MySQL DISTINCT on a GROUP_CONCAT()

154

24

I am doing `SELECT GROUP_CONCAT(categories SEPARATOR ' ') FROM table` . Sample data below:

```
categories
----------
test1 test2 test3
test4
test1 test3
test1 test3
```

However, I am getting `test1 test2 test3 test4 test1 test3` back and I would like to get `test1 test2 test3 test4` back. Any ideas?

Many thanks!

mysql group-concat

asked Jun 21 '10 at 9:37

user371990
771 2 5 3

7 Answers

318

GROUP_CONCAT has DISTINCT attribute:

```
SELECT GROUP_CONCAT(DISTINCT categories ORDER BY categories ASC SEPARATOR ' ') FROM
table
```

edited Jun 21 '10 at 9:47

Daniel Vassallo
277k 61 450 407

answered Jun 21 '10 at 9:41

Naktibalda
10.1k 3 24 43

42

Using DISTINCT will work

```
SELECT GROUP_CONCAT(DISTINCT(categories) SEPARATOR ' ') FROM table
```

REf:- this

answered Jun 21 '10 at 9:44

Salil
34.9k 17 100 138

(notice that `test1` and `test3` are duplicated) while the OP wants to return this string:

```
test1 test2 test3 test4
```

the problem here is that the string `"test1 test3"` is duplicated and is inserted only once, but all of the others are distinct to each other (`"test1 test2 test3"` is distinct than `"test1 test3"` , even if some tests contained in the whole string are duplicated).

What we need to do here is to split each string into different rows, and we first need to create a numbers table:

```
CREATE TABLE numbers (n INT);
INSERT INTO numbers VALUES
(1),(2),(3),(4),(5),(6),(7),(8),(9),(10);
```

then we can run this query:

```
SELECT
  SUBSTRING_INDEX(
    SUBSTRING_INDEX(tableName.categories, ' ', numbers.n),
    ' ',
    -1) category
FROM
  numbers INNER JOIN tableName
  ON
    LENGTH(tableName.categories)>=
    LENGTH(REPLACE(tableName.categories, ' ', ''))+numbers.n-1;
```

and we get a result like this:

```
test1
test4
test1
test1
test2
test3
test3
test3
```

and then we can apply GROUP_CONCAT aggregate function, using DISTINCT clause:

```
SELECT
  GROUP_CONCAT(DISTINCT category ORDER BY category SEPARATOR ' ')
FROM (
  SELECT
    SUBSTRING_INDEX(SUBSTRING_INDEX(tableName.categories, ' ', numbers.n), ' ', -1)
category
  FROM
    numbers INNER JOIN tableName
    ON LENGTH(tableName.categories)>=LENGTH(REPLACE(tableName.categories, ' ',
''))+numbers.n-1
  ) s;
```

Please see fiddle here.


40.4k 12 68 93

It appears your interpretation of OP's question may be right; however, I think it should be pointed out that normalizing the data by creating a "blah_to_categories" and a "categories" table for the appropriate many-to-many relationship would be the best practice here, and would add a lot of flexibility. Still, your answer is a smart workaround for anyone who inherits such a denormalized schema. It also could probably be adapted for the purpose of generating a migration from the old to the normalized schema. – XP84 Jun 1 '16 at 17:11

10

```
SELECT
  GROUP_CONCAT(DISTINCT (category))
FROM (
  SELECT
    SUBSTRING_INDEX(SUBSTRING_INDEX(tableName.categories, ' ', numbers.n), ' ', -1)
category
  FROM
    numbers INNER JOIN tableName
    ON LENGTH(tableName.categories)>=LENGTH(REPLACE(tableName.categories, ' ',
''))+numbers.n-1
  ) s;
```

This will return distinct values like: *test1,test2,test4,test3*

edited Feb 3 '17 at 12:31

pringi
1,058 2 15 21

answered Feb 3 '17 at 10:26

Sainath
111 1 3

5

`DISTINCT` : will gives you unique values.

```
SELECT GROUP_CONCAT(DISTINCT(categories )) AS categories FROM table
```

edited Apr 2 at 10:57

Shree
13.3k 20 76 125

answered Apr 2 at 10:55

Goshika Mahesh
61 1 3

1

I realize this question is old, but I feel like this should be mentioned: group_concat with distinct = performance killer. If you work in small databases, you won't notice, but when it scales - it won't work very well.

answered Jun 28 '17 at 22:24

photocode
345 1 6 18

3 I'm working with a 10 million row table and my query takes the same time with or without DISTINCT. I'm using InnoDB. – ashishduh Aug 10 '17 at 18:07

What data type? How many columns? In my DB, it's heavy on large text fields and there are about 30 some odd columns using distinct. Taking away distinct alone speeds it up dramatically, and it's using

1

You can simply add **DISTINCT** in front.

```
SELECT GROUP_CONCAT(DISTINCT categories SEPARATOR ' ')
```

if you want to sort,

```
SELECT GROUP_CONCAT(DISTINCT categories ORDER BY categories ASC SEPARATOR ' ')
```

answered Apr 1 at 7:04

Silent Spectator
8,349 4 55 65