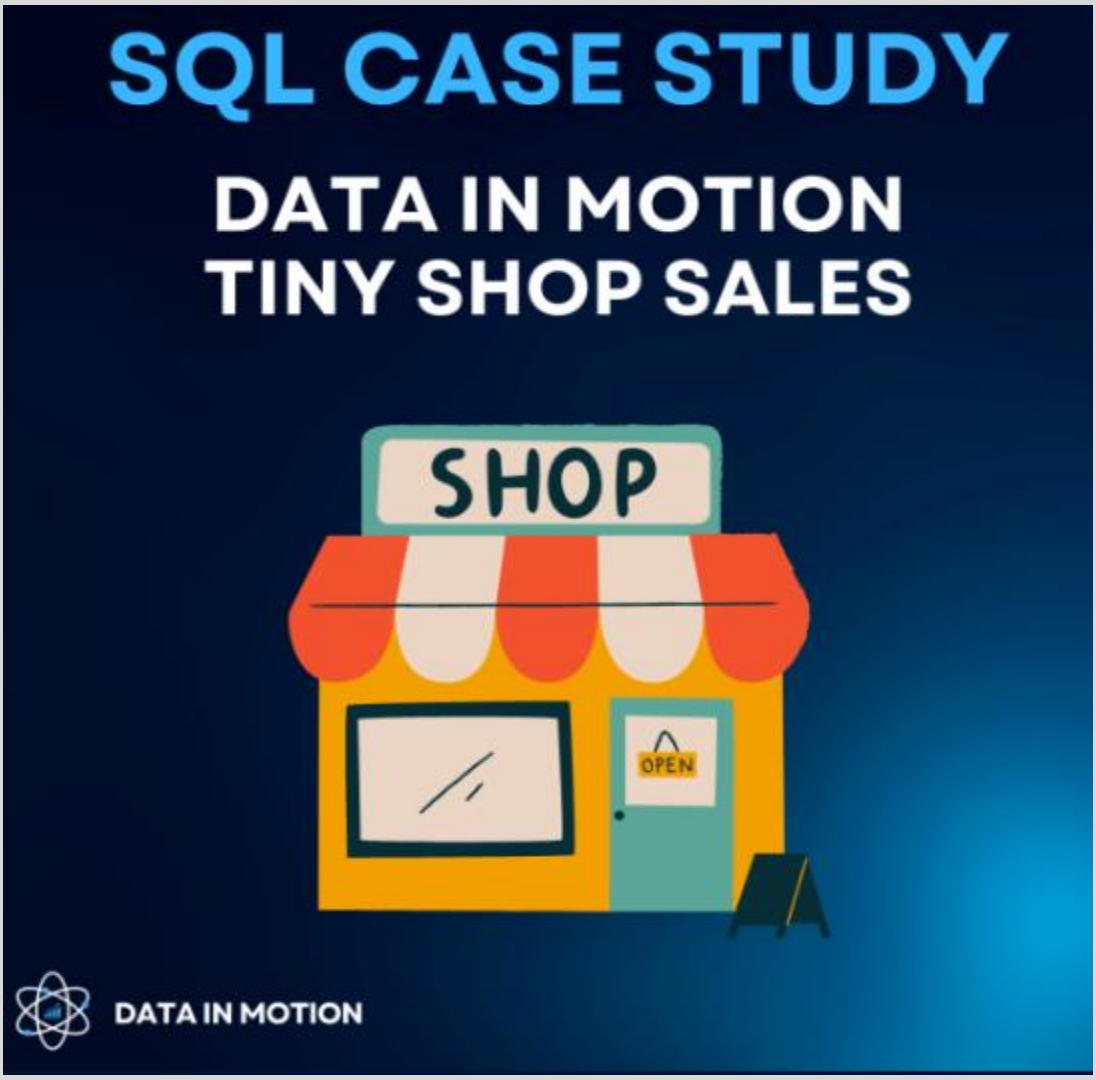

By

FARIJI E. MWAKABWENDA

QUERY RESULTS USING MICROSOFT SQL SERVER STUDIO

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

SQL Case Study Tin...wantake Fariji (53))

```
-- Case Study Questions

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

SELECT TOP 1 product_name,price FROM products
ORDER BY price DESC
```

100 %

Results | Messages

	product_name	price
1	Product M	70

2. Which customer has made the most orders?

SQL Case Study Tin...wantake Fariji (53))*

```
--2) Which customer has made the most orders?

SELECT TOP 1 CUST.customer_id, (CUST.first_name + CUST.last_name) AS CUSTOMER,OI.quantity FROM order_items AS OI
JOIN orders AS ORD
ON OI.order_id = ORD.order_id
JOIN customers AS CUST
ON CUST.customer_id = ORD.customer_id

ORDER BY OI.quantity DESC
```

100 %

Results | Messages

	customer_id	CUSTOMER	quantity
1	1	JohnDoe	4

3. What's the total revenue per product?

SQL Case Study Tin...wantake Fariji (53))*

```
--3) What's the total revenue per product?

SELECT PRO.product_name,(SUM (PRO.price * OI.quantity)) AS [TOTAL REVENUE] FROM products AS PRO
JOIN order_items AS OI
ON PRO.product_id = OI.product_id
GROUP BY product_name
```

100 %

Results | Messages

	product_name	TOTAL REVENUE
1	Product A	50
2	Product B	135
3	Product C	160
4	Product D	75
5	Product E	90
6	Product F	210
7	Product G	120
8	Product H	135
9	Product I	150
10	Product J	330
11	Product K	180
12	Product L	195
13	Product M	420

4. Find the day with the highest revenue.

```
--4) Find the day with the highest revenue.

SELECT TOP 1 ORD.order_date, (SUM (PRO.price * OI.quantity)) AS [TOTAL REVENUE] FROM products AS PRO
JOIN order_items AS OI
ON PRO.product_id = OI.product_id
JOIN ORDERS AS ORD
ON ORD.order_id = OI.order_id
GROUP BY  ORD.order_date
ORDER BY [TOTAL REVENUE] DESC
```

100 %

Results | Messages

	order_date	TOTAL REVENUE
1	2023-05-16	340

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

```sql
--5) Find the first order Find the first order (by date) for each customer for each customer.

WITH CTE_DATE AS
(SELECT (CUST.first_name + ' ' + CUST.last_name) AS CUSTOMER, ORD.order_date AS DATE FROM customers AS CUST
JOIN ORDERS AS ORD
ON CUST.customer_id = ORD.customer_id
JOIN order_items AS OI
ON OI.order_id = ORD.order_id
)
SELECT  CUSTOMER,MIN(DATE) AS [FIRST ORDER (by date)] FROM CTE_DATE
GROUP BY CUSTOMER
```

100 %

Results | Messages

	CUSTOMER	FIRST ORDER (by date)
1	Alice Brown	2023-05-07
2	Bob Johnson	2023-05-03
3	Charlie Davis	2023-05-08
4	Eva Fisher	2023-05-09
5	George Harris	2023-05-10
6	Ivy Jones	2023-05-11
7	Jane Smith	2023-05-02
8	John Doe	2023-05-01
9	Kevin Miller	2023-05-12
10	Lily Nelson	2023-05-13
11	Oliver Patterson	2023-05-14
12	Quinn Roberts	2023-05-15
13	Sophia Thomas	2023-05-16

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

```sql
--6) Find the top 3 customers who have ordered the most distinct products

WITH CTE_PRODUCT AS
(SELECT (CUST.first_name + ' ' + last_name) AS CUSTOMER, product_name,quantity FROM customers AS CUST
JOIN orders AS OD
ON CUST.customer_id = OD.customer_id
JOIN order_items AS OI
ON OI.order_id = OD.order_id
JOIN products AS PRO
ON PRO.product_id = OI.product_id
)
SELECT TOP 3   CUSTOMER,SUM(quantity) AS [TOTAL QUANTITY]  FROM CTE_PRODUCT
GROUP BY CUSTOMER,product_name
ORDER BY [TOTAL QUANTITY]  DESC
```

100 %

Results | Messages

	CUSTOMER	TOTAL QUANTITY
1	John Doe	5
2	Jane Smith	5
3	Bob Johnson	3

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

```sql
--7) Which product has been bought the least in terms of quantity?

SELECT product_name,SUM(quantity) AS QUANTITY FROM order_items AS OI
JOIN products AS PRO
ON OI.product_id = PRO.product_id
GROUP BY product_name
ORDER BY QUANTITY
```

100 %

Results | Messages

	product_name	QUANTITY
1	Product D	3
2	Product E	3
3	Product G	3
4	Product H	3
5	Product I	3
6	Product K	3
7	Product L	3

8. What is the median order total?

```sql
--8) What is the median order total?

SELECT DISTINCT PERCENTILE_CONT(.5) WITHIN GROUP (ORDER BY SUM(OI.quantity * PRO.price)) OVER (PARTITION BY 1) AS MEDIAN
FROM order_items AS OI
JOIN products AS PRO
ON OI.product_id = PRO.product_id
GROUP BY ORDER_ID
```

100 %

Results | Messages

	MEDIAN
1	112.5

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

```sql
--9) For each order, determine if it was ‘Expensive’ (total over 300), ‘Affordable’ (total over 100), or ‘Cheap’.

SELECT ORDER_ID, SUM(OI.quantity * PRO.price) AS [ORDER TOTAL],
CASE WHEN
 SUM(OI.quantity * PRO.price)  > '300' THEN 'EXPENSIVE'
 WHEN
 SUM (OI.quantity * PRO.price)  > '100' THEN 'AFFORDABLE'
ELSE 'CHEAP' END AS [ORDER CATEGORY]
FROM order_items AS OI
JOIN products AS PRO
ON OI.product_id = PRO.product_id
GROUP BY ORDER_ID
ORDER BY [ORDER TOTAL]
```

100 %

Results | Messages

	ORDER_ID	ORDER TOTAL	ORDER CATEGORY
1	1	35	CHEAP
2	3	50	CHEAP
3	5	50	CHEAP
4	6	55	CHEAP
5	2	75	CHEAP
6	4	80	CHEAP
7	12	80	CHEAP
8	7	85	CHEAP
9	9	140	AFFORDABLE
10	14	145	AFFORDABLE
11	8	145	AFFORDABLE
12	13	185	AFFORDABLE
13	15	225	AFFORDABLE
14	11	275	AFFORDABLE
15	10	285	AFFORDABLE
16	16	340	EXPENSIVE

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

```sql
--10) Find customers who have ordered the product with the highest price.

WITH CTE_PRODUCT AS
(SELECT (CUST.first_name + ' ' + last_name) AS CUSTOMER, product_name,PRICE FROM customers AS CUST
JOIN orders AS OD
ON CUST.customer_id = OD.customer_id
JOIN order_items AS OI
ON OI.order_id = OD.order_id
JOIN products AS PRO
ON PRO.product_id = OI.product_id
)
SELECT  TOP 10  CUSTOMER, product_name,PRICE FROM CTE_PRODUCT
ORDER BY PRICE DESC
```

100 %

Results | Messages

	CUSTOMER	product_name	PRICE
1	Ivy Jones	Product M	70
2	Sophia Thomas	Product M	70
3	Ivy Jones	Product L	65
4	Sophia Thomas	Product L	65
5	George Harris	Product K	60
6	Quinn Roberts	Product K	60
7	George Harris	Product J	55
8	Quinn Roberts	Product J	55
9	Eva Fisher	Product I	50
10	Oliver Patterson	Product I	50