Subject: SqlArray Limit and Offset
Posted by zsolt on Thu, 10 Aug 2006 23:03:25 GMT
View Forum Message <> Reply to Message

I have implemented it. I don't know if this is a good interface or not, but I hope so.

SqlArray declaration private section (SqlCtrl.h):

```
unsigned int limit;
bool       limit_set;
int64      offset;
bool       offset_set;
```

SqlArray declaration public section:

```
SqlArray& SetLimit(unsigned int _limit)          { limit = _limit; limit_set = true; return *this; }
SqlArray& SetOffset(int64 _offset)              { offset = _offset; offset_set = true; return *this; }
```

In SqlArray::SqlArray() and SqlArray::Clear()

```
limit_set = false;
offset_set = false;
```

In void SqlArray::AppendQuery(SqlBool where)
I have changed

```
 sql * ::Select(cols).From(table).Where(wh).OrderBy(orderby);
```

to

```
 SqlSet set = ::Select(cols).From(table).Where(wh).OrderBy(orderby);
 if(limit_set)
  if(offset_set) set.Limit(offset, limit);
  else set.Limit(limit);
 else
  if(offset_set) set.Offset(offset);
 sql * set;
```

Subject: Re: SqlArray Limit and Offset
Posted by mirek on Fri, 11 Aug 2006 06:35:42 GMT
View Forum Message <> Reply to Message

zsolt wrote on Thu, 10 August 2006 19:03I have implemented it. I don't know if this is a good

interface or not, but I hope so.

I do not like it too much, but I am unable to invent anything better

Well, in fact, what I would like to have is to make this compatible with Oracle somehow. Well, I believe, with "Limit" it is simple - ignoring in oracle is all we need.

Mirek

Subject: Re: SqlArray Limit and Offset
Posted by mirek on Fri, 11 Aug 2006 07:05:13 GMT
View Forum Message <> Reply to Message

http://www.petefreitag.com/item/451.cfm

What about joining Limit and Offset into single command (in SqlExp)? You are unlikely to use Offset without the Limit...

E.g. Rows(offset, number) and Rows(number) overloads?

Select(FOO).From(BAR).Rows(100, 10);

Mirek

Subject: Re: SqlArray Limit and Offset
Posted by zsolt on Fri, 11 Aug 2006 07:41:47 GMT
View Forum Message <> Reply to Message

Quote:Well, in fact, what I would like to have is to make this compatible with Oracle somehow. Well, I believe, with "Limit" it is simple - ignoring in oracle is all we need.
I don't think, ignorance is a good idea. It should be implemented on client side somehow. Maybe that Rows() method should not use the internal text variable, only some internal limit and offset variables. Sql class should put together a statement or implement limit/offset on client side based on SqlExp's internal limit/offset variables and Database capabilities.

Subject: Re: SqlArray Limit and Offset
Posted by zsolt on Fri, 11 Aug 2006 07:50:28 GMT
View Forum Message <> Reply to Message

An other idea:
I think Limit(offset, limit) and Limit(limit) methods would be better than Rows() methods, because Limit() shows more cleanly what it's purpose is.

Subject: Re: SqlArray Limit and Offset
Posted by rbmatt on Fri, 11 Aug 2006 16:36:10 GMT
View Forum Message <> Reply to Message

zsolt wrote on Fri, 11 August 2006 03:50An other idea:
I think Limit(offset, limit) and Limit(limit) methods would be better than Rows() methods, because Limit() shows more cleanly what it's purpose is.
I agree that this nomenclature is better. I know what it means to limit an sql query, but I would have no clue what Rows() would do.

Subject: Re: SqlArray Limit and Offset
Posted by mirek on Fri, 11 Aug 2006 16:44:08 GMT
View Forum Message <> Reply to Message

OK!

Mirek

Subject: Re: SqlArray Limit and Offset
Posted by mirek on Fri, 11 Aug 2006 21:47:19 GMT
View Forum Message <> Reply to Message

Added (not quite tested

I will defer dialect-independent Limit to new SqlExp.

Mirek

Subject: Re: SqlArray Limit and Offset
Posted by zsolt on Tue, 22 Aug 2006 12:10:13 GMT
View Forum Message <> Reply to Message

Some fixes:

1. changed SqlArray::Limit(int offset, int count); in SqlCtrl.h to
```
 void      Limit(int _offset, int _count)              { offset = _offset; count = _count; }
```

The definition was missing.

2. changed a line in void SqlArray::AppendQuery(SqlBool where) method from
sql * ::Select(cols).From(table).Where(wh).Limit(count).Offset(offset).OrderBy(orderby);
to
sql * ::Select(cols).From(table).Where(wh).OrderBy(orderby).Limit(count).Offset(offset);
LIMIT and OFFSET have to be at the end of the SQL statement.

Subject: Re: SqlArray Limit and Offset
Posted by mirek on Tue, 22 Aug 2006 13:21:02 GMT
View Forum Message <> Reply to Message

Applies (hopefully correctly this time

Mirek

Subject: Re: SqlArray Limit and Offset
Posted by fabio on Wed, 20 Dec 2006 15:49:59 GMT
View Forum Message <> Reply to Message

In oracle limit and offset not work.....

Is very hard implement oracle dialect version of .Limit() and .Offset() why original query be bound to encapsulated in another select.
.....
SELECT * FROM (original query....) WHERE ROWNUM BETWEEN my_offset AND my_offset+my_limit;
.....

In SqlArray is possible to test Dialect and make another query (2 nested select) only for oracle dialect.

Any other idea ?

Fabio

Subject: Re: SqlArray Limit and Offset
Posted by mirek on Wed, 20 Dec 2006 20:00:37 GMT
View Forum Message <> Reply to Message

fabio wrote on Wed, 20 December 2006 10:49In oracle limit and offset not work.....

Is very hard implement oracle dialect version of .Limit() and .Offset() why original query be bound to encapsulated in another select.
.....
SELECT * FROM (original query....) WHERE ROWNUM BETWEEN my_offset AND my_offset+my_limit;
.....

In SqlArray is possible to test Dialect and make another query (2 nested select) only for oracle dialect.

Any other idea ?

Actually, dialects are gone. U++ is now able to "compile" SqlExp for each rdbms (well, able in sense that core of technology is there, but not all variants are implemented).

Anyway, in this case, it is indeed a problem because Oracle lacks this clause.

Mirek

Subject: Re: SqlArray Limit and Offset
Posted by fabio on Thu, 21 Dec 2006 14:17:57 GMT
View Forum Message <> Reply to Message

The 2 level query work only for first time....

the solution is (from http://www.oracle.com/technology/oramag/oracle/06-sep/o56ask tom.html)

... 3 level select.....

```
select * from ( select a.*, ROWNUM as row_num from (
select ..... from .... where ..... order by .....
) a where rownum <= offset+limit) where row_num > offset
```

Upp working solution is ...

1) Change in file: upp\uppsrc\sql\SqlCtrl\SqlArray.cpp

SqlArray::AppendQuery() method:

```
sql * ::Select(cols).From(table).Where(wh).OrderBy(orderby).Limit(count).Offset(offset);
```

in:

```
sql * ::Select(cols).From(table).Where(wh).OrderBy(orderby).Limit(offset,count);
```

2) Change in file: upp\uppsrc\sql\Sql\SqlStatement.cpp

```
SqlSelect& SqlSelect::Limit(int64 offset, int limit) {

   text = SqlCase(ORACLE,"select * from ( select rownum_a.*, ROWNUM as row_num from ( ")()
+ text +
        SqlCase(ORACLE,Format(") rownum_a where rownum <= %d ) where row_num > %d",
```

```
offset+limit,offset))
            (Format(" limit %d, %d",offset,limit));

 return *this;
}
```

this query use rownum reserved word, row_num, rownum_a user word and generate an extra column row_num in the result.

3) optional ?
 Remove SqlSelect::Limit(count)
 Remove SqlSelect::Offset(offset)
 for complete portability

Tested in oracle and sqlite, my be work in MySql (same syntax of sqlite).

Fabio

File Attachments

1) limit.zip, downloaded 520 times

Subject: Re: SqlArray Limit and Offset
Posted by mirek on Thu, 21 Dec 2006 20:49:39 GMT
View Forum Message <> Reply to Message

Should not we rather store limit/offset and use it when final query is generated?

Mirek

Subject: Re: SqlArray Limit and Offset
Posted by fabio on Sat, 23 Dec 2006 09:01:49 GMT
View Forum Message <> Reply to Message

I am sorry for delay.....

Limit(offset,limit) is for actual (complete) query, and work only if used at end of query.

Why store for future use?

I think not possble implement multi-level "limit" in oracle.
This Oracle eccentric method require 2 extra query with extra column (rownum as row_num1...) with unique name. It is a hard work.

I can store not directly in "text" of SqlSelect class, but in new two string like "pre_text" and "post_text" (as in my first only-thinked version of Limit() method), but it also require an adjustment of SqlSelect class.

..... i am lazy .....

Fabio

Subject: Re: SqlArray Limit and Offset
Posted by mirek on Sat, 23 Dec 2006 11:55:23 GMT
View Forum Message <> Reply to Message

fabio wrote on Sat, 23 December 2006 04:01I am sorry for delay.....

Limit(offset,limit) is for actual (complete) query, and work only if used at end of query.

Why store for future use?

I think not possble implement multi-level "limit" in oracle.
This Oracle eccentric method require 2 extra query with extra column (rownum as row_num1...) with unique name. It is a hard work.

I can store not directly in "text" of SqlSelect class, but in new two string like "pre_text" and "post_text" (as in my first only-thinked version of Limit() method), but it also require an adjustment of SqlSelect class.

..... i am lazy .....

Well, thinking about it even more...

Maybe we just stop at the fact that Oracle lacks support for these.

In fact, Oracle/U++ is may daily bread and I never really needed offset/limit. Somehow that is not how Oracle is used.

Mirek

Subject: Re: SqlArray Limit and Offset
Posted by fabio on Wed, 27 Dec 2006 16:09:33 GMT
View Forum Message <> Reply to Message

You can implement "limit and offset" at fetch level.
Many objects (Java, VB...) fetch and discard rows, simulating (one-way) limit and offset.....

In oracle 10g (and 9) is present a new version of OCIStmtFetch for make a clean limit solution:

OCIStmtFetch2(stmthp, errhp, 1, OCI_FETCH_RELATIVE, fetchOffset, OCI_DEFAULT);

This fetches a row from the (scrollable) result set.

But, this oci function, and scrollable cursor feacture, is not present in oci8.

Fabio

Subject: Re: SqlArray Limit and Offset
Posted by mirek on Sun, 31 Dec 2006 08:27:58 GMT
View Forum Message <> Reply to Message

Well, I am still a bit puzzled about what is limit/offset useful for... (Must be all these years with Oracle 7)

Mirek
