QGIS Application - Bug report #18786
Test button in rule based styling returns wrong number of features when calculating (in shapefiles)

2018-04-22 08:39 AM - Erik Meerburg

Status:	Open		
Priority:	Normal		
Assignee:			
Category:	Expressions		
Affected QGIS version:	3.0.1	**Regression?:**	No
Operating System:	Windows 10	**Easy fix?:**	No
Pull Request or Patch supplied:		**Resolution:**	
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	26673

Description

Hi,

When making a rule based legend using a calculation, I ran into this one.

I've got two fields, both Integers. I have two rules here: Field1/field2>1 and Field1/Field2<1. The styling result comes out correct: If Field1 has the larger value, rule 1 applies, and vv.

However, when testing (pushing the Test-button), all objects fall into one of the categories. So the behaviour of the Test-button is not right. That was surprising... first I thought there was something wrong with my rules or the rule mechanism, but that behaviour is correct.

Turns out, this behaviour happens with shapefiles, but not with geopackages. Didn't test any other filetypes.

I've added a test shapefile with just two feautures, and a screenshot where you can see what I mean.

History

#1 - 2018-04-22 09:00 AM - Erik Meerburg

Oh, forgot to mention, but the behaviour of the test button was correct in 2.18.

#2 - 2018-04-22 12:27 PM - Richard Duivenvoorde

Had a short look:

- confirmed that apparently the renderer solves the expressions

    "P_HH_Z_K" / "P_HH_M_K" >1

and

    "P_HH_Z_K" / "P_HH_M_K" <1

right, as you will see two different colors loading the attached shapefile.

But debugging it, you will end up in QgsOgrFeatureIterator, where you can see that there is a switch for the different datasources (which explain why it works differently in gpkg and shp):

https://github.com/qgis/QGIS/blob/master/src/providers/ogr/qgsogrfeatureiterator.cpp#L148

QgsOgrFeatureIterator ends up with the following 'sql where' clause:

    "(("P_HH_Z_K" / CAST(("P_HH_M_K") AS float)) > 1)"

Which if you use that with ogrinfo:

    ogrinfo -sql "select * from test_rulebased where (("P_HH_Z_K" / CAST(("P_HH_M_K") AS float)) < 1)" -geom=NO test_rulebased.shp

NO result gives...

But casting both fields work:

    ogrinfo -sql "select * from test_rulebased where ((CAST(("P_HH_Z_K") AS float) / CAST(("P_HH_M_K") AS float)) < 1)" -geom=NO
    test_rulebased.shp

So either QGIS has to build a smarter 'sql where clause',

Or OGR has to handle given sql better?

Files

| test_rulebased.zip | 205 KB | 2018-04-22 | Erik Meerburg |

https://github.com/qgis/QGIS/blob/master/src/providers/ogr/qgsogrfeatureiterator.cpp#L148

QgsOgrFeatureIterator ends up with the following 'sql where' clause:

    "(("P_HH_Z_K" / CAST(("P_HH_M_K") AS float)) > 1)"