

Re: DBIx::Simple variable interpolation problem

Source: <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2007-07/msg00309.html>

- *From:* Paul Lalli <mriddy@xxxxxxxxxx>
 - *Date:* Fri, 06 Jul 2007 06:20:33 -0700
-

On Jul 6, 6:29 am, Justin C <justin.0...@xxxxxxxxxxxxxxxxxx> wrote:

Yes, it's me again with more of the same... Maybe I should just say "Thank you Paul" now?

I'm tickled that you have that much faith in my powers of debugging, but please don't discount the wealth of other people in this newsgroup who can help you.

I'm querying a database, trying to use a broad query so data from almost any field can be used and all fields are searched to find that data, the search should then return the record numbers that match.

I have the following query statement:

```
my $query = $dataSource->query('SELECT key FROM prospect WHERE ? ~* ?', $field, $sc) ;
```

[http://search.cpan.org/~timb/DBI-1.58/DBI.pm#Placeholders and Bind Values](http://search.cpan.org/~timb/DBI-1.58/DBI.pm#Placeholders_and_Bind_Values)

(I realize you're using DBIx::Simple, but that's just a wrapper around DBI itself).

You can't use place holders for column names. Place holders are used for values only.

I realise you've not got the data to check this, but a full working code snippet is below

EXCELLENT!

('working' meaning that should I change the first ? to a field name then I get results). Anyway, here's the code I have:

Re: DBIx::Simple variable interpolation problem

```
#!/usr/bin/perl

use warnings ;
use strict ;
use DBIx::Simple ;

my ($dataSource, @results) ;
sub db_connect {
my ( $user, $password) = ("justin", "gobble") ;
$dataSource = DBIx::Simple->connect(
'dbi:Pg:database=prospects', $user, $password,
{ RaiseError => 1 , AutoCommit => 1 }
) or die DBIx::Simple->error ;

}

while (@ARGV) {
my $sc = pop @ARGV ; # Search criteria
my @dbFields = qw/contact co_name ad1 ad2 ad3 town county p_code country tel1 tel2/ ;
foreach my $field (@dbFields) {
db_connect();
```

URG. Why are you connecting to the database multiple times? Very very wasteful. Connect once, then run your queries multiple times.

```
my $query = $dataSource->query('SELECT key FROM prospect WHERE ? ~* ?', $field,
$sc) ;
```

Like I said, you can't use placeholders for column names. Just put the columnname directly into your SQL:

```
my $query = $dataSource->query("SELECT key FROM prospect WHERE $field
= ?", $sc);
```

Any suggestions why a field name as a variable makes this not work?

Read the URL I pasted above. The db interface needs to know the actual syntax of the SQL statement before it can be prepared. The column name is needed to validate the SQL.

Paul Lalli

.