

Tuning Your SQL Statements

SQL Tuner provides an easy and optimal way to discover efficient paths for queries that may not be performing as quickly or as efficiently as they could be. Tuner enables the optimization of poorly-performing SQL code through the detection and modification of execution paths used in data retrieval. This is primarily performed through hint injection, and index and statistics analysis.

The screenshot shows the SQL Tuner interface for the user *Matt.tun. The interface is divided into two main sections: Overview and Analysis. The Overview section is currently active, displaying a table of statements and a section for Generated Cases.

Tuning Status: ☒ Generate cases ☒ Perform detail ai ☒ Execute each generated case 3

Statement						Time	
	Name	Schema	Text	Tables	Views	Elapsed (s)	Improved (s)
☒	SELECT 1	MOVIES	select from	5	0	0.63	0.63
☒	SELECT 2	MOVIES	select from	4	0	0.04	0.04
☒	SELECT 3	MOVIES	select from	8	0	0.09	0.09
☒	SELECT 4	MOVIES	select from	5	0	10.10	10.10
☒	SELECT 5	MOVIES	select from	0	0	0.00	0.00

Generated Cases

SQL Statements and Cases			>> Cost	>>Executi...istics	>>
	Name	Text	Value	Elapsed Time (s)	Physic
☒	SELECT 1	select from MOVIES.customer,	21.0	0.63	
☒	SELECT 2	select from MOVIES.customer,	11.0	0.04	
☒	SELECT 3	select from MOVIES.customer,	25.0	0.09	
☒	SELECT 4	select from MOVIES.customer,	22.0	10.10	
☒	SELECT 5	select from MOVIES.customer,	25.0	0.08	
☒	SELECT 6	select from HR.EMP_DETAILS_VIEW			
☒	SELECT 7	select from MOVIES.customer,	21.0	0.07	

For example, different joining methods such as nested loops or hash joins can be used and will be tested, as appropriate. Tuning will select alternate paths, and enable you to change the original path to one of the alternates. Execution paths slower than the original path are eliminated, which enables you to select the quickest of the returned selections and improve query times, overall.

This enables the DBA to correctly optimize queries in the cases where the native optimizer failed.