

QGIS Application - Bug report #1009

boolean, array types not handled reasonably on PostGIS layers

2008-03-25 07:23 PM - Steven Mizuno

Status:	Closed	
Priority:	Low	
Assignee:	Jürgen Fischer	
Category:	Data Provider	
Affected QGIS version:		Regression?: No
Operating System:	All	Easy fix?: No
Pull Request or Patch supplied:		Resolution: fixed
Crashes QGIS or corrupts data:		Copied to github as #: 11069
Description		
<p>A [[PostGIS]] table containing a boolean or any array type (also some other types) causes an empty Attribute table (rows/columns, but no data) and "No features found" using Identify. This worked in 0.8.1.</p> <p>The boolean type is often quite useful.</p> <p>The problem is that a cast to text is used when building the binary cursor query in [[QgsPostgresProvider]], select() and getFeatureAtId() functions, but there are no such casts to text in [[PostgreSQL]] for these types, so the query causes a database error, which is not reported to the user.</p> <p>Reasonable handling would be to use the default string handling in QGIS for any type not specifically handled. This works as long as there is a textual return from the database like there is with a non-binary cursor. Even editing will work.</p> <p>Interestingly, [[PostGIS]] provides a boolean to text function, so a boolean type column could be requested using text(boolean_column) rather than boolean_column::text.</p> <p>A work-around is to create suitable type-to-text functions and casts on the database so the ::text cast is available for these types.</p> <p>I am using [[PostgreSQL]] 8.2.6 / [[PostGIS]] 1.3.2</p>		

Associated revisions

Revision d5bdbb4c - 2008-03-27 01:44 AM - Jürgen Fischer

More work on the postgres provider:

- put common code from select/getNextFeature and getFeatureById methods to new declareCursor and getFeature methods
- unify type handling and support bool, arrays and time types (fixes #1009)
- ignore columns not explicitly supported (might apply to columns previously supported implicitly; please file a bug if you run into one)
- fixes a unreported problem with getFeatureById returning only NULL attributes.

Please test!

git-svn-id: <http://svn.osgeo.org/qgis/trunk/qgis@8285> c8812cc2-4d05-0410-92ff-de0c093fc19c

More work on the postgres provider:

- put common code from select/getNextFeature and getFeatureById methods to new declareCursor and getFeature methods
- unify type handling and support bool, arrays and time types (fixes #1009)
- ignore columns not explicitly supported (might apply to columns previously supported implicitly; please file a bug if you run into one)
- fixes a unreported problem with getFeatureById returning only NULL attributes.

Please test!

git-svn-id: <http://svn.osgeo.org/qgis/trunk@8285> c8812cc2-4d05-0410-92ff-de0c093fc19c

History

#1 - 2008-03-26 05:45 PM - Jürgen Fischer

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

fixed in commit:c598dc27 (SVN r8286). Please test.

#2 - 2008-03-27 09:11 PM - Steven Mizuno

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

Replying to [comment:2 jef]:

| fixed in commit:c598dc27 (SVN r8286). Please test.

I have tested with commit:fa696cda (SVN r8291) and it works as expected for boolean type. Array and money types just show blank cells where there should be values. Even NULL is not displayed for null values.

Since I posted this defect I have come up with a possible solution for handling types that don't have a cast to text. This should fit with the revised [[PostGIS]] provider. However, this may have [[PostgreSQL]] version differences as the functions are not published in the documentation.

Use a function that outputs a cstring for a specified type. I believe that the return from the query is actually a cstring, so there is no need to get the text type. Also be sure to use a column alias to identify the column.

Examples:

boolean type: boolout("column_name") as "column_name"

money type: cash_out("column_name") as "column_name"

array type: array_out("column_name") as "column_name"

array_out should work for any array type, but I have tested only with integer (int4, int8) and text arrays. There is also an anyarray_out() function, but I don't see any significant difference.

If there is a problem handling cstrings the output functions above can be wrapped with a textin() function to get to a text type. I don't believe that there is such a problem, but different versions of [[PostgreSQL]] may have slightly different behavior.

I have attached a patch incorporating the ...out() functions for money, array, and boolean types.

I have tested (but not extensively) the patch on boolean and money types, and integer and text arrays. Identify, Attribute table (display and edit), Feature editing all work.

#3 - 2008-03-27 09:34 PM - Steven Mizuno

I'm sorry - please disregard the part about NULL is not displayed for null values. I rechecked the types and found that NULL is displayed.

#4 - 2008-03-28 04:08 AM - Jürgen Fischer

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

#5 - 2009-08-22 12:54 AM - Anonymous

Milestone Version 0.9.2 deleted

Files

patch_for_bug_1009.txt	1.34 KB	2008-03-27	Steven Mizuno
------------------------	---------	------------	---------------