

QGIS Application - Bug report #20633

QgsVectorLayer from spatialite featureid problem

2018-11-26 04:24 PM - Henrik Spångmyr

Status:	Closed	
Priority:	High	
Assignee:	Julien Cabieces	
Category:	Data Provider/Spatialite	
Affected QGIS version:	3.4.1	Regression?: Yes
Operating System:	Ubuntu 16.04	Easy fix?: No
Pull Request or Patch supplied:	No	Resolution: fixed/implemented
Crashes QGIS or corrupts data:	No	Copied to github as #: 28453
Description		
<p>There seems to be an issue with QgsVectorLayers when the first column (key column?) contains values other than integers.</p> <p>It appears that the Feature ID are set from the first column, and when that column contains non-integer values like letters, the Feature ID is set to 0. The result is that when this feature is selected, all features sharing featureid=0 are also selected.</p> <p>This problem doesn't exist when loading the same table from PostGIS.</p> <p>It workes fine using QGIS 2.18.24.</p> <p>Background:</p> <p>I have a table (same layout in both spatialite and postgis):</p> <pre>CREATE TABLE obs_points (obsid text NOT NULL , name text , ... , ... , PRIMARY KEY (obsid))</pre> <p>Adding geometry for spatialite:</p> <pre>SELECT AddGeometryColumn('obs_points', 'geometry', 3006, 'POINT', 'XY', 0);</pre> <p>And for postgis:</p> <pre>ALTER TABLE obs_points ADD COLUMN geometry geometry(Point,3006);</pre> <p>I load this table to qgis using the QGIS "Add Spatialite Layer" and "Add PostGIS Layer".</p> <p>Adding features in the spatialite layer works fine as long as I only use integers for the obsid column. If I set obsid to letters, "Feature ID" is set to 0. If I set obsid to a float, it gets converted to integer.</p> <p>Ex:</p> <pre>obsid=1 > Feature ID=1 obsid=2 > Feature ID=2 obsid=A > Feature ID=0</pre>		

obsid=b> Feature ID=0
obsid=1.5> Feature ID=1
obsid=2.5> Feature ID=2

Selecting obsid=1.5 also selects obsid=1.

I've also tried to load the spatialite layer using the python api and two different uri-strings:

```
@
uri = QgsDataSourceUri()
    1. key
    layer1 = QgsVectorLayer('""dbname="/path/to/db.sqlite" key="obsid" table="obs_points" (geometry) sql="""', 'obs_points', 'spatialite')
    2. no key
    layer2 = QgsVectorLayer('""dbname="/path/to/db.sqlite" table="obs_points" (geometry) sql="""', 'obs_points', 'spatialite')
@
```

The resulting layer behaves the same way as the one loaded from the QGIS menu.

Linux Mint 19.
QGIS version
3.4.1-Madeira
QGIS code revision
383851c
Compiled against Qt
5.9.5
Running against Qt
5.9.5
Compiled against GDAL/OGR
2.2.3
Running against GDAL/OGR
2.2.3
Compiled against GEOS
3.6.2-CAPI-1.10.2
Running against GEOS
3.6.2-CAPI-1.10.2 4d2925d6
PostgreSQL Client Version
10.3 (Ubuntu 10.3-1)
SpatiaLite Version
4.3.0a
QWT Version
6.1.3
QScintilla2 Version
2.10.2
PROJ.4 Version
493

History

#1 - 2018-11-27 12:35 AM - Giovanni Manghi

- Priority changed from Normal to High
- Operating System deleted (Linux Mint 19)

#2 - 2019-01-14 10:45 AM - Julien Cabieces

- Pull Request or Patch supplied changed from No to Yes

- Status changed from Open to Closed
- Assignee set to Julien Cabieces

Related to #20547

Fixed in master and 3.4 release branch

#3 - 2019-01-14 11:49 AM - Giovanni Manghi

- Resolution set to fixed/implemented

#4 - 2019-02-06 05:58 PM - Henrik Spångmyr

- Status changed from Closed to Reopened

I have some problems with this still both in QGIS 3.4.4 and 3.5.0-master.

What works:

- All labels are visible now.
- Attribute table doesn't appear "filtered".
- I can select any number of rows in the attribute table.

What doesn't work:

- I can't select points on the map properly. Selecting a point sometimes selects another point, and sometimes doesn't select anything.
- If I select a row in the attribute table and press Zoom to selected, I get the error "Cannot zoom to selected feature(s): No extent could be determined."
- If I create a QgsVectorLayer using the python api and tries to add id:s [1, 2, 4] (as specified in the given keycolumn), the resulting feature id:s are [1, 2, 3]. The same happens if I try with keys ['P1', 'P2', 'P3']. I still get feature id:s set to [1, 2, 3].

#5 - 2019-02-06 06:10 PM - Henrik Spångmyr

I have a spatialite database with a table with column obsid.

I create 3 rows with obsid 1, 2 and 4.

I create an uri using obsid as keycolumn.

When QgsVectorLayer is created from this uri, I get feature_ids [1, 2, 3]. All features exist with correct attributes (obsid 1, 2 and 4) when I use QgsVectorLayer.getFeatures().

If I try to select these using QgsVectorLayer.selectByIds([1, 2, 3]), only id:s 1 and 2 get's selected.

Trying QgsVectorLayer.selectByIds([1, 2, 4]) instead changes nothing. Only id:s 1 and 2 are selected.

#6 - 2019-02-07 01:48 PM - Henrik Spångmyr

I have the code:

```
self.uri = QgsDataSourceUri()
self.uri.setDatabase(self.dbpath)
uri.setDataSource("", 'obs_points', 'geometry', "", 'obsid')
self.vlayer = QgsVectorLayer(uri.uri(), 'TestLayer', 'spatialite')
```

```
#self.vlayer is now a QgsVectorLayer with 3 features
```

```
features = self.vlayer.getFeatures()
feature_ids = [feature.id() for feature in features]
self.vlayer.selectByIds(feature_ids)
print("1. feature_ids: " + str(feature_ids))
print("2. QgsVectorLayer.selectedFeatureIds: " + str(self.vlayer.selectedFeatureIds()))
print("3. QgsVectorLayer.getSelectedFeatures: " + str([x.id() for x in self.vlayer.getSelectedFeatures()]))
print("4. QgsVectorLayer.getFeature(): " + str([self.vlayer.getFeature(x).id() for x in feature_ids]))
print("5. QgsVectorLayer.getFeature() type: " + str([str(type(self.vlayer.getFeature(x))) for x in feature_ids]))
print("6. QgsVectorLayer.getFeatures(): " + str([x.id() for x in self.vlayer.getFeatures(feature_ids)]))
```

Gives the result:

```
1. feature_ids: [1, 2, 3]
2. QgsVectorLayer.selectedFeatureIds: [2, 3, 1]
3. QgsVectorLayer.getSelectedFeatures: []
4. QgsVectorLayer.getFeature(): [0, 0, 0]
5. QgsVectorLayer.getFeature() type: ["<class 'qgis._core.QgsFeature'>", "<class 'qgis._core.QgsFeature'>", "<class 'qgis._core.QgsFeature'>"]
6. QgsVectorLayer.getFeatures(): []
```

So:

- The features are loaded to the layer (result from getFeatures()).
- The id seem to be correct when getting the features from getFeatures()
- QgsVectorLayer.selectByIds(feature_ids) seems to be halv-working, since QgsVectorLayer.selectedFeatureIds() returns the correct result.
- QgsVectorLayer.getSelectedFeatures() doesn't seem to return any features at all.
- QgsVectorLayer.getFeature() seems to get a feature of the correct type, but the feature seems to have 0 as feature id. (nr 4. and 5.)
- Selecting features with QgsVectorLayer.getFeatures([1, 2, 3]) doesn't seem to return anything (nr 6.)

#7 - 2019-02-12 02:28 PM - Julien Cabieces

- Status changed from Reopened to Closed

This feature has been backported in 3.4 the 14/01/2019 but the last release (3.4.4) has been done the 18/01/2019, so you have to wait for the next 3.4 release to test this on 3.4.

But I can't reproduce any of the problems you're reporting on the current 3.5-master. Did you make the test with a nightly built after the 14/11/2019 ?

If not, could you please test with the most recent nightly.

I close the issue for now, reopen it if you still encounter the issue on last nightly build

#8 - 2019-02-13 03:13 PM - Henrik Spångmyr

Julien Cabieces wrote:

This feature has been backported in 3.4 the 14/01/2019 but the last release (3.4.4) has been done the 18/01/2019, so you have to wait for the next 3.4 release to test this on 3.4.

But I can't reproduce any of the problems you're reporting on the current 3.5-master. Did you make the test with a nightly built after the 14/11/2019 ?

If not, could you please test with the most recent nightly.

I close the issue for now, reopen it if you still encounter the issue on last nightly build

I can confirm that it works for me also with the latest official 3.5.0 nightly build!

(Somehow it didn't work with my own build from the master branch. Not sure what I did wrong.)