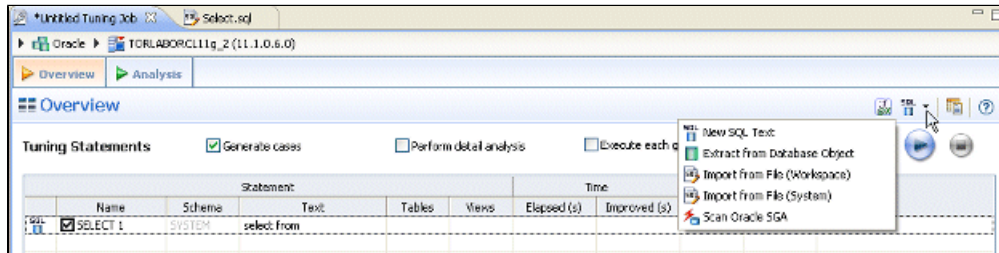


Understanding the Overview tab

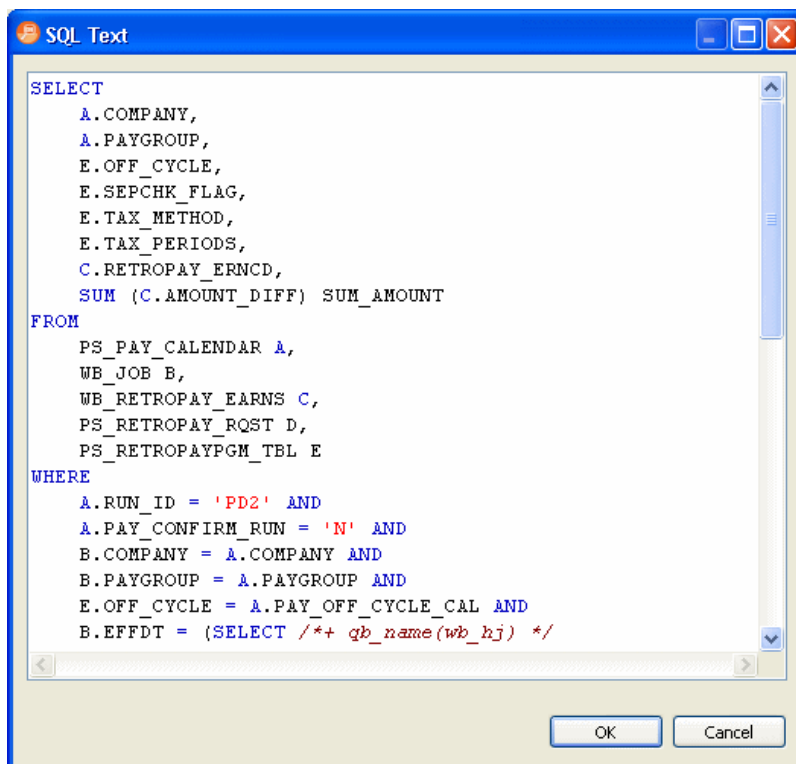
Inputting SQL to tune


Click the SQL button on the **Overview** tab to specify the source of SQL statements you want to tune.



- **New SQL Text:** From the SQL button menu, select **New SQL Text**, and then copy/paste

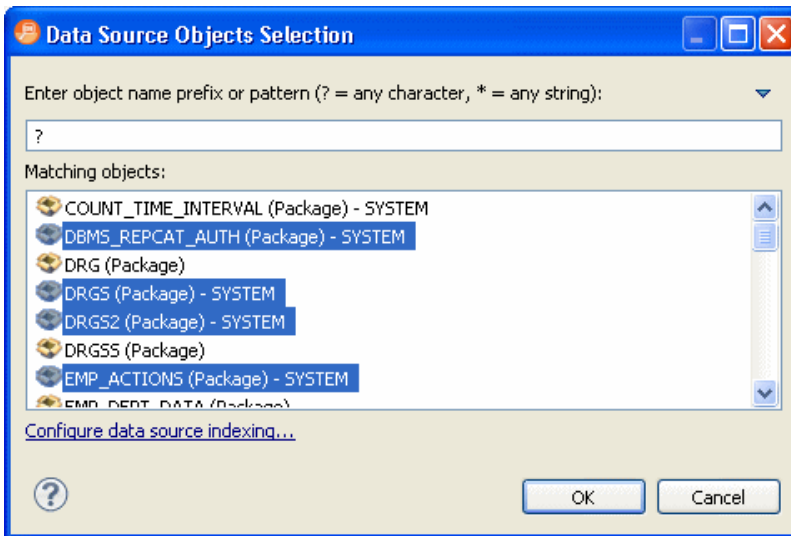
SQL statements to the **SQL Text** dialog or write queries by hand and then click **OK**.



 You can also input SQL by clicking anywhere in the Tuning Statements area and pressing Ctrl-V.

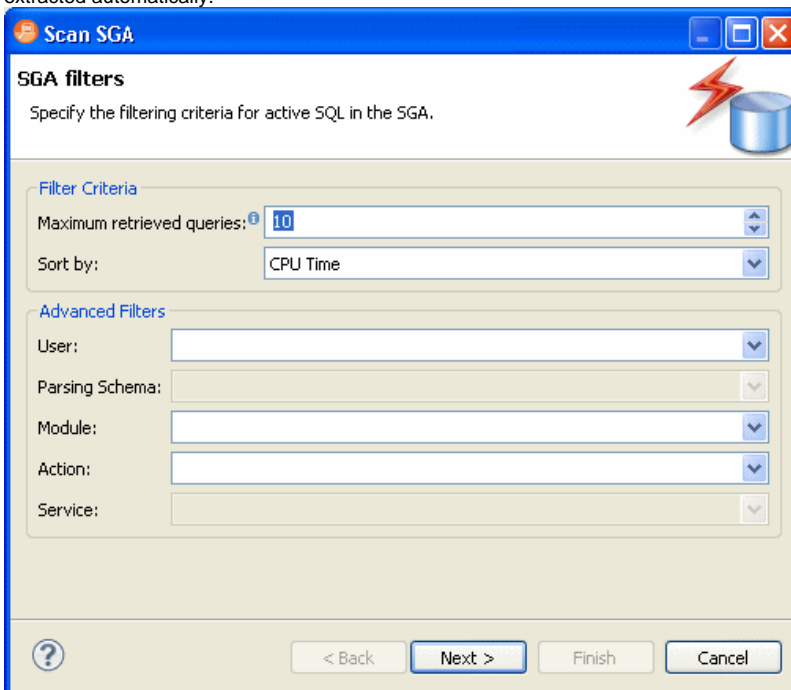
Once you have input the SQL and click **OK**, you can later edit the text by right-clicking an entry in the Tuning Statements area and selecting **Edit**.

- **Extract from Database Objects:** Search for and then select (Ctrl-click) data base objects containing SQL that you want to tune from the selected data source. SQL Query Tuner will search through the database to find objects matching your input and presents matches for you to choose. In order for this option to work, you must enable Data Source Indexing in the properties for the database. If the data source has not already been indexed you will receive a message indexing that no indexing information is available. You can configure the database Properties dialog from the **Data Source Objects Selection** dialog by clicking **Configure data source indexing**....



For information on setting data source indexing properties, see [Specify Data Source Indexing Preferences](#).

- **Import from File (Workspace) and Import from File (System):** Browse the workspace or file system and select an SQL file from which to extract statements to tune.
- **Scan Oracle SGA:** For the Oracle platform only, you can also scan the System Global Area (SGA) for statements to tune. Bind variables are extracted automatically.



- You can also drag and drop Materialized Views, Procedures, and Views from the Data Source Explorer to the Tuning Statements grid and they will be added to the list of statements to tune.

Running a tuning job

Once you click the Run Job icon on the top right-hand side of the Overview tab, the Overview tab provides the list of statements that were analyzed by the Tuner, as well as the cases suggested by the execution process to improve them. Additional information may include statement Name, Text, Source, Cost, and Elapsed Time values, depending on the platform.

Only the Elapsed Time statistic appears on all supported platforms. On Oracle, Execution Statistics and Other Execution Statistics columns will appear. When determining the best possible path using the Overview tab, it is best to use the Elapsed Time value as the guideline. The faster the path, the more optimized the query will become.

Tuning Statements

Name	Schema	Text	Tables	Views	Elapsed (s)	Improved (s)	Cases	Indexes
SELECT 3	SYS	select from sys.job\$					0	
SELECT 4	SYS	select from fet\$, ts\$					0	
SELECT 5	SYS	select from pending_trans\$					0	
SELECT 6	SYS	select from sys.obj\$					0	

Generated Cases

SQL Statements and Cases		Cost	Execution Statistics	Other Execution Statistics		
Name	Text	Value	Elapsed Time (s)	Physical Reads	Logical Reads	CPU Time (s)
SELECT 1	select from SYSTEM.DEF\$ AQCALL	0.56	0.56	0	8	0.00
INDEX_COMBINE		1.0	0.63	0	11	0.00
INDEX		826.0	0.87	0	8	0.00
RULE			0.77	0	4	0.00
SELECT 3	select from sys.job\$					
SELECT 4	select from fet\$, ts\$					
SELECT 5	select from pending_trans\$					
SELECT 6	select from sys.obj\$, sys.user\$, ind\$					

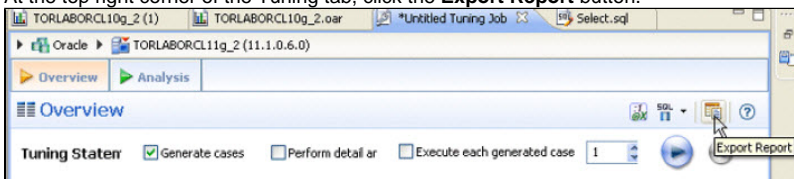
There are three tuning options to choose from before clicking Run Job:

- To analyze the SQL statement, click **Generate cases**.
- To perform the analysis that populates the Analysis tab now, click **Perform detail analysis**. Otherwise, the analysis tab is populated when you click the **Analysis** tab.
- To have the system generate execution statistics, click **Execute each generate case** and then select the number of time the system should execute each generated case. Multiple executions can verify that the case results are not skewed by caching. For example, the first time a query is run, data might be read off of disk, which is slow, and the second time the data might be in cache and run faster. Thus, one case might seem faster than another but it could be just benefiting from the effects of caching. Generally, you only need to execute the cases once, but it may be beneficial to execute the cases multiple times to see if the response times and statistics stay the same.

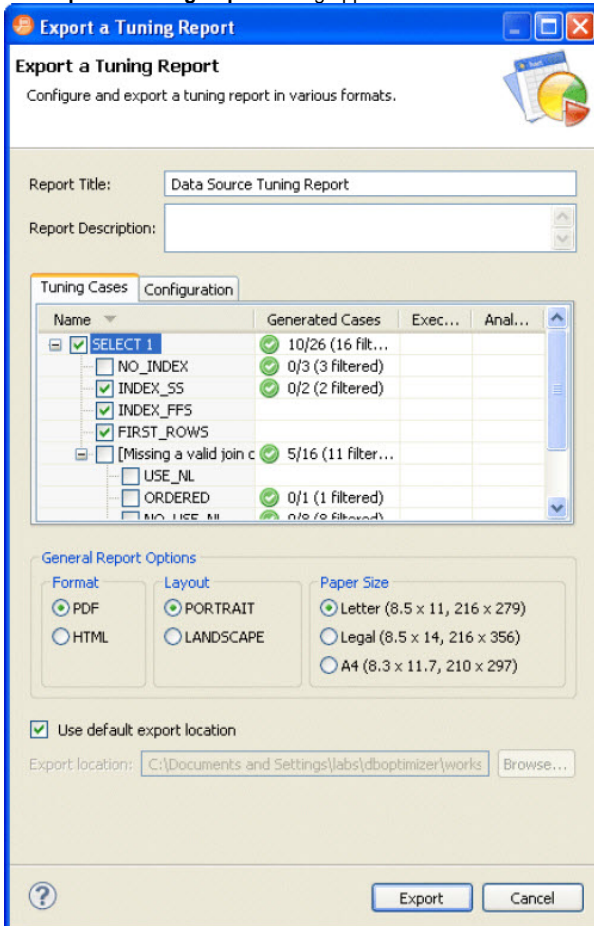
Creating tuning reports

After tuning SQL you can create an HTML or PDF Report of the tuning session. You can choose the details to include in the report.

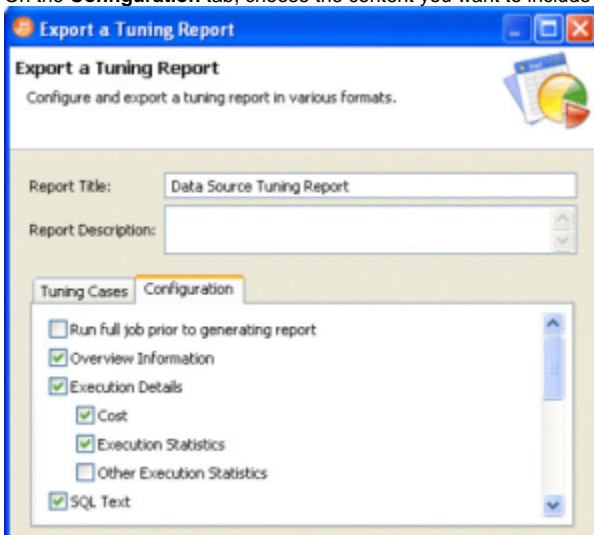
1. At the top right corner of the Tuning tab, click the **Export Report** button.



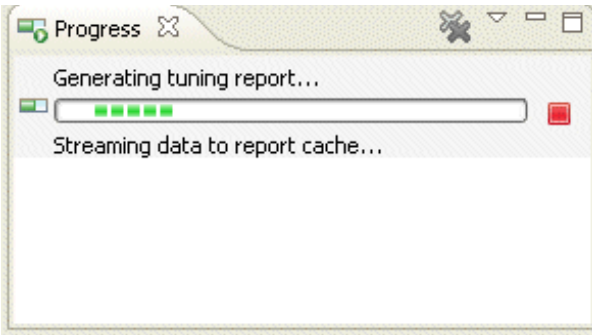
The **Export a Tuning Report** dialog appears.



2. Enter a **Report Title** and **Description**.
3. On the Tuning Cases tab, choose the cases you want to report on. Click the + to expand the cases.
4. On the **Configuration** tab, choose the content you want to include in the report.



5. Select the **General Report** options, enter the location for the report, and then click **Export**. You will see the progress of the report generation in the Progress pane.



When complete, the report is stored at the top level of your workspace.
A report in PDF format will resemble the following:

Bookmarks

- 1 - Data source:
- 2 - Overview
- 3 - Case SELECT 1

Data Source Tuning Report
March 6, 2012 10:07 AM
Test report

1 - Data source:
Name: TORLABORCL11g_2
Platform: Oracle 11.1.0.6.0
Tuning Job Name: Untitled Tuning Job

2 - Overview

Name	Source	Schema	Text	Tables	Views	Elapsed Time(s)	Improved Time(s)	Cases Analyzed	Indexes Analyzed
SELECT 1	Custom Case	SYSTEM	select from HR.EMPLOYEES, HR.DEPARTME...					10	

3 - Case SELECT 1

3.1 - Generated Cases

Name	Text	Cost Result	Elapsed Time (s)	Rows Returned
SELECT 1	select from HR.EMPLOYEES, HR.DEPARTMENTS	41.0		
FIRST_ROWS	Transformation	4.0		
INDEX_FFS		40.0		
INDEX_SS		83.0		

3.2 - SQL Text

```
SELECT *
FROM
  HR.EMPLOYEES,
  HR.DEPARTMENTS;
```

3.3 - Case FIRST_ROWS

3.3.1 - Generated Cases

Name	Text	Cost Result	Elapsed Time (s)	Rows Returned
FIRST_ROWS		4.0		

3.3.2 - SQL Text

```
SELECT /*+ FIRST_ROWS ( 10 ) */
FROM
  HR.EMPLOYEES;
```

[IDERA](#) |
 [Products](#) |
 [Purchase](#) |
 [Support](#) |
 [Community](#) |
 [Resources](#) |
 [About Us](#) |
 [Legal](#)