

Must Be Built

It Must Be Built and This is What I Learnt



About

Posts

20/20 HTML

20/20
CSS

Javascript / jQuery

HTML5

Dreamweaver

PHP

The Useful Link
Page



RSS

Nov 17, 2011

MySQLi Cheatsheet

admin PHP, SQL 6 Comments

Project after project connecting to databases. What I need is a cheatsheet for the PHP MySQLi extension to cover the most common querying scenarios. Here we go:

Give me one record

```
1 $queryFilms = "SELECT fileName, filmDescription FROM movies WHERE filmID = 10";
2 $resultFilms = $mysqli->query($queryFilms);
3 $rowFilms = $resultFilms->fetch_assoc();
4 // then to output
5 echo "<p>{$rowFilms['fileName']}</p>";
```

Give me the whole lot

```
1 $queryFilms = "SELECT * FROM movies ORDER BY movieName";
2 $resultFilms = $mysqli->query($queryFilms);
3 while ($rowFilms = $resultFilms->fetch_assoc()) {
4     echo $rowFilms['movieName'];
5 }
```

One row from user input using prepare

When you have variables coming via \$_POST or \$_GET (\$_GET in this sample) and expect one value back.

```
1 $stmt = $mysqli->prepare("SELECT filmID, fileName FROM movies WHERE filmID = ?");
2 $stmt->bind_param('i', $_GET['filmID']);
3 $stmt->execute();
4 $stmt->bind_result($filmName);
5 $stmt->fetch();
6 $stmt->close();
7 echo $filmName;
```

Notice the use of the flags 'ssdi' in the bind_param() method and the position placeholders (?) in the prepare statement.

For a reminder as to how these relate take a look at this [interactive example](#).

Multiple rows from user input using prepare

When you have variables coming via \$_POST or \$_GET (\$_GET in this sample) and expect multiple values back.

```
1 $stmt = $mysqli->prepare("SELECT filmID, filmName, filmCertificate FROM movies WHERE filmCertificate = ? ORDER
2 BY filmName");
3 $stmt->bind_param('s', $_GET['filmCertificate']);
4 $stmt->execute();
5 $stmt->bind_result($filmID, $filmName, $filmCertificate);
6 while ($stmt->fetch()) {
7     echo "$filmName $filmCertificate";
8 }
```

Count number of rows with prepare

When you need to count the number of rows returned.

```
1 $stmt = $mysqli->prepare("SELECT filmID, filmName, filmCertificate FROM movies WHERE filmCertificate = ? ORDER
2 BY filmName");
3 $stmt->bind_param('s', $_GET['filmCertificate']);
4 $stmt->execute();
5 $stmt->bind_result($filmID, $filmName, $filmCertificate);
6 $stmt->store_result();
7 $numRows = $stmt->num_rows;
8 //optional $stmt->close();
```

INSERT

An INSERT with prepare.

```
1 $stmt = $mysqli->prepare("INSERT INTO movies(filmName,
2 filmDescription,filmImage,filmPrice,filmReview) VALUES (?, ?, ?, ?, ?)");
3 $stmt->bind_param('ssddi', $_POST['filmName'],
4 $_POST['filmDescription'],
5 $_POST['filmImage'],
6 $_POST['filmPrice'],
7 $_POST['filmReview']);
8 $stmt->execute();
9 $stmt->close();
```

UPDATE

An UPDATE with prepare.

```
1 $stmt = $mysqli->prepare("UPDATE movies SET filmName = ?,
2 filmDescription = ?,
3 filmImage = ?,
4 filmPrice = ?,
5 filmReview = ?
6 WHERE filmID = ?");
7 $stmt->bind_param('ssddii',
8 $_POST['filmName'],
9 $_POST['filmDescription'],
10 $_POST['filmImage'],
11 $_POST['filmPrice'],
```

```

12     $_POST['filmReview'],
13     $_POST['filmID']);
14     $stmt->execute();
15     $stmt->close();

```

DELETE

Finally a DELETE statement. Like the UPDATE a WHERE clause ensure the correct record is removed.

```


1     $stmt = $mysqli->prepare("DELETE FROM movies WHERE filmID = ?");
2     $stmt->bind_param('i', $_POST['filmID']);
3     $stmt->execute();
4     $stmt->close();

```

Share
this:



6 Comments



Zoey


JUL 21, 2012 @ 04:32:34

Hey thanks for this! Just saved me a lot of head banging & hair tearing 😊

I have one question tho... I am trying to debug some code I have and I want to see the final sql query that was passed to the execute statement as I am using prepared statements and using the ? placeholder. I tried var_dumping a couple of variables but it is not telling me anything relevant.

How do I go about this? Thanks 😊

REPLY



admin

JUL 21, 2012 @ 16:53:37

Hi Zoey - Thanks for your comments. You could try using the error() method of the mysqli extension. To do so place your \$mysqli->prepare in an 'if' as follows:

```

1     if($stmt = $mysqli->prepare("SELECT filmID, filmName FROM movies WHERE filmXXID = ?")){
2         $stmt->bind_param('i', $_GET['filmID']);
3         $stmt->execute();
4         $stmt->bind_result($filmName, $filmCert);
5         $stmt->fetch();
6         $stmt->close();
7     }else{
8         echo $mysqli->error;
9     }

```

REPLY

In this case you would get an error stating:

"Unknown column 'filmXXID' in 'where clause'"

... flagging up the fact that I had the wrong field name in my SQL.

Hope that helps. You raise an interesting point though and I am going to investigate other methods to use and will put up a post about it shortly.

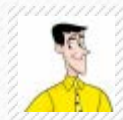
REPLY



Zoey

JUL 21, 2012 @ 20:59:28

Hey thanks for the quick response! You were right, there was an error in the sql statement. I am using the LIKE keyword and didn't put quotes around the placeholder ? (btw, is that the right thing to do? In terms of preventing things like SQL injection, because I thought that eliminating the quotes and using a placeholder is the method to prevent SQL injection)



admin

JUL 22, 2012 @ 07:28:19

Hi Zoey - you are right you don't need to put quotes around a string when using prepared statements with mysqli. Just ensure that you have the 's' flag set up in the bind_param() method. Also with a LIKE because you are using wildcards (%) then it is a idea to concatenate yourself a variable first ie:

```
1 | $stmt = $mysqli->prepare("SELECT filmID, filmName FROM movies WHERE filmName LIKE ?");
2 | //build a variable using query value and wildcards
3 | $filmNameSearch = "%".$GET['filmName']."%";
4 | $stmt->bind_param('s', $filmNameSearch)
```

Note this is a very simple example. Some security / validation against the \$_GET value is always a good idea.

REPLY

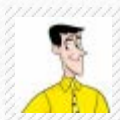


Zoey

JUL 22, 2012 @ 20:13:11

Hey thanks for the feedback... I just implemented this literally 2mins ago before I came here to check out other tips. Now everything is alright 😊 thanks a lot...

REPLY



admin

JUL 25, 2012 @ 21:22:36

Hi Zoey

Just made a new post on the [debugging issue here](#). Includes a link to a [prepare statement checker tool](#) that you might also find useful.

REPLY

Leave a Reply

Your Name *

Your Email *