
SQL Window Functions
365 DataScience

Introduction to MySQL Window Functions

365 DataScience

Introduction to MySQL Window Functions

MySQL Window Functions

an advanced SQL tool performing a calculation for every record in the data set, using other records associated with the specified one from the table

Introduction to MySQL Window Functions

MySQL Window Functions

an advanced SQL tool performing a calculation for every record in the data set, using other records associated with the specified one from the table

- an entire table from a database
- a part of a table
- a result set obtained by using tools such as joins

Introduction to MySQL Window Functions

MySQL Window Functions

an advanced SQL tool performing a calculation for every record in the data set, using other records associated with the specified one from the table

= “the current row”

Introduction to MySQL Window Functions

MySQL Window Functions

an advanced SQL tool performing a calculation for every record in the data set, using other records associated with the specified one from the table

= the *window* over which the given function evaluation will be performed

= "the current row"

Introduction to MySQL Window Functions

MySQL Window Functions

an advanced SQL tool performing a calculation for every record in the data set, using other records associated with the specified one from the table

= "the current row"

= the *window* over which the given function evaluation will be performed

= acts as *the set of rows* on which the given function will be applied

Introduction to MySQL Window Functions

- Using Window Functions is similar yet not identical to using Aggregate Functions

Introduction to MySQL Window Functions

MySQL Window Functions

Introduction to MySQL Window Functions

MySQL Window Functions

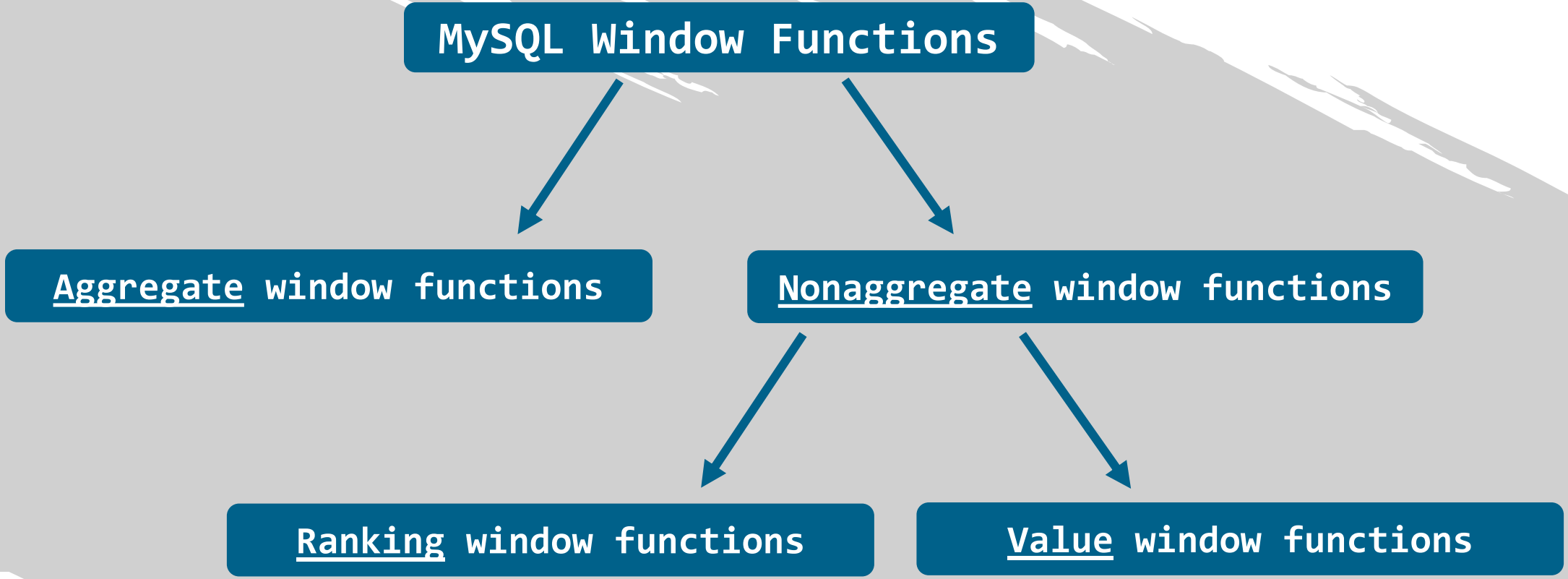
Aggregate window functions

= aggregate functions used in the context of window functions

Introduction to MySQL Window Functions

= aggregate functions used in the context of window functions

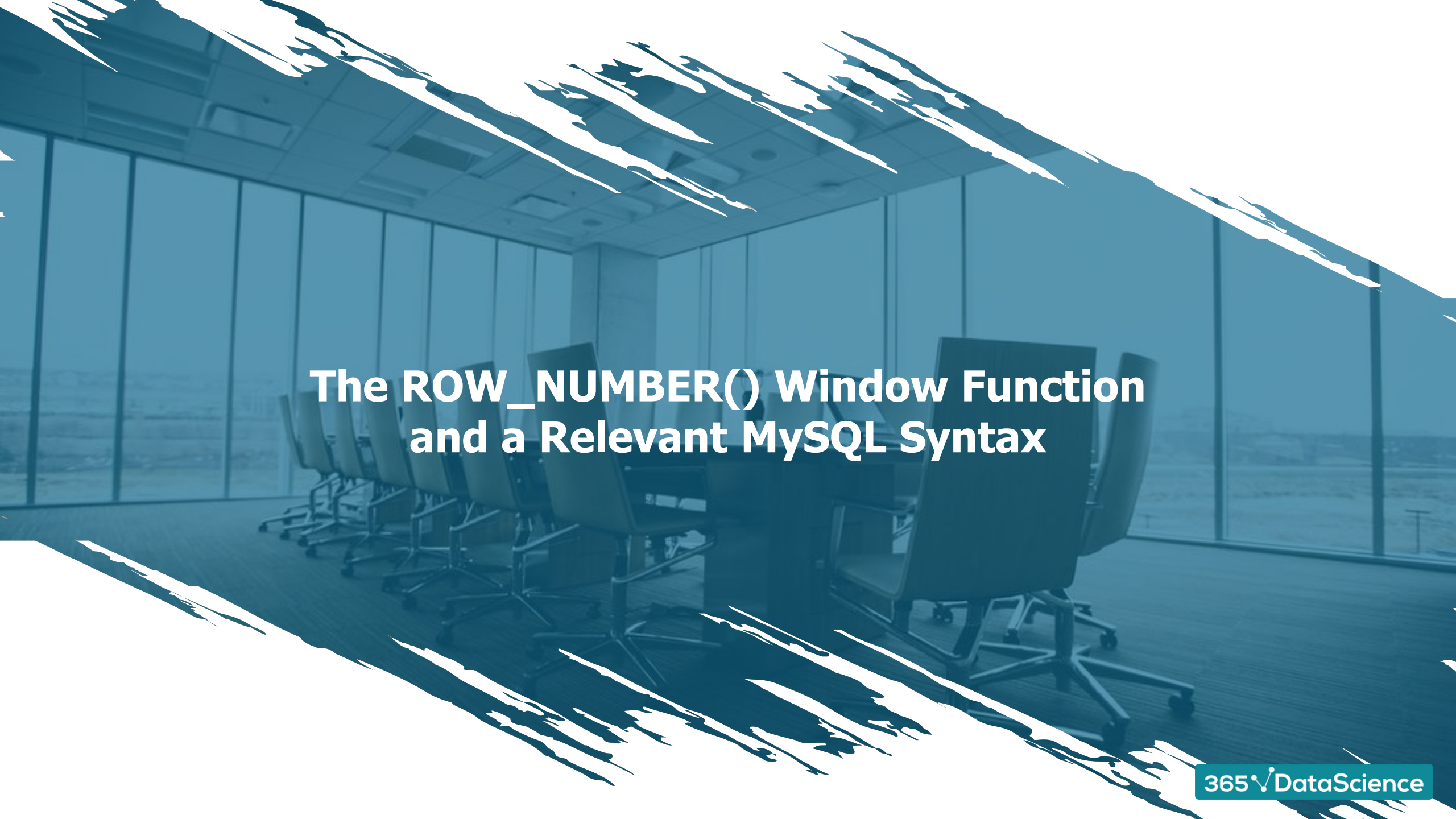
Introduction to MySQL Window Functions



Introduction to MySQL Window Functions

Next:

The `ROW_NUMBER()` Window Function and a Relevant MySQL Syntax



The ROW_NUMBER() Window Function and a Relevant MySQL Syntax

365 DataScience

ROW_NUMBER() and a Relevant MySQL Syntax

- One of the syntax types available for applying window functions:


```sql
SELECT
    ...,
    ROW_NUMBER() OVER () AS ...
FROM
    ...
;
```

ROW_NUMBER() and a Relevant MySQL Syntax

window specification:	action:
None = an *empty* OVER clause	`ROW_NUMBER()` will perform the relevant evaluations on all query rows = a *single* partition
containing `PARTITION BY`	the data will be organized into partitions
containing `ORDER BY`	arrange the values (in an ascending or descending order)

ROW_NUMBER() and a Relevant MySQL Syntax

Next:

- Use several window functions in the same query

MySQL Window Functions Syntax

365 DataScience

MySQL Window Functions Syntax

- The same output can be obtained if we used a WINDOW clause


```sql
SELECT
    ...,
    ROW_NUMBER() OVER () AS ...
FROM
    ...
;
```

MySQL Window Functions Syntax

- The same output can be obtained if we used a WINDOW clause


```sql
SELECT
    ...,
    ROW_NUMBER() OVER alias AS ...
FROM
    ...
WINDOW alias AS ()
;
```

MySQL Window Functions Syntax

- naming windows is way more practical and a sign of best practice when:

  - we have a query employing *several* window functions
  - we need to refer to the same window specification *multiple times* throughout a query

MySQL Window Functions Syntax

a *window* name ≠ a *window function* name

e.g. w

e.g. `ROW_NUMBER()`


The PARTITION BY Clause vs the GROUP BY clause
365 DataScience

PARTITION BY vs GROUP BY


```sql
SELECT
    ...,
    ROW_NUMBER() OVER (PARTITION BY ...) AS ...
FROM
    ...;
```


```sql
SELECT
    ...,
FROM
    ...
GROUP BY;
```

PARTITION BY vs GROUP BY

	PARTITION BY	GROUP BY
Reduces the number of records returned	✗	✓
Affects *how* the window function result will be obtained	✓	✓
can *only* be used within the context of applying window functions	✓	✗

The MySQL RANK() and DENSE_RANK() Window Functions

365 DataScience

RANK() and DENSE_RANK()

- You may prefer to assign the *same* rank to records representing *identical* values

  → What rank values are assigned to the records *subsequent to* the records with an identical value?

	the focus is on:
RANK()	the *number* of values we have in our output
DENSE_RANK()	the *ranking* of the values *itself*

RANK() and DENSE_RANK()

Window Functions in MySQL:

- They all require the use of the `OVER` clause

- The rank values they provide are always assigned sequentially

- The first rank is always equal to the integer *1*, and the subsequent rank values grow *incrementally by 1*, except for the duplicate records potentially

RANK() and DENSE_RANK()

Window Functions in MySQL:

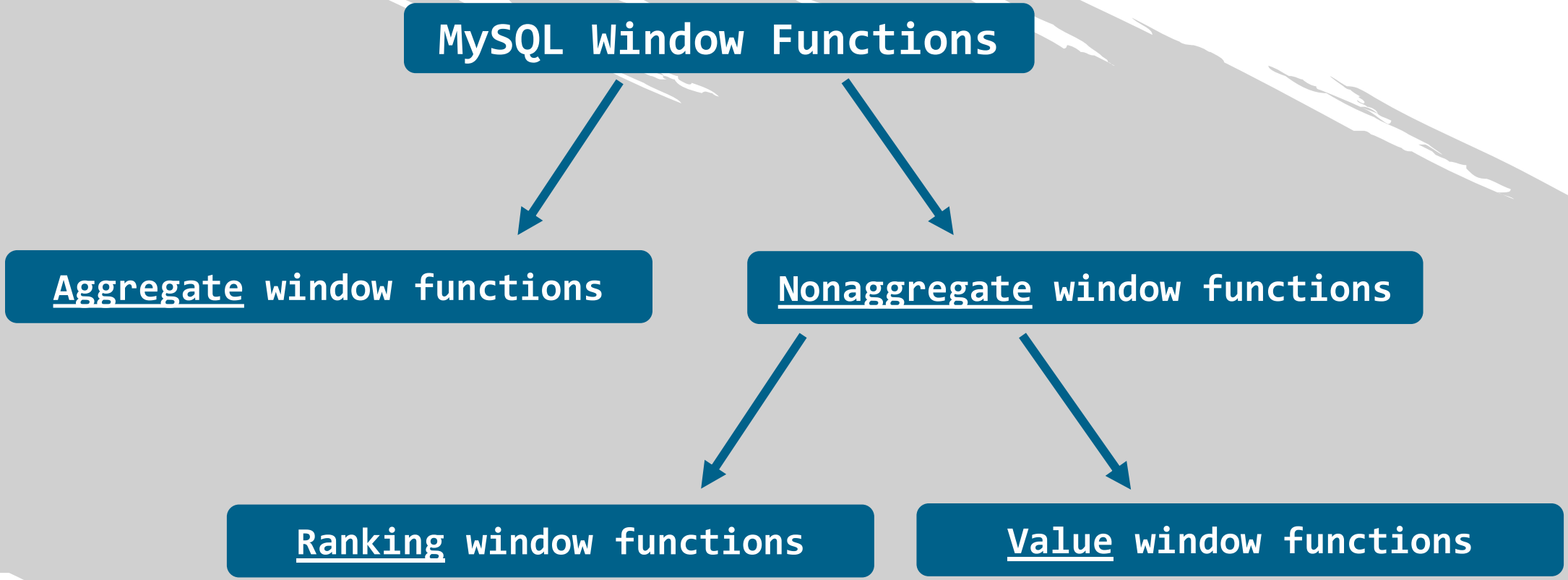
- `RANK()` and `DENSE_RANK()` are only useful when applied on *ordered partitions* (=partitions defined by the use of the `ORDER BY` clause)

	type:	`ORDER BY`
`ROW_NUMBER()`	non-order-sensitive	not necessarily
`RANK()` `DENSE_RANK()`	order-sensitive	more meaningful

The LAG() and LEAD() Value Window Functions

365 DataScience

The LAG() and LEAD() Value Window Functions



The LAG() and LEAD() Value Window Functions

- As opposed to ranking window functions, value window functions return a value *that can be found in the database*

LAG() returns the value from a specified field of a record that *precedes* the current row

= the value that *lags* the current value


```SQL
SELECT
    ...,
    LAG(column_name) OVER () AS ...
FROM
    ...;
```

The LAG() and LEAD() Value Window Functions

- As opposed to ranking window functions, value window functions return a value *that can be found in the database*

LEAD() returns the value from a specified field of a record that *follows* the current row

= the value that *leads* the current value

SQL

```
SELECT
    ...,
    LEAD(column_name) OVER () AS ...
FROM
    ...;
```

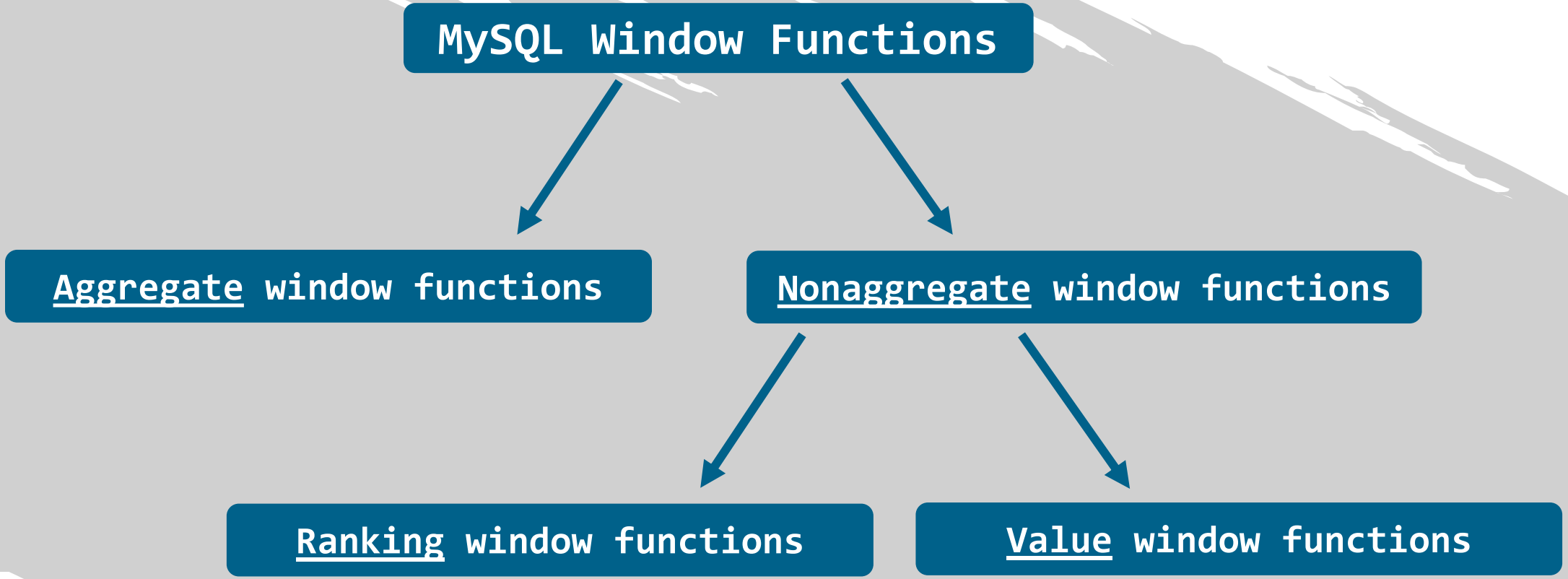

MySQL Aggregate Functions
in the Context of Window Functions
365 DataScience

Aggregate Functions in the Context of Window Functions

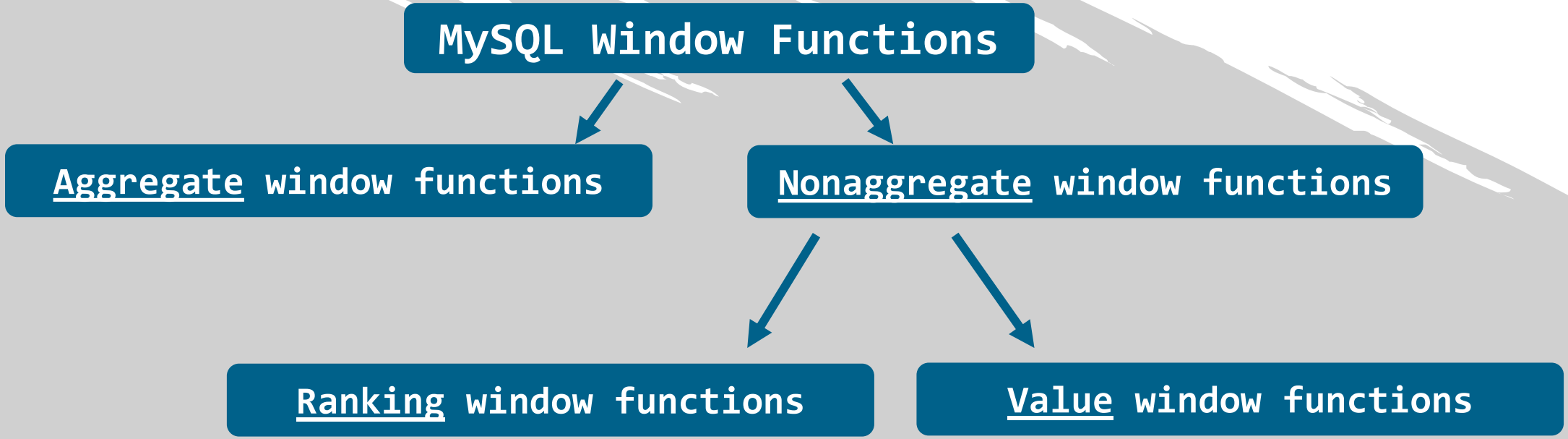
- MySQL aggregate functions in the context of window functions

  ≈ aggregate window functions

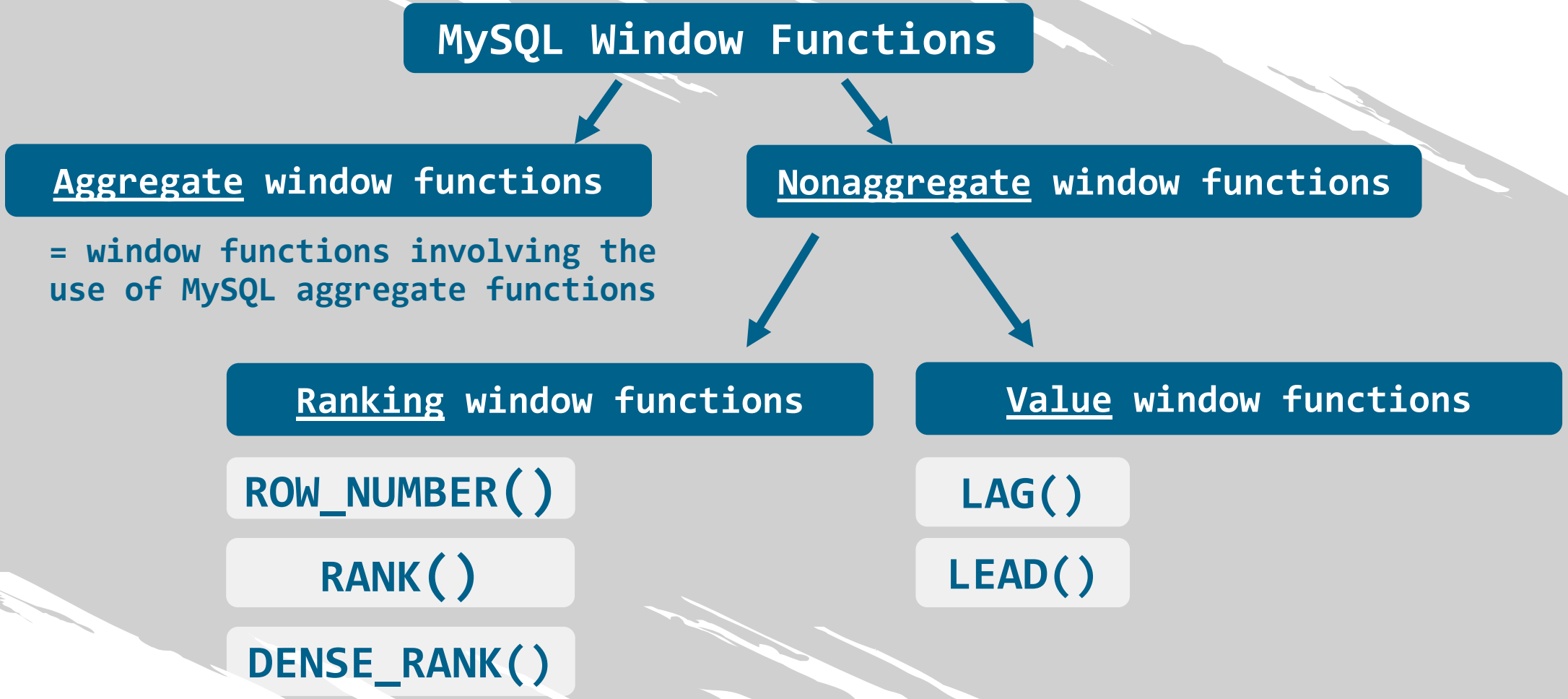
Aggregate Functions in the Context of Window Functions



Aggregate Functions in the Context of Window Functions



Aggregate Functions in the Context of Window Functions



Aggregate Functions in the Context of Window Functions

We must be very careful when utilizing aggregate functions

e.g. `SUM()` `AVG()`

Whether or not the aggregate functions will relate to the window function we are implementing, depends entirely on the way we organize our data and on the syntax we employ

Aggregate Functions in the Context of Window Functions

	MySQL aggregate functions	MySQL aggregate functions in the context of window functions (≈aggregate window functions)
Application of the window function on:	*groups* of values	data *partitions*
Reference to:	the values of a certain column	a window specification
MySQL clause:	GROUP BY	OVER, PARTITION BY, WINDOW
Reduces the number of records returned:	✓	✗