

QGIS Application - Feature request #121

join external tables to geometries

2006-05-19 10:08 AM - mlennert-club-worldonline-be -

Status:	Closed	
Priority:	Low	
Assignee:		
Category:	Data Provider	
Pull Request or Patch supplied:		Resolution: fixed
Easy fix?:	No	Copied to github as #: 10180
Description		
<p>It would be great to be able to join an external table (postgres, dbf, text, ...) to the attributes of a geometries file (such as the join tables feature in Thuban or [[ArcView]]). This would make cartography much easier than having to treat the attribute files directly.</p>		

History

#1 - 2006-05-19 10:09 AM - anonymous -

- Status changed from Open to In Progress

#2 - 2008-02-21 02:36 AM - Markus Neteler

I am also interested...

#3 - 2008-02-26 04:58 PM - Giovanni Allegri

It's a wish coming from many colleagues of mine. It would be useful to support join and then export to a new shapefile with the enlarged attributes table.

#4 - 2008-03-12 04:17 PM - Jürgen Fischer

- Status changed from In Progress to Open

#5 - 2008-07-14 02:23 AM - Tim Sutton

On second thoughts this is probably handled using the enhancement type and not a fake milestone. Returning to 1.0.1 milestone, please ignore last milestone change

#6 - 2008-09-12 10:35 PM - Paolo Cavallini

This is available through a plugin by Carson Farmer <http://www.geog.uvic.ca/spar/carson/cfarmerQgisRepo.xml>

It must be upgraded to 1.0 API though

#7 - 2008-09-13 02:55 AM - mlennert -

Replying to [comment:9 pcav]:

This is available through a plugin by Carson Farmer <http://www.geog.uvic.ca/spar/carson/cfarmerQgisRepo.xml>

It must be upgraded to 1.0 API though

This is already a good start, but IIUC, it creates a new shapefile everytime to join some data. Not really ideal IMHO. Would be nice to have this more 'on-the-fly', i.e. that you can join data from another data source (dbf, RDBMS, etc) to your map within the project, without creating any new files.

Moritz

#8 - 2009-11-20 09:53 PM - Alex Mandel

It just occurred to me that I might know any easy(relatively) fast way to do this. If every vector layer attribute table when loaded, queried or joined (last 2 might be best) was loaded to an in::memory sqlite database. Then we could just use sqlite syntax for doing table joins. These wouldn't be persistent unless we recorded them in the project file on save (which could be done) but they would be really fast to query and would provide robust left, and inner joining. Note, I'm specifically only talking about the attribute tables which should make it easier.

#9 - 2009-11-22 01:45 PM - mlennert -

Replying to [comment:13 wildintellect]:

It just occurred to me that I might know any easy(relatively) fast way to do this. If every vector layer attribute table when loaded, queried or joined (last 2 might be best) was loaded to an in::memory sqlite database. Then we could just use sqlite syntax for doing table joins. These wouldn't be persistent unless we recorded them in the project file on save (which could be done) but they would be really fast to query and would provide robust left, and inner joining. Note, I'm specifically only talking about the attribute tables which should make it easier.

Sounds interesting. But why limit this to attribute tables only ? I would think that the general use case is linking an attribute table to another table, not linking two attribute tables.

Moritz

#10 - 2011-03-02 08:07 AM - Paolo Cavallini

- Status changed from Open to Closed
- Resolution set to fixed

Table join now added in trunk. Please test.

#11 - 2011-03-03 06:56 AM - mlennert -

Replying to [comment:18 pcav]:

Table join now added in trunk. Please test.

I imagine that this is the Joins tab in the layer properties ?

A (very) quick test shows it works nicely, but we'll have to test it more thoroughly.

However, as I said in an earlier comment: what we are really looking for is the possibility of joining a layer attribute table to other data tables, not (only) to other layer attribute data tables. And IIUC, it is impossible at this stage to import a simple data table (e.g. dbf or csv) without geometries, and then join it to an existing layer attribute table, or ?

Moritz

#12 - 2011-03-03 08:25 AM - Marco Hugentobler

It is possible to load geometryless tables. This includes dbfs and csvs via OGR (set Filter to 'all Files' in the dialog) as well as [[PostGIS]] tables without geometry (select the option to show geometryless tables in the postgres connection settings).

#13 - 2011-03-07 02:21 AM - mlennert -

- Status changed from Closed to Feedback

- Resolution deleted (fixed)

Replying to [comment:20 mhugent]:

It is possible to load geometryless tables. This includes dbfs and csvs via OGR (set Filter to 'all Files' in the dialog) as well as [[PostGIS]] tables without geometry (select the option to show geometryless tables in the postgres connection settings).

Cool, never realised this was possible !

However, the join seems to fail once you try to map data:

- 1) Load a shape and a dbf file
- 2) Join the dbf to the shapefile attribute table
- 3) Go to symbology of shapefile and try to make graduated colors based on a joined attribute
=> all attribute values are set to NULL

Better than words: http://geog-pc40.ulb.ac.be/qgis/demo_join.gif. In the test, I use a modified version of the censusblk_swwake file from the grass book demo data: http://www.grassbook.org/data_menu3rd.php. I copied the censusblk_swwake.dbf to censusblk_data.dbf and erased most attributes from censusblk_swwake.dbf via OObase. I'll attach the files here.

Reopening the bug for now...

Moritz

#14 - 2011-03-07 02:24 AM - mlennert -

Replying to [comment:21 mlennert]:

I copied the censusblk_swwake.dbf to censusblk_data.dbf and erased most attributes from censusblk_swwake.dbf via OObase. I'll attach the files here.

Actually too big to attach, so here it is:

http://geog-pc40.ulb.ac.be/qgis/demo_data.zip

#15 - 2011-03-07 05:08 AM - Giovanni Manghi

However, as I said in an earlier comment: what we are really looking for is the possibility of joining a layer attribute table to other data tables, not (only) to other layer attribute data tables. And IIUC, it is impossible at this stage to import a simple data table (e.g. dbf or csv) without geometries, and then join it to an existing layer attribute table, or ?

I agree, the new Join implementation to be complete needs to allow Join also external data tables (dbf, as ftools already does, csv, etc.). It works already fine with geometryless postgres tables.

#16 - 2011-03-07 05:09 AM - Giovanni Manghi

Replying to [comment:20 mhugent]:

It is possible to load geometryless tables. This includes dbfs and csvs via OGR (set Filter to 'all Files' in the dialog) as well as [[PostGIS]] tables without geometry (select the option to show geometryless tables in the postgres connection settings).

sorry, I didn't see this either (I'm browsing the trac mailing lists mails...).

#17 - 2011-03-08 08:33 AM - Marco Hugentobler

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

Thank you for the good example data. The issue should be fixed with .

#18 - 2011-03-10 07:28 AM - mlennert -

- Status changed from Closed to Feedback
- Resolution deleted (fixed)

Replying to [comment:25 mhugent]:

Thank you for the good example data. The issue should be fixed with .

Yes, great !

However, I've found another issue:

- 1) Load a shape and a dbf file (e.g. the attached demo data)
- 2) Join the two
- 3) Save the project
- 4) Close QGIS
- 5) Open QGIS
- 6) Load project

=> The join is still active. Perfect !

But:

- 1) Load a shape and a dbf file (e.g. the attached demo data)
- 2) Join the two
- 3) Map an indicator using Graduated symbol (e.g. Median age)
- 4) Save the project
- 5) Close QGIS
- 6) Open QGIS
- 7) Load project

=> "Handle bad layers" window opens, but I just can't get the shape file to load anymore

This only happens when I use the old symbology, not the new.

Reopening the bug as I believe that this is an integrate part of the feature that needs to work.

Feel free to close again if you don't believe so.

Moritz

#19 - 2011-03-12 07:49 AM - Marco Hugentobler

- *Status changed from Feedback to Closed*

- *Resolution set to fixed*

The old symbology issue should be fixed in commit:489e34bc (SVN r15451). Please reopen the ticket in case of further problems.