Passing an array to a query using a WHERE clause

Asked 15 years, 7 months ago Modified 2 years, 8 months ago Viewed 700k times

Part of PHP Collective

341

Given an array of ids `$galleries = array(1,2,5)` I want to have a SQL query that uses the values of the array in its WHERE clause like:

```
SELECT *
FROM galleries
WHERE id = /* values of array $galleries... eg. (1 || 2 || 5) */
```

How can I generate this query string to use with MySQL?

PHP php mysql

Share

Improve this question

Follow

edited Apr 8, 2022 at 14:35

Braiam

4,484 ● 11 ● 49 ● 81

asked May 25, 2009 at 19:32

Quinn

A few modern/secure/stable alternatives using mysqli are elsewhere on Stack Overflow: Use an array in a mysqli prepared statement: `WHERE .. IN(..)` query and mysqli bind_param for array of strings – mickmackusa Apr 8, 2022 at 13:44

17 Answers

Sorted by: Highest score (default)

358

Locked. Comments on this answer have been disabled, but it is still accepting other interactions. Learn more.

> **BEWARE!** This answer contains a severe SQL injection vulnerability. Do NOT use the code samples as presented here, without making sure that any external input is sanitized.

```
$ids = join("','",$galleries);
$sql = "SELECT * FROM galleries WHERE id IN ('$ids')";
```

Share

edited Aug 22, 2017 at 17:24

answered May 25, 2009 at 19:37

Improve this answer

Follow

Machavity ♦
31.6k ● 27 ● 95 ● 105

Flavius Stef
13.8k ● 2 ● 27 ● 22

2 There were a lot of impassioned comments on this from both angles of SQL injection. The TL;DR here is that this answer DOES work, but, because it takes those raw values and puts them into SQL directly, if the upstream data is untrusted you COULD be opening yourself up to a SQL injection attack. There are other answers to this question that enumerate how to avoid that problem. – Machavity ♦ Apr 8, 2022 at 13:51

Comments disabled on deleted / locked posts / reviews

329

+200

Using PDO:[1]

```
$in = join(',', array_fill(0, count($ids), '?'));
$select = <<<SQL
    SELECT *
    FROM galleries
    WHERE id IN ($in);
SQL;
$statement = $pdo->prepare($select);
$statement->execute($ids);
```

Using MySQLi [2]

```
$in = join(',', array_fill(0, count($ids), '?'));
$select = <<<SQL
    SELECT *
    FROM galleries
    WHERE id IN ($in);
SQL;
$statement = $mysqli->prepare($select);
$statement->bind_param(str_repeat('i', count($ids)), ...$ids);
$statement->execute();
$result = $statement->get_result();
```

Explanation:

Use the SQL `IN()` operator to check if a value exists in a given list.

In general it looks like this:

```
expr IN (value,...)
```

We can build an expression to place inside the `()` from our array. Note that there must be at least one value inside the parenthesis or MySQL will return an error; this equates to making sure that our input array has at least one value. To help prevent

against SQL injection attacks, first generate a `?` for each input item to create a parameterized query. Here I assume that the array containing your ids is called `$ids`:

```
$in = join(',', array_fill(0, count($ids), '?'));

$select = <<<SQL
    SELECT *
    FROM galleries
    WHERE id IN ($in);
SQL;
```

Given an input array of three items `$select` will look like:

```
SELECT *
FROM galleries
WHERE id IN (?, ?, ?)
```

Again note that there is a `?` for each item in the input array. Then we'll use PDO or MySQLi to prepare and execute the query as noted above.

Using the `IN()` operator with strings

It is easy to change between strings and integers because of the bound parameters. For PDO there is no change required; for MySQLi change `str_repeat('i',` to `str_repeat('s',` if you need to check strings.

[1]: I've omitted some error checking for brevity. You need to check for the usual errors for each database method (or set your DB driver to throw exceptions).

[2]: Requires PHP 5.6 or higher. Again I've omitted some error checking for brevity.

Share

Improve this answer

Follow

edited Jun 20, 2020 at 9:12

Community Bot

1 ● 1

answered May 13, 2014 at 20:31

Levi Morrison

19.5k ● 8 ● 67 ● 86

Can anyone clear up what the "..." does or is supposed to be in the mysqli statement? – Chewie The Chorkie Apr 26, 2016 at 18:28

2 If you are referring to `$statement->bind_param(str_repeat('i', count($ids)), ...$ids);` then the `...` is expanding the id's from an array into multiple parameters. If you are referring to `expr IN (value,...)` then that just means that there can be more values eg `WHERE id IN (1, 3, 4)`. There just needs to be at least one. – Levi Morrison Apr 26, 2016 at 19:27

1 I was confused what <<< was but I found a reference: php.net/manual/en/… – Tsangares Jun 16, 2016 at 4:23

1 Also, here is the reference for the `...` : wiki.php.net/rfc/argument_unpacking – Tsangares Jun 16, 2016 at 4:35

60

ints:

```
$query = "SELECT * FROM `$table` WHERE `$column` IN(".implode(',',$array).")";
```

strings:

```
$query = "SELECT * FROM `$table` WHERE `$column`
IN('".implode("','",$array)."')";
```

Share

Improve this answer

Follow

edited Mar 19, 2014 at 13:30

Thomas Ahle

31.6k ● 21 ● 96 ● 118

answered Aug 11, 2011 at 16:48

user542568

777 ● 5 ● 2

31

Assuming you properly sanitize your inputs beforehand...

```
$matches = implode(',', $galleries);
```

Then just adjust your query:

```
SELECT *
FROM galleries
WHERE id IN ( $matches )
```

Quote values appropriately depending on your dataset.

Share Improve this answer Follow

answered May 25, 2009 at 19:38

AvatarKava

15.4k ● 2 ● 28 ● 33

I tried what you are proposing but it just fetched the first key value. I know it doesn't make sense, but if I do it using user542568 example, the damned thing works. – Samuel Ramzan Apr 25, 2020 at 18:16

18

Use:

```
select id from galleries where id in (1, 2, 5);
```

A simple `for each` loop will work.

Flavius/AvatarKava's way is better, but make sure that none of the array values contain commas.

Share
Improve this answer
Follow

edited May 23, 2017 at 12:10
Community Bot
1 • 1

answered May 25, 2009 at 19:36
Matthew Flaschen
284k • 53 • 521 • 552

10

As Flavius Stef's answer, you can use `intval()` to make sure all `id` are int values:

```
$ids = join(',', array_map('intval', $galleries));
$sql = "SELECT * FROM galleries WHERE id IN ($ids)";
```

Share
Improve this answer
Follow

edited May 23, 2017 at 11:55
Community Bot
1 • 1

answered Apr 11, 2015 at 17:48
Duyet Le
187 • 1 • 2 • 6

7

For MySQLi with an escape function:

```
$ids = array_map(function($a) use($mysqli) {
    return is_string($a) ? "'".$mysqli->real_escape_string($a)."'" : $a;
  }, $ids);
$ids = join(',', $ids);
$result = $mysqli->query("SELECT * FROM galleries WHERE id IN ($ids)");
```

For PDO with prepared statement:

```
$qmarks = implode(',', array_fill(0, count($ids), '?'));
$sth = $dbh->prepare("SELECT * FROM galleries WHERE id IN ($qmarks)");
$sth->execute($ids);
```

Share
Improve this answer
Follow

edited Nov 8, 2017 at 9:28
Stephan Richter
1,149 • 13 • 31

answered Apr 15, 2015 at 14:49
artoodetoo
938 • 10 • 55

MySQLi has prepared statements too. Do not escape your input, this is potentially still vulnerable to SQL injection. – Dharman ♦ Feb 25, 2020 at 22:54

We should take care of *SQL injection* vulnerabilities and an *empty condition*. I am going to handle both as below.

6

For a pure numeric array, use the appropriate type conversion viz `intval` or `floatval` or `doubleval` over each element. For string types `mysqli_real_escape_string()` which may also be applied to numeric values if you wish. *MySQL allows numbers as well as date variants as string*.

To appropriately escape the values before passing to the query, create a function similar to:

```
function escape($string)
{
    // Assuming $db is a link identifier returned by mysqli_connect() or
mysqli_init()
    return mysqli_real_escape_string($db, $string);
}
```

Such a function would most likely be already available to you in your application, or maybe you've already created one.

Sanitize the string array like:

```
$values = array_map('escape', $gallaries);
```

A numeric array can be sanitized using `intval` or `floatval` or `doubleval` instead as suitable:

```
$values = array_map('intval', $gallaries);
```

Then finally build the query condition

```
$where  = count($values) ? "`id` = '" . implode("' OR `id` = '", $values) . "'"
: 0;
```

or

```
$where  = count($values) ? "`id` IN ('" . implode("', '", $values) . "')" : 0;
```

Since the array can also be empty sometimes, like `$galleries = array();` we should therefore note that `IN ()` does not allow for an empty list. One can also use `OR` instead, but the problem remains. So the above check, `count($values)`, is to ensure the same.

And add it to the final query:

```
$query  = 'SELECT * FROM `galleries` WHERE ' . $where;
```

> **TIP**: If you want to show all records (no filtering) in case of an empty array instead of hiding all rows, simply replace **0** with **1** in the ternary's false part.

Share
Improve this answer
Follow

edited Oct 18, 2015 at 20:05

Peter Mortensen
31.6k ● 22 ● 109 ● 133

answered Apr 16, 2015 at 9:12

Izhar Aazmi
935 ● 12 ● 25

To make my solution a one-liner *(and ugly one)*, just in case someone needs to: `$query = 'SELECT * FROM galleries WHERE ' . (count($gallaries) ? "id IN ('" . implode("', '", array_map('escape', $gallaries)) . "')" : 0);` – Izhar Aazmi
Apr 17, 2015 at 13:38

6

Safe way without PDO:

```
$ids = array_filter(array_unique(array_map('intval', (array)$ids)));

if ($ids) {
    $query = 'SELECT * FROM `galleries` WHERE `id` IN ('.implode(',',
$ids).');';
}
```

- `(array)$ids` Cast `$ids` variable to array
- `array_map` Transform all array values into integers
- `array_unique` Remove repeated values
- `array_filter` Remove zero values
- `implode` Join all values to IN selection

Share Improve this answer Follow

answered Jan 30, 2018 at 9:12

Lito
1,332 ● 2 ● 18 ● 31

5

Safer.

```
$galleries = array(1,2,5);
array_walk($galleries , 'intval');
$ids = implode(',', $galleries);
$sql = "SELECT * FROM galleries WHERE id IN ($ids)";
```

Share Improve this answer Follow

answered Apr 13, 2015 at 15:41

Filipe

71 ● 1 ● 4

5

Col. Shrapnel's SafeMySQL library for PHP provides type-hinted placeholders in its parametrised queries, and includes a couple of convenient placeholders for working with arrays. The `?a` placeholder expands out an array to a comma-separated list of escaped strings*.

For example:

```
$someArray = [1, 2, 5];
$galleries = $db->getAll("SELECT * FROM galleries WHERE id IN (?a)",
$someArray);
```

* Note that since MySQL performs automatic type coercion, it doesn't matter that SafeMySQL will convert the ids above to strings - you'll still get the correct result.

Share Improve this answer Follow

answered May 17, 2015 at 19:03

Mark Amery

154k ● 89 ● 426 ● 469

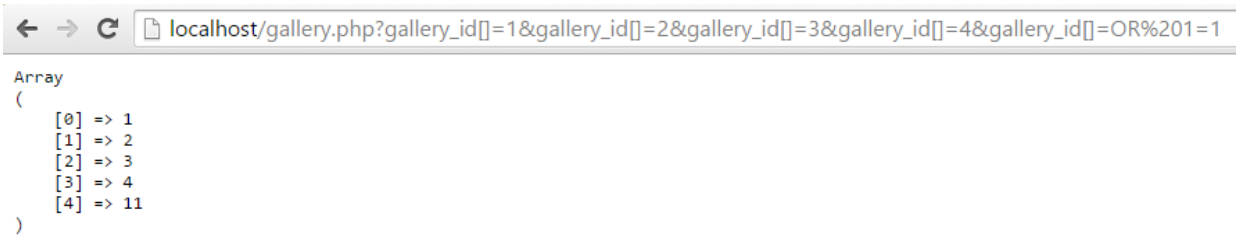
4

We can use this "WHERE id IN" clause if we filter the input array properly. Something like this:

```
$galleries = array();

foreach ($_REQUEST['gallery_id'] as $key => $val) {
    $galleries[$key] = filter_var($val, FILTER_SANITIZE_NUMBER_INT);
}
```

Like the example below:




```
$galleryIds = implode(',', $galleries);
```

I.e. now you should safely use `$query = "SELECT * FROM galleries WHERE id IN ({$galleryIds})";`

Share

Improve this answer

Follow

edited Apr 15, 2015 at 17:15

answered Apr 15, 2015 at 12:57

Supratim Roy

882 ● 11 ● 17

@levi-morrison posted a lot better solution to this. – Supratim Roy Apr 18, 2015 at 7:32

4

You may have table `texts` `(T_ID (int), T_TEXT (text))` and table `test` `(id (int), var (varchar(255)))`

In `insert into test values (1, '1,2,3') ;` the following will output rows from table texts where `T_ID IN (1,2,3)`:

```
SELECT * FROM `texts` WHERE (SELECT FIND_IN_SET( T_ID, ( SELECT var FROM test
WHERE id =1 ) ) AS tm) >0
```

This way you can manage a simple n2m database relation without an extra table and using only SQL without the need to use PHP or some other programming language.

Share

Improve this answer

Follow

edited Oct 18, 2015 at 19:46

Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Jun 3, 2011 at 10:41

SERJOU

57 ● 1

3

More an example:

```
$galleryIds = [1, '2', 'Vitruvian Man'];
$ids = array_filter($galleryIds, function($n){return (is_numeric($n));});
$ids = implode(', ', $ids);

$sql = "SELECT * FROM galleries WHERE id IN ({$ids})";
// output: 'SELECT * FROM galleries WHERE id IN (1, 2)'

$statement = $pdo->prepare($sql);
$statement->execute();
```

Share Improve this answer Follow

answered Aug 19, 2016 at 10:45

Ricardo Canelas

2,460 ● 31 ● 22

2

Besides using the IN query, you have two options to do so as in an IN query there is a risk of an SQL injection vulnerability. You can use **looping** to get the exact data you want or you can use the query with **OR** case

```
1. SELECT *
      FROM galleries WHERE id=1 or id=2 or id=5;


2. $ids = array(1, 2, 5);
   foreach ($ids as $id) {
      $data[] = SELECT *
                     FROM galleries WHERE id= $id;
   }
```

Share

Improve this answer

Follow

edited Oct 18, 2015 at 19:49

Peter Mortensen

31.6k ● 22 ● 109 ● 133

answered Apr 14, 2015 at 5:28

Gaurav Singh

189 ● 2 ● 7

Because the original question relates to an array of numbers and I am using an array of strings I couldn't make the given examples work.

I found that each string needed to be encapsulated in single quotes to work with the `IN()` function.

Here is my solution

```
foreach($status as $status_a) {
        $status_sql[] = '\''.$status_a.'\'';
    }
    $status = implode(',',$status_sql);

$sql = mysql_query("SELECT * FROM table WHERE id IN ($status)");
```

As you can see the first function wraps each array variable in `single quotes (\')` and then implodes the array.

NOTE: `$status` does not have single quotes in the SQL statement.

There is probably a nicer way to add the quotes but this works.

Share

Improve this answer

Follow

edited Aug 4, 2015 at 14:34

Ram Sharma

8,809 ● 7 ● 47 ● 59

answered Jan 27, 2011 at 2:43

RJaus

193 ● 12

1 Or `$filter = "'" . implode("','",$status) . "'";` – Alejandro Salamanca Mazuelo Apr 13, 2015 at 20:06

1 This is injection-vulnerable. – Mark Amery May 17, 2015 at 18:30

Where is escaping of strings? For example `'` inside the string? SQL Injection vulnerable. Use PDO::quote or mysqli_real_escape_string. – 18C Dec 21, 2017 at 13:41

Below is the method I have used, using PDO with named placeholders for other data. To overcome SQL injection I am filtering the array to accept only the values that are integers and rejecting all others.

```
$owner_id = 123;
$galleries = array(1,2,5,'abc');

$good_galleries = array_filter($chapter_arr, 'is_numeric');

$sql = "SELECT * FROM galleries WHERE owner=:OWNER_ID AND id IN
($good_galleries)";
$stmt = $dbh->prepare($sql);
$stmt->execute(array(
    "OWNER_ID" => $owner_id,
```

```
));

$data = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

Share Improve this answer Follow

answered Jan 24, 2018 at 16:31

kojow7

11.3k ● 19 ● 90 ● 146

When you use `is_numeric()` so beware that `0x539` is also a numeric value, so is `0b10100111001` – B001ᛦ May 10, 2021 at 19:53