Subject: PostgreSQL GetInsertedId()
Posted by captainc on Wed, 02 Jul 2008 14:34:14 GMT
View Forum Message <> Reply to Message

I noticed the problem with PostgreSQL's GetInsertedId(). I found this post on the web:
Quote:I would always suggest that people do a select nextval('seq') followed by a suitable insert. It requires no locking, is definitely multi user safe and will *always* correctly identify the row. The only downside is that it requires a quick select before the insert, but if this minor speed sacrifice was that much of an issue"

The definition of nextval() from postgres site:
"Advance the sequence object to its next value and return that value. This is done atomically: even if multiple sessions execute nextval concurrently, each will safely receive a distinct sequence value."
This means that you would have to use the value returned in the insert statement.
Also, currval() should be session-safe, and is probably easier to use.

Like the mysql package, maybe we can use a 'lastid' variable and update the id for each insert statement made.

Otherwise, we might need a 'lastsequence' variable and a String GetLastSequence() function, and as the source comments suggest, a GetInsertedId(const char* sequence) function to go along with it.
Then we could make the call sqlGetInsertedId(sql.GetLastSequence()).

Thoughts?

Subject: Re: PostgreSQL GetInsertedId()
Posted by captainc on Wed, 02 Jul 2008 14:46:12 GMT
View Forum Message <> Reply to Message

Idea 2: Have a mapping of sequences to table names from the .sch file.

Something like:
String seq = m_session.GetPKSequence(TABLENAME)

and we could use it like this:

...insert statement...
sql.GetInsertedId(seq)

and/or

sql.GetNextId(seq)
...insert statement...

Idea 3: Create a separate insert function called InsertSelect which will return 0 (or -1) for a failed insert and otherwise return 1..N which would be the last inserted id using the method described in the above post. So this would mean we would have 1 insert with no overhead and 1 with the overhead of getting the inserted id and returning it.

Subject: Re: PostgreSQL GetInsertedId()
Posted by mirek on Wed, 02 Jul 2008 21:17:31 GMT
View Forum Message <> Reply to Message

Well, I know that the difference between RDBMS is disturbing, but I think neither solution is good enough.

In reality, this seems to be much more problem in theory. In practice, in most cases, you are designing the application for the specific database anyway.

Mirek

Subject: Re: PostgreSQL GetInsertedId()
Posted by captainc on Thu, 03 Jul 2008 11:48:50 GMT
View Forum Message <> Reply to Message

Well we have it in there, but it doesn't do anything, so something has to change.

Subject: Re: PostgreSQL GetInsertedId()
Posted by mirek on Fri, 04 Jul 2008 11:49:39 GMT
View Forum Message <> Reply to Message

captainc wrote on Thu, 03 July 2008 07:48Well we have it in there, but it doesn't do anything, so something has to change.

It is there because it represent the only way how to manage this problem in MySQL and other "autoincrement" databases without sequences.

OTOH, in Oracle, this is nearly impossible.

Mirek

Subject: Re: PostgreSQL GetInsertedId()
Posted by captainc on Mon, 07 Jul 2008 17:45:02 GMT
View Forum Message <> Reply to Message

I understand, but from the perspective of a new Upp user who sees this GetInsertedId() function while using the PostgreSQL package (with assist++ or however) and ends up smacking his head against a wall trying to get it to work correctly, when it is only there as a placeholder and always returns Null... It's just not nice.

Subject: Re: PostgreSQL GetInsertedId()
Posted by unodgs on Mon, 07 Jul 2008 20:15:42 GMT
View Forum Message <> Reply to Message

captainc wrote on Mon, 07 July 2008 13:45I understand, but from the perspective of a new Upp user who sees this GetInsertedId() function while using the PostgreSQL package (with assist++ or however) and ends up smacking his head against a wall trying to get it to work correctly, when it is only there as a placeholder and always returns Null... It's just not nice.
I think we need to add GetInsertedId(const char* seqence) to session interface. Of course one can use currval from sqlexp but I prefer getting id through session object. If I won't find any better solution I will add this new GetInsertedId version.

Subject: Re: PostgreSQL GetInsertedId()
Posted by mirek on Mon, 07 Jul 2008 20:17:55 GMT
View Forum Message <> Reply to Message

unodgs wrote on Mon, 07 July 2008 16:15captainc wrote on Mon, 07 July 2008 13:45I understand, but from the perspective of a new Upp user who sees this GetInsertedId() function while using the PostgreSQL package (with assist++ or however) and ends up smacking his head against a wall trying to get it to work correctly, when it is only there as a placeholder and always returns Null... It's just not nice.
I think we need to add GetInsertedId(const char* seqence) to session interface. Of course one can use currval from sqlexp but I prefer getting id through session object. If I won't find any better solution I will add this new GetInsertedId version.

Does not help, MySQL does not have sequences.

Frankly, this is troublesome problem with many dead ends...

Mirek

Subject: Re: PostgreSQL GetInsertedId()
Posted by unodgs on Mon, 07 Jul 2008 20:46:19 GMT
View Forum Message <> Reply to Message

Quote:
Does not help, MySQL does not have sequences.
Frankly, this is troublesome problem with many dead ends...

I said add not replace :
int GetInsertedId() - for database with autoincrement fields
int GetInsertedId(const char* s) - for database with sequences

or just current GetInsertedId could be replaced with

int GetInsertedId(const char* s = NULL)

if sequence name is null it returns last valid inserted id (of course for databases which supports this way).

This do not solve assist problem - you have to know what getinsertedid to use for your database.

Subject: Re: PostgreSQL GetInsertedId()
Posted by zsolt on Mon, 07 Jul 2008 20:48:52 GMT

View Forum Message <> Reply to Message

I think, GetInsertedId(const char* seqence = 0) would be useful.
MySql and SQLite classes would simply neglect this parameter, PostrgreSQL would use parameter as a sequence name to return "SELECT currval('%s')".

This way we could create portable database software.

Subject: Re: PostgreSQL GetInsertedId()
Posted by zsolt on Mon, 07 Jul 2008 20:49:58 GMT

View Forum Message <> Reply to Message

Oops, Uno, you were faster

Subject: Re: PostgreSQL GetInsertedId()
Posted by mirek on Tue, 08 Jul 2008 05:22:10 GMT

View Forum Message <> Reply to Message

I am not sure what problem this should solve....

For Oracle, PGSQL the insert looks like: Get a new primary key from the SqlSequence, insert the record with this key. (You can use SqlSequence or PgSequence to get the int).

For MySQL it looks like: Insert a record, get a new primary key using GetInsertedId....

Where does fit GetInsertedId with param?

Mirek

Subject: Re: PostgreSQL GetInsertedId()
Posted by zsolt on Tue, 08 Jul 2008 06:55:31 GMT
View Forum Message <> Reply to Message

Yes, but many people are using SERIAL column in PostgreSQL.

Allowing an additional parameter in GetInsertedId, developer would be able to write exactly the same code for PostgreSQL, SQLite and MySql.

I don't know too much about current Oracle features, but I think, there can be some similar solution to "auto incremented" fields.

Subject: Re: PostgreSQL GetInsertedId()
Posted by mirek on Tue, 08 Jul 2008 13:57:25 GMT
View Forum Message <> Reply to Message

zsolt wrote on Mon, 07 July 2008 16:48I think, GetInsertedId(const char* seqence = 0) would be useful.
MySql and SQLite classes would simply neglect this parameter, PostrgreSQL would use parameter as a sequence name to return "SELECT currval('%s')".

This way we could create portable database software.

Well, I sort of see your argument now, but this is fundamentally broken w.r.t concurency.

What if other client inserts a record between your INSERT and SELECT?

Mirek

Subject: Re: PostgreSQL GetInsertedId()
Posted by zsolt on Tue, 08 Jul 2008 14:56:41 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 08 July 2008 15:57
What if other client inserts a record between your INSERT and SELECT?

In PostgreSQL, curval returns the value after last "nextval" in current session.
If some other clients call "nextval" after yours, you will see your original value got from your last "nextval".

Subject: Re: PostgreSQL GetInsertedId()
Posted by captainc on Tue, 08 Jul 2008 19:58:59 GMT

View Forum Message <> Reply to Message

zsolt wrote on Tue, 08 July 2008 10:56luzr wrote on Tue, 08 July 2008 15:57
What if other client inserts a record between your INSERT and SELECT?

In PostgreSQL, curval returns the value after last "nextval" in current session.
If some other clients call "nextval" after yours, you will see your original value got from your last "nextval".

Also, curval and nextval are session specific (session safe). So unless you inserted another item in your session, you will get back the id expected. Other sessions will not interfere with this even if they add information to the database in between.