Reference card

SQL window functions

Window functions

Window functions allow us to **perform certain operations** on a **subset of related rows**, called a **window**, and **return** a **value for each row** in that set.

AGGREGATE

Aggregate window functions are a group of aggregate functions, such as SUM(), COUNT(), AVG(), MAX(), and MIN(), that **calculate aggregate values within a window** and return a result to each row.

```
SELECT
    Column_X,
    Column_Y,
    Column_Z,
    AGG_FUNCTION(Column_Z) OVER (
    PARTITION BY Column_X
    ORDER BY Column_Y) AS Alias
FROM
    Table_name;
```

Divides result set into windows based on Column_X.

Sorts the rows within each partition by the values in Column_Y.

RANKING

Ranking window functions **assign a rank or row number** to each row within a specified window or subset of rows. They work together with the ORDER BY clause.

ROW_NUMBER()

Assigns a unique sequential number to each row within a window partition based on the ordering of a column by the ORDER BY clause. **No two rows are given the same number**.

```
SELECT
    Column_X,
    Column_Y,
    ROW_NUMBER() OVER (
        PARTITION BY Column_X
        ORDER BY Column_Y) AS Alias
FROM
    Table_name;
```

RANK()

Assigns a rank to each row within a window partition based on the ordering of a column by the ORDER BY clause. **Rows with the same values receive the same rank, and the next rank is skipped accordingly**.

```
SELECT
    Column_X,
    Column_Y,
    RANK() OVER (
        PARTITION BY Column_X
        ORDER BY Column_Y) AS Alias
FROM
    Table_name;
```

DENSE_RANK()

Assigns a rank to each row within a window partition based on the ordering of a column by the ORDER BY clause. **Rows with the same values receive the same rank but no ranks are skipped**.

```
SELECT
    Column_X,
    Column_Y,
    DENSE_RANK() OVER (
        PARTITION BY Column_X
        ORDER BY Column_Y) AS Alias
FROM
    Table_name;
```

VALUE-BASED

Value-based window functions are used to **extract values from other rows within a window** and returns the result for each row within the window.

LAG(Column, n)

Allows the access of a value within a column from the **previous nth-row relative to the current row**. The lag value for the first row within a partition will be NULL since there is no previous value.

```
SELECT
    Column_X,
    Column_Y,
    Column_Z,
    LAG(Column_Z, n) OVER (
    PARTITION BY Column_X
    ORDER BY Column_Y) AS Alias
FROM
    Table_name;
```

LEAD(Column, n)

Allows the access of a value within a column from the **following nth-row relative to the current row**. The lead value for the last row within a partition will be NULL since there is nonext value.

```
SELECT
    Column_X,
    Column_Y,
    Column_Z,
    LEAD(Column_Z, n) OVER (
    PARTITION BY Column_X
    ORDER BY Column_Y) AS Alias
FROM
    Table_name;
```

FIRST_VALUE()

Allows the retrieval of the **value of a column from the first row** within a partition.

```
SELECT
    Column_X,
    Column_Y,
    Column_Z,
    FIRST_VALUE(Column_Z) OVER (
    PARTITION BY Column_X
    ORDER BY Column_Y) AS Alias
FROM
    Table_name;
```

LAST_VALUE()

Allows the retrieval of the **value of a column from the last row** within a window frame.

```
SELECT
    Column_X,
    Column_Y,
    Column_Z,
    LAST_VALUE(Column_Z) OVER (
    PARTITION BY Column_X
    ORDER BY Column_Y) AS Alias
FROM
    Table_name;
```