

## QGIS Application - Feature request #3453

### Enable "Use estimated table metadata" by default

2011-01-30 08:22 AM - Aren Cambre

<b>Status:</b>	Closed	<b>Resolution:</b> <b>Copied to github as #:</b> 13513
<b>Priority:</b>	Low	
<b>Assignee:</b>		
<b>Category:</b>	Data Provider	
<b>Pull Request or Patch supplied:</b>	No	
<b>Easy fix?:</b>	No	
<b>Description</b>		
Postgis layer loads start out with this query:		
I think this is needlessly expensive for my 12,000,000 row Postgis dataset for two reasons:		
* DISTINCT isn't necessary because my dataset has no duplicates even though some rows may have identical locations. Do GIS datasets routinely have duplicate data? At least let me override DISTINCT before loading large datasets. I am now worried that other datasets may have had objects missed that had identical locations.		
* Not sure why each geometry object is being checked for geometry type. QGIS knows the geometry type before the Postgis layer is even loaded thanks to the *geometry_columns* table.		

#### History

#1 - 2011-01-30 09:05 AM - Jürgen Fischer

#2 - 2011-01-30 10:15 AM - Aren Cambre

Thanks. Further analysis yields a correction: The query happens when the **Add Postgis Tables(s)** dialog appears after clicking **Add [[PostGIS]] Layer**. While I'm waiting for this interminable query to end, the dialog's **Type** column says **Waiting** for most the candidate layers.

?

And yes, it is issuing that query against the view.

?

If the query could be limited to a reasonable timeframe (LIMIT 100?), that would be a satisfactory resolution to this issue.

?

The query doesn't stop if I close the dialog. QGIS hangs until I manually cancel the query through PGAdmin III's Server Status utility.

#3 - 2011-01-30 10:37 AM - Jürgen Fischer

Replying to [comment:2 arencambre]:



*If the query could be limited to a reasonable timeframe (LIMIT 100?), that would be a satisfactory resolution to this issue.*



Like when checking "Use estimated table metadata" in the postgis connection settings?

**#4 - 2011-01-30 10:52 AM - Aren Cambre**

OK, that fixed it, but is there any reason for this not to be enabled by default, or use some kind of autosensing to make it happen for very large tables? Again, is it likely for views to contain different geometry types?

**#5 - 2011-01-30 11:21 AM - Jürgen Fischer**

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

Replying to [comment:4 arencambre]:



*OK, that fixed it, but is there any reason for this not to be enabled by default, or use some kind of autosensing to make it happen for very large tables? Again, is it likely for views to contain different geometry types?*



It's possible. And autosensing is what the query is about.

**#6 - 2011-01-30 12:46 PM - Aren Cambre**

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

Sorry, meant autosensing for very long queries. But that might get complex.



Better yet, is there any reason why **Use estimated table metadata** shouldn't be enabled by default? If it's likely that views will have a single geometry type in a geometry column, then this expensive query has a real cost and possibly no benefit for most users.

**#7 - 2011-01-30 01:42 PM - Jürgen Fischer**

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

Replying to [comment:6 arencambre]:



Better yet, is there any reason why **Use estimated table metadata** shouldn't be enabled by default? If it's likely that views will have a single geometry type in a geometry column, then this expensive query has a real cost and possibly no benefit for most users.



Well, it produces unreliable results - we need unique geometry types and the estimations don't assure this.



Judging from the user comments so far large datasets and views are uncommon. So we don't know about save assumptions for views.



I consider this an edge case and like to keep the defaults as is.

**#8 - 2011-01-30 04:19 PM - Aren Cambre**

That's fine. Thanks for being willing to discuss.

**#9 - 2013-05-15 06:41 PM - Aren Cambre**

- Resolution deleted (worksforme)
- Assignee deleted (nobody -)
- Target version changed from Version 1.7.0 to Version 2.0.0
- Pull Request or Patch supplied set to No

Does it make sense that **Use estimated table parameters** is enabled by default for MSSQL layers but **Use estimated table metadata** is disabled for

PostGIS layers? How likely will estimation produce unreliable results here? I still wonder if unreliable results is a boundary case?