

## QGIS Application - Bug report #16203

### SQL To Determine Primary Key column\_names does not handle views or situations where constraint\_name <> index\_name

2017-02-16 10:55 PM - Simon Greener

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Category:</b> Data Provider/Oracle	
<b>Affected QGIS version:</b> 2.18.3	<b>Regression?:</b> No
<b>Operating System:</b>	<b>Easy fix?:</b> No
<b>Pull Request or Patch supplied:</b>	<b>Resolution:</b> end of life
<b>Crashes QGIS or corrupts data:</b>	<b>Copied to github as #:</b> 24113

#### Description

All the testing was done by myself on Windows 10 64 bit using QGIS 2.18.3 against Oracle 12.1.

The requests below come from consulting work that I have completed for a customer who is using QGIS Desktop and QGIS Server against Oracle Spatial and are affecting their implementation.

#### Issue 1:

**The current QGIS Oracle code that discovers the column\_name associated with a primary key of a table (or materialized view) does not support situations where the constraint\_name is not equal to the index\_name.**

```
create table foo (  
  fid      number(10,0),  
  fid_int  Integer,  
  fid_num9 number(9,0),  
  fid_num8 number(8,0),  
  fid_num7 number(7,0),  
  an_attribute varchar2(10),  
  geom     sdo_geometry  
);
```

```
ALTER TABLE FOO ADD CONSTRAINT FOO_PK PRIMARY KEY (FID);
```

Current QGIS SQL for detecting PK:

```
SELECT column_name  
FROM all_ind_columns a  
  JOIN all_constraints b  
  ON a.index_name=constraint_name AND a.index_owner=b.owner  
WHERE b.constraint_type='P' AND b.owner='QGIS' AND b.table_name='FOO';  
-- FID
```

However, an alternate and very commonly used method for creating a primary key is:

```
alter table foo drop constraint foo_pk;  
create unique index foo_fid_uidx on foo(fid); /* Create UNIQUE Index Before define primary key /  
alter table foo add constraint foo_pk primary key (fid) using index foo_fid_uidx; / Note that PK uses pre-existing unique index */
```

The current QGIS SQL can be easily modified to cope with this situation:

```
SELECT column_name  
FROM all_ind_columns a  
  INNER JOIN  
  all_constraints b  
  ON ( a.index_owner=b.owner AND a.table_name=b.table_name AND a.index_name=b.index_name )
```

```
WHERE b.constraint_type='P' AND b.owner='QGIS' AND b.table_name='FOO';
```

### Request 1: Current QGIS Oracle code modified to use more complete SQL

**Note:** The above works also for materialized views as they are implemented as tables.

### Issue 2: Views

---

If we create a view over the FOO table:

```
create view vw_foo as select * from foo;
```

We can add primary key metadata as follows:

```
alter view vw_foo add constraint vw_foo_pk primary key (fid) disable;
```

But even the modified SQL for finding a PK does not work.

The SQL that I have created that detects any primary key for a view is:

```
-- Try modified table query
SELECT column_name
FROM all_ind_columns a
  INNER JOIN
  all_constraints b
  ON ( a.index_owner=b.owner AND a.table_name=b.table_name AND a.index_name=b.index_name )
WHERE b.constraint_type='P' AND b.owner='QGIS' AND b.table_name='VW_FOO';
-- no rows selected

-- OK construct more complicated query
select c.owner, c.table_name, d.column_name,
       c.constraint_name, c.constraint_type, c.index_name
from user_constraints a
  inner join
  user_dependencies b
  on ( b.name      = a.table_name
      and b.type    = 'VIEW'
      and b.referenced_type = 'TABLE')
  inner join
  user_constraints c
  on ( c.owner      = b.referenced_owner
      and c.table_name = b.referenced_name
      and c.constraint_type = 'P'
      and c.index_name is not null)
  inner join
  all_ind_columns d
  ON ( d.index_owner= c.owner
      AND d.table_name = c.table_name
      AND d.index_name = c.index_name
      )
where a.table_name = 'VW_FOO'
and a.constraint_type = 'P'
and a.view_related = 'DEPEND ON VIEW';
/*
OWNER TABLE_NAME COLUMN_NAME CONSTRAINT_NAME CONSTRAINT_TYPE INDEX_NAME
-----
QGIS FOO      FID      FOO_PK      P      FOO_FID_UIDX
*/
```

It works.

### Request 2: Could SQL to detect view PKs be implemented in QGIS please?

### Issue 3:

---

Even if a primary key column is detected, QGIS does not display the column\_name in its Layer Dialog.

**\*\*Request 3: Can QGIS be fixed to display the primary key column name?**

I have attached a worked SQL script for all the cases above.

regards  
Simon

## History

---

### #1 - 2017-05-01 01:01 AM - Giovanni Manghi

- *Easy fix? set to No*
- *Regression? set to No*

### #2 - 2017-09-08 01:35 PM - Jürgen Fischer

- *Assignee deleted (Jürgen Fischer)*

### #3 - 2019-03-09 03:09 PM - Giovanni Manghi

- *Resolution set to end of life*
- *Status changed from Open to Closed*

## End of life notice: QGIS 2.18 LTR

### Source:

<http://blog.qgis.org/2019/03/09/end-of-life-notice-qgis-2-18-ltr/>

QGIS 3.4 has recently become our new Long Term Release (LTR) version. This is a major step in our history – a long term release version based on the massive updates, library upgrades and improvements that we carried out in the course of the 2.x to 3x upgrade cycle.

We strongly encourage all users who are currently using QGIS 2.18 LTR as their preferred QGIS release to migrate to QGIS 3.4. This new LTR version will receive regular bugfixes for at least one year. It also includes hundreds of new functions, usability improvements, bugfixes, and other goodies. See the relevant changelogs for a good sampling of all the new features that have gone into version 3.4

Most plugins have been either migrated or incorporated into the core QGIS code base.

We strongly discourage the continued use of QGIS 2.18 LTR as it is now officially unsupported, which means we'll not provide any bug fix releases for it.

You should also note that we intend to close all bug tickets referring to the now obsolete LTR version. Original reporters will receive a notification of the ticket closure and are encouraged to check whether the issue persists in the new LTR, **in which case they should reopen the ticket.**

If you would like to better understand the QGIS release roadmap, check out our roadmap page! It outlines the schedule for upcoming releases and will help you plan your deployment of QGIS into an operational environment.

The development of QGIS 3.4 LTR has been made possible by the work of hundreds of volunteers, by the investments of companies, professionals, and administrations, and by continuous donations and financial support from many of you. We sincerely thank you all and encourage you to collaborate and support the project even more, for the long term improvement and sustainability of the QGIS project.