

Database Queries in PHP 8.2

Craig Francis

Code Auditor

OWASP Chapter Leader, Bristol

WordPress Database Component Maintainer

Code Club Volunteer

Bristol Central Library

Saturdays, 11:45 - 12:45

4th February to 25th March (8 sessions)

For children aged 9 - 13



Use a DB Abstraction

Laravel DB

Doctrine

Propel ORM

RedBean

CakePHP

Zeta Components

Avoids most (not all) SQL Injection issues.

Ref ``literal-string``

```
$qb->select('u')  
  
->from('User', 'u')  
  
->where($qb->expr()->andX(  
  
    $qb->expr()->eq('u.type_id', $_GET['type_id']),  
  
    $qb->expr()->isNull('u.deleted')  
  
));
```

```
$qb->select('u')
```

```
->from('User', 'u')
```

```
->where($qb->expr()->andX(
```

```
    $qb->expr()->eq('u.type_id', $_GET['type_id']), // INSECURE
```

```
    $qb->expr()->isNull('u.deleted')
```

```
));
```

'u.type_id) OR (1 = 1'



```
$qb->select('u')
```

```
->from('User', 'u')
```

```
->where('u.id = :identifier')
```

```
->setParameter('identifier', $_GET['id']);
```

```
$qb->select('u')
```

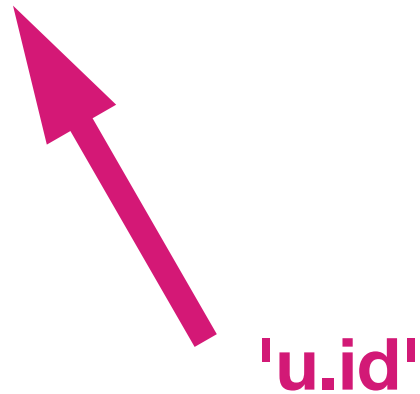
```
->from('User', 'u')
```

```
->where('u.id = ' . $_GET['id']);
```

```
$qb->select('u')
```

```
->from('User', 'u')
```

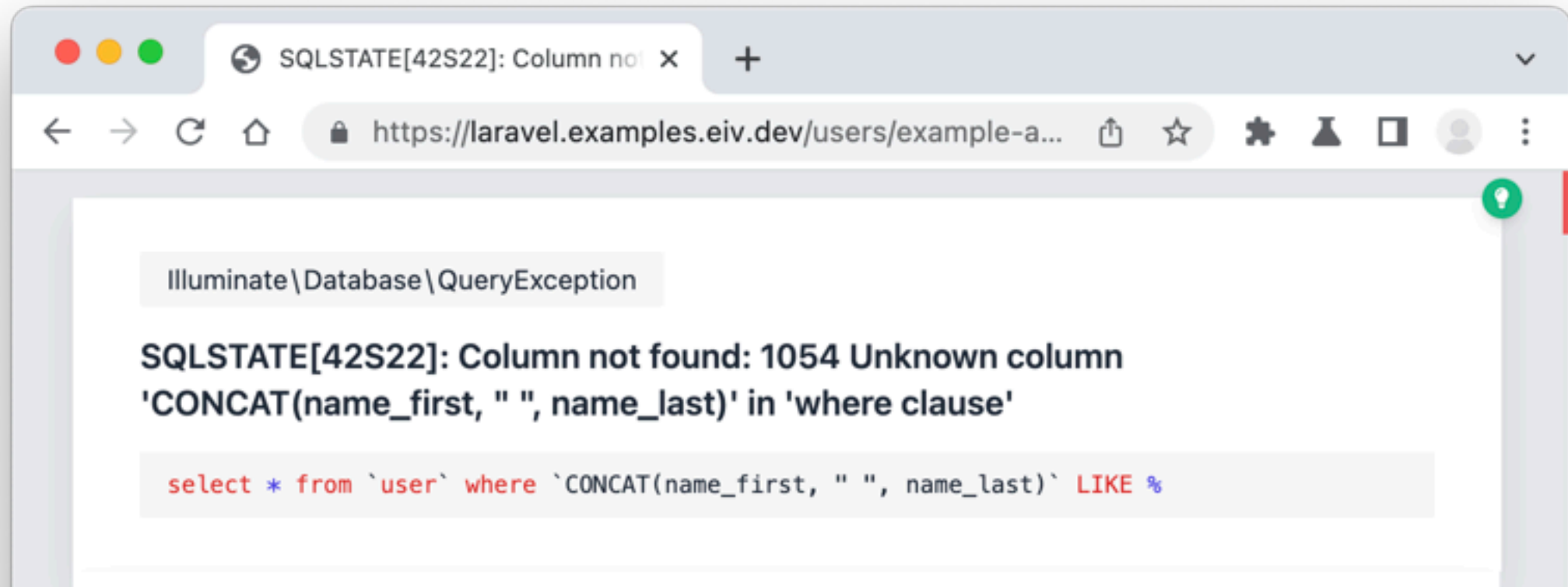
```
->where('u.id = ' . $_GET['id']); // INSECURE
```



```
DB::table('user')  
->where('name', 'LIKE', $search . '%');
```


DB::table('user')

->where('CONCAT(name_first, " ", name_last)', 'LIKE', \$search . '%');



DB::table('user')

->whereRaw('CONCAT(name_first, " ", name_last) LIKE ?', \$search . '%');



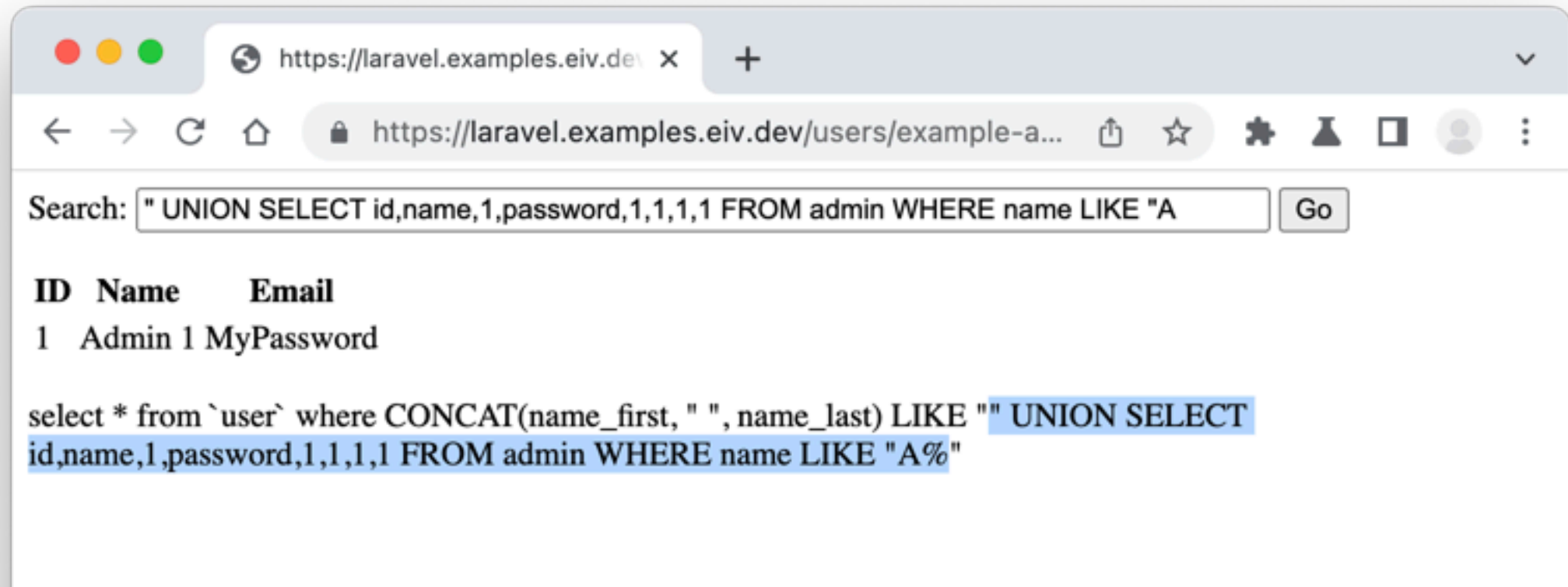
DB::table('user')

->whereRaw('CONCAT(name_first, " ", name_last) LIKE "' . \$search . '%"');



DB::table('user')

->whereRaw('CONCAT(name_first, " ", name_last) LIKE "' . \$search . '%"');




Raw SQL

Pre PHP 8.1

```
1 <?php
2
3 > $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 > $rows = $db->query('SELECT * FROM user WHERE id = ' . $_GET['id']);
6
7 > foreach ($rows as $row) {
8 >     print_r($row);
9 > }
```

```
1 <?php
2
3 > $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 > $rows = $db->query('SELECT * FROM user WHERE id = ' . $db->real_escape_string($_GET['id']));
6
7 > foreach ($rows as $row) {
8 >     print_r($row);
9 > }
```

```
<?php
1
2
3  $db = new mysqli('localhost', 'test', 'test', 'test');
4
5  $rows = $db->query('SELECT * FROM user WHERE id = "' . $db->real_escape_string($_GET['id']) . '"');
6
7  foreach ($rows as $row) {
8      print_r($row);
9  }
```




```
1 <?php
2
3 > $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 > $statement = $db->prepare('SELECT * FROM user WHERE id = ?');
6 > $statement->bind_param('i', $_GET['id']);
7 > $statement->execute();
8
9 > $result = $statement->get_result();
10
11 > while ($row = mysqli_fetch_assoc($result)) {
12 >     print_r($row);
13 > }
14
15
```

```
<?php
1
2
3  $db = new mysqli('localhost', 'test', 'test', 'test');
4
5  $statement = $db->prepare('SELECT * FROM user WHERE type IN (?, ?, ?)');
6  $statement->bind_param('sss', $type1, $type2, $type3);
7  $statement->execute();
8
9  $result = $statement->get_result();
10
11 while ($row = mysqli_fetch_assoc($result)) {
12     print_r($row);
13 }
14
15
```

```
1 <?php
2
3 > $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 > $statement = $db->prepare('SELECT * FROM user WHERE id = ?');
6 > $statement->bind_param('i', 1);
7 > $statement->execute();
8
9 > $result = $statement->get_result();
10
11 > while ($row = mysqli_fetch_assoc($result)) {
12 >     print_r($row);
13 > }
14
15
```

index.php — WebServer

```
<?php
1
2
3  $db = new mysqli('localhost', 'test', 'test', 'test');
4
5  $statement = $db->prepare('SELECT * FROM user WHERE id = ?');
6  $statement->bind_param('i', 1);
7  $statement->execute();
8
9  $result = $statement->get_result();
10
11 while ($row = mysqli_fetch_assoc($result)) {
12     print_r($row);
13 }
14
15
```

Argument #2 cannot be passed by reference

Line: 15 | PHP | Tab Size: 4

```
1 <?php
2
3 > $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 > $ids = [1,2,3];
6
7 > $statement = $db->prepare('SELECT * FROM user WHERE id IN (?, ?, ?)');
8
9 > array_unshift($ids, str_repeat('i', count($ids)));
10
11 > call_user_func_array([$statement, 'bind_result'], $ids);
12
13 > $statement->execute();
14
15 > $result = $statement->get_result();
```

```
1 <?php
2
3 > $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 > $ids = [1,2,3];
6
7 > $statement = $db->prepare('SELECT * FROM user WHERE id IN (?, ?, ?)');
8
9 > array_unshift($ids, str_repeat('i', count($ids)));
10
11 > call_user_func_array([$statement, 'bind_result'], $ids);
12
13 > $statement->execute();
14
15 > $result = $statement->get_result();
16
```

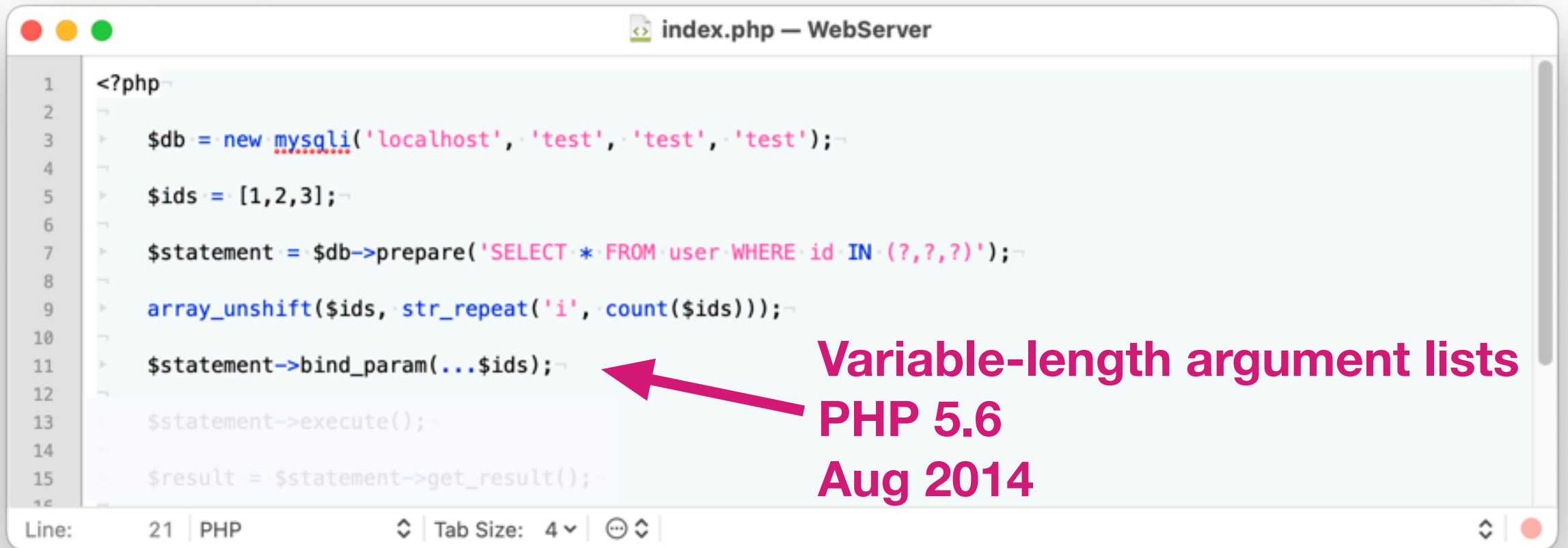
```
1 <?php
2
3 > $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 > $ids = [1,2,3];
6
7 > $statement = $db->prepare('SELECT * FROM user WHERE id IN (?, ?, ?)');
8
9 > array_unshift($ids, str_repeat('i', count($ids)));
10
11 > call_user_func_array([$statement, 'bind_result'], $ids);
12
13 > $statement->execute();
14
15 > $result = $statement->get_result();
16
```

index.php — WebServer

```
1 <?php
2
3 $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 $ids = [1,2,3];
6
7 $statement = $db->prepare('SELECT * FROM user WHERE id IN (?,?,?)');
8
9 array_unshift($ids, str_repeat('i', count($ids)));
10
11 call_user_func_array([$statement, 'bind_result'], $ids);
12
13 $statement->execute();
14
15 $result = $statement->get_result();
```

Line: 22 | PHP Tab Size: 4

Argument #1 to #4 cannot be passed by reference



```
1 <?php
2
3 > $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 > $ids = [1,2,3];
6
7 > $statement = $db->prepare('SELECT * FROM user WHERE id IN (?,?,?)');
8
9 > array_unshift($ids, str_repeat('i', count($ids)));
10
11 > $statement->bind_param(...$ids);
12
13 > $statement->execute();
14
15 > $result = $statement->get_result();
16
```

Line: 21 | PHP ⬆ | Tab Size: 4 ⌵ | ⌵ ⌵ | ⬆ ⬆

Variable-length argument lists
PHP 5.6
Aug 2014

```
1 <?php
2
3 > $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 > $ids = [1,2,3];
6
7 > $statement = $db->prepare('SELECT * FROM user WHERE id IN (?, ?, ?)');
8
9 > array_unshift($ids, str_repeat('i', count($ids)));
10
11 > $statement->bind_param(...$ids);
12
13 > $statement->execute();
14
15 > $result = $statement->get_result();
16
```

PHP: rfc:mysql_execute_param X PHP: rfc:mysql_bind_in_execu X PHP: rfc:mysql_execute_query X +

← → ↺ 🏠

wiki.php.net/rfc/mysql_execute_parameters

🔗 ☆ ⚙️ 🔒 🖨️ 👤 ⋮

php

Edit this page Admin Logout Craig Francis (craigfrancis)

Search

start › rfc › mysql_execute_parameters

PHP RFC: MySQLi Execute with Parameters

- Version: 0.1
- Date: 2020-12-26
- Author: Craig Francis, craig#at#craigfrancis.co.uk
- Status: Draft
- Published at: 🌐 https://wiki.php.net/rfc/mysql_execute_parameters

Introduction

Make *mysqli* easier to use with parameterised queries.

The Problem

Using parameterised queries with *mysqli* is too complicated:

✎ ⌚ 🔗 ✉️ ⬆️

Table of Contents

- PHP RFC: MySQLi Execute with Parameters
- Introduction
- The Problem
- Proposal
- Rough Implementation
- Backward Incompatible Changes
- Proposed PHP Version(s)
- RFC Impact
 - To SAPIs
 - To Existing Extensions
 - To Opcache
- Open Issues
- Alternatives
- Unaffected PHP Functionality
- Future Scope
- Proposed Voting Choices

Proposal

Introduce `mysqli::execute()` (and replace the deprecated `mysqli_execute()` function) to make parameterised queries much easier, e.g.

```
$sql = 'SELECT * FROM user WHERE name LIKE ? AND type = ?';

$parameters = ['%a%', 'admin'];

$result = $db->execute($sql, $parameters);

while ($row = $result->fetch_assoc()) {
    print_r($row);
}
```

PHP: rfc:mysqli_execute_ x PHP: rfc:mysqli_bind_in_ x PHP: rfc:mysqli_execute_ x PHP: mysql:execute_qu x +

← → ↺ 🏠

wiki.php.net/rfc/mysql_i_bind_in_execute

🔖 ☆ ⚙️ 🔒 🗑️ 👤 ⋮

php

Edit this page Admin Logout Craig Francis (craigfrancis)

Search

start › rfc › mysql_i_bind_in_execute

PHP RFC: mysql_i bind in execute

- Version: 1.1
- Date: 2021-02-11
- Author: Kamil Tekiela, dharman@php.net
- Target version: PHP 8.1
- Implementation: 🌐 <https://github.com/php/php-src/pull/6271>
- Status: Implemented

Introduction

PDO has always offered binding values to the prepared statement directly in the execute() call by providing an array with the values. The same functionality was never present in mysql_i, and many users have been confused by that lack of seemingly easy functionality. (See 🌐 [Bug #40891](#), 🌐 [Bug #31096](#))

✎

🕒

🔗

✉

⬆

Table of Contents

- PHP RFC: mysql_i bind in execute
- Introduction
- Proposal
- Backward Incompatible Changes
- Proposed PHP Version(s)
- RFC Impact
 - New Constants
 - php.ini Defaults
- Unaffected PHP Functionality
- Future Scope
- Proposed Voting Choices
- Implementation
- References

```
<?php
1
2
3  $db = new mysqli('localhost', 'test', 'test', 'test');
4
5  $statement = $db->prepare('SELECT * FROM user WHERE type IN (?, ?, ?)');
6  $statement->bind_param('sss', $type1, $type2, $type3);
7  $statement->execute();
8
9  $result = $statement->get_result();
10
11 while ($row = mysqli_fetch_assoc($result)) {
12     print_r($row);
13 }
14
15
```

```
1 <?php
2
3 > $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 > $statement = $db->prepare('SELECT * FROM user WHERE type IN (?, ?, ?)');
6 > $statement->execute([$type1, $type2, $type3]);
7
8 > $result = $statement->get_result();
9
10 > while($row = mysqli_fetch_assoc($result)) {
11 >     print_r($row);
12 > }
13
14
15
```

```
1 <?php
2
3 > $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 > $statement = $db->prepare('SELECT * FROM user WHERE id = ?');
6 > $statement->execute([1]);
7
8 > $result = $statement->get_result();
9
10 > while($row = mysqli_fetch_assoc($result)){
11 >     print_r($row);
12 > }
13
14
15
```


PHP: rfc:mysql_execute_ x | PHP: rfc:mysql_bind_in_ x | PHP: rfc:mysql_execute_ x | PHP: mysql::execute_qui x | +

← → ↺ 🏠

wiki.php.net/rfc/mysql_execute_query

🔖

☆

⚙️

🔥

🖨️

👤

⋮

php

Edit this page Admin Logout Craig Francis (craigfrancis)

Search

start > rfc > mysql_execute_query

PHP RFC: MySQLi Execute Query

- Version: 1
- RFC Started: 2022-04-21
- RFC Updated: 2022-05-11
- Voting Start: 2022-05-11 15:00 UTC / 16:00 BST
- Voting End: 2022-05-25 15:00 UTC / 16:00 BST
- Author: Kamil Tekiela, and Craig Francis [craig#at#craigfrancis.co.uk]
- Status: Accepted
- Target Version: PHP 8.2
- First Published at: 🌐 https://wiki.php.net/rfc/mysql_execute_query
- GitHub Repo: 🌐 <https://github.com/craigfrancis/php-mysqli-execute-query-rfc>
- Implementation: 🌐 [From Kamil Tekiela](#) (proof of concept)

Introduction

✎

🕒

🔗

✉️

⬆️

Table of Contents

- PHP RFC: MySQLi Execute Query
- Introduction
- Proposal
- Notes
 - Function Name
 - Returning false
 - Properties
 - Re-using Statements
 - Updating Existing Functions
 - Why Now
- Backward Incompatible Changes
- Proposed PHP Version(s)
- RFC Impact
 - To SAPIs
 - To Existing Extensions
 - To Opcache
 - New Constants

```
1 <?php
2
3 > $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 > $rows = $db->execute_query('SELECT * FROM user WHERE id = ?', [$_GET['id']]);
6
7 > foreach ($rows as $row) {
8 >     print_r($row);
9 > }
```

```
1 <?php
2
3 > $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 > $rows = $db->query('SELECT * FROM user WHERE id = "' . $db->real_escape_string($_GET['id']) . '"');
6
7 > foreach ($rows as $row) {
8 >     print_r($row);
9 > }
```

```
1 <?php
2
3 > $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 > $ids = [1,2,3];
6
7 > $rows = $db->execute_query('SELECT * FROM user WHERE id IN (?, ?, ?)', $ids);
8
9 > foreach ($rows as $row) {
10 >     print_r($row);
11 > }
12
13
14
15
```

```
1 <?php
2
3 > $db = new mysqli('localhost', 'test', 'test', 'test');
4
5 > $ids = [1,2,3];
6
7 > foreach ($db->execute_query('SELECT * FROM user WHERE id IN (?, ?, ?)', $ids) as $row) {
8 >     print_r($row);
9 > }
```

Thank You

Questions?

<https://eiv.dev/>

@craigfrancis

PHP: rfc:literal_string

PHP: rfc:literal_string

← → ↺ 🏠 🔒 wiki.php.net/rfc/literal_string

🔗 ☆ ⚙️ 🔔 🏠 👤 ⋮

php

Edit this page Admin Logout Craig Francis (craigfrancis)

Search

start › rfc › literal_string

PHP RFC: LiteralString

- Version: 2.0
- Voting Start: ???
- Voting End: ???
- RFC Started: 2022-12-27
- RFC Updated: 2022-12-27
- Author: Craig Francis, craig#at#craigfrancis.co.uk
- Contributors: Joe Watkins, Máté Kocsis
- Status: Draft
- First Published at: https://wiki.php.net/rfc/literal_string
- GitHub Repo: <https://github.com/craigfrancis/php-is-literal-rfc/readme-v2.md>
- Implementation: <https://github.com/php/php-src/compare/master...krakjoe:literals>

Introduction

Add *LiteralString* type, and *is_literal_string()*, to check that a variable contains a “developer defined string”.

✎

🕒

🔗

✉

⬆

Table of Contents

- PHP RFC: LiteralString
- Introduction
- The Problem
- Proposal
- Examples
- Considerations
- Performance
- String Concatenation
- String Splitting
- Frequently Asked Questions
 - FAQ: WHERE IN
 - FAQ: Non-Parameterised Values
 - FAQ: Non-LiteralString Values
 - FAQ: Bypassing It
 - FAQ: Integer Values
 - FAQ: Other Values
 - FAQ: Other Functions
 - FAQ: The Name
 - FAQ: Extensions
 - FAQ: Adoption

