select IS\_BIND\_SENSITIVE from v$sqlarea where SQL\_ID = ‘2spuh8q1dshgv’

Per Mike advise

If avove N

you can go one of two ways. You can check all of the search arguments and see if any of them are on skewed values - in which case adding histogram stats on the column may help the optimizer choose more appropriate plans by marking it as bind sensitive

Or you can search for a baseline plan that gives good average performance

If above Y

cd /home/oracle/tls/sqlt

Pick better plan first

sqlplus as AID

@coe\_xfr\_sql\_profile.sql sqlidxxx planxxx

It should generate sql file as example below. Just run it

@coe\_xfr\_sql\_profile\_actkygvm7u4fy\_2992937973.sql

**Look for lowest Buffer Gets, Lowest Elapsed time...**

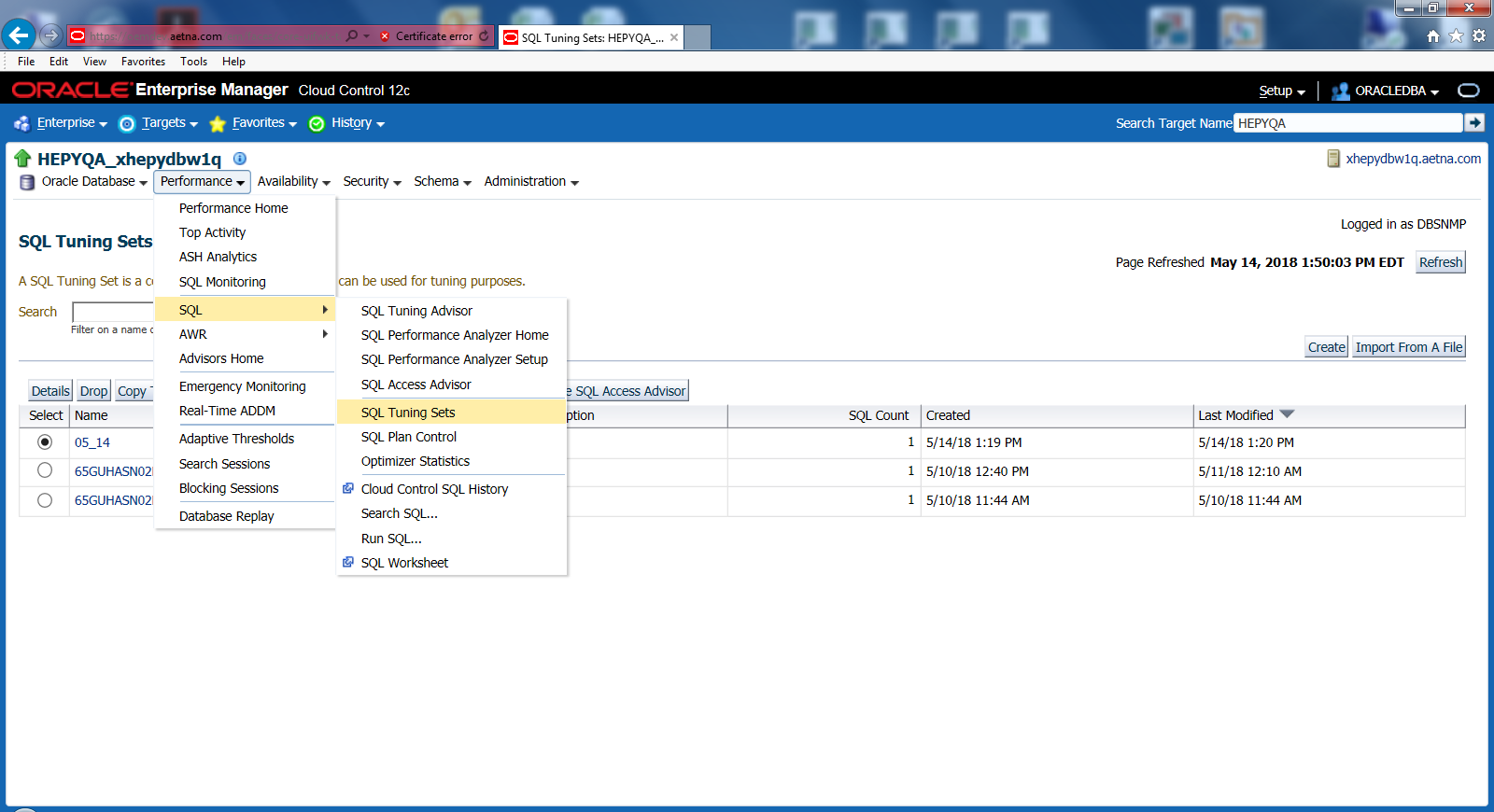
if we take # of execution into consideration last one looks better. Would we just divide Elapsed time / # of executions ?

1271883635 - Executions 1 Elapsed time 353.86 CPU time 138.98 buffer gets 44,350,419

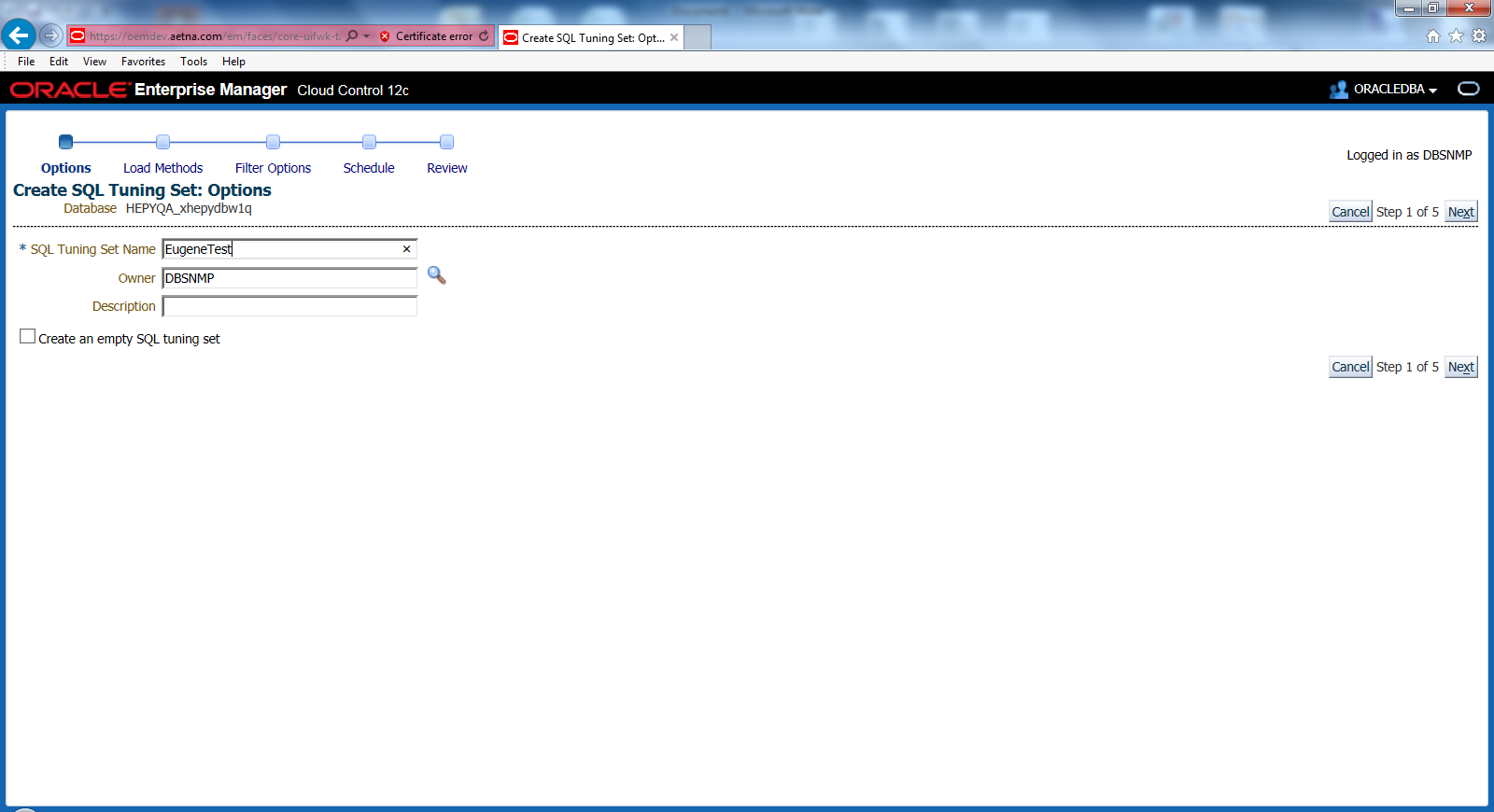
1658741038 - Executions 10 Elapsed time 55,523.15 CPU time 54,324.65 buffer gets 1,114,092,865

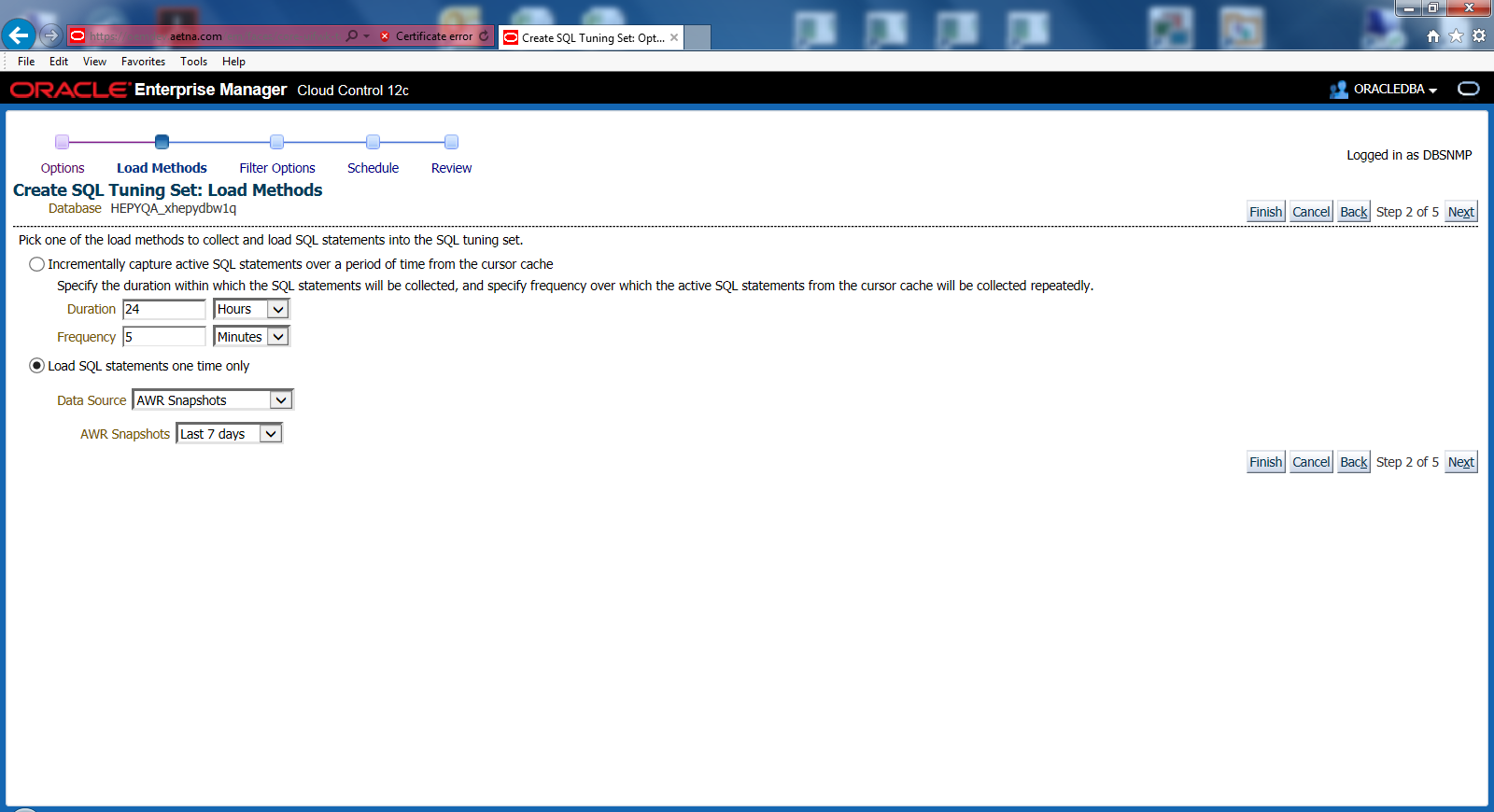
1765638362 - Executions 40 Elapsed time 5,835.67 CPU time 1,374.79 buffer gets 563,431,706

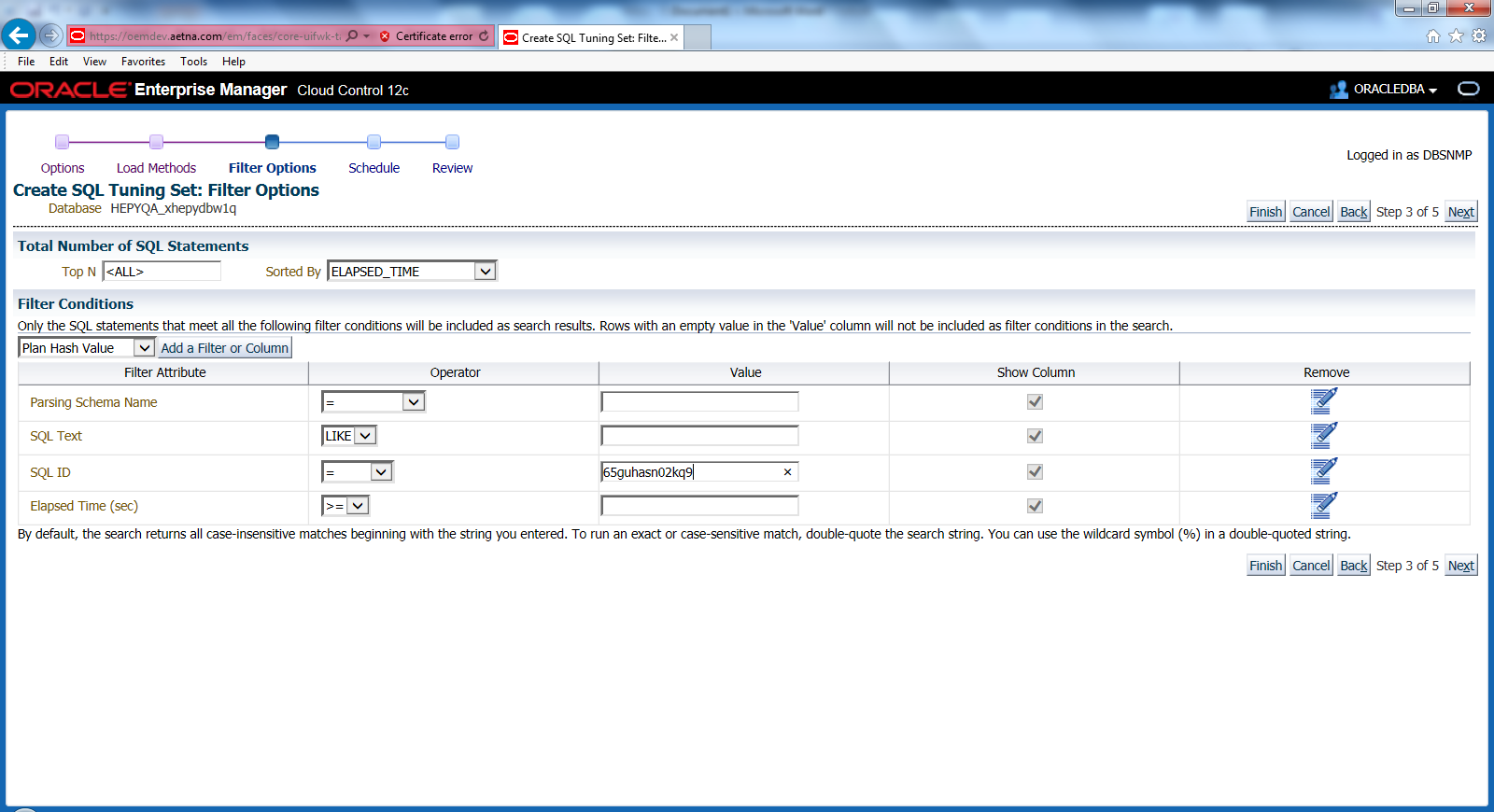
Performance🡪 SQL 🡪SQL Tuning Sets

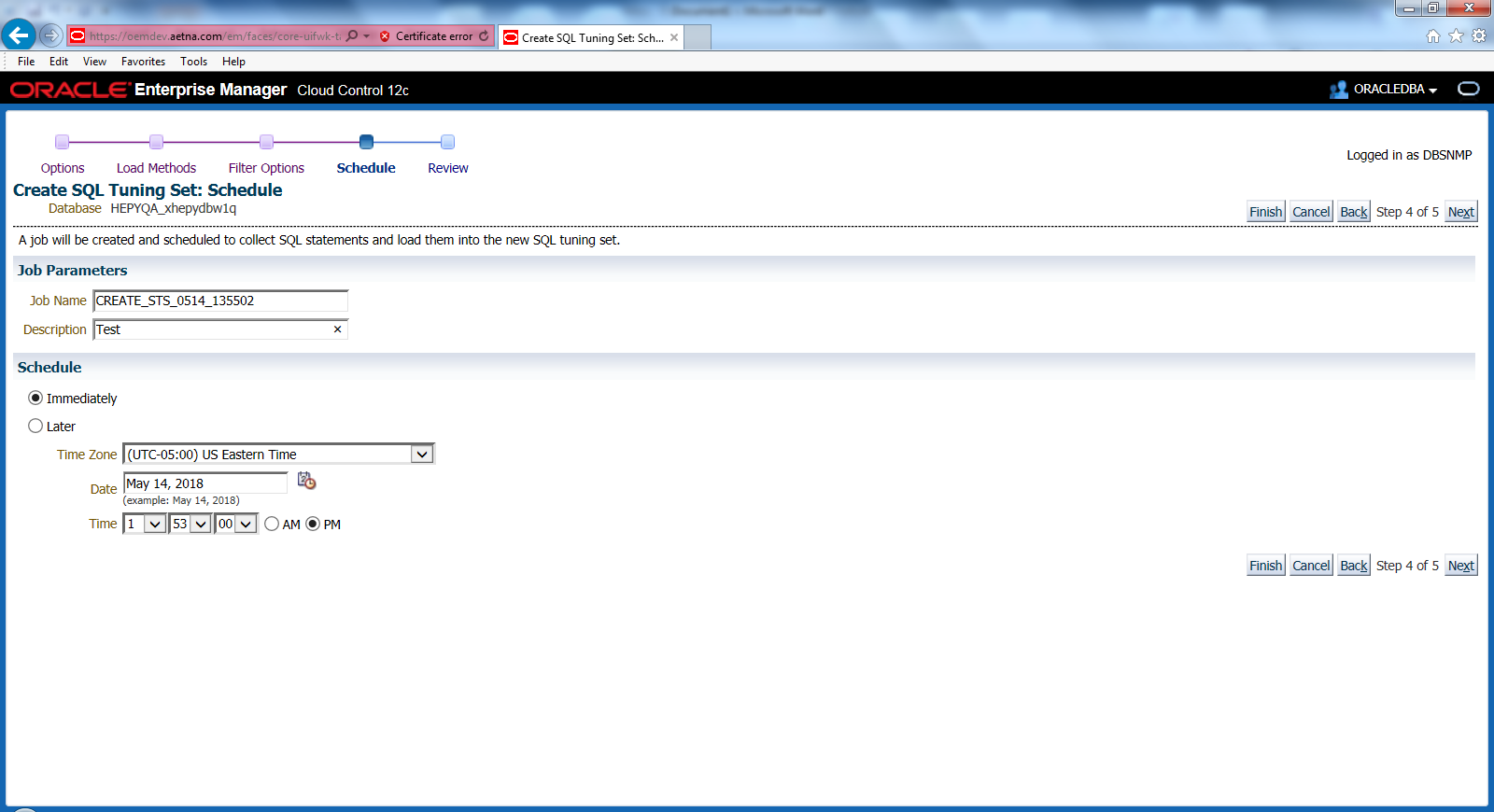


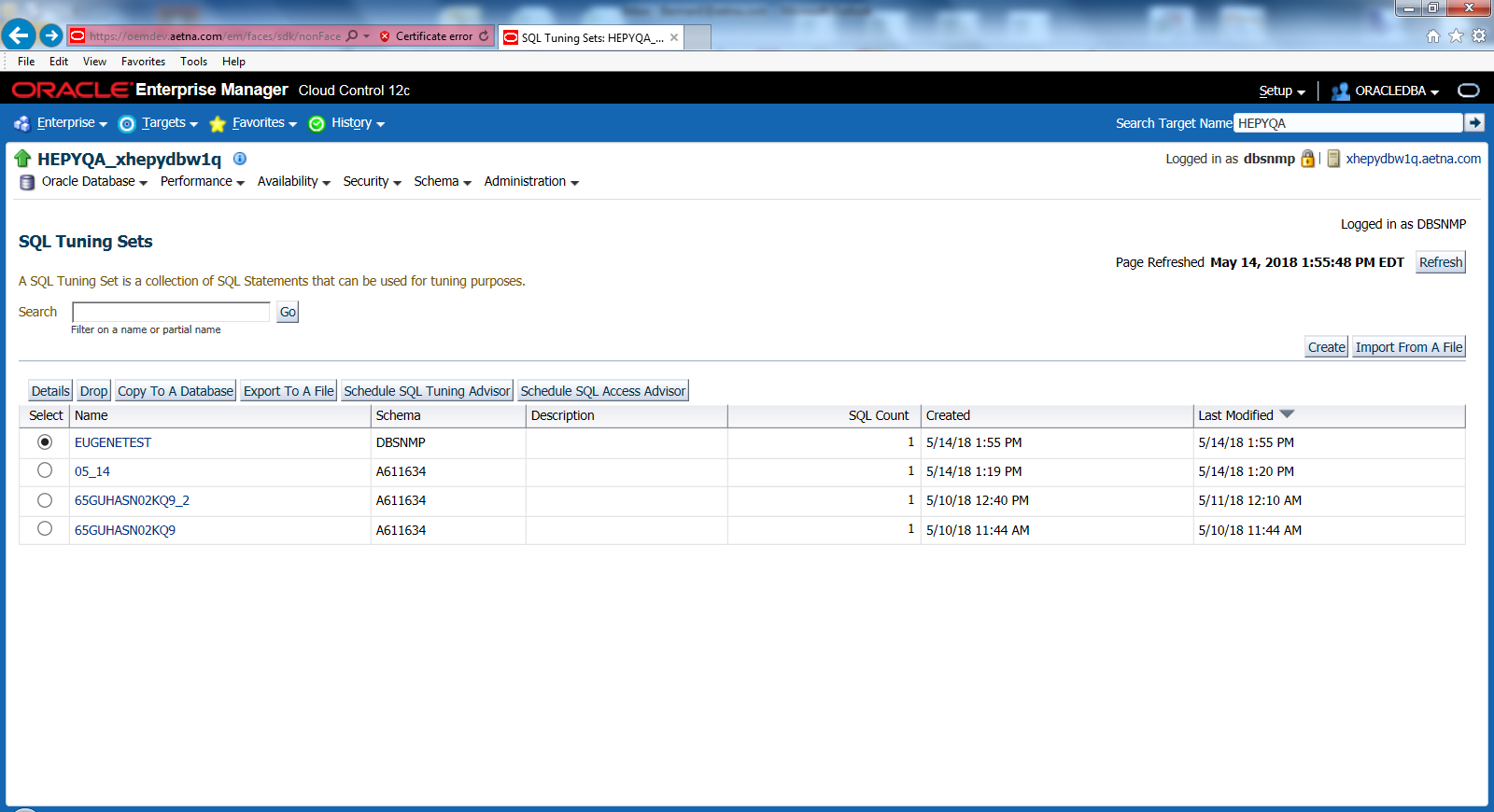
Click Create Button



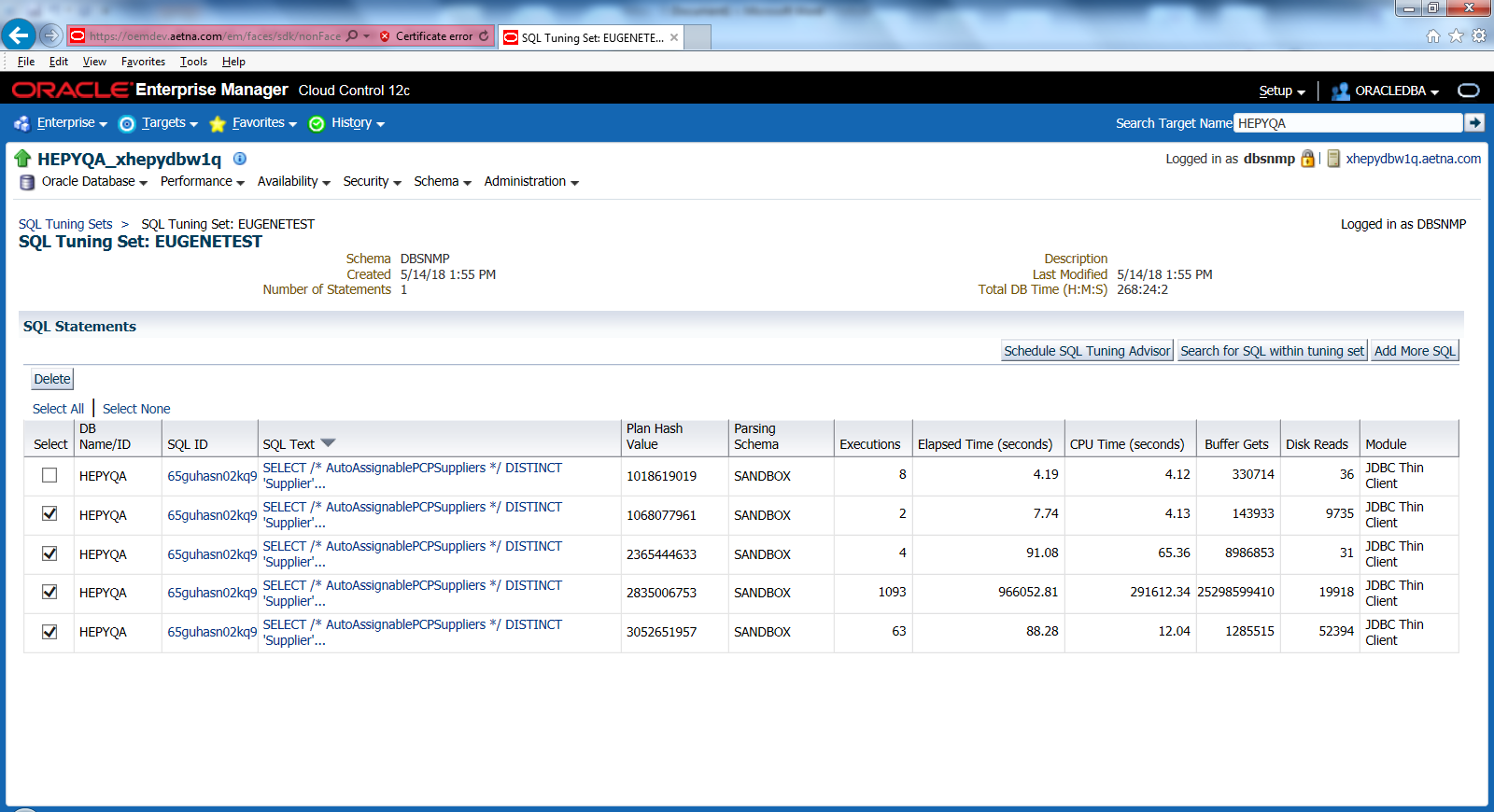


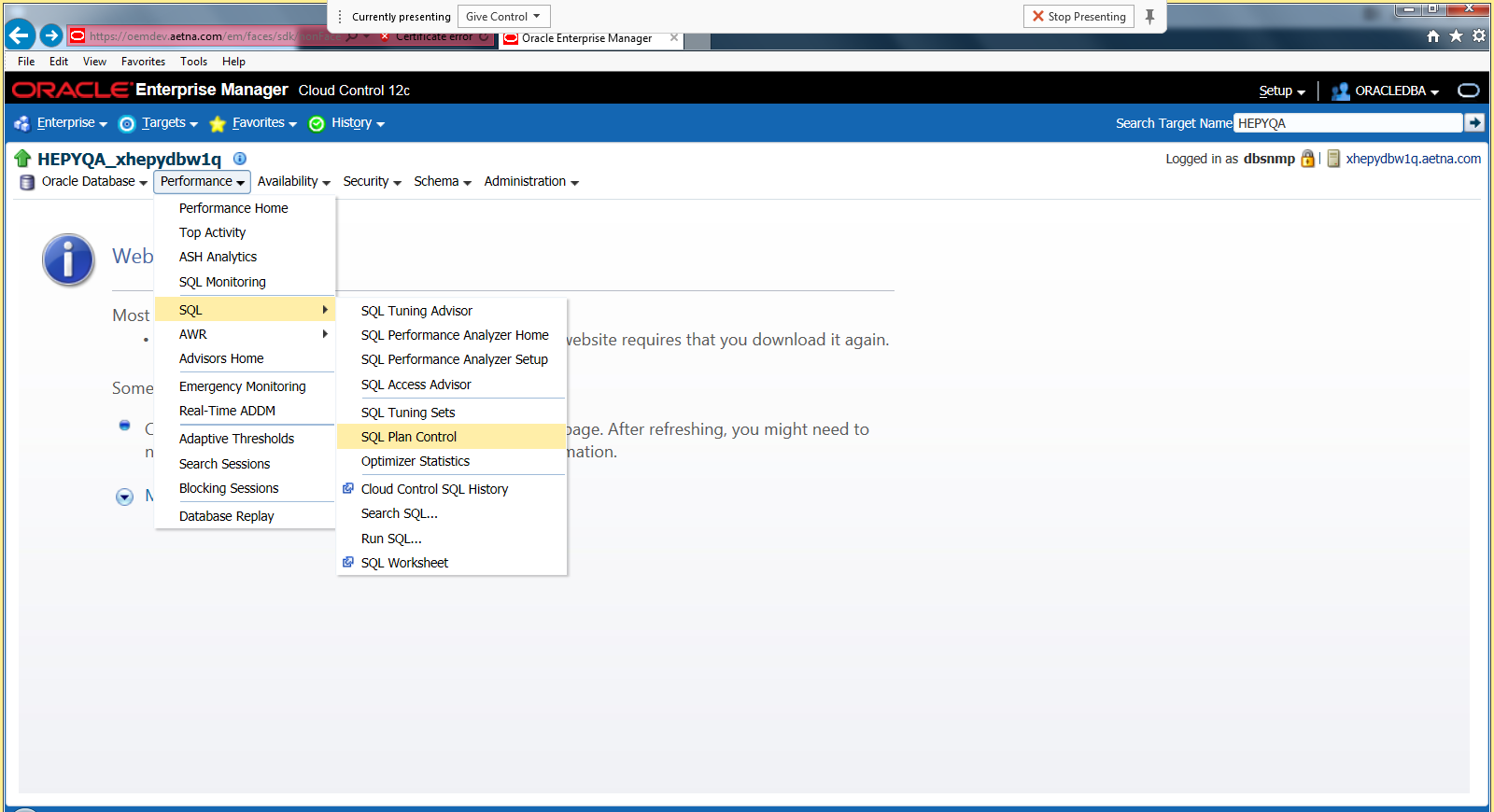




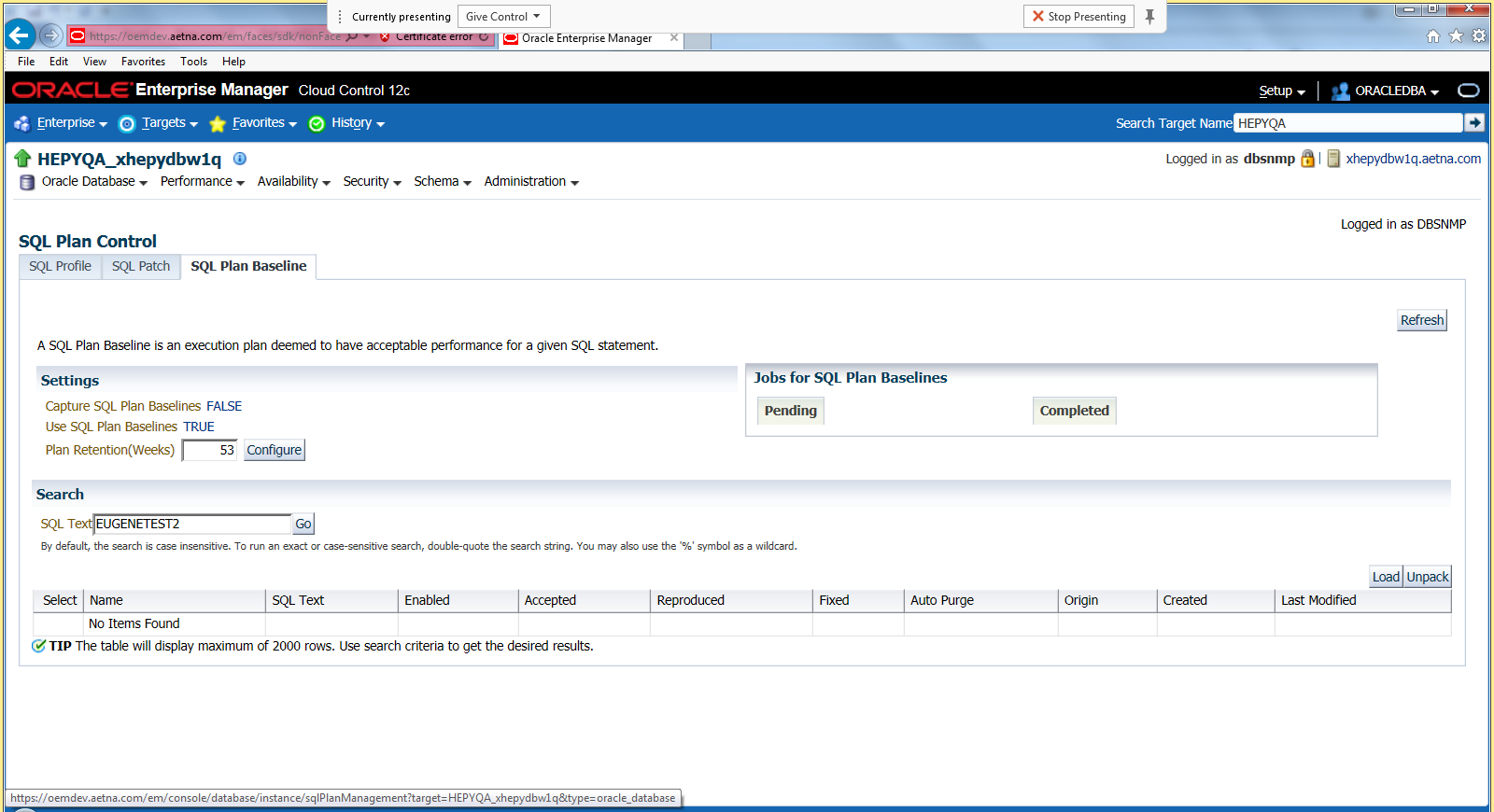


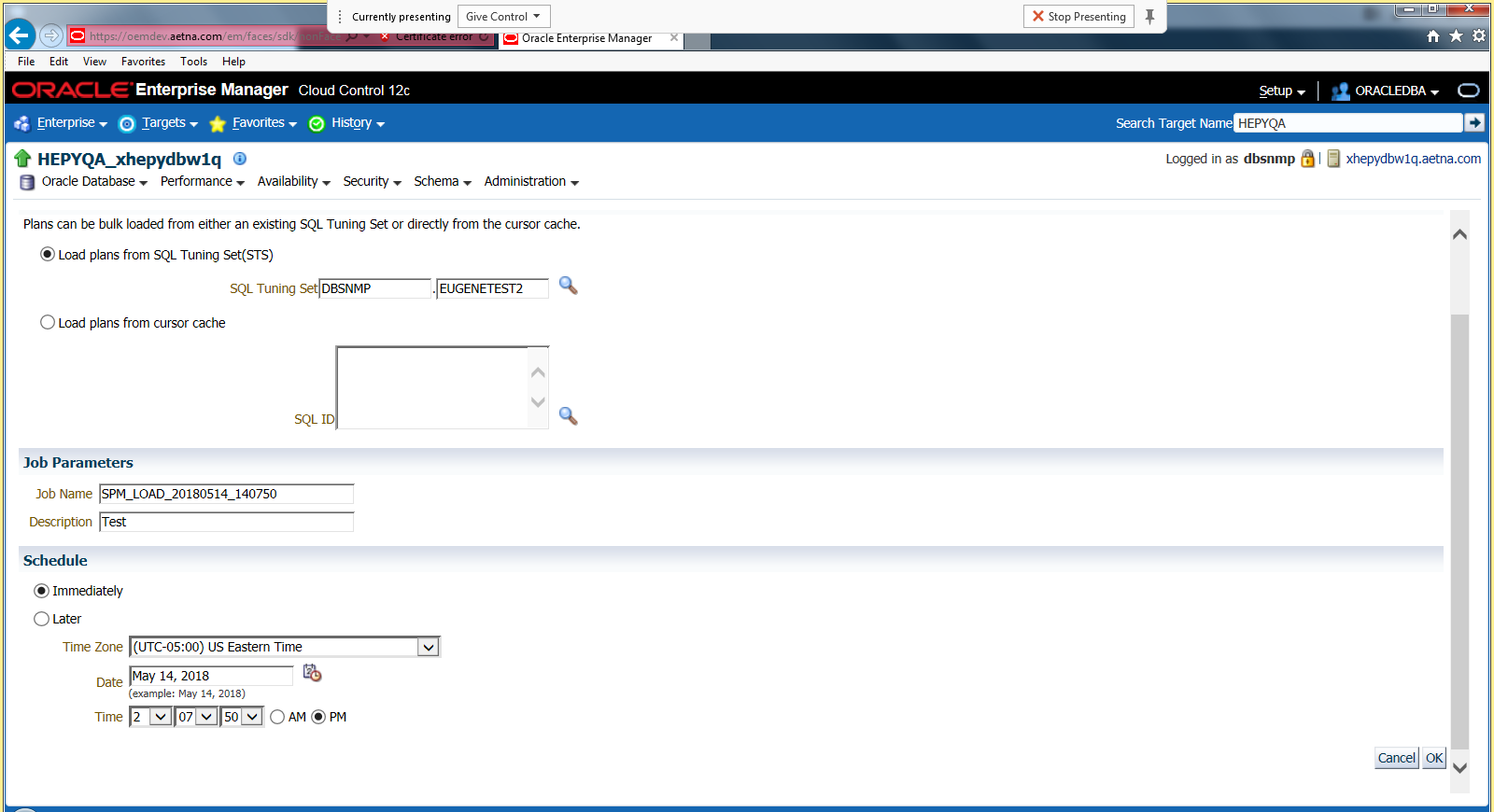
Delete Hash plans that not desirable. Select them and click Delete



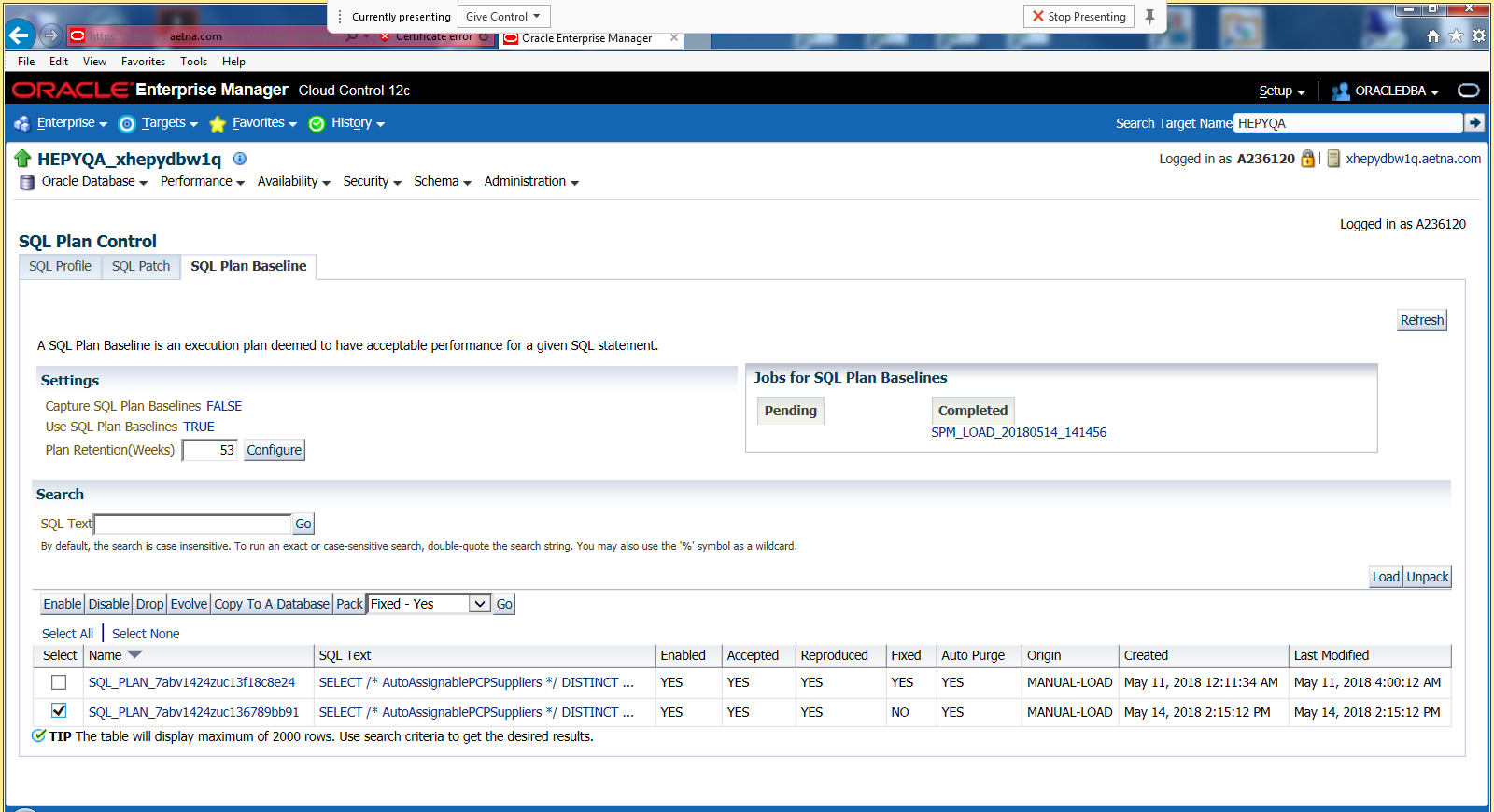


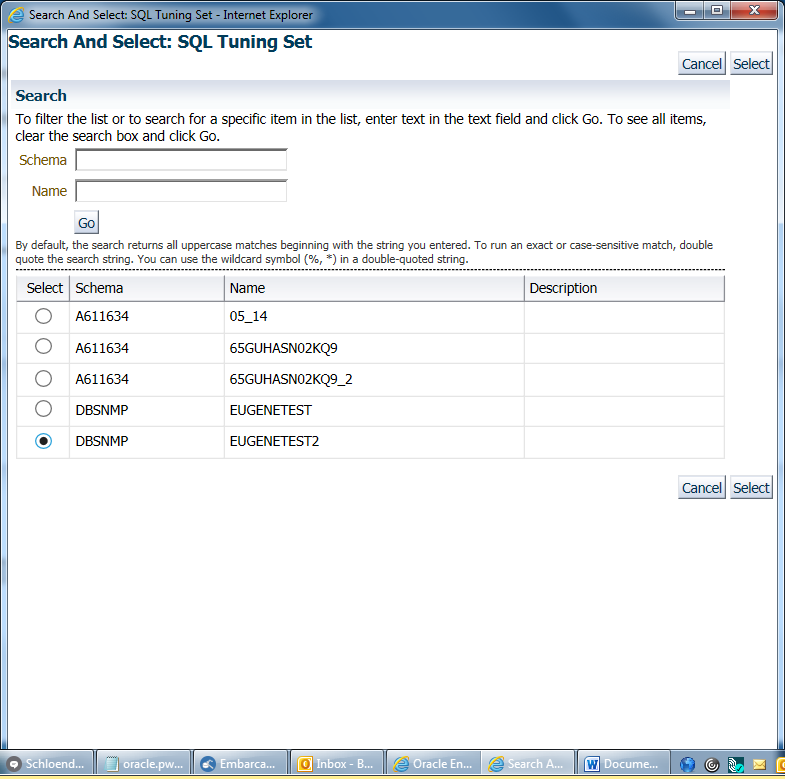
Select SQL Baseline tab and click Load





Select and click Fixed yes and click GO





!! Do this before next query run

select address, hash\_value, executions, loads, version\_count, invalidations, parse\_calls,sql\_plan\_baseline

from v$sqlarea

where sql\_id = ' a7fgy9kkxntq5;

00000000EC82C5B8 313380221 1 4 1 3 1 [NULL]

00000000EC82C5B8 313380221 1 5 1 5 1 [NULL]

exec dbms\_shared\_pool.purge ('00000000884D49F0, 2242762283','C');