Subject: SqlExp and LIMIT
Posted by mezise on Sun, 30 Apr 2006 20:58:58 GMT
View Forum Message <> Reply to Message

Code from reference/SQL_Sqlite3/simple.cpp:

```
 // Test selection:
 sql*Select(row).From(SIMPLE_TEST1);
```

How to apply "LIMIT 15" clause?

There is not something like this:

```
 // Test selection:
 sql*Select(row).From(SIMPLE_TEST1).Limit(15);
```

Michal

Subject: Re: SqlExp and LIMIT
Posted by mirek on Sun, 30 Apr 2006 21:01:01 GMT
View Forum Message <> Reply to Message

mezise wrote on Sun, 30 April 2006 16:58Code from reference/SQL_Sqlite3/simple.cpp:

```
 // Test selection:
 sql*Select(row).From(SIMPLE_TEST1);
```

How to apply "LIMIT 15" clause?

There is not something like this:

```
 // Test selection:
 sql*Select(row).From(SIMPLE_TEST1).Limit(15);
```

Michal

Not yet. To put it straight, so far SqlExp was designed for Oracle and will need further tweaks for other SQL engines.

Mirek

Subject: Re: SqlExp and LIMIT
Posted by mezise on Mon, 01 May 2006 01:06:17 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 30 April 2006 17:01Not yet. To put it straight, so far SqlExp was designed for Oracle and will need further tweaks for other SQL engines.

Mirek

OFFSET-LIMIT clause is quite necessary in MySQL, SQLite and PostgreSQL.
So, could this code be put into next dev5 release?

into Sqlexp.h:

```
 SqlSet&         Limit(const unsigned int limit);
 SqlSet&         Limit(const unsigned int offset, const unsigned int limit);
 SqlSet&         Offset(const unsigned int offset);
```

into Sqlexp.cpp:

```
SqlSet& SqlSet::Limit(const unsigned int limit) {
 text << " limit " << limit;
 return *this;
}

SqlSet& SqlSet::Limit(const unsigned int offset, const unsigned int limit) {
 text << " limit " << offset << ", " << limit;
 return *this;
}

SqlSet& SqlSet::Offset(const unsigned int offset) {
 text << " offset " << offset;
 return *this;
}
```

After OrderBy it would be the right place.

Michal

Subject: Re: SqlExp and LIMIT
Posted by mirek on Mon, 01 May 2006 07:58:09 GMT
View Forum Message <> Reply to Message

mezise wrote on Sun, 30 April 2006 21:06luzr wrote on Sun, 30 April 2006 17:01Not yet. To put it straight, so far SqlExp was designed for Oracle and will need further tweaks for other SQL engines.

Mirek

OFFSET-LIMIT clause is quite necessary in MySQL, SQLite and PostgreSQL.
So, could this code be put into next dev5 release?

Definitely! Done. Thank you, that is the kind of response I like

BTW, speaking about it, you should be aware of Sql refactoring plan....

The problem there now is with dialects AND using multiple engines at once (like Oracle and MySql in single app).

There are two problems - SQL dialects differ AND of course, "implicit session" are now two...

Well, several month ago plan was to bind the dialect with "execution" Sql - that would involve storing SqlExp into some intermediate language form and then "compiling" it for target engine. Also, implicit session and default SQL cursor (which in fact are related) would be gone in multisession apps.

However, further thinking revealed that it would still be difficult to use - you would have to quote session when makeing any Sql cursor. So the current plan, somewhat less elegant but I believe more productive, is simply to have per-thread "current session" variable and change it using RAII push/pops, something like

```
void MyFn() {
   UseSqlSession __(MySqlSession);
   // now until __ destructor, MySqlSession and MySQL dialect are used for everything, SQL
reffers to MySqlSession
}
```

maybe, using macro hackery, we could also introduce form

```
void MyFn() {
   SQLCONTEXT(MySqlSession) {
.....
   }
}
```

That also means that those "dialect" members of SqlS are likely to be gone in favor of single per-thread global variable.

Mirek