SQL Interview Question

PWC Data Engineer Interview

"Write a query to find each customer's **latest order amount** along with the **second latest order** amount."

📥 Input Data

order_id	customer_id	order_date	order_amount
1	101	2024-01-10	150.00
2	101	2024-02-15	200.00
3	101	2024-03-20	180.00
4	102	2024-01-12	200.00
5	102	2024-02-25	250.00
6	102	2024-03-10	320.00
7	103	2024-01-25	400.00
8	103	2024-02-15	420.00

📤 Expected Output

customer_id	latest_order	second_latest
101	180.00	200.00
102	320.00	250.00
103	420.00	400.00

The Core Pattern

Nth Value with ROW_NUMBER()

1

Rank Orders by Date

Assign row numbers for each customer's orders, newest first

```
ROW_NUMBER() OVER(PARTITION BY customer_id ORDER BY order_date DESC)
```

2

Filter Latest Orders

Select rows where rank is 1 (latest) or 2 (second latest)

```
WHERE rn IN (1, 2)
```

3

Pivot with CASE

Use CASE to create separate columns for latest and second latest

```
CASE WHEN rn = 1 THEN order_amount END AS latest_order
```

4

Aggregate per Customer

Group by customer_id and get MAX of each column

```
MAX(latest_order),
MAX(second_latest)
```

💡 **The Row Number Pivot Trick**

Instead of complex self-joins or subqueries, use ROW_NUMBER() to rank orders, filter for ranks 1 and 2, then pivot using CASE statements.

Optimized SQL Solution

MySQL 8.0+

```sql
-- Step 1: Rank orders for each customer by date (newest first)
WITH ranked_orders AS (
    SELECT
        customer_id,
        order_amount,
        order_date,
        ROW_NUMBER() OVER(
            PARTITION BY customer_id
            ORDER BY order_date DESC
        ) AS rn
    FROM orders
),
-- Step 2: Pivot latest and second latest orders using CASE
pivoted_data AS (
    SELECT
        customer_id,
        CASE WHEN rn = 1 THEN order_amount END
            AS latest_order,
        CASE WHEN rn = 2 THEN order_amount END
            AS second_latest
    FROM ranked_orders
    WHERE rn IN (1, 2)
)
-- Step 3: Aggregate to get one row per customer
SELECT
    customer_id,
    MAX(latest_order) AS latest_order,
    MAX(second_latest) AS second_latest
FROM pivoted_data
GROUP BY customer_id
ORDER BY customer_id;
```

One-Pass Solution with NTH_VALUE()

MySQL 8.0+

```sql
-- NTH_VALUE picks specific ranked value

SELECT DISTINCT
    customer_id,
    NTH_VALUE(order_amount, 1) OVER(
        PARTITION BY customer_id
        ORDER BY order_date DESC
        ROWS BETWEEN UNBOUNDED PRECEDING
        AND UNBOUNDED FOLLOWING
    ) AS latest_order,
    NTH_VALUE(order_amount, 2) OVER(
        PARTITION BY customer_id
        ORDER BY order_date DESC
        ROWS BETWEEN UNBOUNDED PRECEDING
        AND UNBOUNDED FOLLOWING
    ) AS second_latest
FROM orders
ORDER BY customer_id;
```

Master SQL Patterns

Ace Your Data Engineer Interviews

The Nth Value Pattern

One Pattern, Infinite Business Applications

- ✅ E-commerce: Latest vs previous purchases
- ✅ Finance: Recent vs earlier transactions
- ✅ Healthcare: Current vs past readings
- ✅ IoT: Latest vs historical sensor data

🔄 **Repost this so your boss knows you are leveling up in SQL.**

💡 Pro Tip: Master both ROW_NUMBER() and NTH_VALUE() approaches!