QGIS Application - Bug report #5425
Remote PostGIS layer not compatible with features from local shapefile layer with same schema

2012-04-20 01:27 PM - Eric Jarvies

Status:	Closed		
Priority:	High		
Assignee:			
Category:	Data Provider/PostGIS		
Affected QGIS version:	master	**Regression?:**	No
Operating System:	OS X	**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	15082

Description

If I load two layers that share the **exact same schema**, one being a remote PostGIS layer(layer 1), and the other being a local shapefile layer(layer 2), when I copy a feature from layer 2, and paste it into layer 1, the attributes to not transfer properly. I believe this has something to do with the remote PostGIS layer using 'gid' and 'the_geom' columns, which of course is needed in the PostgreSQL database, but NOT in the QGIS layer stack along side shapefile layers that do NOT have "gid" and "the_geom" columns.

When pasting a feature from the local shapefile layer 2 into the remote PostGIS layer 1, the layer 2 attributes of copied/pasted feature of course do not have 'gid' and 'the_geom columns, and thus this results in a data offset of 2columns. If you use an export utility like pgsql2shp, the 'gid' and 'the_geom' columns do not get exported into the shapefile's tabular attributes, and rightly so as these are PostgreSQL/PostGIS specific, but QGIS seems to not take this into account, and seems to load the remote PostGIS layer with those two columns present.

When I create a shapefile and add columns to it, it does not create a 'gid' and 'the_geom' columns, it only does this after I use SPIT to import it into PostgreSQL/PostGIS, thus other then these two columns, the schema is identical such as the following example;

PostGIS layer 1 schema:
id
clave
name
address
tel
gid(pgsql only)
the_geom(pgsql only)

shapefile layer 2 schema:
id
clave
name
address
tel

So back in QGIS, the loaded remote PostgreSQL/PostGIS layer 1 is treating these two columns(gis and the_geom) as normal columns, so now when compared with my local shapefile layer 2, they are no longer identical in terms of their schema. So this now introduces problems, such as, if I wish to copy a feature from the local shapefile layer 2 and paste it into the remote PostgreSQL/PostGIS layer 1, the geometry itself(the feature) will paste into the layer 2 with no problems, but the attributes on the other hand, those end up getting screwed up, where an offset of 2 is introduced... here is an example:

When a feature(geometry) containing these attributes from the local shapefile **layer 2** is **COPIED**:
clave; 666
name; JOE BLOW

address; 123 ANYSTREET
tel: 867-5309
email; joe@blow.com

and then subsequently **PASTED** into the remote PostgreSQL/PostGIS **layer 1**, we end up with an offset that looks like this:
clave;
name;
address; 666
tel: JOE BLOW
email; 123 ANYSTREET

So, the actual data is being offset by two columns, and the two columns that are passed over, the data from those columns do not show up.

This makes any type of merging of features between any local shapefile layer with any remote PostgreSQL/PostGIS layer not possible.

If I have not made sense, or this is not clear, please advise and I will attempt to clarify better.

Thank you,

Eric Jarvies

Associated revisions

Revision 46adf1a0 - 2012-04-21 12:40 AM - Jürgen Fischer

remap attribute by name not index on paste (fixes #5425)

History

#1 - 2012-04-20 01:58 PM - Eric Jarvies

Another observation; When viewing PROPERTIES > FIELDS of the remote PostgreSQL/PostGIS layer, the LENGTH and PRECISION columns show all columns as having '-1' as their value, which of course is not correct, meaning the remote PostgreSQL/PostGIS table itself as is accessed/viewed directly ala Terminal app/SQL commands, clearly shows those columns with their correct values, so this is limited to QGIS.

#2 - 2012-04-20 03:42 PM - Jürgen Fischer

- Status changed from Open to Closed

Fixed in changeset commit:"46adf1a017f2aa4cba8ec43f2a9b772442ce7754".