Subject: SQLITE3: Bind parameter by name
Posted by tojocky on Wed, 12 May 2010 13:55:17 GMT
View Forum Message <> Reply to Message

Hello all,

How about to implement in SQLITE bind parameter by name? not only by index.

example of sqlite text:
"SELECT * from sqlite_master where type=table and name =:table_name"

in SQL class need to add meber:
Sql::SetParamName(String p_name, int p_index);
or add direct:
Sql::SetParam(int i, const Value& val, String p_name)
or
Sql::SetParam(String p_name, const Value& val) // auto increment parameter index
by using:
const char *sqlite3_bind_parameter_name(sqlite3_stmt*, int);

Can I add this functionality and propose for adding in library?

Any opinion are welcome!

Best regards, Ion Lupascu (tojocky)

Subject: Re: SQLITE3: Bind parameter by name
Posted by mirek on Mon, 17 May 2010 12:48:40 GMT
View Forum Message <> Reply to Message

tojocky wrote on Wed, 12 May 2010 09:55Hello all,

How about to implement in SQLITE bind parameter by name? not only by index.

example of sqlite text:
"SELECT * from sqlite_master where type=table and name =:table_name"

in SQL class need to add meber:
Sql::SetParamName(String p_name, int p_index);
or add direct:
Sql::SetParam(int i, const Value& val, String p_name)
or
Sql::SetParam(String p_name, const Value& val) // auto increment parameter index
by using:
const char *sqlite3_bind_parameter_name(sqlite3_stmt*, int);

Can I add this functionality and propose for adding in library?

Any opinion are welcome!

Best regards, Ion Lupascu (tojocky)

As SqlExp seems to be dominant choice for doing SQL, I see little benefit here. Also, not every DB engine supports this and you would require a common interface.

I am strongly opposed to this.

Mirek

Subject: Re: SQLITE3: Bind parameter by name
Posted by tojocky on Tue, 18 May 2010 05:22:14 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 17 May 2010 15:48tojocky wrote on Wed, 12 May 2010 09:55Hello all,

How about to implement in SQLITE bind parameter by name? not only by index.

example of sqlite text:
"SELECT * from sqlite_master where type=table and name =:table_name"

in SQL class need to add meber:
 Sql::SetParamName(String p_name, int p_index);
or add direct:
 Sql::SetParam(int i, const Value& val, String p_name)
or
 Sql::SetParam(String p_name, const Value& val) // auto increment parameter index
by using:
const char *sqlite3_bind_parameter_name(sqlite3_stmt*, int);

Can I add this functionality and propose for adding in library?

Any opinion are welcome!

Best regards, Ion Lupascu (tojocky)

As SqlExp seems to be dominant choice for doing SQL, I see little benefit here. Also, not every DB engine supports this and you would require a common interface.

I am strongly opposed to this.

Mirek

I use classic version of Sql because: in a function I can construct the sql text, but in other I can pass the parameters.

Parameters support SQLITE, PostgreSQL, Oracle. Not sure about MySQL.
Other databases?

I can do the common interface and propose for acceptance.

Best regards, Ion Lupascu (tojocky)

Subject: Re: SQLITE3: Bind parameter by name
Posted by mirek on Tue, 18 May 2010 07:26:28 GMT
View Forum Message <> Reply to Message

tojocky wrote on Tue, 18 May 2010 01:22luzr wrote on Mon, 17 May 2010 15:48tojocky wrote on Wed, 12 May 2010 09:55Hello all,

How about to implement in SQLITE bind parameter by name? not only by index.

example of sqlite text:
"SELECT * from sqlite_master where type=table and name =:table_name"

in SQL class need to add meber:
Sql::SetParamName(String p_name, int p_index);
or add direct:
Sql::SetParam(int i, const Value& val, String p_name)
or
Sql::SetParam(String p_name, const Value& val) // auto increment parameter index
by using:
const char *sqlite3_bind_parameter_name(sqlite3_stmt*, int);

Can I add this functionality and propose for adding in library?

Any opinion are welcome!

Best regards, Ion Lupascu (tojocky)

As SqlExp seems to be dominant choice for doing SQL, I see little benefit here. Also, not every DB engine supports this and you would require a common interface.

I am strongly opposed to this.

Mirek

I use classic version of Sql because: in a function I can construct the sql text, but in other I can pass the parameters.

Well, I see the point.

Anyway, for constructing SQL, we have designed SqlExp

BTW, cannot you just pass parameters to the function where you construct the SQL? In the past 10 years, that worked quite well for me (using SqlExp)

Quote:
Parameters support SQLITE, PostgreSQL, Oracle. Not sure about MySQL.

MySQL does not support it.

http://dev.mysql.com/doc/refman/5.0/en/mysql-stmt-prepare.ht ml

I also believe that PGSQL does not support it too.

BTW, with litte effort, you can add this feature "externally", using only parameter positions.

All you need to do is parse the SQL command, replace your named parameter placeholders with "?" and create the Index of names in the process, then simply translate parameter name to position... Much simpler than patching all SQL clients...

Mirek

Subject: Re: SQLITE3: Bind parameter by name
Posted by tojocky on Tue, 18 May 2010 15:27:42 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 18 May 2010 10:26

Well, I see the point.

Anyway, for constructing SQL, we have designed SqlExp

BTW, cannot you just pass parameters to the function where you construct the SQL? In the past 10 years, that worked quite well for me (using SqlExp)

Quote:
Parameters support SQLITE, PostgreSQL, Oracle. Not sure about MySQL.

MySQL does not support it.

http://dev.mysql.com/doc/refman/5.0/en/mysql-stmt-prepare.ht ml

I also believe that PGSQL does not support it too.

BTW, with litte effort, you can add this feature "externally", using only parameter positions.

All you need to do is parse the SQL command, replace your named parameter placeholders with "?" and create the Index of names in the process, then simply translate parameter name to position... Much simpler than patching all SQL clients...

Mirek

Thank you Mirek for hint!

Subject: Re: SQLITE3: Bind parameter by name
Posted by frankdeprins on Fri, 21 May 2010 07:06:18 GMT
View Forum Message <> Reply to Message

Hello,

I have a question related to this as well.
I am creating a database tool for sqlite where the user can launch arbitrary statements and, before executing them, I have to check whether there are bind variables (and how many).
As you say, the interface of the Sql class is quite general and this seems to be lacking.
I tried the ParseForArgs function by copying it in my sources, but it is not really fool proof, as it does not take comments in the statement in account.
e.g. the function returns 1 for this statement:

```
select
  field -- just some field or not?
from
  table
```

although the '?' in this case is not really a bind variable.

Any suggestions?

frank

Subject: Re: SQLITE3: Bind parameter by name
Posted by tojocky on Fri, 21 May 2010 10:31:39 GMT
View Forum Message <> Reply to Message

frankdeprins wrote on Fri, 21 May 2010 10:06Hello,

I have a question related to this as well.
I am creating a database tool for sqlite where the user can launch arbitrary statements and, before executing them, I have to check whether there are bind variables (and how many).
As you say, the interface of the Sql class is quite general and this seems to be lacking.

I tried the ParseForArgs function by copying it in my sources, but it is not really fool proof, as it does not take comments in the statement in account.
e.g. the function returns 1 for this statement:
```
select
  field -- just some field or not?
from
  table
```
although the '?' in this case is not really a bind variable.

Any suggestions?

frank

To be sure, remove comments form sql statement!

Subject: Re: SQLITE3: Bind parameter by name
Posted by frankdeprins on Fri, 21 May 2010 11:36:06 GMT
View Forum Message <> Reply to Message

Yes, but when a user of the tool posts a statement, there can be comments in it. You have no control over what he posts as statement. Of course you can programatically remove the comments, but that is not always OK. Sometimes, with DDL statements, you want the comments in the database. I do this often so that the comments are available when I ask the creation statement of an object in sqlite. See it as a form of database documentation.
Anyway, I think that any statement that is valid for sqlite, should be accepted by the Ultimate wrapper as well, without mistaking question marks as bind variables when they aren't.
