



Window functions in deep





Who am I

Andrii Sukhoi

- Backend developer
- 5 year of experience
- 3 years in enterprise development
- Tech stack: php, postgresql, vuejs, mysql, apache

 **GitHub** <https://github.com/sukhoi94>

 **LinkedIn** <https://www.linkedin.com/in/andrii-sukhoi-0a3773125/>

 **stackoverflow** <https://stackoverflow.com/users/6478501/andrii-sukhoi>

manchester216@gmail.com



Dummy table posts

+ Options

<div><div><div></div><div></div><div></div></div></div>			id	title	body	author	category	views	created_at
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	1	The Wall	Awesome ablum	Andrii	Rock	40	2020-12-13 12:52:58
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	2	Wish you were here	Wish you were here song	Jane	Rock	50	2020-12-13 12:52:58
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	3	Loosing my religion	R.E.M song	Andrii	Rock	25	2020-12-13 12:52:58
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	4	Window functions	Window functions	Bob	sql	12	2020-12-13 12:52:58
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	5	Stupid stuff	no body :D	Bob	Other	40	2020-12-13 12:52:58
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	6	SELECT clause	SELECT clause body	Jane	sql	1	2020-12-13 12:52:58
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	7	Church chants	Church chants body	Jane	Chants	25	2020-12-13 12:59:13
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	8	loops in php	loops in php	Bob	php	45	2020-12-13 12:59:13
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	9	if condition in php	php if	Andrii	php	522	2020-12-13 12:59:16
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	10	loops in php, part 2	loops in php part 2	Jane	php	59	2020-12-13 12:59:16

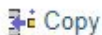


☐ Check all

With selected:



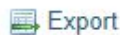
Edit



Copy



Delete



Export



Init dummy table


https://github.com/sukhoy94/window_functions_cheatsheet


	sukhoy94 Update README.md	a69429b 3 minutes ago	🕒 23 commits
	README.md	Update README.md	3 minutes ago
	dummy_table_setup.txt	queries upper case	2 days ago


No description provided.

 Readme




Releases

 main [window_functions_cheatsheet / dummy_table_setup.txt](#) Go to file ...

 andrii queries upper case Latest commit 204b8a0 2 days ago [History](#)

 0 contributors

33 lines (31 sloc) | 1.04 KB

[Raw](#) [Blame](#)   

```
1 DROP DATABASE IF EXISTS wfunctions;
2 CREATE DATABASE wfunctions;
3
4 CREATE OR REPLACE TABLE wfunctions.posts
5 (
6     id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
7     title VARCHAR(200) NOT NULL,
8     body TEXT NOT NULL,
9     author VARCHAR(50) NOT NULL,
```

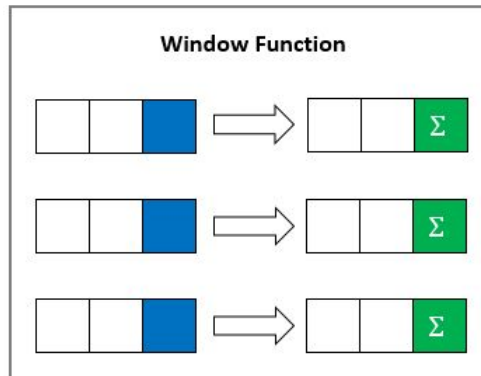
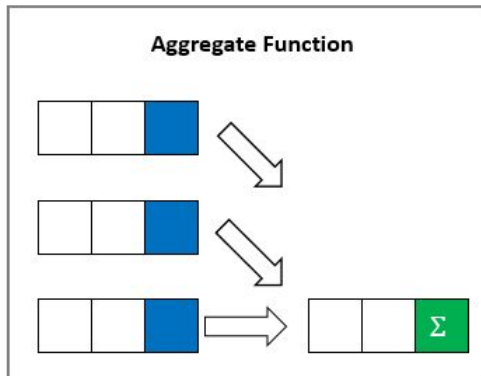


What is window function ?

Window functions allow access to data in the records right before and after the current record. A window function defines a *frame* or *window* of rows with a given length around the current row, and performs a calculation across the set of data in the window

A window function performs an aggregate-like operation on a set of query rows. However, whereas an aggregate operation groups query rows into a single result row, a window function produces a result for each query row:

- The row for which function evaluation occurs is called the current row.
- The query rows related to the current row over which function evaluation occurs comprise the window for the current row.





What is frame in window function ?

A frame is a subset of the current partition - keywords PARTITION and ORDER

+ Options

				id	title	body	author	category	views	created_at
<input type="checkbox"/>				1	The Wall	Awesome ablum	Andrii	Rock	40	2020-12-13 12:52:58
<input type="checkbox"/>				2	Wish you were here	Wish you were here song	Jane	Rock	50	2020-12-13 12:52:58
<input type="checkbox"/>				3	Loosing my religion	R.E.M song	Andrii	Rock	25	2020-12-13 12:52:58
<input type="checkbox"/>				4	Window functions	Window functions	Bob	sql	12	2020-12-13 12:52:58
<input type="checkbox"/>				5	Stupid stuff	no body :D	Bob	Other	40	2020-12-13 12:52:58
<input type="checkbox"/>				6	SELECT clause	SELECT clause body	Jane	sql	1	2020-12-13 12:52:58
<input type="checkbox"/>				7	Church chants	Church chants body	Jane	Chants	25	2020-12-13 12:59:13
<input type="checkbox"/>				8	loops in php	loops in php	Bob	php	45	2020-12-13 12:59:13
<input type="checkbox"/>				9	if condition in php	php if	Andrii	php	522	2020-12-13 12:59:16
<input type="checkbox"/>				10	loops in php, part 2	loops in php part 2	Jane	php	59	2020-12-13 12:59:16

☐ Check all With selected: Edit Copy Delete Export



Aggregate functions example

An aggregate function is a function that summarizes the rows of a group into a single **value**.

- `AVG()`
- `BIT_AND()`
- `BIT_OR()`
- `BIT_XOR()`
- `COUNT()`
- `JSON_ARRAYAGG()`
- `JSON_OBJECTAGG()`
- `MAX()`
- `MIN()`
- `STDDEV_POP()`, `STDDEV()`, `STD()`
- `STDDEV_SAMP()`
- `SUM()`
- `VAR_POP()`, `VARIANCE()`
- `VAR_SAMP()`

```
SELECT
    SUM(views) as sum
FROM
    Posts;
```

```
SELECT
    AVG(views) as sum
FROM
    posts;
```

```
SELECT
    SUM(views) as sum_by_author
FROM
    posts
GROUP BY author;
```



Window functions list

Name	Description
<u>CUME_DIST ()</u>	Cumulative distribution value
<u>DENSE_RANK ()</u>	Rank of current row within its partition, without gaps
<u>FIRST_VALUE ()</u>	Value of argument from first row of window frame
<u>LAG ()</u>	Value of argument from row lagging current row within partition
<u>LAST_VALUE ()</u>	Value of argument from last row of window frame
<u>LEAD ()</u>	Value of argument from row leading current row within partition
<u>NTH_VALUE ()</u>	Value of argument from N-th row of window frame
<u>NTILE ()</u>	Bucket number of current row within its partition.
<u>PERCENT_RANK ()</u>	Percentage rank value
<u>RANK ()</u>	Rank of current row within its partition, with gaps
<u>ROW_NUMBER ()</u>	Number of current row within its partition



Window functions common syntax

```
FUNCTION_NAME([params]) OVER ([PARTITION, ORDER, NULL_TREATMENT])
```

```
LAST_VALUE (views) OVER (  
    PARTITION BY category  
    ORDER BY id  
    RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING  
) as last_views
```

```
ROW_NUMBER() OVER (PARTITION BY category ORDER BY id DESC)
```



Aggregate with w-functions

```
SELECT
  id,
  author,
  title,
  body,
  category,
  views,
  SUM(views) OVER() as total_views
FROM
  posts;
```

```
SELECT
  id,
  author,
  title,
  body,
  category,
  views,
  SUM(views) OVER(PARTITION BY author) as
total_views_by_author
FROM
  posts;
```

The `OVER` clause is permitted for many aggregate functions, which therefore can be used as window or nonwindow functions, depending on whether the `OVER` clause is present or absent:

- `AVG()`
- `BIT_AND()`
- `BIT_OR()`
- `BIT_XOR()`
- `COUNT()`
- `JSON_ARRAYAGG()`
- `JSON_OBJECTAGG()`
- `MAX()`
- `MIN()`
- `STDDEV_POP()`, `STDDEV()`, `STD()`
- `STDDEV_SAMP()`
- `SUM()`
- `VAR_POP()`, `VARIANCE()`
- `VAR_SAMP()`

MySQL also supports nonaggregate functions that are used **only as window functions**. For these, the `OVER` clause is mandatory:

- `CUME_DIST()`
- `DENSE_RANK()`
- `FIRST_VALUE()`
- `LAG()`
- `LAST_VALUE()`
- `LEAD()`
- `NTH_VALUE()`
- `NTILE()`
- `PERCENT_RANK()`
- `RANK()`
- `ROW_NUMBER()`

CUME_DIST() over_clause

Formula = RowNo/TotalRows

```
SELECT
  id,
  author,
  title,
  body,
  category,
  views,
  CUME_DIST() OVER (ORDER BY views) AS
views_distribution
FROM
  posts
```

Cumulate the probability of some value in a group of rows



Output Result 10						
	id	author	title	body	category	views
1	6	Jane	SELECT clause	SELECT clause body	sql	1
2	4	Bob	Window functions	Window functions	sql	12
3	3	Andrii	Loosing my religion	R.E.M song	Rock	25
4	7	Jane	Church chants	Church chants body	Chants	25
5	1	Andrii	The Wall	Awesome ablum	Rock	40
6	5	Bob	Stupid stuff	no body :D	Other	40
7	8	Bob	loops in php	loops in php	php	45
8	2	Jane	Wish you were here	Wish you were here song	Rock	50
9	10	Jane	loops in php, part 2	loops in php part 2	php	59
10	9	Andrii	if condition in php	php if	php	522

Real case

Sometimes, you may want to create a report that shows the top or bottom x% values from a data set, for example, top 1% of products by revenue.

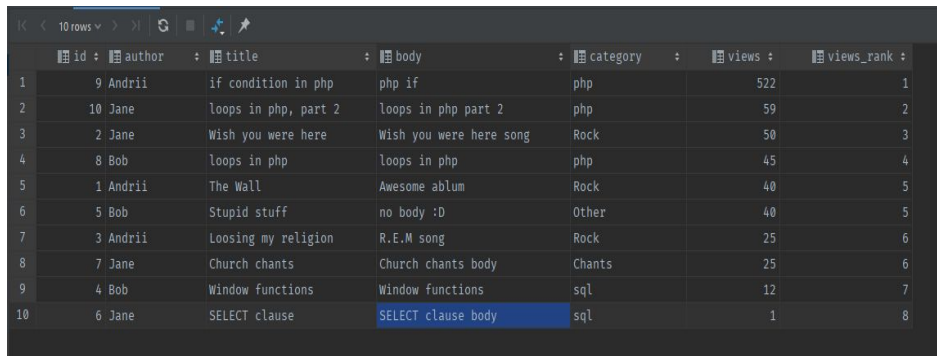
DENSE_RANK() over_clause

Returns the rank of the current row within its partition, without gaps. Peers are considered ties and receive the same rank. This function assigns consecutive ranks to peer groups; the result is that groups of size greater than one do not produce noncontiguous rank numbers.

For an example, see the [RANK\(\)](#) function description.

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

```
SELECT
  id,
  author,
  title,
  body,
  category,
  views,
  DENSE_RANK() over (ORDER BY views DESC)
as views_rank
FROM
  posts
ORDER BY views_rank
```



	id	author	title	body	category	views	views_rank
1	9	Andrii	if condition in php	php if	php	522	1
2	10	Jane	loops in php, part 2	loops in php part 2	php	59	2
3	2	Jane	Wish you were here	Wish you were here song	Rock	50	3
4	8	Bob	loops in php	loops in php	php	45	4
5	1	Andrii	The Wall	Awesome ablum	Rock	40	5
6	5	Bob	Stupid stuff	no body :D	Other	40	5
7	3	Andrii	Loosing my religion	R.E.M song	Rock	25	6
8	7	Jane	Church chants	Church chants body	Chants	25	6
9	4	Bob	Window functions	Window functions	sql	12	7
10	6	Jane	SELECT clause	SELECT clause body	sql	1	8

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.

```
SELECT
    id,
    author,
    title,
    body,
    category,
    views,
    RANK() over (ORDER BY views DESC) as views_rank
FROM
    posts
ORDER BY views_rank
```

id	author	title	body	category	views	views_rank
9	Andrii	if condition in php	php if	php	522	1
10	Jane	loops in php, part 2	loops in php part 2	php	59	2
2	Jane	Wish you were here	Wish you were here song	Rock	50	3
8	Bob	loops in php	loops in php	php	45	4
1	Andrii	The Wall	Awesome ablum	Rock	40	5
5	Bob	Stupid stuff	no body :D	Other	40	5
3	Andrii	Loosing my religion	R.E.M song	Rock	25	7
7	Jane	Church chants	Church chants body	Chants	25	7
4	Bob	Window functions	Window functions	sql	12	9
6	Jane	SELECT clause	SELECT clause body	sql	1	10

FIRST_VALUE (**expr**) [*null_treatment*] *over_clause*

```
SELECT
    id,
    author,
    title,
    body,
    category,
    views,
    FIRST_VALUE (views) OVER (PARTITION BY category) as first_views
FROM
    posts
```

id	author	title	body	category	views	first_views
7	Jane	Church chants	Church chants body	Chants	25	25
5	Bob	Stupid stuff	no body :D	Other	40	40
8	Bob	loops in php	loops in php	php	45	45
9	Andrii	if condition in php	php if	php	522	45
10	Jane	loops in php, part 2	loops in php part 2	php	59	45
1	Andrii	The Wall	Awesome ablum	Rock	40	40
2	Jane	Wish you were here	Wish you were here song	Rock	50	40
3	Andrii	Loosing my religion	R.E.M song	Rock	25	40
4	Bob	Window functions	Window functions	sql	12	12
6	Jane	SELECT clause	SELECT clause body	sql	1	12

LAST_VALUE (**expr**) [*null_treatment*] *over_clause*

```
SELECT
    id,
    author,
    title,
    body,
    category,
    views,
    LAST_VALUE(views) OVER (
        PARTITION BY category
        ORDER BY id
        RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING
    ) as last_views
FROM
    posts
ORDER BY category, id
```

FRAME_CLAUSE

RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW - default value

RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING

id	author	title	body	category	views	last_views
7	Jane	Church chants	Church chants body	Chants	25	25
5	Bob	Stupid stuff	no body :D	Other	40	40
8	Bob	loops in php	loops in php	php	45	59
9	Andrii	if condition in php	php if	php	522	59
10	Jane	loops in php, part 2	loops in php part 2	php	59	59
1	Andrii	The Wall	Awesome ablum	Rock	40	25
2	Jane	Wish you were here	Wish you were here song	Rock	50	25
3	Andrii	Loosing my religion	R.E.M song	Rock	25	25
4	Bob	Window functions	Window functions	sql	12	1
6	Jane	SELECT clause	SELECT clause body	sql	1	1

LAG(*expr* [, *N* [, *default*]]) [*null_treatment*] *over_clause*

Returns the value of *expr* from the row that lags (precedes) the current row by *N* rows within its partition. If there is no such row, the return value is *default*. For example, if *N* is 3, the return value is *default* for the first two rows. If *N* or *default* are missing, the defaults are 1 and NULL, respectively.

N must be a literal nonnegative integer. If *N* is 0, *expr* is evaluated for the current row.

```
SELECT
    id,
    author,
    title,
    body,
    category,
    views,
    LAG(views) OVER (
        PARTITION BY category ORDER BY id
    ) as views_lag
FROM
    posts
ORDER BY category, id
```

	id	author	title	body	category	views	views_lag
1	7	Jane	Church chants	Church chants body	Chants	25	<null>
2	5	Bob	Stupid stuff	no body :D	Other	40	<null>
3	8	Bob	loops in php	loops in php	php	45	<null>
4	9	Andrii	if condition in php	php if	php	522	45
5	10	Jane	loops in php, part 2	loops in php part 2	php	59	522
6	1	Andrii	The Wall	Awesome ablum	Rock	40	<null>
7	2	Jane	Wish you were here	Wish you were here song	Rock	50	40
8	3	Andrii	Loosing my religion	R.E.M song	Rock	25	50
9	4	Bob	Window functions	Window functions	sql	12	<null>
10	6	Jane	SELECT clause	SELECT clause body	sql	1	12

```

SELECT
    id,
    author,
    title,
    body,
    category,
    views,
    row_number() OVER w AS rating,
    lag(views) OVER w - views AS views_lag
FROM
    posts
WINDOW w AS (ORDER BY views DESC)
ORDER BY rating

```

	id	author	title	body	category	views	rating	views_lag
	9	Andrii	if condition in php	php if	php	522	1	<null>
	10	Jane	loops in php, part 2	loops in php part 2	php	59	2	463
	2	Jane	Wish you were here	Wish you were here song	Rock	50	3	9
	8	Bob	loops in php	loops in php	php	45	4	5
	1	Andrii	The Wall	Awesome ablum	Rock	40	5	5
	5	Bob	Stupid stuff	no body :D	Other	40	6	0
	3	Andrii	Loosing my religion	R.E.M song	Rock	25	7	15
	7	Jane	Church chants	Church chants body	Chants	25	8	0
	4	Bob	Window functions	Window functions	sql	12	9	13
	6	Jane	SELECT clause	SELECT clause body	sql	1	10	11

LEAD(*expr* [, *N* [, *default*]]) [*null_treatment*] over *clause*

```
SELECT
    id,
    author,
    title,
    body,
    category,
    views,
    LEAD(views) OVER (
        PARTITION BY category ORDER BY id
    ) as views_lead
FROM
    posts
ORDER BY category, id
```

Returns the value of *expr* from the row that leads (follows) the current row by *N* rows within its partition. If there is no such row, the return value is *default*. For example, if *N* is 3, the return value is *default* for the last two rows. If *N* or *default* are missing, the defaults are 1 and NULL, respectively.

N must be a literal nonnegative integer. If *N* is 0, *expr* is evaluated for the current row.

id	author	title	body	category	views	views_lead
7	Jane	Church chants	Church chants body	Chants	25	<null>
5	Bob	Stupid stuff	no body :D	Other	40	<null>
8	Bob	loops in php	loops in php	php	45	522
9	Andrii	if condition in php	php if	php	522	59
10	Jane	loops in php, part 2	loops in php part 2	php	59	<null>
1	Andrii	The Wall	Awesome ablum	Rock	40	50
2	Jane	Wish you were here	Wish you were here song	Rock	50	25
3	Andrii	Loosing my religion	R.E.M song	Rock	25	<null>
4	Bob	Window functions	Window functions	sql	12	1
6	Jane	SELECT clause	SELECT clause body	sql	1	<null>

`NTH_VALUE(expr, N) [from_first_last] [null_treatment] over_clause`

Returns the value of *expr* from the *n*-th row of the window frame. If there is no such row, the return value is NULL.

n must be a literal positive integer.

from_first_last is part of the SQL standard, but the MySQL implementation permits only FROM FIRST (which is also the default). This means that calculations begin at the first row of the window. FROM LAST is parsed, but produces an error. To obtain the same effect as FROM LAST (begin calculations at the last row of the window), use ORDER BY to sort in reverse order.

```
SELECT
  id,
  author,
  title,
  body,
  category,
  views,
  NTH_VALUE(views, 1) OVER (
    PARTITION BY category ORDER BY id
  ) as views_second_in_frame
FROM
  posts
ORDER BY category, id
```

id	author	title	body	category	views	views_second_in_frame
7	Jane	Church chants	Church chants body	Chants	25	25
5	Bob	Stupid stuff	no body	Other	40	40
8	Bob	loops in php	loops in php	php	45	45
9	Andrii	if condition in php	php if	php	522	45
10	Jane	loops in php, part 2	loops in php, part 2	php	59	45
1	Andrii	The Wall	Awesome ablum	Rock	40	40
2	Jane	Wish you were here	Wish you were here song	Rock	50	40
3	Andrii	Loosing my religion	R.E.M song	Rock	25	40
4	Bob	Window functions	Window functions	sql	12	12
6	Jane	SELECT clause	SELECT clause body	sql	1	12

NTILE (*N*) over_clause

Divides a partition into *N* groups (buckets), assigns each row in the partition its bucket number, and returns the bucket number of the current row within its partition. For example, if *N* is 4, NTILE () divides rows into four buckets. If *N* is 100, NTILE () divides rows into 100 buckets. *N* must be a literal positive integer. Bucket number return values range from 1 to *N*.

```
SELECT
    id,
    author,
    title,
    body,
    category,
    views,
    NTILE(2) OVER (
        ORDER BY id
    ) as ntile_views
FROM
    posts
ORDER BY category, id
```

	id	author	title	body	category	views	ntile_views
1	1	Andrii	The Wall	Awesome ablum	Rock	40	1
2	2	Jane	Wish you were here	Wish you were here song	Rock	50	1
3	3	Andrii	Loosing my religion	R.E.M song	Rock	25	1
4	4	Bob	Window functions	Window functions	sql	12	1
5	5	Bob	Stupid stuff	no body	Other	40	1
6	6	Jane	SELECT clause	SELECT clause body	sql	1	2
7	7	Jane	Church chants	Church chants body	Chants	25	2
8	8	Bob	loops in php	loops in php	php	45	2
9	9	Andrii	if condition in php	php if	php	522	2
10	10	Jane	loops in php, part 2	loops in php, part 2	php	59	2

PERCENT_RANK() over_clause

Returns the percentage of partition values less than the value in the current row, excluding the highest value. Return values range from 0 to 1 and represent the row relative rank, calculated as the result of this formula, where *rank* is the row rank and *rows* is the number of partition rows:

$$(rank - 1) / (rows - 1)$$

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

```
SELECT
  id,
  author,
  title,
  body,
  category,
  views,
  PERCENT_RANK() OVER (ORDER BY views) AS
  'percent_rank'
FROM
  posts
```

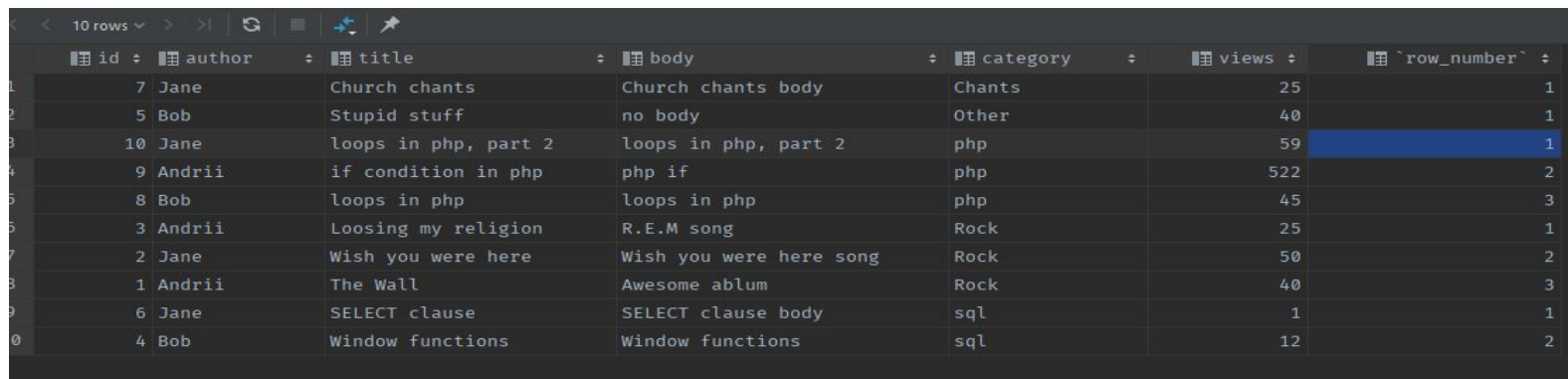
	id	author	title	body	category	views	'percent_rank'
1	6	Jane	SELECT clause	SELECT clause body	sql	1	0
2	4	Bob	Window functions	Window functions	sql	12	0.1111111111
3	3	Andrii	Loosing my religion	R.E.M song	Rock	25	0.2222222222
4	7	Jane	Church chants	Church chants body	Chants	25	0.2222222222
5	1	Andrii	The Wall	Awesome ablum	Rock	40	0.4444444444
6	5	Bob	Stupid stuff	no body	Other	40	0.4444444444
7	8	Bob	loops in php	loops in php	php	45	0.6666666667
8	2	Jane	Wish you were here	Wish you were here song	Rock	50	0.7777777778
9	10	Jane	loops in php, part 2	loops in php, part 2	php	59	0.8888888889
10	9	Andrii	if condition in php	php if	php	522	1

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.

```
SELECT
  id,
  author,
  title,
  body,
  category,
  views,
  ROW_NUMBER() OVER (PARTITION BY category
ORDER BY id DESC) AS 'row_number'
FROM
  posts
```



	id	author	title	body	category	views	row_number
1	7	Jane	Church chants	Church chants body	Chants	25	1
2	5	Bob	Stupid stuff	no body	Other	40	1
3	10	Jane	loops in php, part 2	loops in php, part 2	php	59	1
4	9	Andrii	if condition in php	php if	php	522	2
5	8	Bob	loops in php	loops in php	php	45	3
6	3	Andrii	Loosing my religion	R.E.M song	Rock	25	1
7	2	Jane	Wish you were here	Wish you were here song	Rock	50	2
8	1	Andrii	The Wall	Awesome ablum	Rock	40	3
9	6	Jane	SELECT clause	SELECT clause body	sql	1	1
0	4	Bob	Window functions	Window functions	sql	12	2



Thanks