QGIS Application - Feature request #8143
Add tooltip about why users cannot delete columns in Spatialite layers

2013-06-24 05:32 AM - Giovanni Manghi

Status:	Closed		
Priority:	Normal		
Assignee:			
Category:	GUI		
Pull Request or Patch supplied:	No	**Resolution:**	invalid
Easy fix?:	No	**Copied to github as #:**	16971

Description

subject says it all.

I would add that from a user point of view it would be really appreciated a workaround to allow delete CL columns, like the one that the plugin Table Manager used for shapes, before OGR allowing deleting columns.

Related issues:

Related to QGIS Application - Bug report # 15832: cannot delete fields on a s...	**Closed**	**2016-11-10**

History

#1 - 2013-06-24 06:04 AM - Sandro Furieri

- Status changed from Open to Closed

- Resolution set to wontfix

please note well: this is not a specific defect affecting the SpatiaLite data provider, it simply is a direct consequence of the SQLite's own design; DROP COLUMN is not a supported operation.

there is absolutely no way in SQLite to delete (or even rename) a column once it has been created, simply because the underlying SQL storage engine forbids such an operation.

for any further detail on this topic, please see:
http://stackoverflow.com/questions/8442147/how-to-delete-or-add-column-in-sqlite

bye Sandro

#2 - 2013-06-24 07:07 AM - Giovanni Manghi

- Resolution deleted (wontfix)

- Tracker changed from Bug report to Feature request

- Subject changed from Cannot delete columns in Spatialite layers to Add tooltip about why users cannot delete columns in Spatialite layers

- Category changed from Data Provider/SpatiaLite to GUI

#3 - 2013-06-24 07:55 AM - Sandro Furieri

some tricky workaround is surely possible: e.g.

1. begin a huge monolithic transaction
2. rename the already existing table (e.g. as "dummy_something")
3. create a new table, using the same name as before but this time omitting any column to be suppressed
4. create any Index or Spatial Index (or even Trigger) exactly as it was previously defined
5. copy all rows from the old to the new table

6. drop the old table
7. commit the still pending transaction
8. and finally perform a full VACUUM because your DB now will surely include lots of wasted space

an approach like the above one is already implemented by the spatialite_gui tool; it works nicely in many simpler cases, but will eventually lead to a non-functioning DB if some Foreign Keys or Views (or even Triggers) depend on the affected table in some unpredictable way.

and you should carefully consider that for a table containing many million rows a very long time could be required in order to perform all required steps

#4 - 2013-06-24 09:21 AM - Jürgen Fischer

The providers can only expose which capabilities they have (eg. DeleteAttributes), but can't tell upper layers why they don't have others.

#5 - 2013-06-25 04:18 AM - Giovanni Manghi

Sandro Furieri wrote:

> *some tricky workaround is surely possible: e.g.*
>
> *1. begin a huge monolithic transaction*
> *2. rename the already existing table (e.g. as "dummy_something")*
> *3. create a new table, using the same name as before but this time omitting any column to be suppressed*
> *4. create any Index or Spatial Index (or even Trigger) exactly as it was previously defined*
> *5. copy all rows from the old to the new table*
> *6. drop the old table*
> *7. commit the still pending transaction*
> *8. and finally perform a full VACUUM because your DB now will surely include lots of wasted space*
>
> *an approach like the above one is already implemented by the spatialite_gui tool; it works nicely in many simpler cases, but will eventually lead to a non-functioning DB if some Foreign Keys or Views (or even Triggers) depend on the affected table in some unpredictable way.*
>
> *and you should carefully consider that for a table containing many million rows a very long time could be required in order to perform all required steps*

will the geopackage format support such operations (renaming and in particular deleting columns)?

#6 - 2013-06-25 04:20 AM - Giovanni Manghi

- Resolution set to invalid

Jürgen Fischer wrote:

> *The providers can only expose which capabilities they have (eg. DeleteAttributes), but can't tell upper layers why they don't have others.*

no problem then... I just see a lot of puzzled users asking me why they can't delete a column. But that is anyway not our issue...

#7 - 2013-06-25 04:49 AM - Sandro Furieri

Geopackage will simply be based on the top of SQLite's own SQL dialect. Accordingly to this, Geopackage as well will be absolutely unable to support any ALTER TABLE RENAME COLUMN or ALTER TABLE DROP COLUMN statement.

SQLite is an absolutely unconventional DBMS; it certainly has lights and shadows, you can love it or you can hate it: anyway you cannot change it.
I easily understand that for many users it could be eventually unpleasant or puzzling discovering that renaming/deleting column is forbidden, or discovering that all columns are substantially "typeless" (except for Primary Key columns).
But these are not at all "defects": they are intended as explicit and rationally well motivated (and well documented) design choices.

#8 - 2013-06-25 05:43 AM - Giovanni Manghi

Sandro Furieri wrote:

> *Geopackage will simply be based on the top of SQLite's own SQL dialect. Accordingly to this, Geopackage as well will be absolutely unable to support any ALTER TABLE RENAME COLUMN or ALTER TABLE DROP COLUMN statement.*
>
> *SQLite is an absolutely unconventional DBMS; it certainly has lights and shadows, you can love it or you can hate it: anyway you cannot change it.*
> *I easily understand that for many users it could be eventually unpleasant or puzzling discovering that renaming/deleting column is forbidden, or discovering that all columns are substantially "typeless" (except for Primary Key columns).*
> *But these are not at all "defects": they are intended as explicit and rationally well motivated (and well documented) design choices.*

Just for the sake of discussion and with no intent whatsoever to make any kind of critique to sqlite, splite or qgis.

My point of view is all about free software and open formats.

I gave (and will give) gis training courses to hundreds of people and I try to explain why using open formats is good and right. 98% of those people needs to do simple things, like for example deleting a column in a table. They don't care if it is a feature of the format or a feature of the software, and I perfectly understand their point of view. Those people do not use DB like PostGIS or other enterprises DB, they don't have the skills and probably they do not even need to. But many of this people do use Arc* and many of them are keen to leave it behind. In Arc* they now use geodatabases (personal, file, whatever) and it is very handy, let's face it. Any computed/imported vector/raster goes into the DB and they can their simple things, like renaming/deleting columns, adding simple triggers, etc. without having to issue any cli command or use any other piece of software.

That is my dream for qgis, a closer synergy with the geodatabase of open formats (to allow for example store the computed vector/raster layers into a DB instead of shp+tif) and one way or another the challenge will be allow do simple things in a simple way.

#9 - 2014-06-18 04:07 AM - Saber Razmjooei

I wonder if it is possible to add the functionality (dropping a column) from:
http://sqliteman.com/
to the DB Manager.

Renaming the column will still be an issue

#10 - 2016-03-12 03:09 PM - R. R.

I agree with Giovanni's remarks and conclusion. In spatialite_gui both dropping and renaming a column seems to work fine. I wish QGIS would provide the same functionality via the DB Manager and the attribute table dialogue.