QGIS Application - Bug report #818
PostGIS Tables with type GEOMETRY not supported

2007-11-18 08:06 PM - Frank Warmerdam -

Status:	Closed			
Priority:	Low			
Assignee:	nobody -			
Category:	Build/Install			
Affected QGIS version:			**Regression?:**	No
Operating System:	Debian		**Easy fix?:**	No
Pull Request or Patch supplied:			**Resolution:**	fixed
Crashes QGIS or corrupts data:			**Copied to github as #:**	10877

Description

If I try and add layer from postgis where the constraint is only for the geometry column to be at least of type "GEOMETRY" things don't work. I get the error:

Column wkb_geometry in "public"."mexico" has a geometry type of GEOMETRY, which Qgis does not currently support.

A value of "GEOMETRY" in the "type" field of GEOMETRY_COLUMNS indicates that any sort of geometry is permitted in the geometry column.

The problem was experienced with the raster transparency branch code, but is presumably the case in all modern source.

Associated revisions

Revision 38838ca1 - 2007-11-19 10:26 PM - Jürgen Fischer

support for GEOMETRY column, fixes #818

git-svn-id: http://svn.osgeo.org/qgis/trunk/qgis@7611 c8812cc2-4d05-0410-92ff-de0c093fc19c

Revision 6ba8b2a6 - 2007-11-19 10:26 PM - Jürgen Fischer

support for GEOMETRY column, fixes #818

git-svn-id: http://svn.osgeo.org/qgis/trunk@7611 c8812cc2-4d05-0410-92ff-de0c093fc19c

History

#1 - 2007-11-19 01:28 PM - Jürgen Fischer

- Status changed from Open to Closed

- Resolution set to fixed

support for GEOMETRY added in commit:6ba8b2a6 (SVN r7612)

#2 - 2007-11-19 03:03 PM - Frank Warmerdam -

Sweet,

I have confirmed this works.

However, this logic:

```
2527      valid = true;
2528      if(fType == "GEOMETRY")
2529      {
2530        // check to see if there is a unique geometry type
2531        sql = QString("select distinct "
2532              "case"
2533              " when geometrytype(%1) IN ('POINT','MULTIPOINT') THEN 'POINT'"
2534              " when geometrytype(%1) IN ('LINESTRING','MULTILINESTRING') THEN 'LINESTRING'"
2535              " when geometrytype(%1) IN ('POLYGON','MULTIPOLYGON') THEN 'POLYGON'"
2536              " end "
2537              "from %2").arg(geometryColumn).arg(mSchemaTableName);
2538        if(mUri.sql!="")
2539            sql += " where " + mUri.sql;
2540
2541        result = executeDbCommand(connection, sql);
2542
2543        if (PQntuples(result)==1)
2544        {
2545            fType = PQgetvalue(result, 0, 0);
2546        }
2547        PQclear(result);
```

does not seem to be having the effect I assume it is intended to. That is, even though my layers have only polygon and multipolygon geometries, I am still given a selector with all three geometry groups (and defaulting to point).

#3 - 2007-11-19 03:30 PM - Gavin Macaulay -

I note that the recent fix to this ticket has introduced a couple of extra columns to the 'Add [[PostGIS]] Tables' dialog box and has also altered the way in which in the user now selects layers to be added.

I think that the new tickbox column is unnecessary - highlighting the names of the layers to be loaded worked well (for me at least) without the added complication of a tickbox. The tickboxes have also removed some of the multiple selection options that were available (e.g., to select all tables now requires the user to individually select all of the layers, compared to shift-clicking the top and bottom table names).

The 'Import As' column is redundant for columns that aren't of type 'geometry', and I suspect would confuse most users in that situation. Also, as mentioned above, when the layer is of geometry type, the selection given in the 'import as' column isn't restricted to the actual geometry types in the table. This would probably be expensive to find out for all spatial tables in a large database. Perhaps the selection of 'import as' geometry should be deferred to when the layer is loaded and where the code only needs to find out the geometry type for just the selected layers?

Gavin

#4 - 2007-11-19 04:21 PM - Jürgen Fischer

Replying to [comment:2 warmerdam]:

> *However, this logic:*

> *[...]*
>
> *does not seem to be having the effect I assume it is intended to.*

The dialog currently doesn't care about what the GEOMETRY columns actually carries. The user is expected to select what feature type he want's to see from the layer and the dialog then add a filter via where clause to just select these features.

The logic you're refering to is in the postgresprovider and actually triggered when the layer is added. It verifies that there is only one feature type in the (filtered) table. As the postgresprovider is not only use from the dialog.

#5 - 2007-11-19 04:40 PM - Jürgen Fischer

Replying to [comment:3 g_j_m]:

> *I think that the new tickbox column is unnecessary - highlighting the names*
> *of the layers to be loaded worked well (for me at least) without the added*
> *complication of a tickbox. The tickboxes have also removed some of the*
> *multiple selection options that were available (e.g., to select all tables*
> *now requires the user to individually select all of the layers, compared to*
> *shift-clicking the top and bottom table names).*

> *The 'Import As' column is redundant for columns that aren't of type*
> *'geometry', and I suspect would confuse most users in that situation. Also,*
> *as mentioned above, when the layer is of geometry type, the selection given*
> *in the 'import as' column isn't restricted to the actual geometry types in*
> *the table. This would probably be expensive to find out for all spatial*
> *tables in a large database. Perhaps the selection of 'import as' geometry*
> *should be deferred to when the layer is loaded and where the code only needs*
> *to find out the geometry type for just the selected layers?*

You are certainly right about the loss of usablity.   There's already a thread
that determines the feature types of views, which might also be expensive.

This could be extended to the GEOMETRY columns. In case GEOMETRY column only
contains a unique feature type everything could stay as it was. In case the
GEOMETRY column contain multiple feature type a combobox that actually contains
the available feature type could be used instead of the icon. Thereby avoiding
the need for an extra column.

I added the checkbox because the combobox don't look to well in selected lines.
By moving the combobox to the first column that reason might vanish to.

Thanks for the input. I'll try that.

#6 - 2007-11-19 07:48 PM - Tim Sutton
- Status changed from Closed to Feedback
- Resolution deleted (fixed)

Hi jef

I'm just catching up on this thread. So reading between the lines here am I right in my interpretation of your motivations for this change here:
- you added the import as column to cater for mixed geometries
- this requires manipulating the contents of the import as cell
- the selection behaviour was select by row which meant you then couldnt change
  individual cells - in particular the import as cell
- to address that you changed selection mode to 'by cell'
- then you needed to add a cbx now to let you 'select rows'

If I get that all correctly I would like to suggest taking a different approach:

Since the case where table has poorly defined geometry type is probably an edge case it makes more sense to me that we just prompt the user after ok is clicked to make a determination of geometry type for ambiguous layers. This will let us leave the dialog with its original behaviour and remove the new elements from the layers table which in most cases will be redundant.

I'm also a bit concerned that making changes to the GUI will have knock on effects for document writers - translation of the documentation has begun and it is going to cause difficulties if we change procedural / workflow elements of the application that have already been committed to the current documentation.

What do you think? Can we roll back and just pop up a dialog if the user needs to be prompted for the geometry type?

I've reopened the ticket for now pending further discussion.

Best Regards

Tim

#7 - 2007-11-20 04:05 AM - Jürgen Fischer

- Status changed from Feedback to Closed

- Resolution set to fixed

fix makeover in commit:c9c3efb2 (SVN r7619)

The table layout and selection is now as it was, unless a table with multiple geometry types appears. In that case instead of the type icon a type combobox with the found geometry types is presented instead. The user is expected to selected which geometry he wants retrieved.

The type lookup is only done when the GEOMETRY type is found in geometry_columns and the type lookup is not restricted to that table in the server settings.

If such a columns is found, it's handled like the geometry column found in view (which are not listed in geometry_columns), ie. the determination what geometry type the table contains is done in a thread.

#8 - 2009-08-22 12:52 AM - Anonymous

Milestone Version 0.9.1 deleted