SQL WINDOW FUNCTIONS

A QUICK GUIDE TO RANKING AND PARTITIONING AND BEYOND !

shekhar888

ROW_NUMBER()

DEFINITION: Assigns a unique sequential integer to rows within a partition of a result set, starting at 1 for the first row in each partition.

```sql
SELECT
    employee_id,
    department_id,
    salary,
    ROW_NUMBER() OVER (PARTITION BY department_id ORDER BY salary DESC) AS row_num
FROM
    employees;
```

OUTPUT:

EMPLOYEE_ID	DEPARTMENT_ID	SALARY	ROW_NUM
103	10	9000	1
102	10	8500	2
101	20	10000	1
104	20	9500	2

EXPLANATION:

The ROW_NUMBER() function assigns a unique row number to each employee within the same department (department_id). The rows are ordered by salary in descending order, so the highest-paid employee in each department gets a row_num of 1. Since there are two departments, the numbering restarts at 1 for each department.

RANK()

DEFINITION: Assigns a rank to each row within a partition of a result set, with gaps in rank values if there are ties.

```sql
SELECT
    employee_id,
    department_id,
    salary,
    RANK() OVER (PARTITION BY department_id ORDER BY salary DESC) AS rank
FROM
    employees;
```

OUTPUT:

EMPLOYEE_ID	DEPARTMENT_ID	SALARY	RANK
103	10	9000	1
102	10	8500	2
101	20	10000	1
104	20	9500	2

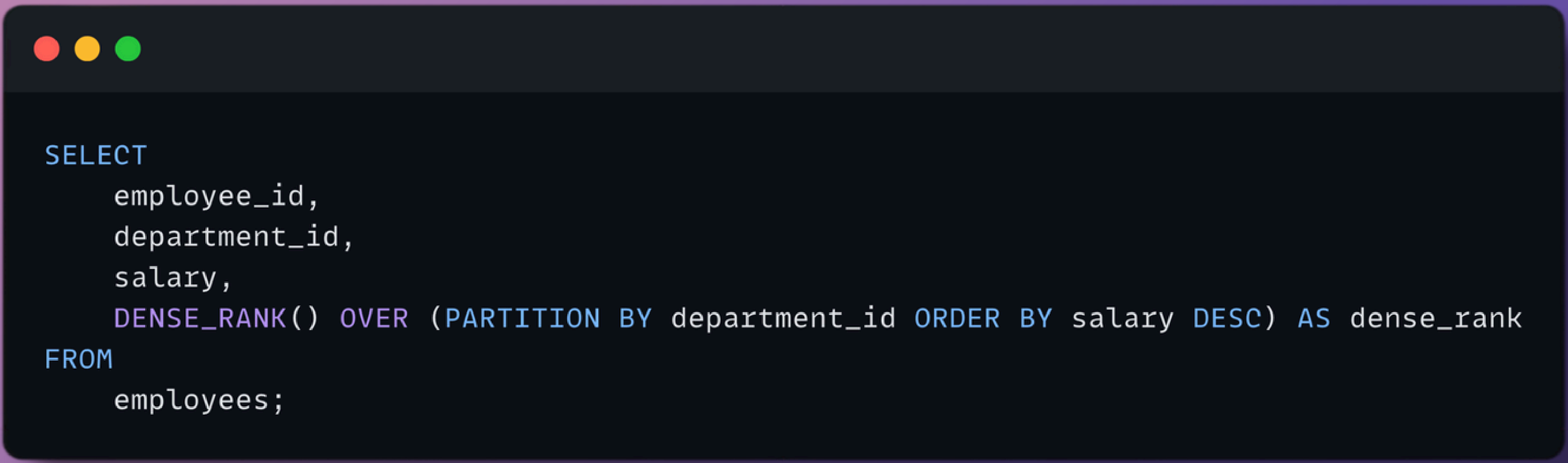
EXPLANATION:

The RANK() function assigns a rank to employees based on their salary within each department. Since there are no ties in the salary, the ranks are sequential. If there had been a tie, the same rank would have been assigned to the tied rows, and the next rank would have been skipped, creating a gap.

shekhar888

DENSE_RANK()

DEFINITION: Similar to RANK() but without gaps in the ranking sequence when there are ties.



```
SELECT
    employee_id,
    department_id,
    salary,
    DENSE_RANK() OVER (PARTITION BY department_id ORDER BY salary DESC) AS dense_rank
FROM
    employees;
```

OUTPUT:

EMPLOYEE_ID	DEPARTMENT_ID	SALARY	DENSE_RANK
103	10	9000	1
102	10	8500	2
101	20	10000	1
104	20	9500	2

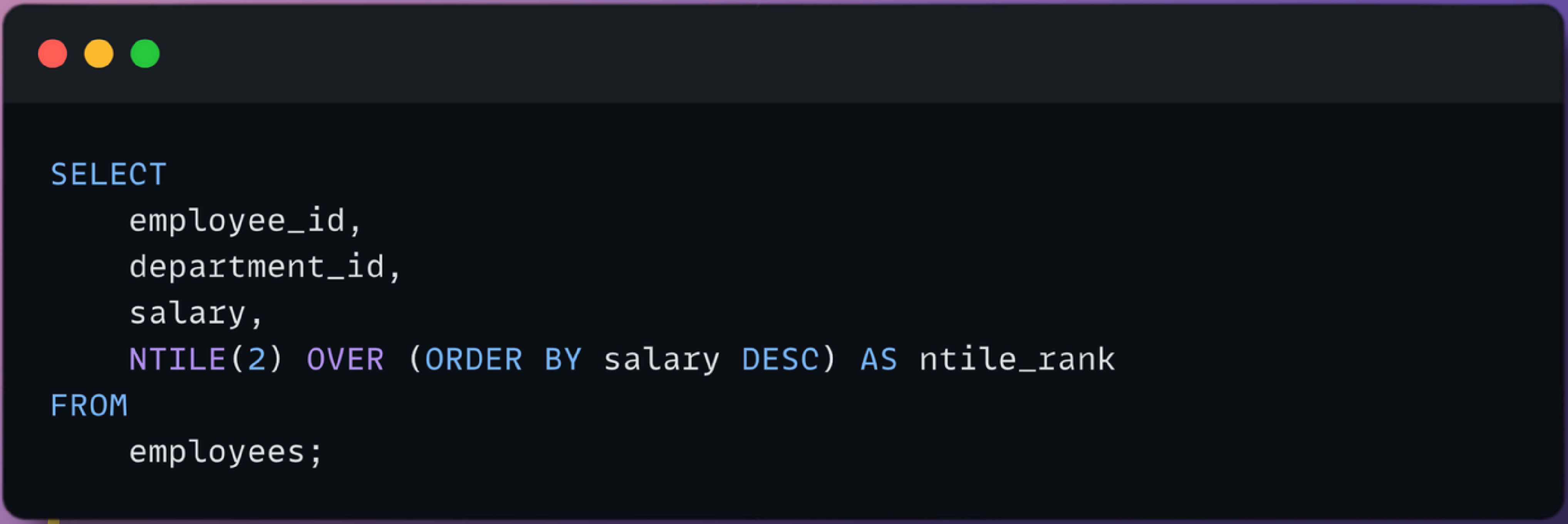
EXPLANATION:

The DENSE_RANK() function works similarly to RANK() but ensures no gaps in the ranking sequence. Since there are no ties, the result is the same as with RANK(). If there were ties, DENSE_RANK() would assign consecutive ranks without skipping any numbers.

in shekhar888

NTILE()

DEFINITION: Distributes rows of a result set into a specified number of roughly equal groups.



```sql
SELECT
    employee_id,
    department_id,
    salary,
    NTILE(2) OVER (ORDER BY salary DESC) AS ntile_rank
FROM
    employees;
```

OUTPUT:

EMPLOYEE_ID	DEPARTMENT_ID	SALARY	NTILE_RANK
101	20	10000	1
104	20	9500	1
103	10	9000	2
102	10	8500	2

EXPLANATION:

The NTILE(2) function divides the rows into two groups, ordered by salary in descending order. The highest-paid employees go into the first group (ntile_rank 1), and the next set of employees goes into the second group (ntile_rank 2). Since there are four rows, each group contains two employees.

LAG()

DEFINITION: Provides access to a value in a prior row within the same result set.

```sql
SELECT
    employee_id,
    salary,
    LAG(salary, 1, 0) OVER (ORDER BY salary DESC) AS previous_salary
FROM
    employees;
```

OUTPUT:

EMPLOYEE_ID	SALARY	PREVIOUS_SALARY
101	10000	0
104	9500	10000
103	9000	9500
102	8500	9000

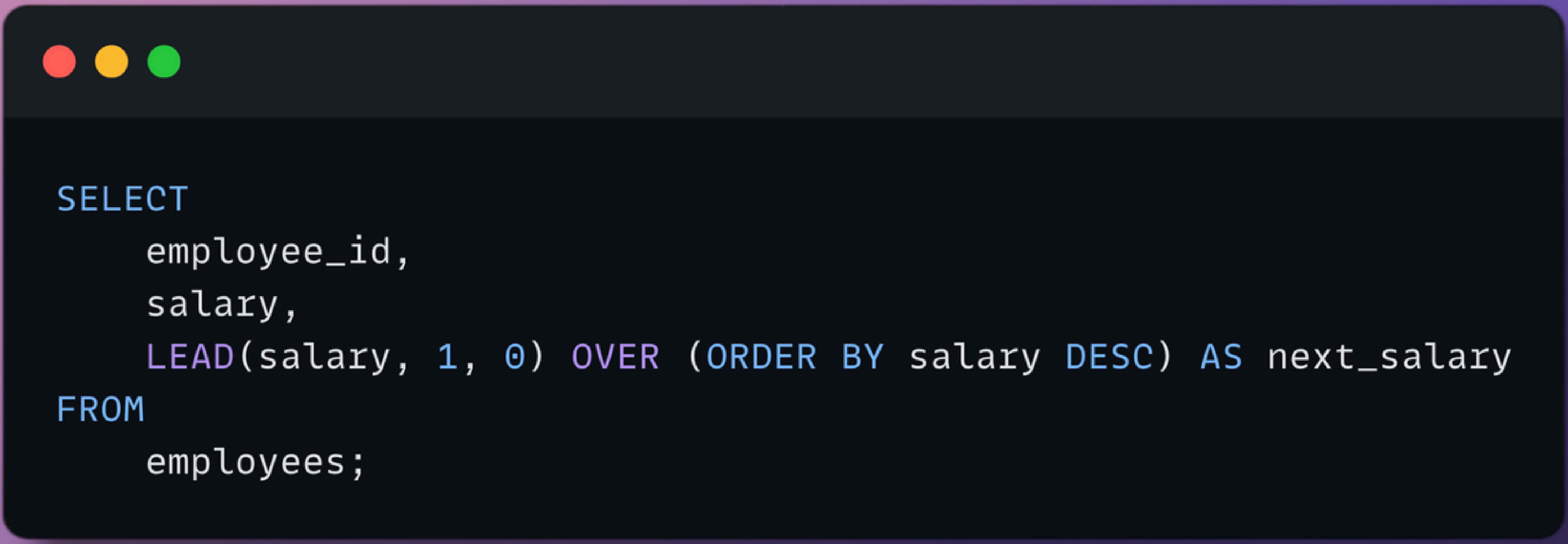
EXPLANATION:

The 'LAG()' function retrieves the salary of the previous row based on the descending order of salaries. For the first row, there is no previous row, so the default value '0' is used. For each subsequent row, it fetches the salary from the row above.

shekhar888

LEAD()

DEFINITION: Provides access to a value in a subsequent row within the same result set.



```
SELECT
    employee_id,
    salary,
    LEAD(salary, 1, 0) OVER (ORDER BY salary DESC) AS next_salary
FROM
    employees;
```

OUTPUT:

EMPLOYEE_ID	SALARY	NEXT_SALARY
101	10000	9500
104	9500	9000
103	9000	8500
102	8500	0

EXPLANATION:

The LEAD() function retrieves the salary of the next row based on the descending order of salaries. For the last row, there is no subsequent row, so the default value 0 is used. For each row, it fetches the salary from the row below.

shekhar888