	Affordable
9	280	Affordable
10	570	Expensive
11	550	Expensive
12	160	Affordable
13	370	Expensive
14	290	Affordable
15	450	Expensive
16	680	Expensive



10. Find customers who have ordered the product with the highest price.

```
SELECT
DISTINCT i.order_id,
CONCAT(c.first_name," ",c.last_name) AS Customer_name,
i.quantity
FROM order_items AS i
LEFT JOIN products AS p
    ON i.product_id = p.product_id
LEFT JOIN orders AS o
    ON i.order_id = o.order_id
LEFT JOIN customers AS c
    ON o.customer_id = c.customer_id
WHERE i.product_id = (SELECT product_id
FROM products ORDER BY PRICE DESC LIMIT 1);
```

order_id	Customer_name	quantity
11	Ivy Jones	3
16	Sophia Thomas	3

~: Thank You: ~