**QUESTION**

# Most Recent Transaction

Medium

10 Points

Given a table of customer sales in a retail store with columns id, transaction_value, and created_at representing the date and time for each transaction, write a query to get the last transaction for each day.

The output should include the ID of the transaction, datetime of the transaction, and the transaction amount. Order the transactions by datetime.

**Output Schema:**

| Column | Type |
|---|---|
| id | INT |
| created_at | DATETIME |
| transaction_value | FLOAT |

## TABLE SCHEMA

```sql
CREATE TABLE customer_sales (
id INT PRIMARY KEY,
transaction_value DECIMAL(10, 2),
created_at DATETIME
);

INSERT INTO customer_sales (id, transaction_value, created_at)
VALUES
(1, 50.00, '2025-01-23 10:15:00'),
(2, 30.00, '2025-01-23 15:45:00'),
(3, 20.00, '2025-01-23 18:30:00'),
(4, 45.00, '2025-01-24 09:20:00'),
(5, 60.00, '2025-01-24 22:10:00'),
(6, 25.00, '2025-01-25 11:30:00'),
(7, 35.00, '2025-01-25 14:50:00'),
(8, 55.00, '2025-01-25 19:05:00');
```

## SOLUTION


## OUTPUT

| id | created_at | transaction_value |
|---|---|---|
| 3 | 2025-01-23 18:30:00 | 20 |
| 5 | 2025-01-24 22:10:00 | 60 |
| 8 | 2025-01-25 19:05:00 | 55 |

## My Thought Process:

To solve this, I used the ROW_NUMBER() function to rank each transaction per day based on the time they occurred (latest first).

Then, I picked the top-ranked (i.e., latest) transaction for each day.

I used DATE(created_at) to group the data by day and ignored the time portion.

## Business Impact:

This same logic can be used beyond sales for example, to track the last time each employee clocked out every day or the first time they clocked in.

Perfect for attendance systems, shift tracking, or even analyzing work-from-home patterns!