

It is always frustrating when SQL code is executed at run time and you cannot see an explicit error message in the LMS to explain the blank results. Some years ago, we used to have a way to turn on SQL logs inside the LMS and we COULD see the error generated. But not in this day and age.

You have performed the problem-solving steps that I would have suggested, but I am still guessing that you missed a minor error that is preventing the report from selecting data using your parameter at run time.

Here is my advice – for which I cannot find a job aid, so save this!

In Report Designer, be sure that you are running the query on a known set of data where you are guaranteed a small, correct result – such as 3 rows

*Remind yourself that when you run the report in Report Designer, **the parameter is never executed** because it is in the **green commented text**.*

This commented text is always ignored by Report Designer (because the `/** ... */` tells Report Designer that it should never read the code). Only at RUN time in the LMS will the LMS strip out the commented code and read/execute.

NOW that you know the RIGHT answer (3 rows should be returned) – you want to test YOUR parameter query in Report Designer by TEMPORARILY making your search query **and error codes** VISIBLE from within Report Designer

Read the paragraph above once or twice more until the light bulb goes off in your head!

Then when you realize how smart this is and how it will work, continue:

- a. Copy the green commented text from the query
- b. PASTE the green commented text into the query – I do this DIRECTLY before or after my commented text to be sure it is executed **in the same place** as the parameter will be interpreted at run time
- c. Now edit the green commented text so that it will be executed by Report Designer as the next statement in the query:
 - Remove the `/**` and `*/` code and then the text will not be green anymore, so will be executed by Report Designer
 - Always good to use criteria you know should work

An example is shown below to help illustrate.

```

15 , psr.inst_id, ins.fname as inst_fname, ins.lname as inst_lname
16
17 -- ADD FIELD TO TEST FOR PROGRAM TITLE
18 , cpnt.cpnt_title as cpntdesc
19
20 from pa_cpnt_evthst a, pa_cmpl_stat b, pa_student c
21 , pv_course cp, pa_sched sch, pa_cpnt cpnt,
22 pa_facility fac, ps_schd_resources psr, pa_inst ins
23
24 where a.cmpl_stat_id = b.cmpl_stat_id
25 and a.stud_id = c.stud_id
26 and cp.cpnt_typ_id(+) = a.cpnt_typ_id
27 and cp.cpnt_id(+) = a.cpnt_id
28 and cp.rev_dte(+) = a.rev_dte
29 and sch.schd_id(+) = a.schd_id
30 and sch.facility_id = fac.facility_id(+)
31 and cpnt.cpnt_id(+) = a.cpnt_id
32 and cpnt.cpnt_typ_id(+) = a.cpnt_typ_id
33 and cpnt.rev_dte(+) = a.rev_dte
34 and sch.schd_id = psr.schd_id (+)
35 and psr.inst_id = ins.inst_id (+)
36 /** and (a.cpnt_typ_id, a.cpnt_id, a.rev_dte) in [ItemSearch] */
37
38 /** and c.stud_id in [UserSearch]*/
39
40 /** and b.cmpl_stat_id in [CompletionStatusSearch]*/
41 /** and a.cmpl_dte >= [FromDate] */
42 /** and a.cmpl_dte < [ToDate]*/
43 /** and [security:pa_student c] */

```

BEFORE

```

15 , psr.inst_id, ins.fname as inst_fname, ins.lname as inst_lname
16
17 -- ADD FIELD TO TEST FOR PROGRAM TITLE
18 , cpnt.cpnt_title as cpntdesc
19
20 from pa_cpnt_evthst a, pa_cmpl_stat b, pa_student c
21 , pv_course cp, pa_sched sch, pa_cpnt cpnt,
22 pa_facility fac, ps_schd_resources psr, pa_inst ins
23
24 where a.cmpl_stat_id = b.cmpl_stat_id
25 and a.stud_id = c.stud_id
26 and cp.cpnt_typ_id(+) = a.cpnt_typ_id
27 and cp.cpnt_id(+) = a.cpnt_id
28 and cp.rev_dte(+) = a.rev_dte
29 and sch.schd_id(+) = a.schd_id
30 and sch.facility_id = fac.facility_id(+)
31 and cpnt.cpnt_id(+) = a.cpnt_id
32 and cpnt.cpnt_typ_id(+) = a.cpnt_typ_id
33 and cpnt.rev_dte(+) = a.rev_dte
34 and sch.schd_id = psr.schd_id (+)
35 and psr.inst_id = ins.inst_id (+)
36 /** and (a.cpnt_typ_id, a.cpnt_id, a.rev_dte) in [ItemSearch] */
37
38 /** and c.stud_id in [UserSearch]*/
39 AND c.stud_id in ('LDCOURY', 'DESOUZA', 'PLUNEZ')
40
41 /** and b.cmpl_stat_id in [CompletionStatusSearch]*/
42 /** and a.cmpl_dte >= [FromDate] */
43 /** and a.cmpl_dte < [ToDate]*/
44 /** and [security:pa_student c] */

```

AFTER

- d. Now preview the query in Report Designer – I am fairly certain that you will get zero rows AND I hope that you will get an error that you can decipher

THIS MEANS THAT YOU MUST BE PATIENT and read the Report Designer error log rather carefully.

FYI: Report Designers errors tend to be hidden out in the far right of the log and might be a little cryptic. But at least it IS an error message!

For example, here is a screen cap of the error message when the alias I used for the **parameter** in my previous example was not correct for THIS query (or for the sub-select statement of this query).

```

org.eclipse.birt.report.engine.api.EngineException:
Error happened while running the report.
at org.eclipse.birt.report.engine.api.impl...
at org.eclipse.birt.report.engine.api.impl...
.
.
.
org.eclipse.birt.report.data.oda.jdbc.JDBCException:
SQL statement does not return a ResultSet
object.

SQL error #1:java.sql.SQLException:
java.sql.SQLException: ORA-00904: "B"."STUD_ID":
invalid identifier

```

That is not UNCOMMON – easy to make a mistake in a complex query.

In this case, the correct alias was “a”, not “b”. I was able to make the quick fix, test again. **THEN fix the parameter**, and my report worked again with the search parameter in the LMS.

One more example follows with syntax for testing date parameters, amongst the trickiest of our missions!

– Here is my green commented text

```
/** and trunc(sc.req_dte) >= [FromDate] */  
/** and not (sc.req_dte - to_dsinterval( '+0 23:59:59' )) > [ToDate] */
```

And following is the code I used to test:

```
-- REMINDER - COMMENT OUT NEXT TWO LINES WHEN TESTING IS COMPLETE  
and trunc(sc.req_dte) >= '01-JUL-15'  
and not (sc.req_dte - to_dsinterval( '+0 23:59:59' )) > '30-JUN-16'
```

* AGAIN – I knew the RIGHT answer ahead of time: That I should get 5 records within this date range

And yes, I DO put in that highlighted statement to remind myself to fix this before I save, zip, and import. That is the most common mistake I make when editing queries!