
Subject: sql session in a global variable?

Posted by [Wolfgang](#) on Tue, 13 Dec 2011 21:55:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

I have a method in which my sqlite session is being connected:

```
.h
class sqltry
{
public:
    typedef sqltry CLASSNAME;
    sqltry();
private:
    Sqlite3Session sqlite3;
    Sql sqlPointer;

.cpp
bool sqltry::connect(String path)
{
    if (!sqlite3.Open(path))
    {
        return false;
    }
    Sql sql(sqlite3);
    sqlPointer = sql;
}
```

But if i try to use the sqlPointer in another method of the same class my app stops working and crashes...

```
.cpp
bool sqltry::atest(S)
{
    if (sqlPointer.Execute("create table t2 (t1key INTEGER PRIMARY KEY,data TEXT,num
double,timeEnter DATE);"))
        PromptOK("DONE!");
    return true;
}
```

Whats wrong with it?

Subject: Re: sql session in a global variable?

Posted by [mr_ped](#) on Wed, 14 Dec 2011 09:29:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

Didn't try the sources, so just by quick reading it...

The "sql" in connect procedure is local variable, so it gets destroyed once it gets out of scope. I'm afraid this is making it's content invalid. The assignment to "sqlPointer", unless the Sql class has very clever copy constructor, will not help to preserve it. I'm too lazy to check actual Sql source, sorry.

Anyway, I don't like the "sqlPointer" variable name, because it is not pointer, but an object, so the variable name is misleading.

I would suggest to completely remove sqlPointer class member variable, and create new instance of Sql every time you need it.

See [http://www.ultimatepp.org/srcdoc\\$Sql\\$tutorial\\$en-us.html](http://www.ultimatepp.org/srcdocSqltutorial$en-us.html) , especially step 2 may help you. After you set global "SQL", you can then get new instance by simple "Sql sql;" in your code.

Subject: Re: sql session in a global variable?

Posted by [Wolfgang](#) on Wed, 14 Dec 2011 09:33:05 GMT

[View Forum Message](#) <> [Reply to Message](#)

you're right, the "pointer" in var name is misleading... was just my first thought that is is one.

I know about the global SQL but this is it, i want to avoid - I would prefer a variable defined by myself because maybe i need more than one database connection.. and so i tried to handle it.

Subject: Re: sql session in a global variable?

Posted by [mr_ped](#) on Wed, 14 Dec 2011 10:03:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

well, after reading that tutorial myself I think your first version may work if you change connect:

```
bool sqltry::connect(String path)
{
    if (!sqlite3.Open(path))
    {
        return false;
    }
    sqlPointer = sqlite3;
}
```

And of course rename that sqlPointer to something more verbose, like

"someDatabaseForThisAndThatSql", so when you read it two years later, you will get idea what is it good for.

(to get pointers you would need to start to use "*", "&" and new/delete where appropriate. I don't

believe you want to do that in this case.

edit - addition:

you can of course use fresh instances as needed even with multiple databases. I would maybe put the most used one into SQL, so then Sql instances for that one would be "Sql sql;", Sql instances for other databases would be created like "Sql sql(other_db_session);"

Subject: Re: sql session in a global variable?

Posted by [mirek](#) on Thu, 15 Dec 2011 07:29:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Ehm, I have created a reply here, but it seems I forgot to hit "Send".

Well, the problem here is that you cannot assign Sql nor SqlSession. It was a minor bug that copy constructor and operator= were not made private.

So what you can now is something like:

```
class sqltry
{
public:
    typedef sqltry CLASSNAME;
    sqltry();
private:
    Sqlite3Session sqlite3;
    One<Sql> sql;
    Sql& SQL() {
        if(!sql)
            sql = new Sql(sqlite3);
        return *sql;
    }
}
```

...and then use SQL()...

Subject: Re: sql session in a global variable?

Posted by [Wolfgang](#) on Thu, 15 Dec 2011 08:51:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

I see what u mean but upp tells me

Quote:e:\ultimate++\myapps\urlaub\SQL.h(25) : error C2091: function returns function

e:\ultimate++\myapps\urlaub\SQL.h(27) : error C2440: 'return' : cannot convert from 'Upp::Sql' to

```
'Upp::Sql &(__cdecl *)(void)'
```

```
Sqlite3Session sqlSession;  
One<Sql> sql;  
Sql& SQL()  
{  
    if(!sql) sql = new Sql(sqlSession);  
    return *sql;  
}
```

Line 27 is return *sql;

Subject: Re: sql session in a global variable?
Posted by [mirek](#) on Thu, 15 Dec 2011 10:23:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Sorry, name clash with SQL define, rename to e.g. ISQL.

Subject: Re: sql session in a global variable?
Posted by [Wolfgang](#) on Thu, 15 Dec 2011 10:26:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

ok... but after all the thinking about a good design for such a sqlite class i decided not to write one. The use of sqlite is such simple with the existing sqlite class in upp that there is no need for an additional class.

Sorry for the trouble i caused!

Subject: Re: sql session in a global variable?
Posted by [mirek](#) on Thu, 15 Dec 2011 10:56:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Anyway, based on this conversation, I have found to be a good (and simple to implement) idea to have Sql::SetSession method.

So your example would be now basically possible to write as:

```
struct SqlTry {  
    Sqlite3Session sqlite3;  
    Sql sql;  
  
    bool connect(String path)  
    {
```

```
if (!sqlite3.Open(path))
{
    return false;
}
    sql.SetSession(sqlite3);
}
```

Subject: Re: sql session in a global variable?
Posted by [Wolfgang](#) on Thu, 15 Dec 2011 10:58:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

and then, is Sql sql usable in the whole class after this?

Subject: Re: sql session in a global variable?
Posted by [mirek](#) on Thu, 15 Dec 2011 11:49:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

"whole class" is somewhat strange definition, but "yes".

Further development: Perhaps rather use Sql0 instead of Sql... (both should work, but Sql might create some DB connection to global SQL session that is then deleted in SetSession).

Subject: Re: sql session in a global variable?
Posted by [Alboni](#) on Thu, 12 Sep 2013 20:14:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Thu, 15 December 2011 11:56 Anyway, based on this conversation, I have found to be a good (and simple to implement) idea to have Sql::SetSession method.

So your example would be now basically possible to write as:

```
[code]
.....<snip>
    sql.SetSession(sqlite3);
```

But..... Sql::SetSession is a private member.....

Subject: Re: sql session in a global variable?
Posted by [coolman](#) on Fri, 12 Jun 2020 20:41:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I know this is old thread but why the SetSession() is private?

Thanks, Radek

Subject: Re: sql session in a global variable?
Posted by [mirek](#) on Sun, 14 Jun 2020 08:18:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

coolman wrote on Fri, 12 June 2020 22:41Hi,

I know this is old thread but why the SetSession() is private?

Thanks, Radek

Because it might be difficult / impossible to do such change for Sql when some operations are in progress. Or it seemed that way at the time.

Mirek
