Subject: Some Sql addidtions
Posted by unodgs on Thu, 26 Jun 2008 12:56:09 GMT
View Forum Message <> Reply to Message

Recently I made possible to write select in select fields list:

```
SQL * ::Select(
   ID,
   NAME,
   ::Select(NAME, AGE).From(PARENTS).Where(KID_ID == ID),
   PHONE
).From(KIDS)
```

..and conditions

```
SQL * ::Select(
  X == Y,
  NUMBER < 100,
).From(TABLE)
```

SqlNvl has now As() method so it's possible to write:

```
SQL * ::Select(
   (QUANTITY * PRICE).As(FINAL_PRICE)
   (NAME | SURNAME).As(FULL_NAME)
).From(PRICES)
```

It's also possible to write cases:

```
SQL * ::Select(
  ID,
  Case(AGE < 5, "Baby")
      (Between(AGE, 5, 10), "Kid")
      ("Young man"),
  NAME
.From(PERSONS)
```

Subject: Re: Some Sql addidtions
Posted by zsolt on Thu, 26 Jun 2008 18:13:01 GMT
View Forum Message <> Reply to Message

This is very-very useful, thanks!

Subject: Re: Some Sql addidtions
Posted by tojocky on Fri, 27 Jun 2008 21:16:45 GMT
View Forum Message <> Reply to Message

How about to write SQL queries as a simple text?
for example:
Quote:Query *q;
q->SetText ("SELECT SomeColumns FROM SomeTable WHERE SomeCondition = &SomeParameter GROUP BY SomeGoups HAVING SomeHavingCondition...")?
and after then we can set parameter value;
Quote:q->SetParameter ("SomeParameter", parameter_value);

after then we can do:

Quote:QResult *q_result = q->Execute ();

QChoose q_chouse = q_result->Choose ();

while (q_chouse->Next ()) {...}

or

Quote:Array q_res = q->Execute ()->Unload ();

How about thi idea? I thing that this is more simple... and will be possible for sent to an server application only query text!

Subject: Re: Some Sql addidtions
Posted by zsolt on Sat, 28 Jun 2008 08:25:35 GMT
View Forum Message <> Reply to Message

Sorry, but Ultimate++ is much better, than your suggestion

Just read "SQL programming" section in Overview.

Subject: Re: Some Sql addidtions
Posted by cbpporter on Sat, 28 Jun 2008 10:18:10 GMT
View Forum Message <> Reply to Message

If you need something like that, check out mysql++.

Subject: Re: Some Sql addidtions
Posted by mirek on Sat, 28 Jun 2008 11:56:54 GMT
View Forum Message <> Reply to Message

tojocky wrote on Fri, 27 June 2008 17:16How about to write SQL queries as a simple text?
for example:
Quote:Query *q;
q->SetText ("SELECT SomeColumns FROM SomeTable WHERE SomeCondition = &SomeParameter GROUP BY SomeGoups HAVING SomeHavingCondition...")?
and after then we can set parameter value;
Quote:q->SetParameter ("SomeParameter", parameter_value);

after then we can do:

Quote:QResult *q_result = q->Execute ();

QChoose q_chouse = q_result->Choose ();

while (q_chouse->Next ()) {...}

or

Quote:Array q_res = q->Execute ()->Unload ();

How about thi idea? I thing that this is more simple... and will be possible for sent to an server application only query text!

You can. That is U++'s "low-level".

E.g.

Sql sql;
sql.Execute("select foo from bar");
....

Mirek

Subject: Re: Some Sql addidtions
Posted by tojocky on Sat, 28 Jun 2008 21:30:02 GMT
View Forum Message <> Reply to Message

Thank you Mirek! I like this "low level" query!

I tried to compile example package and get folow error:

Quote:C:\uppdev\examples\SQLApp\query.cpp(17) : error C2593: 'operator ==' is ambiguous
c:\uppdev\uppsrc\sql\Sqlexp.h(346): could be 'Upp::SqlBool Upp::operator ==(const Upp::SqlVal &,const Upp::SqlSet &)'
c:\uppdev\uppsrc\sql\Sqlexp.h(308): or 'Upp::SqlBool Upp::operator ==(const Upp::SqlVal &,const Upp::SqlVal &)'

while trying to match the argument list '(Upp::SqlId, Upp::SqlSelect)'

Subject: Re: Some Sql addidtions
Posted by bytefield on Sat, 28 Jun 2008 22:46:58 GMT
View Forum Message <> Reply to Message

Well, maybe someone should do a more consistent tutorial on how is made SQL interface in upp, what you can do with and how to do, because the Overview is a bit unclear in explaining SQL stuff. And what about those macros (TABLE, INT, INT_), where exist an explication on how to use SQL schema with upp because examples are complicated (them use SQL without any explanation) or them don't use all features of SQL. I bet that upp don't implement all SQL stuff defined by standard so sometimes is good to use "low-level interface" specially when you don't have e good explanation of "high-level interface" or when you have to do your work in the hard way . So will be there an experienced upp user and a well-wisher which will do a tutorial or you think that the hard way of learning something is always the best?
Without any annoyance guys, that's my opinion.

Andrei

Subject: Re: Some Sql addidtions
Posted by mirek on Sun, 29 Jun 2008 16:15:16 GMT
View Forum Message <> Reply to Message

tojocky wrote on Sat, 28 June 2008 17:30Thank you Mirek! I like this "low level" query!

I tried to compile example package and get folow error:

Quote:C:\uppdev\examples\SQLApp\query.cpp(17) : error C2593: 'operator ==' is ambiguous
c:\uppdev\uppsrc\sql\Sqlexp.h(346): could be 'Upp::SqlBool Upp::operator ==(const Upp::SqlVal &,const Upp::SqlSet &)'
c:\uppdev\uppsrc\sql\Sqlexp.h(308): or 'Upp::SqlBool Upp::operator ==(const Upp::SqlVal &,const Upp::SqlVal &)'
while trying to match the argument list '(Upp::SqlId, Upp::SqlSelect)'

Please, when posting a problem like this, mention the compiler (and platform, but it is obvious here

Mirek

Subject: Re: Some Sql addidtions
Posted by tojocky on Sun, 29 Jun 2008 19:03:08 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 29 June 2008 19:15tojocky wrote on Sat, 28 June 2008 17:30Thank you Mirek!
I like this "low level" query!

I tried to compile example package and get folow error:

Quote:C:\uppdev\examples\SQLApp\query.cpp(17) : error C2593: 'operator ==' is ambiguous
c:\uppdev\uppsrc\sql\Sqlexp.h(346): could be 'Upp::SqlBool Upp::operator ==(const Upp::SqlVal &,const Upp::SqlSet &)'
c:\uppdev\uppsrc\sql\Sqlexp.h(308): or 'Upp::SqlBool Upp::operator ==(const Upp::SqlVal &,const Upp::SqlVal &)'
while trying to match the argument list '(Upp::SqlId, Upp::SqlSelect)'

Please, when posting a problem like this, mention the compiler (and platform, but it is obvious here

Mirek
Sorry!

platform is Win32 xp, Compiler is MSC8 optimal!

Subject: Re: Some Sql addidtions
Posted by mirek on Mon, 30 Jun 2008 07:15:36 GMT
View Forum Message <> Reply to Message

unodgs wrote on Thu, 26 June 2008 08:56Recently I made possible to write select in select fields list:

```
SQL * ::Select(
   ID,
   NAME,
   ::Select(NAME, AGE).From(PARENTS).Where(KID_ID == ID),
   PHONE
).From(KIDS)
```

..and conditions

```
SQL * ::Select(
  X == Y,
  NUMBER < 100,
).From(TABLE)
```

SqlNvl has now As() method so it's possible to write:

```
SQL * ::Select(
   (QUANTITY * PRICE).As(FINAL_PRICE)
   (NAME | SURNAME).As(FULL_NAME)
).From(PRICES)
```

It's also possible to write cases:

```
SQL * ::Select(
  ID,
  Case(AGE < 5, "Baby")
      (Between(AGE, 5, 10), "Kid")
      ("Young man"),
  NAME
.From(PERSONS)
```

Well, you have broken SqlExp in the process

```
/*
SqlVal::SqlVal(const SqlSelect& x) {
 SetHigh('(' + ((SqlStatement) x).GetText() + ')');
}

SqlVal::SqlVal(const SqlBool& x) {
 SetHigh(~x);
}
*/
```

I had to comment these, otherwise I am getting a lot of errors like the one tojocky mentions.

Mirek

Subject: Re: Some Sql addidtions
Posted by mirek on Mon, 30 Jun 2008 07:36:49 GMT
View Forum Message <> Reply to Message

Well, to avoid this situation in the future, I have moved my "SqlExp documentation examples" package to reference/SqlExp.

This should have two nice purposes:

When you add something new to SqlExp, add example of it to this package. And, by compiling it, there should be a bit of safeguarding against these kinds of bugs...

Mirek

Subject: Re: Some Sql addidtions
Posted by unodgs on Mon, 30 Jun 2008 10:05:57 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 30 June 2008 03:36Well, to avoid this situation in the future, I have moved my "SqlExp documentation examples" package to reference/SqlExp.

This should have two nice purposes:

When you add something new to SqlExp, add example of it to this package. And, by compiling it, there should be a bit of safeguarding against these kinds of bugs...

Mirek
Good idea. I'll try to fix that (In my app there are so many sqls so I thought I hadn't broken anything..)

Subject: Re: Some Sql addidtions
Posted by unodgs on Mon, 30 Jun 2008 19:04:26 GMT
View Forum Message <> Reply to Message

Another solution (at least for SqlApp) is to comment

inline SqlBool operator==(const SqlVal& a, const SqlSet& b) { return In(a, b); }
inline SqlBool operator!=(const SqlVal& a, const SqlSet& b) { return NotIn(a, b); }

and using directly In or NotIn instead of SqlVal !=/== SqlSet. IMO it's more readable, but for now I'll leave it as it is. I'll try to find solution without compromise.

Subject: Re: Some Sql addidtions
Posted by mirek on Mon, 30 Jun 2008 19:31:06 GMT
View Forum Message <> Reply to Message

unodgs wrote on Mon, 30 June 2008 15:04and using directly In or NotIn instead of SqlVal !=/== SqlSet. IMO it's more readable, but for now I'll leave it as it is. I'll try to find solution without compromise.

Well, that would lead to about 1000 errors in my code

Not an option!

Besides, != and == are really quite intuitive. Tom introduced them originally and I resisted... tried to use In / NotIn but than catched myself thinking in "==" "!=" terms (is there any value in the set equal?)

Mirek