
SQL scripts fail or run slow after optimizer changes

- **Article Type:** General
 - **Product:** Aleph
 - **Product Version:** 18.01
-

Description:

We implemented the changes outlined in the "Oracle Optimizer Changes for Aleph" document (see KB 14852). I understand these steps are intended to delete the statistics that Oracle 10 typically uses for cost-based optimization, as Aleph uses Oracle "hints", instead. While this seems to work okay with our online system, several of our SQL scripts now fail, run for many hours without returning results, or run until Oracle runs out of temp space (generating an ORA-01652 error). Is there a way to make our SQL scripts run successfully after implementing the optimizer changes?

Resolution:

I believe I have found a solution by adding a hint to the script, like this:

<cut text>

```
...  
FROM (SELECT /*+ DYNAMIC_SAMPLING(2) ALL_ROWS */  
SUBSTR(Z13_TITLE,1,(INSTR(Z13_TITLE,'/',3,1)-1)) TITLE,  
SUBSTR(Z00R_TEXT, INSTR(Z00R_TEXT,'(')+1, INSTR(Z00R_TEXT, ')')- (INSTR(Z00R_TEXT,'(')+1 )) RTIME,  
<cut text>
```

This seems to force dynamic sampling without requiring ongoing table analysis statistics. I ran a SQL script that had been failing since our changes and it now completes in 13 minutes.

Thanks to Mike Rogers, University of Tennessee, for sharing this information

- **Article last edited:** 10/8/2013