

QGIS Application - Bug report #19211

Connect to postgis database fails in "Layer - Add Postgis layer" on databases with weird table names

2018-06-17 01:34 PM - Jens Astrom

Status:	Open	
Priority:	Normal	
Assignee:		
Category:	Data Provider/PostGIS	
Affected QGIS version:	3.0.0	Regression?: No
Operating System:		Easy fix?: Yes
Pull Request or Patch supplied:		Resolution:
Crashes QGIS or corrupts data:		Copied to github as #: 27040
Description		
I accidentally created a table in PostgreSQL called public."tt["species"]". This causes an "invalid name syntax" message when I try to connect to the database through the Add postgis layer route (which tries to list the available tables). The problem does not affect the DB-manager. The problem seems to lie in the query QGIS performs. Specifically, the concatenation in the has_table_privilege part. <pre>SELECT c.relname, n.nspname, a.attname, c.relkind, CASE WHEN t.typname IN ('geometry','geography','topogeometry') THEN t.typname ELSE b.typname END AS coltype,obj_description(c.oid) FROM pg_attribute a JOIN pg_class c ON c.oid=a.attrelid JOIN pg_namespace n ON n.oid=c.relnamespace JOIN pg_type t ON t.oid=a.atttypid LEFT JOIN pg_type b ON b.oid=t.typbasetype WHERE c.relkind IN ('v','r','m') AND has_schema_privilege(n.nspname, 'usage') AND has_table_privilege("" n.nspname "." c.relname "", 'select') AND (t.typname IN ('geometry','geography','topogeometry') OR b.typname IN ('geometry','geography','topogeometry','pcpatch')) AND (n.nspname,c.relname,a.attname) NOT IN (SELECT f_table_schema,f_table_name,f_geometry_column FROM geometry_columns)</pre> Rewriting the query like this makes it work: <pre>SELECT c.relname, n.nspname, a.attname, c.relkind, CASE WHEN t.typname IN ('geometry','geography','topogeometry') THEN t.typname ELSE b.typname END AS coltype,obj_description(c.oid) FROM pg_attribute a JOIN pg_class c ON c.oid=a.attrelid JOIN pg_namespace n</pre>		

```

ON n.oid=c.relnamespace JOIN pg_type t
ON t.oid=a.atttypid LEFT JOIN pg_type b
ON b.oid=t.typtype
WHERE c.relkind IN ('v','r','m')
AND has_schema_privilege( n.nspname, 'usage' )
AND has_table_privilege(QUOTE_IDENT(n.nspname) || '.' || QUOTE_IDENT(c.relname) , 'select' )
AND (t.typname IN ('geometry','geography','topogeometry') OR b.typname IN ('geometry','geography','topogeometry','pcpatch'))
AND (n.nspname,c.relname,a.attname) NOT IN (SELECT f_table_schema,f_table_name,f_geometry_column FROM
geometry_columns)

```

If there are no downsides, consider updating the query to the last version, which might be more robust to bad table naming. I'm not skilled enough in Python to test it out myself I'm afraid.

Tested on QGIS 2.18.0, postgis client version 9.2.4 AND QGIS 3.0.0-Girona , PostgreSQL Client Version 9.2.4

PostgreSQL version: "PostgreSQL 9.5.2 on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 4.8.2-19ubuntu1) 4.8.2, 64-bit"

```

ENCODING = 'UTF8'
TABLESPACE = pg_default
LC_COLLATE = 'en_US.UTF-8'
LC_CTYPE = 'en_US.UTF-8'
CONNECTION LIMIT = -1;

```

As well as on "PostgreSQL 9.6.9 on x86_64-pc-linux-gnu (Ubuntu 9.6.9-2.pgdg16.04+1), compiled by gcc (Ubuntu 5.4.0-6ubuntu1~16.04.9) 5.4.0 20160609, 64-bit"

```

ENCODING = 'UTF8'
TABLESPACE = gisdata
LC_COLLATE = 'en_US.UTF-8'
LC_CTYPE = 'en_US.UTF-8'
CONNECTION LIMIT = -1;

```