QGIS Application - Bug report #4569
Primary keys of type TEXT always null after creating feature in spatialite layer

2011-11-27 02:15 PM - Goyo D

Status:	Closed		
Priority:	Normal		
Assignee:	Jürgen Fischer		
Category:	Vectors		
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 #:**	14480

Description

Steps to teproduce:

1. Create a spatialite table with a primary key of type TEXT.
2. Load the spatialite table in QGIS.
3. Add a feature and type a non null value into the primary key.
4. Save changes.
5. The primary key is null.

Alternatively:

1. Download the attached archive and extract all files.
2. Open the sample.qgs.
3. Add a feature to Points1. You always get null values in point_id (TEXT PRIMARY KEY).
4. Add a feature to Points2. You always get a constraint error (because of the NOT NULL constraint added).
5. As a workaround Points3 has a NOT NULL UNIQUE constraint.

QGIS master, Ubuntu 11.10.

Associated revisions

Revision 72d7a238 - 2012-07-04 11:33 PM - Jürgen Fischer

spatialite provider:

- fix #4569

- select newly added spatialite database

History

#1 - 2011-12-16 02:09 PM - Giovanni Manghi

- Target version set to Version 1.7.4

#2 - 2012-04-16 06:32 AM - Paolo Cavallini

- Crashes QGIS or corrupts data set to No

- Target version changed from Version 1.7.4 to Version 1.8.0

- Affected QGIS version set to master

#3 - 2012-07-04 07:37 AM - Jürgen Fischer

- Assignee set to Jürgen Fischer

#4 - 2012-07-04 08:10 AM - Sandro Furieri

The SpatiaLite data provider always assumes a

Primary Key of the INTEGER AUTOINCREMENT type.

please note: the AUTOINCREMENT clause "magically"
transforms any NULL value into some unique numeric
value not yet used as a key-value, thus effectively
hiding any related complexity.
the same behaviour isn't obviously possible to
support when the Primary Key is of the TEXT type.

I hope this will help to understand why using any
Primary Key not defined as INTEGER AUTOINCREMENT
isn't a good idea.
(and AFAIK, the same is for PostGIS)

bye Sandro

#5 - 2012-07-04 01:32 PM - Goyo D

Thsnks, Sandro. The problem is that this enforces artificial restrictions on the database schema. I'm happy so far using autoincrement primary keys but I can think of some contexts where it can be an issue.
Anyway if this behavoir is a strong design decision feel free to close the ticket as wontfix.

#6 - 2012-07-04 02:35 PM - Jürgen Fischer
- Status changed from Open to Closed

Fixed in changeset commit:"72d7a238b76c2d9a137da7a7eebe9d7833509561".

#7 - 2012-07-04 02:48 PM - Jürgen Fischer

Jürgen Fischer wrote:

> *Fixed in changeset commit:"72d7a238b76c2d9a137da7a7eebe9d7833509561".*

The value of the primary key still defaults to NULL unless the user changes it. The provider used to overwrite the primary key with NULL, even if the user decided to enter a different value.

Accidental changes could still be avoided by manually setting the edit widget of the field to "Hidden" or "Immutable".

BTW the postgres provider presets the value with the default expression (eg. nextval('table_id_seq')). So the user has a similar option to change that or leave it as is.

Files

sample.tar.gz	241 KB	2011-11-27	Goyo D