

# LIMIT 1 OFFSET 1 fails the interview.

D8A.ACADEMY/RESOURCES/SECOND-HIGHEST-SALARY

## THE QUESTION

|                |                             |
|----------------|-----------------------------|
| second highest | salary in employees         |
| really tests   | ties, NULLs, one-row tables |
| ask first      | ← distinct value or person? |

## THE NAIVE ANSWER

|                   |                        |
|-------------------|------------------------|
| ORDER BY ... DESC | LIMIT 1 OFFSET 1       |
| works when        | every salary is unique |
| and when          | there are 2+ rows      |

## WHY IT BREAKS

|                |                           |
|----------------|---------------------------|
| tie at the top | returns the top again     |
| one row only   | no row, not NULL          |
| OFFSET         | ← skips rows, not values. |

## ROBUST ANSWERS

|                   |                       |
|-------------------|-----------------------|
| DENSE_RANK() = 2  | ties share a rank     |
| MAX below MAX     | NULL if no second one |
| DISTINCT + OFFSET | wrap in a subquery    |

## NTH, PER DEPARTMENT

|                   |                        |
|-------------------|------------------------|
| rank = n          | same query, any n      |
| PARTITION BY dept | second in each team    |
| MAX below MAX     | stops scaling at n = 3 |

## GOTCHAS

|              |                           |
|--------------|---------------------------|
| RANK()       | no rank 2 after a tie     |
| ROW_NUMBER() | splits ties arbitrarily   |
| NULL salary  | first in DESC on Postgres |

## WORTH MEMORIZING

### 01 The answer that survives a tie

Rank the salary values, not the rows. Ties share a rank, and an empty result becomes NULL.

```
-- second highest distinct salary, NULL if none
SELECT MAX(salary) AS second_highest
FROM (
  SELECT salary, DENSE_RANK() OVER (
    ORDER BY salary DESC NULLS LAST) AS rk
  FROM employees
) ranked
WHERE rk = 2;

-- nth highest: change the 2
-- per department: PARTITION BY department_id
```

→ MAX() around the result does two jobs: several people on rank 2 collapse to one value, and a table with a single salary returns NULL instead of zero rows.

### 02 Trap: same question, three tables

The naive query is right on clean data. Interview data is never clean.

| SALARIES IN THE TABLE | LIMIT 1 OFFSET 1 | DENSE_RANK = 2 |
|-----------------------|------------------|----------------|
| 100, 100, 90          | returns 100      | returns 90     |
| 100 (one row)         | returns no row   | returns NULL   |
| NULL, 100, 90         | returns 100      | returns 90     |

→ Interview phrasing: do you want the second distinct salary, and what should come back when there is none? Then I rank with DENSE\_RANK.