

QGIS Application - Bug report #6798

when copying a table by d&d from one schema to another the serial property is lost in the pk

2012-11-30 11:39 AM - Giovanni Manghi

Status: Closed	
Priority: Normal	
Assignee:	
Category: DB Manager	
Affected QGIS version: master	Regression?: No
Operating System:	Easy fix?: No
Pull Request or Patch supplied: No	Resolution: end of life
Crashes QGIS or corrupts data: No	Copied to github as #: 15946
Description	
When copying (by d&d) a PG table from one schema to another, it seems that the primary key column loses the autocremental property (in the new table), resulting in a error when adding features.	
Related issues:	
Related to QGIS Application - Bug report # 13823: DB Manager - Renaming const...	Closed 2015-11-13
Duplicated by QGIS Application - Bug report # 14104: DB Manager: sequence iss...	Closed 2016-01-14

History

#1 - 2012-12-31 01:23 AM - Paolo Cavallini

- Assignee changed from Giuseppe Sucameli to anonymous -

#2 - 2013-07-12 03:41 PM - Jürgen Fischer

- Assignee deleted (anonymous -)

#3 - 2013-07-22 07:55 AM - Giovanni Manghi

it happens only when in the dialog the user does not specify the primary key by enabling the "primary key" checkbox, so while it a very boring bug for real work, it is not that critical.

#4 - 2013-07-22 07:57 AM - Paolo Cavallini

Why not setting the box checked by default?

#5 - 2014-06-12 04:21 PM - Giovanni Manghi

- Target version changed from Version 2.0.0 to Version 2.4

#6 - 2014-06-28 07:45 AM - Jürgen Fischer

- Target version changed from Version 2.4 to Future Release - High Priority

#7 - 2015-12-21 11:34 AM - Giovanni Manghi

- Subject changed from primary key issue when copying a table from one schema to another to when copying a table by d&d from one schema to another the serial property is lost in the pk

- Target version deleted (Future Release - High Priority)

Giovanni Manghi wrote:

|

2024-09-20

1/5

it happens only when in the dialog the user does not specify the primary key by enabling the "primary key" checkbox, so while it a very boring bug for real work, it is not that critical.

this workaround does not work anymore.

#8 - 2015-12-23 04:45 AM - Bo Victor Thomsen

A more specific explanation of the problem:

A "serial" data type in PostgreSQL is syntactic sugar for "integer NOT NULL DEFAULT nextval('some_sequence'::regclass)", combined with the implicit creation of the sequence used in the DEFAULT clause.

So the D&D function **do** create the correct base type for the column, but fails to create the corresponding sequence and the default clause for the column.

#9 - 2016-03-13 06:58 AM - Paul Kanelli

See also #14104

#10 - 2016-03-21 12:35 AM - Paul Kanelli

This issue is quite annoying and it's unfixed for 3 years. Can someone set the priority to high or blocker?

#11 - 2016-03-21 11:17 AM - Giovanni Manghi

Paul Kanelli wrote:

This issue is quite annoying and it's unfixed for 3 years. Can someone set the priority to high or blocker?

we usually don't raise priority based on age of tickets. Blockers are regressions, and high priority is usually reserved for issues causing crash or data corruption (that are not regressions). This issue does not fit either.

#12 - 2016-03-22 02:42 PM - Paul Kanelli

Thx for the info. Here comes the workaround:

```
SELECT S E T V A L('<schema><table>_<column>_seq', (SELECT M A X(<column>) FROM <schema>.<table>));
```

```
ALTER TABLE <schema>.<table>
```

```
ALTER COLUMN <column> SET DEFAULT nextval('<schema><table>_<column>_seq'::regclass);
```

Please ignore the spaces in SETVAL and MAX.

Example:

```
SELECT S E T V A L('public.buildings_gid_seq', (SELECT M A X(gid) FROM public.buildings));
```

```
ALTER TABLE public.buildings
ALTER COLUMN gid SET DEFAULT nextval('public.buildings_gid_seq'::regclass);
```

#13 - 2016-03-31 03:06 PM - Paul Kanelli

- Assignee set to *Giuseppe Sucameli*

#14 - 2016-04-20 09:38 PM - Paul Kanelli

- Assignee deleted (*Giuseppe Sucameli*)

#15 - 2016-05-21 11:30 AM - Jason Wheatley

Paul - Thanks for the work around. I'll have to test that.

This is definitely an annoying bug. I only found this out recently when we decided to switch to PostGIS/QGIS as our main production workflow. Anytime you need to move a table from one location to another, you have to create a new primary key column (check primary key box and type in a new column name) otherwise you are unable to add any new records to those imported tables.

I would love to see this one fixed!

#16 - 2016-05-21 11:33 AM - Giovanni Manghi

Paul Kanelli wrote:

Thx for the info. Here comes the workaround:

```
SELECT SETVAL('<table>_<column>_seq', (SELECT MAX(<column>) FROM <schema>.<table>));
```

```
ALTER TABLE <schema>.<table>
```

```
ALTER COLUMN <column> SET DEFAULT nextval('<table>_<column>_seq'::regclass);
```

Please ignore the spaces in SETVAL and MAX.

if this is a patch that can be applied in DB manager code please submit it via pull request in the QGIS github repository.

#17 - 2016-05-21 02:21 PM - Paul Kanelli

Hi Giovanni, it's just a SQL query to sync the primary key. I've no programming skills.

Related: #14712

#18 - 2016-07-03 02:13 AM - R. R.

Is there any chance that we get this issue fixed soon?

#19 - 2016-08-26 01:18 AM - R. R.

(In some cases the name of the primary key is placed in quotes when copying a table by D&D (see also: #15226). After removing the quotes in the "Import vector layer" dialog, adding new features works properly. To me, this is really mysterious.)

EDIT: fixed

#20 - 2016-11-21 11:43 AM - R. R.

In QGIS 2.18.0 the sequence is completely missing after copying a table by D&D, but maybe this is better than a broken sequence.

#21 - 2016-11-21 11:48 AM - R. R.

- File *bug_6798_qgis_2_18.mp4* added

#22 - 2016-12-18 02:29 AM - R. R.

See also: #15991

#23 - 2016-12-18 02:55 AM - R. R.

- File *6798_rename_primary_key.mp4* added

It seems that renaming the primary key is the only way to create a working sequence (see screencast *6798_rename_primary_key.mp4*).

#24 - 2017-03-04 01:06 PM - R. R.

- File *6798_create_sequence.mp4* added

- File *pg_create_sequence_qgis_2x.py* added

Here comes a workaround for the missing sequence ...created with a lot of help from gis.stackexchange.com.

#25 - 2017-03-06 04:02 AM - Giovanni Manghi

Reinhard Reiterer wrote:

| *Here comes a workaround for the missing sequence ...created with a lot of help from gis.stackexchange.com.*

it would be nice if you can do a PR to add this script to

<https://github.com/ggis/QGIS-Processing>

#26 - 2017-03-12 06:09 AM - R. R.

- File *6798_create_sequence_version_2.mp4* added

- File *pg_create_sequence_qgis_2x_version_2.py* added

Hi Giovanni, I will check GitHub as soon as I find some time. At the moment, I do not know how a PR works. However, I have implemented a small improvement. There is no longer need to reload the layer manually.

#27 - 2017-05-01 01:09 AM - Giovanni Manghi

- Regression? set to No
- Easy fix? set to No

#28 - 2019-03-09 04:04 PM - Giovanni Manghi

- Status changed from Open to Closed
- Resolution set to end of life

End of life notice: QGIS 2.18 LTR

Source:

<http://blog.qgis.org/2019/03/09/end-of-life-notice-qgis-2-18-ltr/>

Files

bug_6798_qgis_2_18.mp4	869 KB	2016-11-21	R. R.
6798_rename_primary_key.mp4	1.37 MB	2016-12-18	R. R.
pg_create_sequence_qgis_2x.py	1.57 KB	2017-03-04	R. R.
6798_create_sequence.mp4	4.57 MB	2017-03-04	R. R.
6798_create_sequence_version_2.mp4	2.3 MB	2017-03-12	R. R.
pg_create_sequence_qgis_2x_version_2.py	1.75 KB	2017-03-12	R. R.