

QGIS Application - Bug report #16048

Spatial select doesn't work properly on filtered sqlite layers

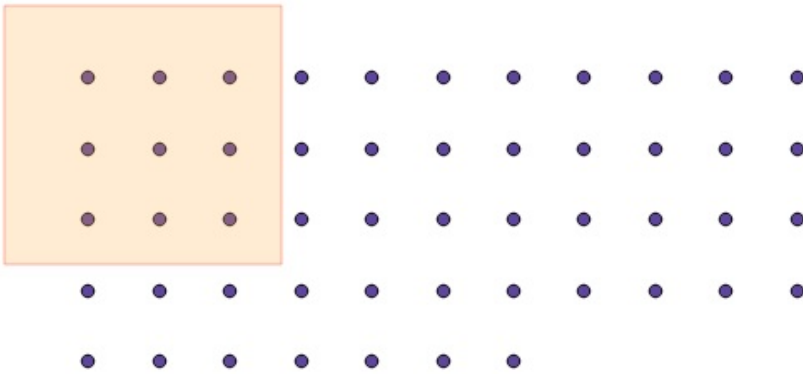
2017-01-08 06:41 PM - John Bryant

Status: Closed	
Priority: Low	
Assignee:	
Category: Data Provider/SpatialLite	
Affected QGIS version: 2.18.2	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 #: 23963

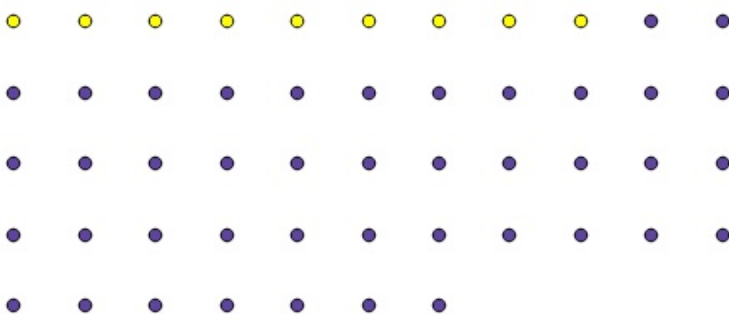
Description

When a sqlite layer is filtered, spatial select (eg. select by rectangle) selects the wrong features, ie. ones that are outside the selection area. See screenshots:

Select by rectangle:



Incorrect results:



The number of features is correct, but they are in the wrong location. This problem doesn't seem to occur with other formats, eg. shapefile.

The attached QGIS project and data demonstrate this issue.

This issue makes sqlite a problematic format for usage in QGIS.

Related issues:

Related to QGIS Application - Bug report # 16330: "Save as" to Spatialite doe...

Closed

2017-03-09

History**#1 - 2017-01-08 11:27 PM - Giovanni Manghi***- Category set to Data Provider/Spatialite*

Confirmed also on 2.14 and 2.8.

#2 - 2017-01-09 01:10 AM - Giovanni Manghi*- Priority changed from High to Normal***#3 - 2017-03-08 04:24 AM - Bernd Marcus**

The wrong selection of features working with filtered and non-filtered spatialite tables usually appear by using tables with no primary key.

BUT in the example file the table has such a key. Checking the Geometry and the CREATE statement show two critical issues:

1. the SRID is 0; this is never a good idea, cause there is no coordinate reference system defined (and no EPSG code); in such case QGIS will setup the reference system to the default CRS;
2. the main problem is the "Geometry" column, due to its declared type of 'POINT'. This might be the right way in other databases but using sqlite you should follow its type rules (see [<http://www.sqlite.org/datatype3.html>]) and define the feature type later on by recovering the geometry.

Attached is the example file with an additional spatial table, where I used "blob" as the column type of the geom. Even with a filter on the table selections work correct now.

#4 - 2017-03-08 04:27 AM - Bernd Marcus*- File pts_sqlite.sqlite.zip added*

couldn't upload the database file, one more try.

#5 - 2017-03-08 05:32 AM - Giovanni Manghi*- Subject changed from Spatial select doesn't work properly on filtered sqlite layers to Spatial select doesn't work properly on filtered sqlite layers when SRID=0**- Priority changed from Normal to Low*

Bernd Marcus wrote:

The wrong selection of features working with filtered and non-filtered spatialite tables usually appear by using tables with no primary key.

BUT in the example file the table has such a key. Checking the Geometry and the CREATE statement show two critical issues:

- 1. the SRID is 0; this is never a good idea, cause there is no coordinate reference system defined (and no EPSG code); in such case QGIS will setup the reference system to the default CRS;*
- 2. the main problem is the "Geometry" column, due to its declared type of 'POINT'. This might be the right way in other databases but using sqlite you should follow its type rules (see [<http://www.sqlite.org/datatype3.html>]) and define the feature type later on by recovering the geometry.*

Attached is the example file with an additional spatial table, where I used "blob" as the column type of the geom. Even with a filter on the table selections work correct now.

many thanks for this notes. It seems then that this issue is much less important than I supposed.

#6 - 2017-03-08 06:56 AM - Harrissou Santanna

I do not totally get the subtlety between sqlite and spatialLite formats (so bear with me, please) but if the original (and wrong) layer can be created from QGIS, then it can be seen as an important issue.

I mean, Does QGIS sqlite/spatialLite creation mechanism ensure that noone (namely beginners) will create a layer with the critics raised by Bernd? John didn't precise where the file comes from, though.

#7 - 2017-03-08 07:00 AM - Giovanni Manghi

Harrissou Santanna wrote:

I do not totally get the subtlety between sqlite and spatialLite formats (so bear with me, please) but if the original (and wrong) layer can be created from QGIS, then it can be seen as an important issue.
I mean, Does QGIS sqlite/spatialLite creation mechanism ensure that noone (namely beginners) will create a layer with the critics raised by Bernd? John didn't precise where the file comes from, though.

both when you create a SL db from the "save as" dialog or from the "new spatialite layer" one you cannot avoid choosing a CRS for the data.

#8 - 2017-03-08 07:20 AM - Bernd Marcus

Dear Giovanni,

the new title of this ticket is not correct. You can work with spatialite tables with a SRID = 0 in QGIS, but you can run into deep trouble with those 'layers'. The SRID = 0 is not the main reason for the wrong selected features.

In the database I've attached in the previous post, I haven't changed the SRID. But I've rebuilt the table and changed the type of the geometry column from "POINT" to "BLOB". Afterwards I assigned the 'POINT' geometry type to the geometry column with RecoverGeometryColumn(). And THIS solved the problem with the wrong selections.

@John Bryant:

How did you create the pts_sqlite table? Did you save the shape file as a spatialite table with QGIS's 'save as' dialog or with drag 'n' drop in the db-manager or manually with ogr2ogr or in a completely other fancy way? Would be interesting to know why you get an geometry type of 'Point' straight on.

#9 - 2017-03-08 07:45 AM - Giovanni Manghi

- Subject changed from Spatial select doesn't work properly on filtered sqlite layers when SRID=0 to Spatial select doesn't work properly on filtered sqlite layers

#10 - 2017-03-08 02:48 PM - John Bryant

- File qgis_sqlite_issue_test2.zip added

Interesting! Thanks a lot Bernd for looking into this. In this case I created the table using the QGIS 'Save as' dialog. But in my workflow I often use other tools, such as FME, to create Spatialite databases.

I've created an updated version of the test case with an explicitly named sqlite table 'saveas_spatialite' to illustrate. I can confirm the problem still occurs in exactly the same way with this new table.

So, if you're right that the issue is with the geometry type (POINT vs BLOB), then possibly the QGIS export function is the source of the issue?

Also, I'm a bit confused about the SRID=0 issue you raised - in QGIS I can see an SRID of 4326 for these tables. I'm not very familiar with the sqlite database structure so I don't know where to look for the SRID in a sqlite browser. In any case, again, the table was exported from QGIS so perhaps an issue with the 'Save as' function.

#11 - 2017-03-08 02:58 PM - Giovanni Manghi

- Status changed from Open to Feedback

John Bryant wrote:

Interesting! Thanks a lot Bernd for looking into this. In this case I created the table using the QGIS 'Save as' dialog. But in my workflow I often use other tools, such as FME, to create Spatialite databases.

I've created an updated version of the test case with an explicitly named sqlite table 'saveas_spatialite' to illustrate. I can confirm the problem still occurs in exactly the same way with this new table.

So, if you're right that the issue is with the geometry type (POINT vs BLOB), then possibly the QGIS export function is the source of the issue?

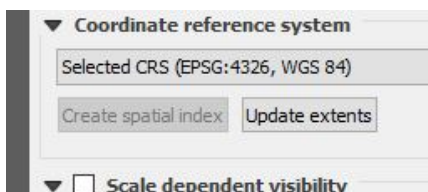
Also, I'm a bit confused about the SRID=0 issue you raised - in QGIS I can see an SRID of 4326 for these tables. I'm not very familiar with the sqlite database structure so I don't know where to look for the SRID in a sqlite browser. In any case, again, the table was exported from QGIS so perhaps an issue with the 'Save as' function.

after adding a spatial index on the SL layer the selection works as expected also when filtered.

#12 - 2017-03-08 03:02 PM - John Bryant

- File spatial_index_greyed_out.JPG added

But how does one add a spatial index in QGIS? That option is greyed out.



#13 - 2017-03-08 03:10 PM - John Bryant

I checked the QGIS export dialog - there is a SPATIAL_INDEX Layer Option, which defaults to YES. But even confirming this is so, the exported table still exhibits the same problematic behaviour.

#14 - 2017-03-09 01:42 AM - Giovanni Manghi

John Bryant wrote:

|

| But how does one add a spatial index in QGIS? That option is greyed out.

spatial indexes for rdbms data must be created within the same datasource. In QGIS for example you can use DB Manager.

#15 - 2017-03-09 02:11 AM - Giovanni Manghi

- Status changed from Feedback to Closed
- Resolution set to invalid

John Bryant wrote:

| I checked the QGIS export dialog - there is a SPATIAL_INDEX Layer Option, which defaults to YES. But even confirming this is so, the exported table still exhibits the same problematic behaviour.

so I see two issues here:

- using "save as" to Spatialite does not seem to create the spatial index even if the option is checked
- With DB Manager the spatial index can be created, but apparently DB Manager has never had implemented a way to show you that the SI for SL is present (like it does for PostGIS layers).

I'm closing this and open two new ones.

#16 - 2017-03-09 04:47 AM - Bernd Marcus

- Status changed from Closed to Reopened

I did a small exercise and checked different options to import data into a sqlite database, to figure out when the described selection problem arises. I used the pts_shp.shp from John as the base layer and did the following:

1. exporting the points to a new sqlite db with the "save as..." dialog in QGIS
2. drag 'n' drop pts_shp into an existing db using db-manager
3. drag 'n' drop pts_shp into an existing db using the browser windows
4. import the pts_shp into an existing db using Spatialite_GUI
5. using ogr2ogr to transfer the shp to a Spatialite table.

Task 2 didn't work (-> error dialog: invalid layer; could not load pts_shp).

In all other cases I used indexed and unindexed geometries. A spatial index is not part of the problem. Also not a strict type of BLOB for the geometry column, so I have to correct myself from yesterday. By importing a shp into a sqlite db with Spatialite_GUI the feature type (in our case POINT) is used for the geometry column type.

Apart from case 1 the other three created spatial tables had no selection issues when filtered.

The only remarkable difference between the 4 tables is the kind of how they are loaded into QGIS. When exporting a vector layer with the "save as..." dialog, QGIS will load the new created layer/table automatically if the option is not switched off. In such case we will get corrupted selections on filtered spatialite tables.

Loading the same table manually in QGIS using an existing database connection, everything works well.

Deactivate the automatic layer load option to avoid confusion when working with sqlite/spatialite tables (maybe other vector formats as well).

#17 - 2017-05-01 01:01 AM - Giovanni Manghi

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

#18 - 2017-08-09 01:43 PM - Andy Harfoot

Just encountered this issue myself on 2.18.11 on Windows 7 x64

I agree with Marcus that the problem appears to manifest in the loading of the data

The working method I have found uses the following steps:

1. Load the Spatialite / SQLite DB layer through DB Manager (Right click > Add to Canvas) OR Drag and drop from a Spatialite DB connection in Browser Window
2. Apply the filter through the QGIS Layer properties

Applying the filter in the DB manager using SQL prior to loading the layer results in the bug regardless of whether the 'Avoid selecting by feature id' option is used.

Loading the DB layer by dragging and dropping the Spatialite file from Explorer or the Browser window, treating it as a file based datasource, results in the bug

The Layer source strings reported in the layer properties generated by the three methods are as follows:

1. Database connection load, QGIS layer properties filter (no select bug)

```
dbname='<DatabaseName>' table="<TableName>" (geometry) sql="natural" = 'spring'
```

1. DB Manager query and load (select bug)

```
dbname='<DatabaseName>' table="(SELECT * FROM <TableName> WHERE \"natural\" = 'spring')" (geometry) sql=
```

1. QGIS File (OGR) based load, QGIS layer properties filter (select bug)

```
<DatabaseName>|layerid=2|subset="natural" = 'spring'
```

#19 - 2019-03-09 03:09 PM - Giovanni Manghi

- Status changed from Reopened to Closed
- Resolution changed from invalid 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/>

QGIS 3.4 has recently become our new Long Term Release (LTR) version. This is a major step in our history – a long term release version based on the massive updates, library upgrades and improvements that we carried out in the course of the 2.x to 3x upgrade cycle.

We strongly encourage all users who are currently using QGIS 2.18 LTR as their preferred QGIS release to migrate to QGIS 3.4. This new LTR version will receive regular bugfixes for at least one year. It also includes hundreds of new functions, usability improvements, bugfixes, and other goodies. See the relevant changelogs for a good sampling of all the new features that have gone into version 3.4

Most plugins have been either migrated or incorporated into the core QGIS code base.

We strongly discourage the continued use of QGIS 2.18 LTR as it is now officially unsupported, which means we'll not provide any bug fix releases for it.

You should also note that we intend to close all bug tickets referring to the now obsolete LTR version. Original reporters will receive a notification of the ticket closure and are encouraged to check whether the issue persists in the new LTR, **in which case they should reopen the ticket.**

If you would like to better understand the QGIS release roadmap, check out our roadmap page! It outlines the schedule for upcoming releases and will help you plan your deployment of QGIS into an operational environment.

The development of QGIS 3.4 LTR has been made possible by the work of hundreds of volunteers, by the investments of companies, professionals, and administrations, and by continuous donations and financial support from many of you. We sincerely thank you all and encourage you to collaborate and support the project even more, for the long term improvement and sustainability of the QGIS project.

Files

qgis_sqlite_issue_test.zip	413 KB	2017-01-08	John Bryant
select_area.jpg	16.1 KB	2017-01-08	John Bryant
results.jpg	14.6 KB	2017-01-08	John Bryant
pts_sqlite.sqlite.zip	411 KB	2017-03-08	Bernd Marcus
qgis_sqlite_issue_test2.zip	823 KB	2017-03-08	John Bryant
spatial_index_greyed_out.JPG	15.7 KB	2017-03-08	John Bryant