

QGIS Application - Bug report #5475

Problem to insert splitted geometries in postgis

2012-04-23 01:20 PM - Luca Lanteri

Status:	Closed	
Priority:	Severe/Regression	
Assignee:	Marco Hugentobler	
Category:		
Affected QGIS version:	master	Regression?: No
Operating System:		Easy fix?: No
Pull Request or Patch applied:	No	Resolution:
Crashes QGIS or corrupts data:	No	Copied to github as #: 15099
Description		
<p>When geometry of a postgis layes is splitted in 2 parts Qgis assign the same old gid to the new geometries instead of leaving to postgres to assign the correct new gid.</p> <p>The problem was tested on qgis 1.9 on windows. In Qgis 1.7.4 it works fine.</p>		
Related issues:		
Related to QGIS Application - Bug report # 5758: Primary key issue when using...		Closed 2012-06-07
Related to QGIS Application - Bug report # 6422: QGIS delete Postgis features...		Closed 2012-09-27
Related to QGIS Application - Bug report # 7550: Split feature on spatialite ...		Closed 2013-04-08
Duplicated by QGIS Application - Bug report # 17185: Data loss when discardin...		Closed 2017-09-22

History

#1 - 2012-04-23 03:16 PM - Tim Sutton

- File Error_001.jpeg added

Note that the issue is actually that the pkey is duplicated and thus the insert fails.

#2 - 2012-04-24 07:10 AM - Giovanni Manghi

Tim Sutton wrote:

| Note that the issue is actually that the pkey is duplicated and thus the insert fails.

is this a regression that is present also in the code freezed for the 1.8 release? if yes please tag this as blocker.

#3 - 2012-04-24 07:10 AM - Giovanni Manghi

- Status changed from Open to Feedback

#4 - 2012-04-24 07:24 AM - Jürgen Fischer

- Assignee set to Jürgen Fischer

- Priority changed from Normal to Severe/Regression

Giovanni Manghi wrote:

| is this a regression that is present also in the code freezed for the 1.8 release? if yes please tag this as blocker.

yes.

#5 - 2012-04-24 07:24 AM - Jürgen Fischer

- Status changed from Feedback to Open

#6 - 2012-04-24 07:29 AM - Giovanni Allegri

It seems a problem when generating the gid param value for the insert statement. AFAIU the attributes are by default the same as the original feature. When committing the provider should set an appropriate value for the new gid.

Looking at the code, in previous versions (e.g. branch 1.7.4) it seems it was done using a mPrimaryKeyDefault which was parametrized to be the next value beyond the maximum (as here <http://bit.ly/I9I3rb>). In the master branch I only see that the return value from QgsPostgresProvider::paramValue is always the original value if the default value is null (<http://bit.ly/I9ImIM>).

#7 - 2012-05-04 10:40 AM - Mayeul Kauffmann

- Crashes QGIS or corrupts data changed from No to Yes

Hi,

I updated as "cause data corruption" as it actually replaces the original line by a shorter line at least for me). Even if you say "do not save" when you leave the edit session, some data are lost

#8 - 2012-05-18 02:53 AM - Marco Hugentobler

If the insert of the new features fails, the user receives a descriptive error message and still has the possibility to adapt the primary key values, then commit again.

While I see it can be inconvenient, it is questionable if this is a bug at all. A table can have all sorts of constraints on it (not only on the pkey), and it is a difficult requirement for QGIS to detect that automatically. Furthermore, some users require that it should be possible to modify the pkey value in the attribute dialog. Therefore, the pkey is handled just like a normal attribute.

| Even if you say "do not save" when you leave the edit session, some data are lost

I guess you already said 'do save', otherwise you would not have the pkey unique problem.

#9 - 2012-05-18 03:09 AM - Giovanni Allegri

When I insert a new feature, the PK attribute is automatically filled with the following string:

```
nextval('testsplit_gid_seq'::regclass)
```

Then, if I try to split the geometry, the attribute window doesn't show up, because this operation is supposed to copy the same attributes values in the new feature. But, applying changes raises the following error:

Could not commit changes to layer testsplit

Errors: ERROR: 1 feature(s) not added.

SUCCESS: 1 geometries were changed.

Provider errors:

PostGIS error while adding features: ERROR: duplicate key value violates unique constraint "testsplit_pkey"

The problem is that the original geometry has been already modified. If I toggle editing off, also refusing to commit changes, the original feature is modified, and the other part (splitted) is lost.

Being impossible for Qgis to judge how to manage PKs, it should be left to the backend to manage it, i.e. let PG apply the changes, automatically increasing the PK from the sequence, and return the value back (eventually).

Otherwise a new window should be opened, leaving the user to enter the values, perhaps suggesting the same string as during the insertion of a new feature.

#10 - 2012-05-18 03:20 AM - Jürgen Fischer

Giovanni Allegri wrote:

Being impossible for Qgis to judge how to manage PKs, it should be left to the backend to manage it, i.e. let PG apply the changes, automatically increasing the PK from the sequence, and return the value back (eventually).

That would happen if the pk attribute would have the default value in it. Now it has a concrete value from the splitted feature in it. Currently there is no way to know which attributes make up the primary key outside of the provider. And so the split tool has no way to decide which attributes to change to their default value, which would in turn be replaced by the postgres provider.

The offending pk can however still be changed and committed after you have seen the error. No data should be lost unless you explicitly discard it.

Otherwise a new window should be opened, leaving the user to enter the values, perhaps suggesting the same string as during the insertion of a new feature.

There's also no way to tell which features have problems outside of the provider. But as said you can still change the offending feature (via the attribute table or the edit form).

#11 - 2012-05-18 05:40 AM - Giovanni Allegri

Now it has a concrete value in it. Currently there is no way to know which attributes make up the primary key outside of the provider

Actually this is a provider concern. The provider has the ability to know what kind of PK it is, hasn't it? So I imagine it could remove the duplicated value from PK in case the PK is a sequence, and let the RDBMS manage the correct insertion.

#12 - 2012-05-18 05:55 AM - Jürgen Fischer

Giovanni Allegri wrote:

Actually this is a provider concern. The provider has the ability to know what kind of PK it is, hasn't it? So I imagine it could remove the duplicated value from PK in case the PK is a sequence, and let the RDBMS manage the correct insertion.

It could, but that might also be considered a bug - if someone wants to set an explicit key (see #5608).

#13 - 2012-05-18 05:56 AM - Jürgen Fischer

- Crashes QGIS or corrupts data changed from Yes to No

#14 - 2012-05-18 06:05 AM - Giovanni Allegri

It could, but that might also be considered a bug - if someone wants to set an explicit key (see #5608).

Mmm, you're right, but whatever is being chosen I don't think that the current implementation is correct from an user experience perspective. If you don't open the attribute table and change the PK value manually, you obtain data corruption as soon as you try to save the editing as it is. If my table has a serial PK, I would naturally expect go on and save, without having to think about the fact that the provider doesn't take a decision and tries to do something wrong (i.e. try to save with an existing PK value).

If we want keep the responsibility out of the provider, I think we should advice the user about that, i.e. suggest to set the PK value BEFORE committing to DB, not AFTER the geometry has been written, otherwise we loose data and consistency.

#15 - 2012-05-18 07:33 AM - Jürgen Fischer

Giovanni Allegri wrote:

If we want keep the responsibility out of the provider, I think we should advice the user about that, i.e. suggest to set the PK value BEFORE committing to DB, not AFTER the geometry has been written, otherwise we loose data and consistency.

Split is the opposite of merge. When merging the user gets to decide which values of the original features the new merged feature should have. I think there should be something similar for the split case, where the user can decide which attributes the new features should have. They should start with the values of the original feature, but allow changes including a reset to the default value (ie. `nextval('testsplit_gid_seq'::regclass)` in this case).

But I think that would actually be a new feature and not just a fix.

#16 - 2012-05-18 07:48 AM - Giovanni Allegri

I agree with you Jürgen. Anyway, I think that a good design is to resume from a commit if an error happens. In this case the error makes the original feature being changed and losing the other part. I see it as a unique transaction, which should commit only if it success in its integrity.

#17 - 2012-05-18 10:55 AM - Jürgen Fischer

Giovanni Allegri wrote:

I agree with you Jürgen. Anyway, I think that a good design is to resume from a commit if an error happens. In this case the error makes the original feature being changed and losing the other part. I see it as a unique transaction, which should commit only if it success in its integrity.

Adding of features, deleting of features and changing of attributes and of geometries are independent operations on provider level and are therefore handled in separate database transactions in the postgres provider. Therefore the split "transaction" actually consists of several database transactions and the successful transactions are removed from the current edit session and cannot be rolled back, while the unsuccessful transactions are rolled back and stay in the current edit session.

#18 - 2012-05-18 01:29 PM - Giovanni Allegri

Ok, but I think that (if it would be possible/paid/sponsored/etc) it should be different. A split operation should be treated as a single transaction. It's like for a money exchange: you would never treat the withdrawal and the accounting as two independent transactions (I hope!) :)

#19 - 2012-05-18 01:36 PM - Jürgen Fischer

- File fix5475.diff added

Giovanni Allegri wrote:

Ok, but I think that (if it would be possible/paid/sponsored/etc) it should be different. A split operation should be treated as a single transaction. It's like for a money exchange: you would never treat the withdrawal and the accounting as two independent transactions (I hope!) :)

That would be a pretty huge change - changing stuff in the vector layer and all vector data providers and in turn the API.

Attached patch at least allows to reset attributes of new feature to their default value via a context menu of the field in the edit form. But it changes the API - not sure if it should be applied to 1.8.

#20 - 2012-05-18 01:36 PM - Jürgen Fischer

- Pull Request or Patch supplied changed from No to Yes

#21 - 2012-05-19 09:07 AM - Jürgen Fischer

Other ideas:

1. leave the attributes of the splitted feature untouched and require the user to use the merge attributes tool, if he wants the new feature to have the same attributes.
2. leave attributes marked as hidden or readonly out of the attribute assignment to the new splitted off feature (the user could flag the pk attribute as hidden/readonly in that case)

#22 - 2012-05-20 03:24 AM - Giovanni Allegri

The contextual menu patch can be a useful tool, anyway I would open the form dialog at the end of the split operation, as it happens when inserting a new feature.

Question: what about adding a capability to QgsVectorDataProvider to know the PK of a vector layer? In case the dataprovider supports it, when saving the edits a warning could be raised to acknowledge the user that a duplicated PK has been detected. The user will be free to discard the saving or force the commit.

#23 - 2012-05-20 11:21 PM - Marco Hugentobler

Another possible solution would be to show QgsMergeAttributesDialog after a split operation (the attribute form is less suitable since the split operation may create an arbitrary number of new features). That would not change the API. It would need some changes in the merge dialog to favor the default value of an attribute and some small changes in the split tool (which both should be possible in pre-release state).

What do you think about it? (I could do the implementation, then Jürgen may fix the node tool)

#24 - 2012-05-21 12:07 AM - Jürgen Fischer

Marco Hugentobler wrote:

Another possible solution would be to show QgsMergeAttributesDialog after a split operation (the attribute form is less suitable since the split operation may create an arbitrary number of new features). That would not change the API. It would need some changes in the merge dialog to favor the default value of an attribute and some small changes in the split tool (which both should be possible in pre-release state). What do think about it? (I could do the implementation, then Jürgen may fix the node tool)

Hey, that's my idea (see comment 15). I'd even think that it doesn't need a change in API.

Anyway, fine with me.

#25 - 2012-05-21 12:17 AM - Marco Hugentobler

(see comment 15)

Ah true, the comment list is too long to keep the overview :-)

#26 - 2012-05-21 12:27 AM - Jürgen Fischer

- Assignee changed from Jürgen Fischer to Marco Hugentobler

#27 - 2012-05-21 08:50 AM - Giovanni Allegri

I agree that having a dialog similar to QgsMergeAttributesDialog after the split operation would be simpler approach for the user. Obviously the dialog should be adapted to the different function.

I think that two key points for future developments are emerged from this ticket:

- having an improved transaction management, to be able to aggregate multiple operations in one single transaction, so that an error on one of them causes the rollback of all of them
- be able to be informed by data providers in case their data sources have PKs/unique values fields. It would help to inform/prevent the user from committing operations that would cause constraints violations.

#28 - 2012-05-22 12:11 AM - Marco Hugentobler

- Status changed from Open to Closed

Thinking about the split features dialog, I realise that it is important to insert default values into the dialog to make it convenient for the user. But we can do that directly in the core code without a dialog (and also without problems with API and string freeze). So now it is the other way round, default value is preferred over copied value, otherwise the user needs to change that with the info-Tool before committing the changes.

So hopefully the issue is fixed for now (a more user friendly dialog can always be added after the release): commit:5745fb90

#29 - 2017-09-22 04:41 PM - Jürgen Fischer

- Duplicated by Bug report #17185: Data loss when discarding changes in a PostGIS layer added

Files

Error_001.jpeg	31.2 KB	2012-04-23	Tim Sutton
----------------	---------	------------	------------

