

Unable to add new features in QGIS for PostGIS table using sequence for gid. Not providing nextval('sequencename'::regclass) in editor form or attribute table

Status:	Open	
Priority:	Normal	
Assignee:		
Category:	Data Provider/PostGIS	
Affected QGIS version:	3.6.0	Regression?: No
Operating System:	Debian 8.11	Easy fix?: No
Pull Request or Patch supplied:	No	Resolution:
Crashes QGIS or corrupts data:	No	Copied to github as #: 29489

Last Friday I encountered a new error that I've never seen before. When editing a new PostGIS table in QGIS 3.6.0 I'm unable to save my edits as the 'gid' column auto populates it's value as NULL. This behaviour seems to occur when drawing a new spatial feature, as well as when copy and pasting new features from a separate shapefile without 'gid' column.

Could not commit changes to layer tablename

Provider errors:

DETAIL: Failing row contains (null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null,
ull, null,
106000020E6100000010000000103000000010000000400000060015885FD2F...).

Looking into the database the table is using gid SERIAL PRIMARY KEY to define the "gid" column. I've tried dropping and re-adding the "gid" column, creating new tables, even manually creating a sequence and manually defining a new gid column as gid integer NOT NULL DEFAULT nextval('sequencename'::regclass) but no luck.

The strange thing is other tables within our PostgreSQL/PostGIS database do respond properly, providing the `nextval('sequencename'::regclass)` within the editor form and attribute table when creating new features or copy and pasting features in from another source.

PostgreSQL 9.4.19 on x86_64-unknown-linux-gnu, compiled by gcc (Debian 4.9.2-10+deb8u1) 4.9.2, 64-bit

POSTGIS="2.1.4 r12966" GEOS="3.4.2-CAPI-1.8.2 r3921" PROJ="Rel. 4.8.0, 6 March 2012" GDAL="GDAL 1.10.1, released 2013/08/26" LIBXML="2.9.1" LIBJSON="UNKNOWN" TOPOLOGY RASTER

I suspect that the database may have recently been upgraded to this version 9.4.19 which maybe related to this new glitch. Older tables in our database seem to function properly, recreating those same database tables using new tablename seems to run into this new glitch.

When searching online I came across an older ticket - Bug report #19564 which seems to describe the same behaviour. As described in this post, tables that do function properly seem to identify the 'gid' column as a 'Text Edit' widget type, whereas the new tables that are broken seem to identify 'gid' column as a 'Range' widget type. Looking at the database both tables were created the same manner using gid SERIAL PRIMARY KEY. However, somehow QGIS is detecting a difference between these two tables.

 $\frac{1}{3}$

column using a sequence properly, it makes our database significantly harder to use within QGIS. This issue seems to be affecting more users so I decided to file a new issue to see if any others have been able to find a suitable solution.

I've attached some screenshots for your reference. It seems that either QGIS, the PostgreSQL database or the way in which SERIAL columns are being communicated from one to the other has somehow changed recently, but I'm not sure as to where the change exists or how to resolve the behaviour. I'm hoping someone with more knowledge of the inner workings of QGIS and the PostGIS layer management will have some more understanding of what is happening here. This is the strangest behaviour I have ever come across in my 10 years of working with PostgreSQL/PostGIS and I'm absolutely stumped.

History

#1 - 2019-03-25 07:08 PM - David Mancini

- *File working_table_definition.png added*
- *File not_working_table_definition.png added*
- *File not_working_editor_form.png added*
- *File working_editor_form.png added*

Here are some more screenshots of the behaviour. I have some of the editor form popup upon a new feature being created, I also took some screenshots of how the tables are currently defined in PostgreSQL/PostGIS within pgadmin III.

I hope these are helpful references.

#2 - 2019-03-26 03:34 PM - Giovanni Manghi

- *Priority changed from High to Normal*
- *Crashes QGIS or corrupts data changed from Yes to No*

#3 - 2019-05-22 12:17 PM - Sylvia Preuss

Hi David,

I have got the same behaviour with QGIS 2.18 Las Palmas

"Konnte Änderungen am Layer test nicht festschreiben

Fehler: FEHLER: Ein Objekt nicht hinzugefügt.

Datenanbieterfehler:

PostGIS-Fehler beim Attributhinzufügen: FEHLER: NULL-Wert in Spalte »gid« verletzt Not-Null-Constraint

DETAIL: Fehlgeschlagene Zeile enthält (0102000020E86400000300000000641324E764C1841E0A576F61D92554134CA55..., null)."

PostGIS:

CREATE TABLE myschema.test

```
(  
  the_geom geometry(LineString,25832),  
  gid serial NOT NULL,  
  CONSTRAINT test_pkey PRIMARY KEY (gid)  
)
```

WITH (

OIDS=FALSE

);

... same behaviour with OIDS=TRUE

Did you found a solution for this problem?

#4 - 2019-05-22 06:37 PM - David Mancini

Hi Sylvia,

The only work around I was able to come up with within QGIS 3.6.x was setting a Default expression as follows: maximum("gid") + 1

Not ideal, but does at least allow me to save new features and copy / paste new features in from tables without gid. Having a look into the database I can see that this workaround still does not utilize the currently defined sequence as the Next Value does not continue to grow as new features are added in this way.

I hope this helps you out!

I'm curious about what version your PostgreSQL/PostGIS server you are running and whether or not it was recently upgraded?

#5 - 2019-05-23 10:56 AM - Sylvia Preuss

Hi David,

PostGIS 2.2 USE_GEOS=1 USE_PROJ=1 USE_STATS=1
Postgres PostgreSQL 9.5.4, compiled by Visual C++ build 1800, 64-bit

thank you,
there was no upgrade.
I renamed a table myschema.test into myschema.testnew but the name of the PRIMARY KEY of this table didn't change as well. I couldn't rename test_pkey into testnew_pkey. Maybe something was wrong with myschema.test. I think the database was confused by that. But I'm not really sure if this was the reason. Unfortunately it doesn't even help to create a new database.
On a separate server the serial - datatype in QGIS is still working. Maybe you need a new server

Files			
working_gid_sequence.png	46.5 KB	2019-03-25	David Mancini
not_working_gid_sequence.png	53 KB	2019-03-25	David Mancini
working_editor_form.png	9.16 KB	2019-03-25	David Mancini
not_working_editor_form.png	6.88 KB	2019-03-25	David Mancini
working_table_definition.png	21.4 KB	2019-03-25	David Mancini
not_working_table_definition.png	21.6 KB	2019-03-25	David Mancini