QGIS Application - Bug report #4685
Misleading error message for postgis tables with null geometry values

2011-12-20 06:51 AM - Sandro Santilli

Status:	Closed		
Priority:	Normal		
Assignee:			
Category:			
Affected QGIS version:	master	**Regression?:**	No
Operating System:		**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	fixed
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	14571

Description

The message is:

The problem is there's only null geometries in the set.

Something like this improves it slightly:

```
--- a/src/providers/postgres/qgspostgresprovider.cpp
+++ b/src/providers/postgres/qgspostgresprovider.cpp
@@ -3169,7 +3169,7 @@ bool QgsPostgresProvider::getGeometryDetails()
   QgsDebugMsg( "Getting geometry column: " + sql );
   result = connectionRO->PQexec( sql );

-  QgsDebugMsg( "geometry column query returned " + QString::number( PQntuples( result ) ) );
+  QgsDebugMsg( "geometry column query returned " + QString::number( PQntuples( result ) ) + " rows");

   if ( PQntuples( result ) > 0 )
   {
@@ -3255,10 +3255,10 @@ bool QgsPostgresProvider::getGeometryDetails()
         sql += " where " + sqlWhereClause;
       }

+     QgsDebugMsg( "Finding actual geometry column type: " + sql );
      result = connectionRO->PQexec( sql );

-     if ( PQntuples( result ) == 1 )
-     {
+     if ( PQntuples( result ) == 1 && !PQgetisnull(result, 0, 0) )
       fType = QString::fromUtf8( PQgetvalue( result, 0, 0 ) );
      }
     }
```

History

#1 - 2011-12-20 06:52 AM - Sandro Santilli

- Target version set to Version 1.8.0

Hopefully in better shape now:

```
--- a/src/providers/postgres/qgspostgresprovider.cpp
+++ b/src/providers/postgres/qgspostgresprovider.cpp
@@ -3169,7 +3169,7 @@ bool QgsPostgresProvider::getGeometryDetails()
   QgsDebugMsg( "Getting geometry column: " + sql );
   result = connectionRO->PQexec( sql );

-  QgsDebugMsg( "geometry column query returned " + QString::number( PQntuples( result ) ) );
+  QgsDebugMsg( "geometry column query returned " + QString::number( PQntuples( result ) ) + " rows");

   if ( PQntuples( result ) > 0 )
   {
@@ -3255,10 +3255,10 @@ bool QgsPostgresProvider::getGeometryDetails()
         sql += " where " + sqlWhereClause;
       }

+     QgsDebugMsg( "Finding actual geometry column type: " + sql );
      result = connectionRO->PQexec( sql );

-     if ( PQntuples( result ) == 1 )
-     {
+     if ( PQntuples( result ) == 1 && !PQgetisnull(result, 0, 0) )
       fType = QString::fromUtf8( PQgetvalue( result, 0, 0 ) );
      }
    }
```

#2 - 2011-12-20 08:09 AM - Sandro Santilli

With the patch above the message would say that the column is of unsupported type "geometry".
In PostGIS "geometry" means any kind, while "geometrycollection" means exactly a collection, but I think for what concerns qgis they could both mean the same thing, which is: an unconstrained mixed bag of geometry types.

So maybe we could list all these cases (geometry and geometrycollection) under the same name.
Either "geometrycollection" hijacking existing name or "mixed" to create a new name.
Sounds like something for Marco to consider in his refactoring to support collections...

#3 - 2011-12-20 08:55 AM - Jürgen Fischer

geometry columns with unsupported types are automatically removed from the selection dialog, so that error message should only appear if you add an invalid layer through the API. So I'd consider this an edge case, that should be handled and avoided by the plugin.

#4 - 2011-12-20 10:01 AM - Sandro Santilli

Ok, it makes sense. Then I guess the way to reproduce this is by having two geometry columns. One record. One geometry NULL and the other NOT NULL.
That's exactly my setup, which is what you get created by GEOS's XMLTester to debug failing tests with a single operand ...

#5 - 2012-01-27 06:12 AM - Alexander Bruy

- Crashes QGIS or corrupts data set to No

- Status changed from Open to Closed

- Resolution set to fixed

- Affected QGIS version set to master

Applied in commit:ac93c0c30f