

Ranking Window Functions: Takeaways

by Dataquest Labs, Inc. - All rights reserved © 2024

Syntax

- **ROW_NUMBER()**: Assigns a unique sequential integer to each row within the result set, starting from 1 for the first row.

```
ROW_NUMBER() OVER (  
  [PARTITION BY col(s) ]  
  ORDER BY col(s) [ASC | DESC], ...  
)
```

- **RANK()**: Assigns a unique rank to each row within the result set, with the same rank assigned to rows with equal values, leaving gaps between ranks for non-unique values.

```
RANK() OVER (  
  [PARTITION BY col(s) ]  
  ORDER BY col(s) [ASC | DESC], ...  
)
```

- **DENSE_RANK()**: Assigns a unique rank to each row within the result set, with the same rank assigned to rows with equal values but without gaps between ranks for non-unique values.

```
DENSE_RANK() OVER (  
  [PARTITION BY col(s) ]  
  ORDER BY col(s) [ASC | DESC], ...  
)
```

- **NTILE(n)**: Divides the result set into a specified number (n) of approximately equal-sized groups, assigning each row to a group with a unique integer identifier ranging from 1 to n.

```
NTILE(n) OVER (  
  [PARTITION BY col(s) ]  
  ORDER BY col(s) [ASC | DESC], ...  
)
```

Concepts

- SQL Ranking window functions assign a ranking, or row number, to each row in a result set based on a specified ordering.
- The four ranking window functions we discussed in this lesson are `ROW_NUMBER()` , `RANK()` , `DENSE_RANK()` , and `NTILE(n)` .
- These functions are helpful in analytical queries and are often used with other functions to generate complex ranking and grouping queries.
- The `OVER()` clause defines the window or partition over which the ranking is calculated, and the `PARTITION BY` and `ORDER BY` subclauses are used to group and sort the data within the window.

References

- [Ranking Functions](#)
- [Window Functions in PostgreSQL](#)

Takeaways by Dataquest Labs, Inc. - All rights reserved © 2024