QGIS Application - Bug report #13947
postgis tables with inherits can lead to not unique QgsFeature.id

2015-12-04 09:16 AM - Alessandro Pasotti

Status:	Closed		
Priority:	Normal		
Assignee:	Jürgen Fischer		
Category:	Data Provider		
Affected QGIS version:	master	**Regression?:**	No
Operating System:		**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	21962

Description

Here is a test that reproduce the issue:

https://github.com/elpaso/QGIS/commit/614db661295ad3ddb4ca699ac92f6af548975f00

Associated revisions

Revision fc7fea55 - 2015-12-04 09:08 PM - Jürgen Fischer

postgres provider: verify uniqueness of parent table primary keys (fixes #13947)

Revision 269f7092 - 2015-12-08 11:05 AM - Jürgen Fischer

[FEATURE] postgres provider: refine warning about estimated metadata (fixes #13947)

History

#1 - 2015-12-04 12:59 PM - Jürgen Fischer

- Status changed from Open to Closed

Fixed in changeset commit:"fc7fea55ce1291d033a1b07455d374ba175413b3".

#2 - 2015-12-05 07:19 AM - Sebastian Dietrich

Please also note that your test creates the tables in a way that makes duplicate values much more likely:

When creating the child tables you define the inherited columns again. For the *gid* column this creates a **new sequence** for **each child table**. Duplicate values are almost guaranteed.

You should rather define only new columns and skip the inherited ones. The *gid* column in the child table then uses the same sequence as the parent table and duplicate values can only happen when manually entering values for *gid*.

Bad:

    CREATE TABLE qgis_test.child_table
    (
      gid serial NOT NULL,

```
  geom geometry(Point,4326),
  code character varying,
  CONSTRAINT child_pkey PRIMARY KEY (gid)
)
INHERITS ( qgis_test.base_table)
```

Good:

```
CREATE TABLE qgis_test.child_table
(
  CONSTRAINT child_pkey PRIMARY KEY (gid)
)
INHERITS ( qgis_test.base_table)
```

#3 - 2015-12-07 03:25 AM - Alessandro Pasotti

Yes, pretty bad design but that was exactly the purpose of the test. Unfortunately, it is a stripped down real case, from a large organization in the public sector that manages millions of records in that way with potentially catastrophic results when used in QGIS yielding duplicated QgsFeature ids.

With Juerghen's fix the layer will be invalid. Still unsure if the patch really solve that problem though.

I'll try to elaborate the test a bit more.

#4 - 2015-12-08 12:36 AM - Alessandro Pasotti

- Status changed from Closed to Reopened

The patch does not cover the case in which use estimated metadata is true.

#5 - 2015-12-08 02:06 AM - Jürgen Fischer

- Status changed from Reopened to Closed

Fixed in changeset commit:"269f70928571be08fe67d14eb43bda77945d8773".

#6 - 2015-12-08 04:35 AM - Sebastian Dietrich

Nice fix :-)

The actual problem is the incomplete support for inheritance in PostgreSQL. *Inheritance* together with *Partitioning* is such a mighty feature when working with huge datasets, but you give up the valuable good of *consistency*. No foreign keys, no unique indexes, exclusion constraints and many other features we are used to.

Since QGIS relies on consistent datasets you should treat inherited tables like (non-materialized) views: You have total responsibility for consistency. Usually this means you need to set up database triggers to enforce uniqueness across child tables and so on.