

- `RANK() over_clause`

Returns the rank of the current row within its partition, with gaps. Peers are considered ties and receive the same rank. This function does not assign consecutive ranks to peer groups if groups of size greater than one exist; the result is noncontiguous rank numbers.

This function should be used with `ORDER BY` to sort partition rows into the desired order. Without `ORDER BY`, all rows are peers.

`over_clause` is as described in [Section 12.21.2, “Window Function Concepts and Syntax”](#).

The following query shows the difference between `RANK()`, which produces ranks with gaps, and `DENSE_RANK()`, which produces ranks without gaps. The query shows rank values for each member of a set of values in the `val` column, which contains some duplicates. `RANK()` assigns peers (the duplicates) the same rank value, and the next greater value has a rank higher by the number of peers minus one. `DENSE_RANK()` also assigns peers the same rank value, but the next higher value has a rank one greater. For reference, the query also displays row numbers using `ROW_NUMBER()`:

```
mysql> SELECT
    val,
    ROW_NUMBER() OVER w AS 'row_number',
    RANK() OVER w AS 'rank',
    DENSE_RANK() OVER w AS 'dense_rank'
FROM numbers
WINDOW w AS (ORDER BY val);
```

val	row_number	rank	dense_rank
1	1	1	1
1	2	1	1
2	3	3	2
3	4	4	3
3	5	4	3
3	6	4	3
4	7	7	4
4	8	7	4
5	9	9	5

- `ROW_NUMBER() over_clause`

Returns the number of the current row within its partition. Rows numbers range from 1 to the number of partition rows.

`ORDER BY` affects the order in which rows are numbered. Without `ORDER BY`, row numbering is nondeterministic.

`ROW_NUMBER()` assigns peers different row numbers. To assign peers the same value, use `RANK()` or `DENSE_RANK()`. For an example, see the `RANK()` function description.

`over_clause` is as described in [Section 12.21.2, “Window Function Concepts and Syntax”](#).

## 12.21.2 Window Function Concepts and Syntax

This section describes how to use window functions. Examples use the same sales information data set as found in the discussion of the `GROUPING()` function in [Section 12.20.2, “GROUP BY Modifiers”](#):

```
mysql> SELECT * FROM sales ORDER BY country, year, product;
```

year	country	product	profit
------	---------	---------	--------