Run a tuning job

As you add SQL statements to the job on the Overview tab of the tuner, tuning-supported DML statements are parsed from the statements and added to the Overview tab in preparation for the tuning function execution.

Each tuning source statement is listed by Name, Schema, Text, Tables and Views. Additionally, each statement will have Time and Analysis values that approximate how efficiently they execute on the specified data source.

In the Generated Cases area of the Overview tab of a tuning job, the Cost and Execution Statistics columns let you compare the relative efficiency of SQL statements or statement cases. While the explain plan Cost for a statement or case is calculated when you add SQL to a tuning job, the Elapsed Time and Execution Statistics (and Other Execution Statistics columns, if available) columns are not populated until you execute that statement or case.

If the Tuning Status Indicator indicates that a statement or case is ready to execute, you can execute one or more statements on the Overview tab. Alternatively, the Tuning Status Indicator may show that you have to correct the SQL or set bind variables before you can execute.

Once the tuning job has run, the Overview tab provides a series of cases, per statement, that you can select and modify based on your results.

In some cases, automatic case generation might be disabled (via the Preferences panel). If this is true, or if you have otherwise modified the Generated Cases table and can no longer generate a specific case, you can instead explicitly generate a case for specific statements.



To execute a tuning job

- 1. Ensure you have registered and selected a data source. For more information, see Register data sources and Specify a data source.
- 2. Ensure you are connected to the database by double clicking the database name in the Data Source Explorer.
- 3. Click the tuning icon on the toolbar, or click File > New > Tuning Job.
- 4. On the Overview tab, specify the SQL you want to tune:
- Modify the number of times to execute each statement in the **Execute each generated case** field at the top right of the tuner, as needed.
- 5. Click the execution button [7] on the right side of the case generation field.
- 6. The tuning job runs, exacting and analyzing each statement and providing values in the appropriate columns.

To explicitly generate a case for a specific statement

- 1. Ensure you are connected to the database by double clicking the database name in the Data Source Explorer.
- 2. Click the Overview tab.
- 3. In the Generated Cases area, right-click in the Name field of a statement or transformation case and select Generate Cases from the context menu, or click the Overview Run Job icon. The specified case is generated.

To view the generated cases for a specific statement

1. In the Tuning Statements area, click the checkbox to the left of the tuning source statement name.

A check mark appears in the checkbox and the cases displayed in the Generated Cases area are filtered to display only those cases related to the selected source statement.

• 01	verview Ana	lysis								
0	verview Ler	or detected							(a)	0 🕆 - 📬
unii	ng Statements	Ge	nerate cases	Perform de	tail analysis	V	Execute each gers	erated case 3	🗧 times	
			Statemant				Time	Analy	sis	
	Name	Schema	Text	Tables	Views	Elaosed	(s) Improved (s)	Cases	Indexes	
1	SELECT 3	SYS	select from sys.job\$					0		
5	SELECT 4	SYS	select from fet\$, ts\$					0	21	
		-	1.17					0		
1	SELECT 5	SYS	select from pending_trans;					0		
ien i	SELECT 5	SYS SYS	select from sys.obi\$.					0		
ien	SELECT 5	SYS SYS SQL Stat	seet from pending_trans; select from svs.obi\$; ements and Cases		×.	Tost	»Executiistics))))	Ther Execution Stat	: Istics
ien	SELECT 5 SELECT 6 erated Cases	SYS SYS SQL Stat	select from pending_trans; select from svs.obi\$; ements and Cases Text	·	>> c	Cost labe	>>Executiistics Elapsed Time (s)	0 >>> C Physical Reads	Other Execution Stat	istics CPU Time (s)
ien	V SELECT 5 SELECT 6 erated Cases Name B 62 SELECT 1	SYS SYS SQL Stat	select from pending_trans; select from sys.obj\$, ements and Cases Text select from SYSTEM.OEP#_AQ	CALL	>> (>> (Tost lake	>>Executiistics Elapsed Time (s) 0.56	>>> c Physical Reads	Other Execution Stat Logical Reads	latics CPU Time (s) 0.
ien	SELECT 5 SELECT 6 erated Cases Neme B & SELECT 1 NDEX_COM	SYS SYS SQL Stat	select from svs. obi\$. select from svs. obi\$. ements and Cases Text select from SYSTEM.DEP\$_AQ	CALL	>> (>> (Cost Value 1.0	>>Executiistics Elapsed Time (s) 0.56 0.63	>>> c Physical Reads 0 0	Other Execution Stat Logical Reads 1 8	istics CPU Time (s) 0. 0.
を ien 和目目	Ø SELECT 5 Ø SELECT 6 Prated Cases Neme B ∰ SELECT 1 NDEX_COM	SYS SYS SQL Stat BDNE	select from panding_transi select from sys.obi\$ ements and Cases Test select from SYSTEM.DEPB_AQ	CALL	>> (>> (Cost /alue 1.0 826.0	>>Executiistics Elapsed Time (s) 0.63 0.63 0.87	>> c Physical Reads 0 0 0	Other Execution Stat Logical Reads 0 11 0 8	istics CPU Time (s) 0./ 0./ 0./
を ien 和目目目	SELECT 5 SELECT 6 Prated Cases Name SelECT 1 NDEX_COM NDEX NDEX	SYS SYS SQL Stat BDNE	select from panding_transi select from sys.obi\$. ements and Cases Text select from SYSTEM.DEPB_AQ	CAIL	>> (>> (Cost (alue 1.0 826.0	≫Executiistics Elapsed Time (s) 0.63 0.87 0.77	>>> c Physical Reads 0 0 0 0 0 0	Other Execution Stat Logical Reads 9 11 9 8 9 4	Istics CPU Time (s) 0. 0. 0. 0.
もん ien 加加目目目	SELECT 5 SELECT 6 erated Cases Name B & SELECT 1 INDEX_COM INDEX BULE SELECT 3	SYS SYS SQL Stat BDNE	select from sys.obi\$ enerts and Cases Text select from SYSTEM.DEP5_AQ select from sys.jeb\$	CAIL	× (Cost /alue 1.0 826.0	>>Executiistics Elepsed Time (s) 0.56 0.63 0.87 0.77	>>> C Physical Reads 0 0 0 0 0 0	Other Execution State Logical Reads 0 111 0 8 0 4	istics CPU Time (s) 0, 0, 0,
	☑ SELECT 5 ☑ SELECT 6 erated Cases ■ SELECT 1 ■ INDEX_COM ■ INDEX_COM ■ NULE ★ SELECT 3 ★ SELECT 4	SYS SYS SQL Stat BDNE	select from sys.obi\$ emerts and Cases Text select from SYSTEM.DEP\$_AQ select from SYSTEM.DEP\$_AQ select from sys.job\$	CALL	>> () >> () >> ()	Cost /akue 1.0 826.0	>>Executiistics Elapsed Time (s) 0.63 0.63 0.87 0.77	>>> c Physical Reads 0 0 0 0 0	Other Execution Stat Logical Reads 0 111 0 8 1 4	Istics CPU Time (s) 0. 0. 0. 0.
	SELECT 5 SELECT 6 erated Cases Read B ← SELECT 1 INDE_COM INDE_COM RULE SELECT 3 - SELECT 4 - SELECT 5	SYS SYS SQL Stat BDAE	select from panding_transf select from sys.obi\$. ments and Cases Text select from SYSTEM.DEP5_AQ select from sys.job\$ select from sys.job\$ select from sys.job\$ select from sys.job\$	CAIL	>> () >> ()	Tost /alue 1.0 826.0	WExecutiistics Elapsed Time (c) 0.65 0.63 0.87 0.87	>> C Physical Reads 0 0 0 0 0 0 0 0	Rher Execution Stat Logical Reads 0 111 8 0 4	3 CPU Time (s) 0. 0. 0. 0.
Sen Jugggggggg	SELECT 5	SYS SYS SQL Stat	select from pending_transi select from sys.obi\$. ments and Cases Text select from SYSTEM.DEP\$_AQ select from sys.job\$ select from sys.job\$ select from for4, to\$ select from pending_trans\$	CALL	× 4	Tost Value 1.0 826.0	₩Executiistics Elapsed Time (c) 0.56 0.63 0.87 0.87	>> c Physical Reads 0 0 0 0	The Execution State Logical Reads 0 11 0 8 0 4	istics CPU Time (