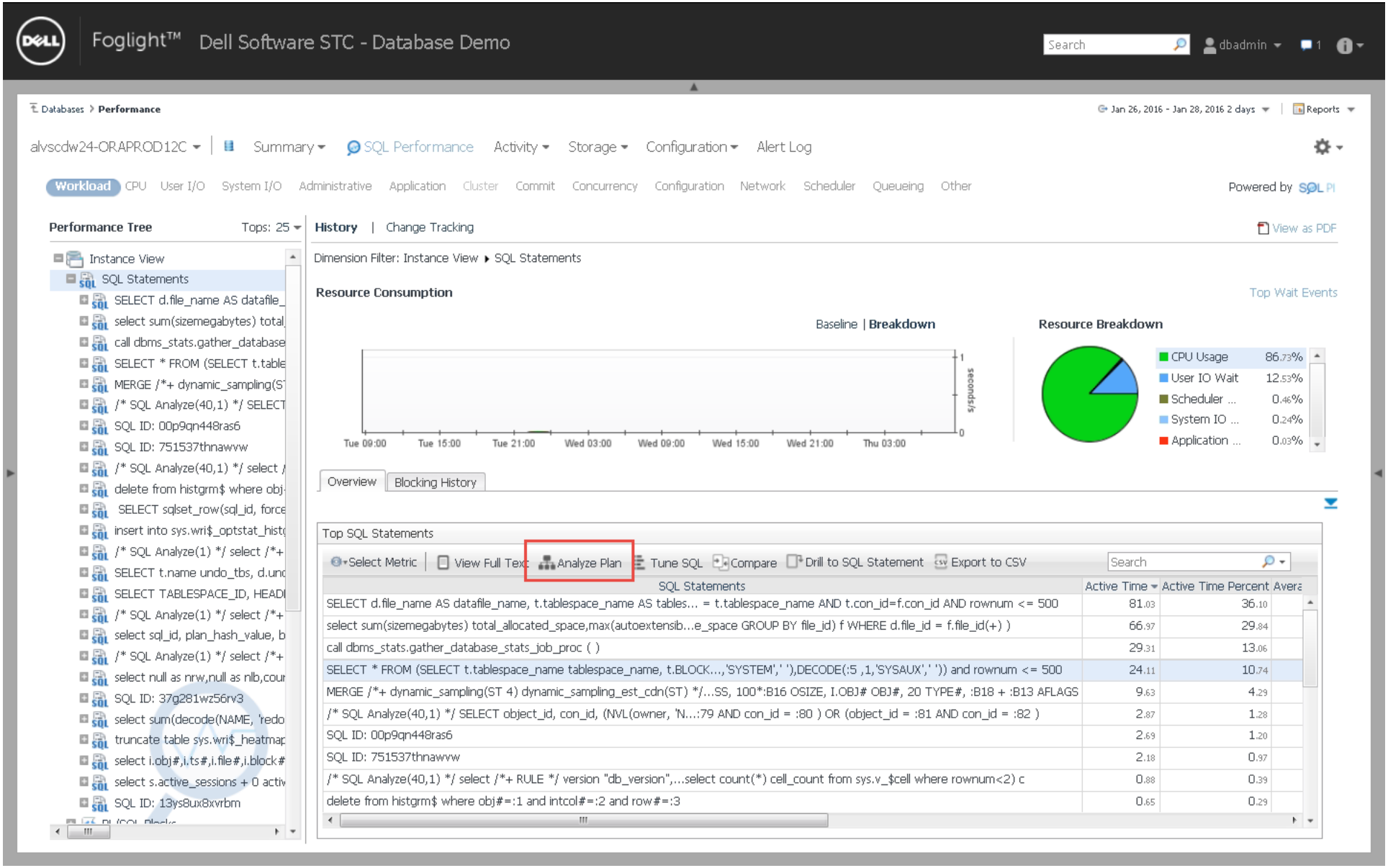
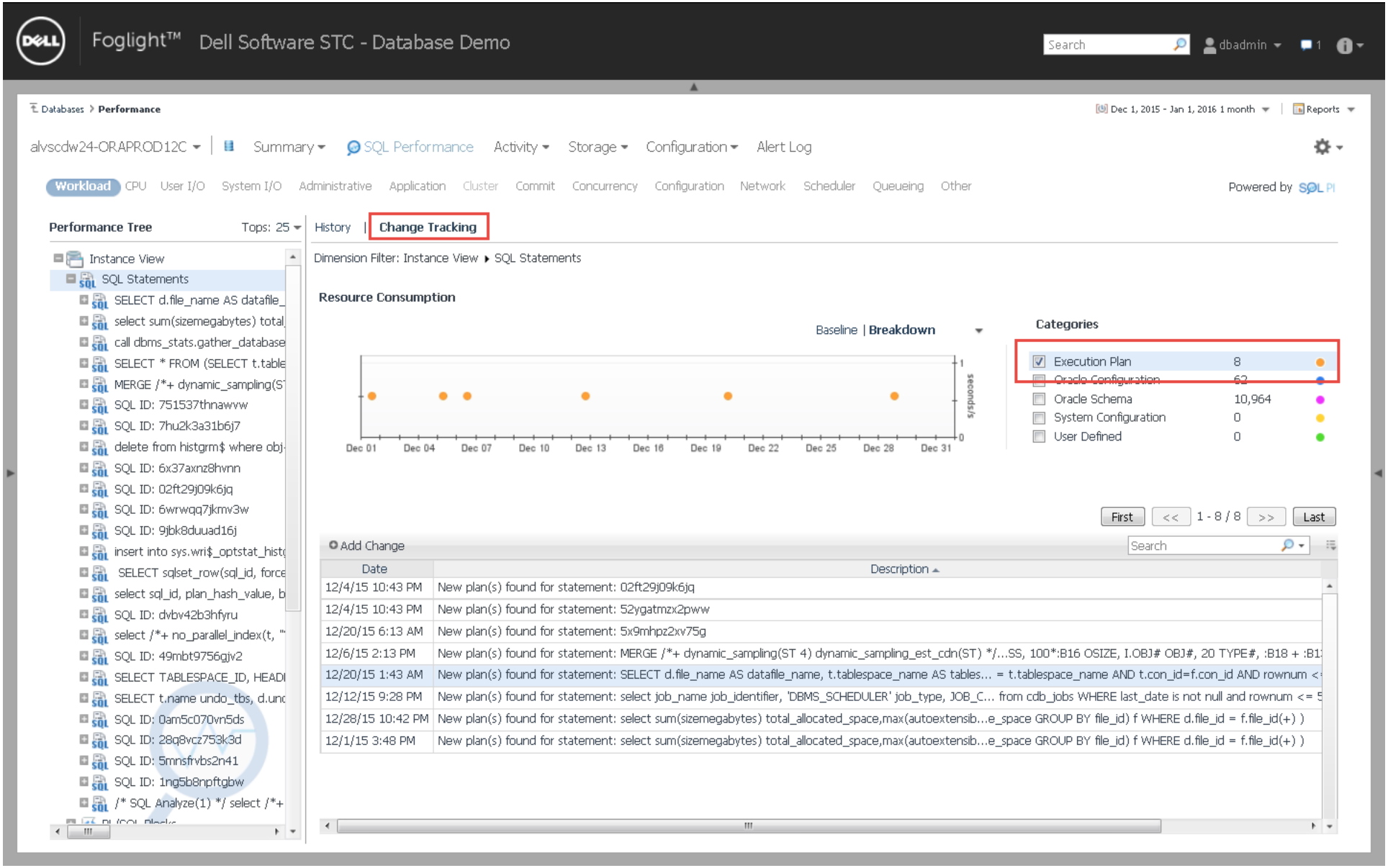
What Did that SQL Query Do?

Oracle Edition

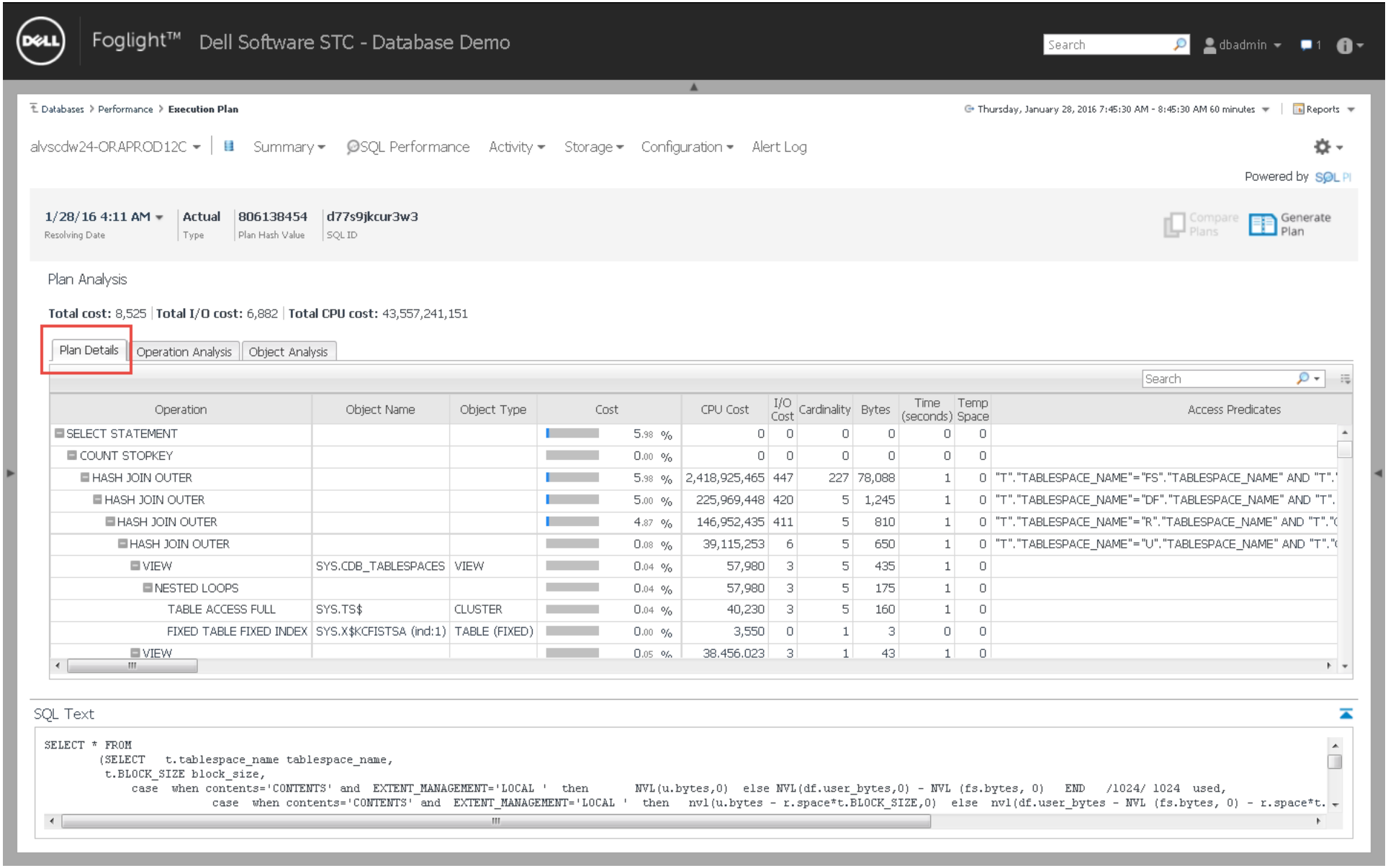
When investigating database performance issues related to SQL queries, it is often useful to know the execution plan of the statement. However, unless it was executed recently, we only have the execution plan that we can generate via the explain plan command. This may or may not be the actual plan that was used when the SQL ran. Using Foglight Performance Investigator for Oracle, we can easily go back and review *actual* plans that were used when the statement was executed. In addition, we can compare different plans as they change over time due to database object changes, optimizer statistics, etc.



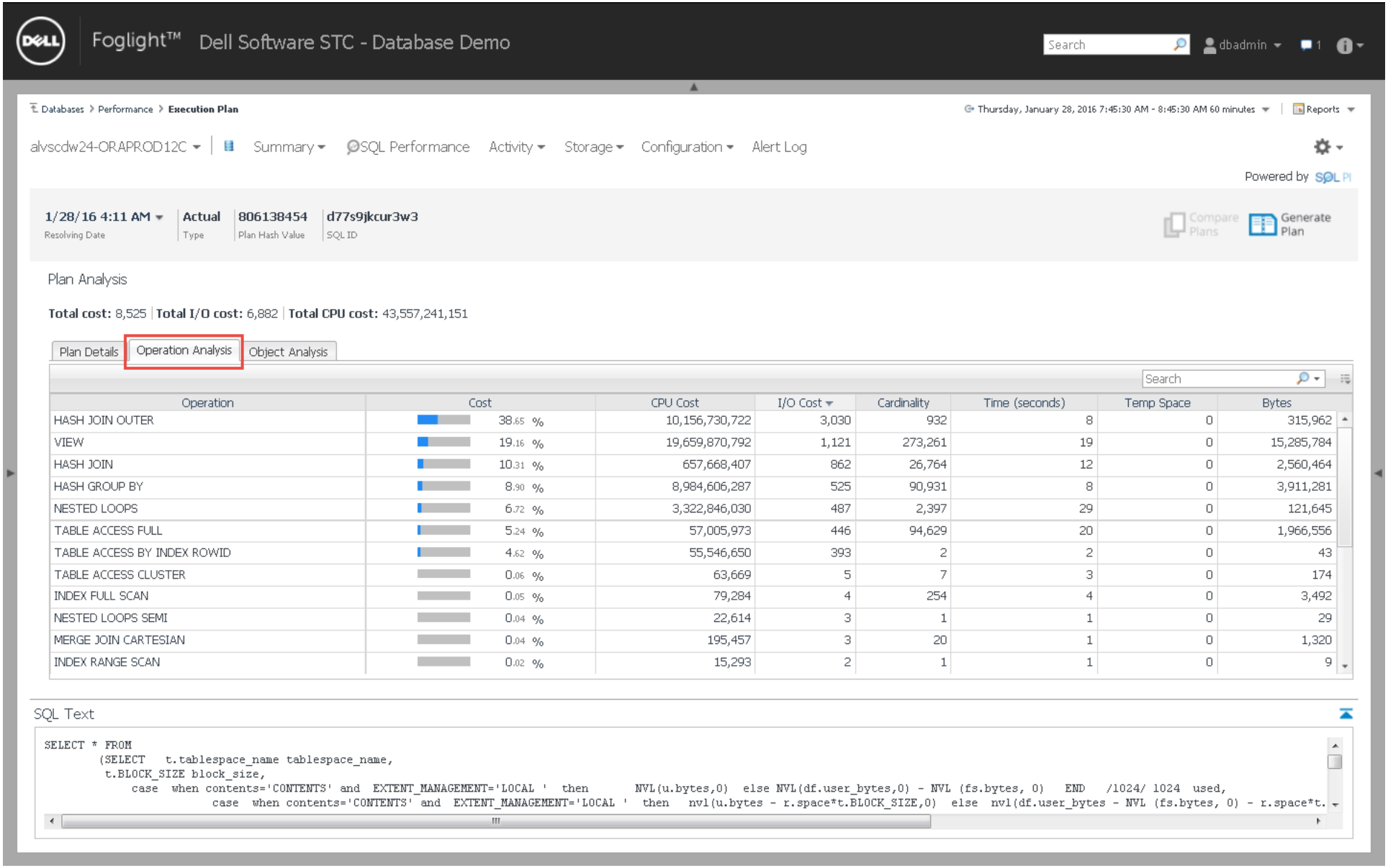
In the screenshot above, a time period was isolated and the top SQL statements sorted by Active Time are shown. To review the execution plans, all you need to do is pick a SQL and click “Analyze Plan”.

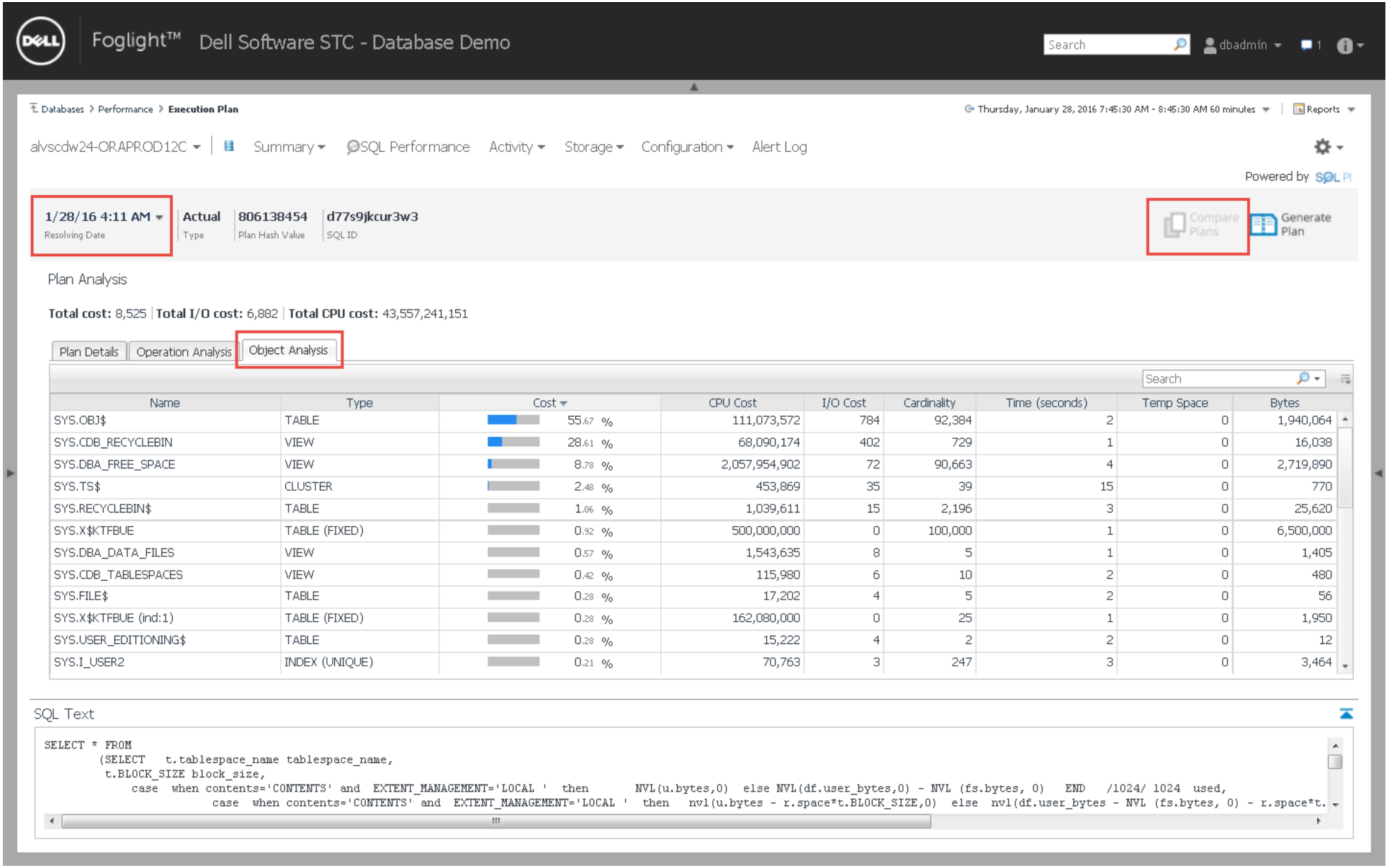


You can also review changes to execution plans by switching to “Change Tracking” from the “History” view. Execution plan changes are overlaid on the timeline and workload baseline graph, and you can simply click into one of the changes in the grid.

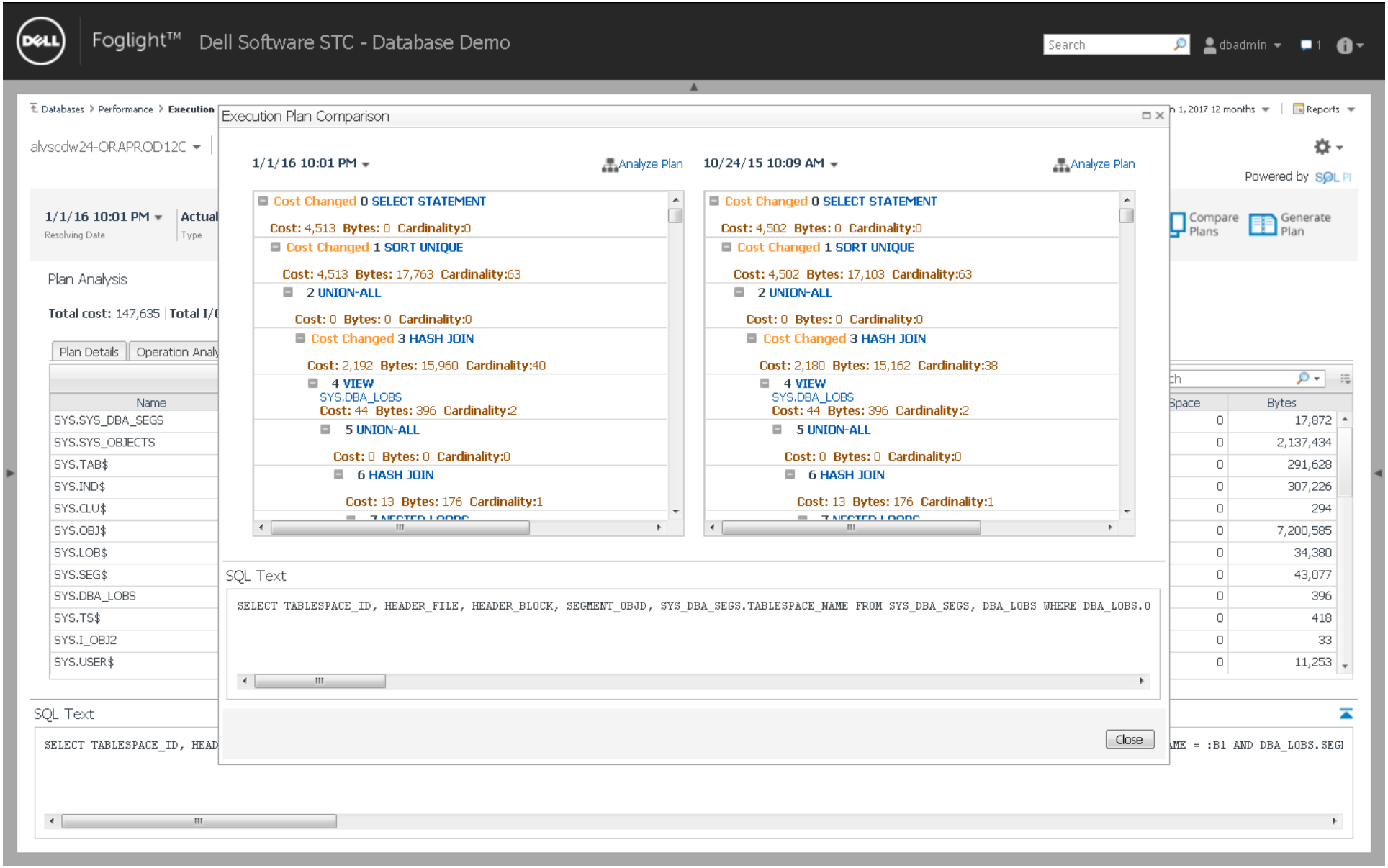


This drilldown shows you Plan Details, Operator Analysis and Object Analysis on separate tabs.

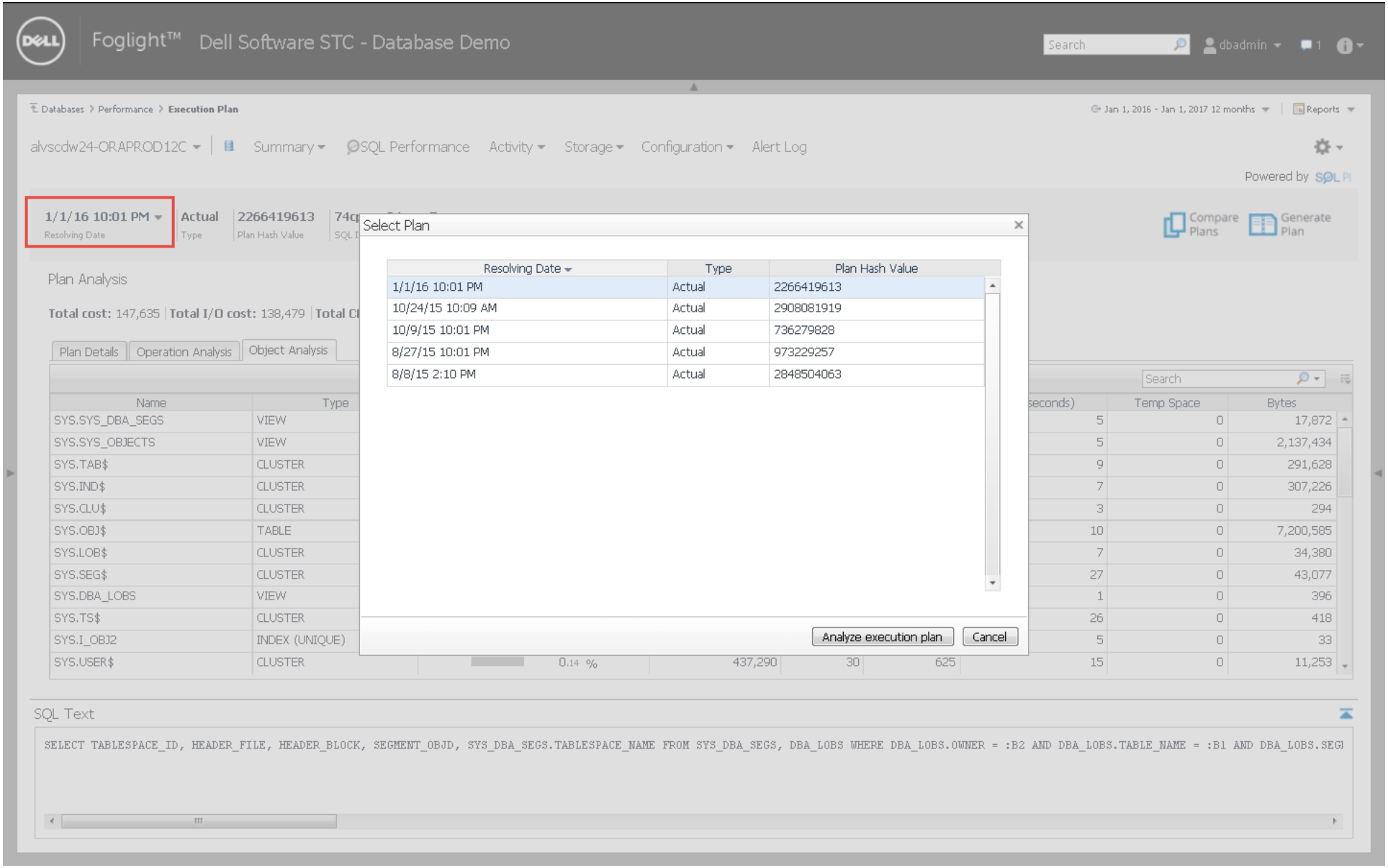

Foglight™ Dell Software STC - Database Demo
Search
dbadmin
1
Databases > Performance > Execution Plan
Thursday, January 28, 2016 7:45:30 AM - 8:45:30 AM 60 minutes
Reports
alvscdw24-ORAPROD12C
Summary
SQL Performance
Activity
Storage
Configuration
Alert Log
Powered by SQL PI
1/28/16 4:11 AM
Resolving Date
Actual
Type
806138454
Plan Hash Value
d77s9jkcur3w3
SQL ID
Compare Plans
Generate Plan
Plan Analysis
Total cost: 8,525 | Total I/O cost: 6,882 | Total CPU cost: 43,557,241,151
Plan Details
Operation Analysis
Object Analysis
Search
Operation
Cost
CPU Cost
I/O Cost
Cardinality
Time (seconds)
Temp Space
Bytes
HASH JOIN OUTER 38.65 % 10,156,730,722 3,030 932 8 0 315,962
VIEW 19.16 % 19,659,870,792 1,121 273,261 19 0 15,285,784
HASH JOIN 10.31 % 657,668,407 862 26,764 12 0 2,560,464
HASH GROUP BY 8.90 % 8,984,606,287 525 90,931 8 0 3,911,281
NESTED LOOPS 6.72 % 3,322,846,030 487 2,397 29 0 121,645
TABLE ACCESS FULL 5.24 % 57,005,973 446 94,629 20 0 1,966,556
TABLE ACCESS BY INDEX ROWID 4.62 % 55,546,650 393 2 2 0 43
TABLE ACCESS CLUSTER 0.06 % 63,669 5 7 3 0 174
INDEX FULL SCAN 0.05 % 79,284 4 254 4 0 3,492
NESTED LOOPS SEMI 0.04 % 22,614 3 1 1 0 29
MERGE JOIN CARTESIAN 0.04 % 195,457 3 20 1 0 1,320
INDEX RANGE SCAN 0.02 % 15,293 2 1 1 0 9
SQL Text
SELECT * FROM
(SELECT t.tablespace_name tablespace_name,
t.BLOCK_SIZE block_size,
case when contents='CONTENTS' and EXTENT_MANAGEMENT='LOCAL ' then NVL(u.bytes,0) else NVL(df.user_bytes,0) - NVL (fs.bytes, 0) END /1024/ 1024 used,
case when contents='CONTENTS' and EXTENT_MANAGEMENT='LOCAL ' then nvl(u.bytes - r.space*t.BLOCK_SIZE,0) else nvl(df.user_bytes - NVL (fs.bytes, 0) - r.space*t.



If there is more than 1 historical execution plan available, the “Compare Plans” button will be active. You can click into that drilldown to compare historical plans. Note that there will only be a historical plan stored for the first time it was detected as being changed.



You can then pick from the date/time pulldowns different historical plans to compare.



You may also analyze a specific plan by picking from the date/time dropdown shown above. This is great functionality to have because it lets you review plans that may have executed weeks or months ago, and also makes it very easy for you to detect when a plan has changed. If you also manage SQL Server databases, I've posted a similar tutorial on the site as well.