

QGIS Application - Feature request #20623

More documentation covering the limitations of Spatialite use

2018-11-25 10:58 AM - Dan Isaacs

Status: Feedback	
Priority: Normal	
Assignee:	
Category: Data Provider/SpatialLite	
Pull Request or Patch supplied:	Resolution:
Easy fix?: No	Copied to github as #: 28443
Description	
<p>I'd do this myself, but my knowledge of the area is not wide enough to ensure accuracy, but in the current QGIS documentation, no mention is made of the limits to QGIS integration with Spatialite and having just fallen victim to spending a couple of weeks setting up an entire project as a Spatialite database and finding just such limitations, it would have been advantageous to have them written up somewhere. The ones I've encountered so far are;</p>	
<ol style="list-style-type: none">1. Layers will not import from any other format in a useful manner. No matter whether the import is from shapefiles or csv's and regardless of the type of data within the tables, all data is imported as TEXT. This renders all further querying pointless as no maths can be done with the figures. You can workaround this by multiplying everything by 1.0, but that's a real hassle for a large number of expressions.2. Layers do not display correctly in QGIS (particularly labels, which randomly disappear at certain zoom levels). I think this may have something to do with...3. Indexes are not maintained or rebuilt properly. Indexes remain in the database for every layer that's ever been created regardless of later deletion. If a geometry layer has been added, the columns remain forever in the geometry tables and I think this might be what causes problems for later tables which happen to have the same name as a previous one.4. Attribute tables do not always show, often they show only one feature. I think this has something to do with the assignment of unique IDs, but ensuring each feature has a unique ID after importing does not fix the problem. A workaround for viewing only is to update the SQL but this renders the layer as a 'view' which are not editable and...5. The Identify tool doesn't work on views. The attributes are there, but the tool just doesn't bring up a form. This is also the behaviour on those layers which are mysteriously filtered to only one attribute.6. If you should make a mistake in setting up any of your layers you have to start all over again because columns cannot be deleted or altered, you could export the table as a shapefile or csv and alter the columns that way but (as per 1.) all your fields are converted to text and all your carefully created views are rendered useless.7. A more minor thing, but some widgets don't work with spatialite layers. The slider widget on 'Range', for example, displays properly but cannot adjust the value.8. Even when one has finally worked out convoluted workarounds for any of these issues, Spatialite throws another wrench by treating data only tables differently from geometry tables. A workaround I found for the importing of data types, for example, is to export as a shapefile, change the datatype in the dbf (plus add some example data of that type) import it, add it the canvas, then re-import it. This works OK for geometry layers, however it fails to work for plain dbf files.9. Importing layers from other databases sometimes mysteriously loses all the features. I haven't yet got as far as to work out what the common factor is here, but it's clearly not intuitive. Load a perfectly working layer into the canvas, import it to another database from the canvas and sometimes all the features are gone. I've even had a couple where some of the features are gone but others remain.	
<p>I've no doubt that some of these problems have solutions and I'm just unaware of them, or I'm doing things wrong, but even so, the system is far from intuitive and the current documentation (that I've been able to find) simply presents Spatialite as if it were just an alternative</p>	

means of storing the data to do pretty much the same thing as shapefiles, which it clearly is not.

The reason I've put this as a feature request for documentation is that it would clearly take ages to fix all these problems, and some of them are to do with Sqlite, not QGIS, and so can't be fixed anyway, it's about making people more aware before they just presume Spatialite is a viable option for their project. It's a very limited option compared to shapefiles, and I don't think that's made clear enough.

History

#1 - 2018-11-26 11:00 AM - Dan Isaacs

Oh, and I forgot to add the seemingly random 'Could not commit changes to layer' which crops up every now and again without any given reason or even a clue as to why which has suddenly rendered an, until now, perfectly functioning layer completely useless. Again, I'm well aware that I'm probably doing something wrong (which is why this isn't a bug report), but if simply changing some geometry with the node editor and altering the styling (which are the only two tasks I'd carried out) need to be performed in some specific manner in order to avoid bricking the whole layer then this should really be written quite clearly somewhere.

#2 - 2018-11-26 11:46 PM - Giovanni Manghi

- Status changed from Open to Feedback

Hi,

you should really:

- 1) search the bug tracker and check if any of these issues have been already reported (I'm almost sure a few have already been)
- 2) split the the issues that have not yet been reported in separated tickets, one issue per ticket

for each ticket you must add specific steps describing how to replicate the issue, as also attaching sample data/projects.

thanks

#3 - 2018-11-27 08:24 AM - Dan Isaacs

Giovanni Manghi wrote:

Hi,

you should really:

- 1) search the bug tracker and check if any of these issues have been already reported (I'm almost sure a few have already been)*
- 2) split the the issues that have not yet been reported in separated tickets, one issue per ticket*

for each ticket you must add specific steps describing how to replicate the issue, as also attaching sample data/projects.

thanks

As per the last paragraph, some of these problems have been reported before (the identify tool issue), some are to do with Sqlite itself (so I recently discovered), and some are intermittent problems which only occur after the database has been used for some time (as I think they may be caused by the indexes getting easily corrupted). All in all, it is very difficult to present these issues as bugs (or they already have been). Those that remain replicable bugs I will make some test files for and report them, but that's not the main point. The main point is that, even if reported, it would take quite a while to sort them

all out and until then Spatialite is not really a viable option for many case uses. The only purpose behind this request is to suggest that, until these issues are fixed, that limitation is made clear somewhere to prevent people from investing a lot of time structuring their projects in Spatialite only to find it doesn't work for their scenario.

My use, for example, involves working on a large number of sites, each with its own GIS project. A new analysis, layer, or measurement done on one site has to be transferred to all the others so the projects all look similar. As my work involves a lot of complex analysis, it seemed (from the documentation) that spatialite would do a better job than shapefiles with virtual columns. The problems with data types, adjusting columns and import/export, have, however, made spatialite virtually unusable in this scenario. It would have been helpful to me (and so presumably to others) if these limitations had been made clear in the documentation, just being realistic about the chances of them all being fixed/updated in the near future.

#4 - 2018-11-27 08:36 AM - Giovanni Manghi

Dan Isaacs wrote:

if these limitations had been made clear in the documentation, just being realistic about the chances of them all being fixed/updated in the near future.

feel free to submit a documentation pull request with some notes.

#5 - 2018-11-27 06:25 PM - Dan Isaacs

Giovanni Manghi wrote:

Dan Isaacs wrote:

if these limitations had been made clear in the documentation, just being realistic about the chances of them all being fixed/updated in the near future.

feel free to submit a documentation pull request with some notes.

Forgive my ignorance, but I don't really know what a pull request is, let alone how to submit one. Happy to do so if someone can point me in the right direction.

#6 - 2018-11-28 07:56 AM - Giovanni Manghi

Forgive my ignorance, but I don't really know what a pull request is, let alone how to submit one. Happy to do so if someone can point me in the right direction.

https://docs.qgis.org/2.18/en/docs/documentation_guidelines/first_contribution.html

#7 - 2018-11-28 10:07 AM - Dan Isaacs

Giovanni Manghi wrote:

Forgive my ignorance, but I don't really know what a pull request is, let alone how to submit one. Happy to do so if someone can point me in the right direction.

Thanks for the link, but I don't think it would be an appropriate means for me progressing with this issue.

I don't really have the technical expertise to know what it is I'm coming up against, I only know it doesn't work, not *why*, which I think would be needed for a proper addition to the documentation.

I was imagining something like; "Spatialite layers can be useful for... but because {whatever the reason is} they are not entirely suitable for applications where tables might need to be modified or imported regularly." and then maybe separately; "Spatialite views can be loaded into QGIS but cannot be edited or identified because...", and "Spatialite tables require {whatever it is they require} in order to load in a format which can be edited".

As you can see, I think I'd need to know why these things are the way they are before committing anything to the documentation and I'm afraid I'm just not that knowledgeable. I imagine, from the types of problems I've been encountering, that Spatialite works best with already existing datasets which require no further alteration, but having not tried that myself all I could honestly write at the moment is "Spatialite layers don't work as you might expect them to".

I checked the bug tracker before adding this and most of these issues have been raised before, most from more than 2 years ago and are still open. Seemingly these problems are either too difficult to solve or not a priority for developers. Either way, the fact that some fairly blocking Spatialite issues remain unresolved from over two years ago really emphasises the need for some clarity that Spatialite is not really a fully functioning option as someone migrating from ESRI shapefiles might expect.

I think this issue <https://issues.qgis.org/issues/15300> might actually be a significant part of the problem. I can see lack of feature `_ids` causing quite a few of the issues I've found, yet it's had zero attention since it was raised more than two years ago. And here's <https://issues.qgis.org/issues/16945> what I think might be the root of the data types problem, again with no action to address it from more than a year ago.

I understand that developer's time is limited, but that doesn't mean they can't just be honest about which aspects are actually supported and which aren't. Users of Qgis are generally a busy bunch too and it's a massive hassle to create a project which later doesn't really work because of some basic flaws that developers have known about since 2016.

If no-one with more knowledge than me has the time or inclination to take this issue any further then it will just have to remain as it is. It concerns me as a long-time QGIS user that quite serious core issues remain unsolved, whilst development time is spent on adding whole new features (like 3D mapping). Personally, I'm not sure that's the best direction to be heading.

#8 - 2018-11-28 11:13 AM - Harrissou Santanna

Hi Dan,

I understand your point and your frustration. I agree the issues you point are real ones that complicate working with SpatialLite data.

Fixing a bug requires a lot of parameters:

- It needs someone to report it (already done here)
- the availability of a developer that knows the internals of the feature (all the parts of QGIS code are not known by all the devs)
- the developer is aware of the report
- the bug priority among all the hundreds of bugs he can help to fix
- (we often forget it) some funds to do the work
- and once fixed someone to test and confirm

If no-one with more knowledge than me has the time or inclination to take this issue any further then it will just have to remain as it is.

If I can do a suggestion, do a compilation of the issues (should be nice to separate what is an issue to what is a limitation of the format if any) and write a mail either to `qgis-user` or `qgis-developer` list to voice these usability issues. It should get more audience (and I hope, attention) and if there are people that can help to fix/fund a fix or whatever, you'll know.

It concerns me as a long-time QGIS user that quite serious core issues remain unsolved, whilst development time is spent on adding whole new

features (like 3D mapping). Personally, I'm not sure that's the best direction to be heading.

There are real efforts done to reduce the bugs (with a bug fixing dedicated period which does not avoid fixing bug anytime) but QGIS can't avoid releasing new features. Users need new features and I agree they as well need working features.

Because you mention the 3D mapping, may I remind that this was the [community vote](#) during the grant proposal.

#9 - 2018-11-28 11:27 AM - Dan Isaacs

Fixing a bug requires a lot of parameters:

Absolutely. As you say, it all takes time and money, neither of which are in infinite supply, so it's a matter of priorities, and if those priorities are set by the community, then all the better. All I'm saying is that where there are decisions being made about those priorities (whether deliberately or inadvertently) it would really help us users if we knew about them and could adjust our plans accordingly. I don't know if Spatialite functioning is just unpopular (compared to something like 3D mapping) or whether it's just really hard and we don't have any experts on the development team. It could be either. But one thing is for sure (from the many-year-old open bug reports) is that it's not well supported right now (compared to other areas) and I just thought it would be helpful to be upfront about that somewhere so people didn't run into major blocking problems half way through their projects.

If I can do a suggestion, do a compilation of the issues (should be nice to separate what is an issue to what is a limitation of the format if any) and write a mail either to qgis-user or qgis-developer list to voice these usability issues. It should get more audience (and I hope, attention) and if there are people that can help to fix/fund a fix or whatever, you'll know.

Thanks, and if there's anything I can do to assist with this, please let me know.

#10 - 2018-12-01 11:25 PM - Jochen Huber

- File *Test_Spatialite.zip* added

Hi Dan,

I regularly use Spatialite databases in QGIS projects and have struggled often with problems like you have. A few rules have helped me to get it working most of the time. It is all by experience so maybe something isn't really necessary (anymore), but it works for me:

- Use lower case names for columns with id / pkey and geometry (snake_case should be okay), preferably also for other field names
- Every Spatialite table that is to be used in QGIS needs a unique index field. This field has to be numeric (preferably integer). Text won't work in my experience. Composite primary keys do not work. It is possible to import tables with a composite pkey from PostGIS databases (with limitations and only in one way), but not from Spatialite DBs (probably a db provider issue).
- To import a layer/dataset to the Spatialite DB: Add the data to be imported to QGIS; verify that all fields are imported correctly (eg, fields containing numerical values as integer/real, not as string - especially when using csv files); select the target database in the QGIS database manager; use the import layer/file button; be sure to select an id field and a geometry field (usually the option to create a spatial index should be activated)
- To add layers from the database: open the DB Manager, right click on the layer **→ add to map**

Views are a difficult issue. It is possible to create views with geometry that are usable, but I have not figured out a reliable way. You usually have to edit the auxiliary tables manually.

A possible workaround: Add a new virtual layer in QGIS. Use an SQL query to define the virtual layer. Select the id as first field (if you make a one-to-many-join: use the id from the "many" side to avoid duplicates), the geometry field as second field (it seems to be always called "geometry", even if in the db the field is eg "geom"), then all the attributes or newly created query-specific fields you need.

- One-to-many-joins: "many" table always on the left side of join, "one" table on the right

The most important point is the id field - if it is not unique and not numeric or missing at all, objects won't be drawn, attribute tables are incomplete etc.

I agree with you that working with Spatialite DBs in QGIS is not at all intuitive. For more users to embrace the advantages of Spatialite, it has to be way

more robust and straightforward. I think there are three aspects to address:

1. Fix bugs, add features - needs time and funding and, most of all, developers who know Spatialite, the libraries used and QGIS
2. Add checks / warnings / error messages and disable options causing problems (eg regarding a missing id field). This should be less difficult than 1. and prevent frustration
3. Document the limitations and add tips for working with Spatialite databases. I don't have any experience with documentation editing in QGIS yet and I don't have much time currently, but I would be willing to help here (since I am not a developer, so I can't do anything about 1. and 2.).

An example (Shapefile, DB and project file) is attached.

#11 - 2018-12-02 08:41 AM - Dan Isaacs

Jochen,

Thanks for the tips, I was beginning to think I was the only person using Spatialite. I'd not noticed the lower case primary key thing, it may well be responsible for a number of the issues I'm having. It's very frustrating to have to work these things out by trial and error, especially when working to a deadline as I am at the moment, so I'm very grateful for your timely input.

I know this isn't technically the purpose of these trackers, but having had no luck by any other route, do you mind if I just take the opportunity to ask a fellow Spatialite user if you've found a way around a particularly blocking problem I'm having at the moment?

One of the most annoying elements of the current Spatialite implementation is its propensity to turn every numerical value into unusable text at every opportunity. Having spent two days importing layers from an old project in the most unbelievably convoluted manner to get numbers to show up as REAL, not NUMERIC (or even TEXT), I've come to produce a view based off one of them that I want to display on QGIS with a graduated style. Views in Spatialite created from REAL columns seem to behave as REAL (ie they can be used in mathematical operations), but when imported into QGIS, the graduated style doesn't recognise them as real. Frustratingly, one of the columns in my view has been rendered REAL whilst the other is TEXT and I've no idea why.

I've tried CAST but when loaded into QGIS is still sees it as text. I've tried multiplying it by 1.0, still TEXT. I've tried exporting it as a Shapefile, loading that to QGIS and importing it through DB Manager, still TEXT. Finally I got it to work by exporting it as a Shapefile, editing the DBF to N,19,1 datatype, deleting all the data and writing it back in manually with 0.1 added to each value and then loading to QGIS. Now I need to update a value and have to go through the whole charade again.

Do you happen to know of any way of forcing QGIS to see numerical values in views as, well, numerical values?

Back to the topic, is there somewhere we can compile all these issues/workarounds so that they can be added to the documentation as one unified section rather than as a result of several separate additions?

#12 - 2018-12-04 06:58 AM - Jochen Huber

Hi Dan,

since one of the major advantages of using views is that the data is always up-to-date, having to manually import/export/tweak data is of course not acceptable...

I did not have many problems with data types in Spatialite so far, so I'm afraid I don't really have a solution for you. I checked projects at work yesterday, but I have to admit that I do have a tendency to upsize data to PostGIS when the problems with Spatialite become too many. I now mostly use Spatialite as a better Shapefile replacement for smaller projects without the need for extensive queries etc.

Do you import the data set in a new Spatialite table? Since data types in Spatialite are only preferences, I found it to be more reliable to create the table first (with the desired data types) and then import data so QGIS does not have to guess the types. Maybe this helps for underlying views, intensive use of typecasting could also be useful (at least I would have expected this, your experiences tell otherwise).

Would using Postgres/PostGIS be an option for you?

I think the discussion would be better placed in one of the QGIS mailing lists or a discussion group - maybe it would be possible to collect experiences / problem reports by more users there and condense them into separate tickets / bug reports.

Files

Test_Spatialite.zip

410 KB

2018-12-01

Jochen Huber