# Practices for Lesson 33: Analyzing SQL and Optimizing Access Paths

Practices for Lesson 33: Overview

Overview

In these practices, you will use the SQL Tuning Advisor to optimize SQL performance.

Practice 33-1: Using the SQL Tuning Advisor

Overview

In this practice, you optimize the performance of a costly SQL statement by using the SQL Tuning Advisor through Enterprise Manager Database Express (EM Express).

Note: All advisors are also available through Enterprise Manager Cloud Control.

Assumptions

You are logged in as the Oracle user. Practice 2-1: Using Enterprise Manager Database Express to Manage Performance has been performed and EM Database has been enabled for Flash operation.

Tasks

Initiate a SQL Load

Open a new terminal window and source the oraenv script.

Execute the $HOME/labs/DBMod\_MonTune/PERF\_setup\_tuning.sh shell script. This script creates a user named OE, a tablespace named TBS\_APP, and a schema named OE in the TBS\_APP tablespace in ORCLPDB1. It does the same thing in ORCLPDB2. Wait for the setup script to finish. It may take a couple of minutes to run. You can ignore any error messages because they are expected.

Start an application workload in ORCLPDB1 and ORCLPDB2. The PERF\_loop.sh script runs a SQL script named PERF\_loop.sql eight times in ORCLPDB1 as the OE user. It then runs the same SQL script eight times in ORCLPDB2 as the SYSTEM user. Let this script run several minutes before moving onto the next step.

Use EM Express to Tune the SQL Based on Statistics

In this section, you review but do not implement the first recommendation of the SQL Tuning Advisor, which is based on statistics.

Open a browser and launch Enterprise Manager Database Express by entering the following URL: https://localhost:5500/em or using the EM Express link on the bookmark toolbar.

If a message appears saying: This Pluging is vulnerable and should be updated… Click on the red circle and then **Allow** button on the popup.

On the Login page, enter the username **SYS** and the password. Leave the Container Name box empty, select **as sysdba**, and click **Login**. See the “*Course Practice Environment:*

*Security Credentials*” document in your Activity Guide for the password.

Select **Performance** and then **Performance Hub**.

From the Performance Hub page, click the **Activity** tab.

The current SQL executions are listed in the table at the bottom of the page. In this example, you can see that there is one consuming SQL execution (the first SQL ID in the list).

Position your cursor over the SQL ID. The following code should appear. If your result looks different, wait a moment and refresh EM Express. Click the SQL ID of the greatest consumer.

On SQL Details for … page, click the **Execution Statistics** tab.

The Plan Details area at the bottom of the page shows you the current plan for executing the SQL.

To return to where you were last, select **Performance** and then **Performance Hub**.

Once back on the main Performance Hub page, click the **Activity** tab.

In the Activity column, click the row to select the SQL statement to tune and click the **Tune SQL** button to launch the SQL Tuning Advisor.

Question: What may the SQL Tuning Advisor suggest?

Answer: It can suggest indexing columns, SQL profiles implementation, restructuring the SQL statement, and collecting missing or stale object statistics.

In the Tune SQL dialog box, enter the task name **MySQLTune** and click **OK**.

While the analysis task is completing, you are automatically brought to the SQL Tuning Advisor page, which has two tabs:

The Automatic tab lists the automatic tasks executed every night.

The Manual tab lists the manually created SQL tuning tasks, click the **Manual** tab.

The task you just created is listed. You may need to wait a moment for it to complete its processing if the status shows the running icon

If status shows completed icon , then click your MySQLTune to read the

recommendations.

Examine the Turing Results page for your session/statement and read the recommendations.

In the SQL Text area, you can view the full SQL statement to be tuned. Scroll down to view all of it.

In the **Select Recommendation** area, notice that there are two recommendations: Statistics and SQL Profile. For the Statistics recommendation, SQL Tuning Advisor's findings say that Optimizer statistics are not up-to-date for objects referenced in the SQL statement. It identifies OE tables in the SQL statement, which are not analyzed.

Question: How do tables get statistics collected automatically?

Answer: There is an automated task that automatically gathers Optimizer statistics every night. You can configure the settings that are used for Optimizer statistics gathering.

In the **Select Recommendations** area, click the **Statistics** link.

On the Recommendation Details page, view the list of recommendations and then click the **Implement** button at the top of the page. Notice that the Optimizer recommends gathering statistics for the tables in your SQL statement.

In the Gather Statistics dialog box, click **Show SQL**.

View the SQL package and procedure that would gather the statistics and click **OK**.

In the Gather Statistics dialog box, click **Cancel** because you will ask the Optimizer Advisor in the next practice to help you implement the statistics collection in the best way.

Use EM Express to Tune the SQL Using a SQL Profile

In this section, you implement the second recommendation, which suggests the usage of a SQL profile. This option provides a better execution plan.

Select **Performance** and then **SQL Tuning Advisor**.

Click the **Manual** tab, if needed.

Click **MySQLTune** (or the name you gave your tuning task).

In the **Select Recommendation** section, at the bottom of the Benefit column, view the benefit percentage value for using the SQL Profile. In this example, the SQL profile would increase performance by almost 97%. Your value may be different. In the Select Recommendation section, click the SQL Profile link.

On the Original Plan tab, which is displayed by default, view the SQL execution plan. You saw this plan earlier in the practice.

Click the Plan Using SQL Profile tab and view its SQL execution plan. Notice the differences between it and the original plan.

At the top of the page, click the **Implement** button

In the Create SQL Profile dialog box, click **Show SQL**.

A Confirmation dialog box shows you the generated SQL statement. Click **OK**.

In the Create SQL Profile dialog box, click **OK** to implement the new profile.

The SQL profile is created. In the Confirmation dialog box, click **OK**.

In the terminal window, if your workload script is still running, press Ctrl+c to stop the activity.

Rerun the SQL Script and Verify the Performance Benefit

In this section, you re-execute the PERF\_loop.sh script and verify that the SQL tuning you just implemented made the query consume fewer resources in the database.

Return to the terminal window.. Set Start SQL\*Plus and connect to the CDB root as the SYS

user with the SYSDBA privilege.

When testing SQL, it is a good idea to periodically flush the shared pool entries to remove older execution plans.

Remove any blocks of the tables (selected in the query) from the buffer cache.

Exit SQL\*Plus.

Run the application workload again in ORCLPDB1 and ORCLPDB2. The PERF\_loop.sh script runs a SQL script named PERF\_loop.sql eight times in ORCLPDB1 as the OE user. It then runs the same SQL script eight times in ORCLPDB2 as the SYSTEM user.

Wait at least 3 minutes then return to EM Express.

If a Warning dialog box is dispayed stating that a particular SQL ID is no longer from Cursor Cache, click **OK**.

Select **Performance** and then **Performance** Hub.

Click the **Activity** tab.

At the bottom of the page, note the value in the Activity column.

Question: How does the value in the Activity column now compare to the value in the Activity column prior to SQL tuning? Is there a performance benefit to the SQL tuning that you just did?

Answer: In this example, the average active sessions value went down from .11 to .01, so yes, there is a performance benefit from tuning the SQL. Your values may differ.

Click the link for the SQL ID. The Summary tab is displayed for the SQL ID.

Click the **Execution Statistics** tab.

In the Top Plans area, ensure the row with the Container Name ORCLPDB1 is selected.

In the Plan Details area, notice that the plan now used is the plan that uses the SQL Profile (refer back to ***step 6*** in the previous section in this practice).

Click **Log Out** to exit EM Express and close the browser window.

In the terminal window, press **Ctrl+C** to stop the activity.

Practice 33-2: Using the Optimizer Statistics Advisor

Overview

In this practice, you learn how to improve optimizer statistics collection quality by using the Optimizer Statistics Advisor.

The advisor task runs automatically in the maintenance window, but you can also run it on demand. If the advisor makes findings and then recommendations, then in some cases you can run system-generated scripts to implement them. Optimizer statistics play a significant part in determining the execution plan for queries. Therefore, it is critical for the optimizer to gather and maintain accurate and up-to-date statistics. All findings are derived from rules, but not all rules generate findings.

Tip

Because you use several windows at the same time in this practice, you may find it helpful to change the name of each of them in their banner at the top.

To set a title for a terminal window:

In the terminal window's menu, select **Terminal** and then **Set Title**. A Set Title dialog box is displayed.

In the Title box, enter the name “**Window ”** and a number

Click **OK**.

Assumptions

You are logged in as the oracle user.

Tasks

Window 1: Start an Application Workload

In the open terminal, name it Window 1, the execute the

$HOME/labs/DBMod\_MonTune/PERF\_setup\_tuning.sh shell script. Wait for the setup script to finish. You can ignore any error messages because they are expected.

Start an application workload in ORCLPDB1 and ORCLPDB2.

Window 2: Use the Optimizer Statistics Advisor to Generate Recommendations

In this section, you create an object filter for an Optimizer Statistics Advisor task, create and execute the task, generate a report with recommendations, and then implement those recommendations.

Open another new terminal window. This window will be referred to as Window 2.

Start SQL\*Plus and connect to ORCLPDB1 as the SYS user. Refer to *Practice Environment: Security Credentials* for the password value.

Execute the $HOME/labs/DBMod\_MonTune/OPTADV\_1.sql script, which is an object filter for an Optimizer Advisor Task. This filter disables statistics collection recommendations for all objects except those in the OE schema. In the previous practice, using SQL Tuning Advisor, the OE tables were included in queries that required tuning.

CREATE OR REPLACE PROCEDURE sh\_obj\_filter(p\_tname IN VARCHAR2) IS v\_retc CLOB;

BEGIN

v\_retc := DBMS\_STATS.CONFIGURE\_ADVISOR\_OBJ\_FILTER (p\_tname,'EXECUTE',NULL,NULL,NULL,'DISABLE'); v\_retc := DBMS\_STATS.CONFIGURE\_ADVISOR\_OBJ\_FILTER

(p\_tname,'EXECUTE',NULL,'OE',NULL,'ENABLE'); END;

/

Create and execute an advisor task named my\_task by executing the

$HOME/labs/DBMod\_MonTune/OPTADV\_2.sql script.

DECLARE

v\_tname VARCHAR2(128) := 'my\_task'; v\_ename VARCHAR2(128) := NULL;

v\_report CLOB := null; v\_script CLOB := null; v\_implementation\_result CLOB; BEGIN

v\_tname := DBMS\_STATS.CREATE\_ADVISOR\_TASK(v\_tname); sh\_obj\_filter(v\_tname);

v\_ename := DBMS\_STATS.EXECUTE\_ADVISOR\_TASK(v\_tname); END;

/

Verify that the procedure completed successfully.

Query the USER\_ADVISOR\_TASKS view.

Query the USER\_ADVISOR\_EXECUTIONS view. The results below are formatted for easier viewing. Your dates and numbers of rows will be different from those shown below. Make note of the value in the EXECUTION\_NAME column for MY\_TASK. In this example, the value is EXEC\_165.

Execute the $HOME/labs/DBMod\_MonTune/OPTADV\_3.sql script to run the following commands:

VAR b\_report CLOB DECLARE

v\_tname VARCHAR2(32767); BEGIN

v\_tname := 'my\_task';

:b\_report := dbms\_stats.report\_advisor\_task(v\_tname, type => 'TEXT', section=>'ALL', level=>'ALL');

END;

/

Execute the $HOME/labs/DBMod\_MonTune/OPTADV\_4.sql script to run the following commands:

DECLARE

v\_len NUMBER(10); v\_offset NUMBER(10) :=1;

v\_amount NUMBER(10) :=10000; BEGIN

v\_len := DBMS\_LOB.getlength(:b\_report); WHILE (v\_offset < v\_len)

LOOP

DBMS\_OUTPUT.PUT\_LINE(DBMS\_LOB.SUBSTR(:b\_report,v\_amount,v\_offset

));

v\_offset := v\_offset + v\_amount; END LOOP;

END;

/

Edit the script $HOME/labs/DBMod\_MonTune/OPTADV\_5.sql with vi. Be sure to edit the script and enter the correct value for F.EXECUTION\_NAME as determined in step 5b.

Execute the $HOME/labs/DBMod\_MonTune/OPTADV\_5.sql script to view the findings.

Generate the script before a possible implementation.

Execute the $HOME/labs/DBMod\_MonTune/OPTADV\_6.sql script to run the following commands:

SET SERVEROUTPUT ON

VARIABLE b\_script CLOB DECLARE

v\_tname VARCHAR2(32767); BEGIN

v\_tname := 'my\_task';

:b\_script := DBMS\_STATS.SCRIPT\_ADVISOR\_TASK(v\_tname); END;

/

Execute the $HOME/labs/DBMod\_MonTune/OPTADV\_7.sql script to run the following commands:

DECLARE

v\_len NUMBER(10); v\_offset NUMBER(10) :=1;

v\_amount NUMBER(10) :=10000; BEGIN

v\_len := DBMS\_LOB.getlength(:b\_report); WHILE (v\_offset < v\_len)

LOOP

DBMS\_OUTPUT.PUT\_LINE(DBMS\_LOB.SUBSTR(:b\_script, v\_amount, v\_offset));

v\_offset := v\_offset + v\_amount; END LOOP;

END;

/

-- No scripts will be provided for the rule USEAUTOJOB.Please check the report for more details.

-- No scripts will be

provided for the rule COMPLETEAUTOJOB.Please check the report for more details.

-- No scripts will be provided for the rule MAINTAINSTATSHISTORY.Please check the report for more details.

-- No scripts will be provided for the rule

TURNONSQLPLANDIRECTIVE.Please check the report for more details.

-- No scripts will be provided for the rule AVOIDSETPROCEDURES.Please check the report for more details.

-- No scripts will be provided for the rule USEDEFAULTPARAMS.Please

check the report for more details.

-- No scripts will be provided for the rule USEGATHERSCHEMASTATS.Please check the report for more

details.

-- No scripts will be provided for the rule AVOIDINEFFICIENTSTATSOPRSEQ.Please check the report for more details.

-- No

scripts will be provided for the rule AVOIDUNNECESSARYSTATSCOLLECTION.Please check the report for more details.

-- No scripts will

be provided for the rule GATHERSTATSAFTERBULKDML.Please check the report for more details.

-- No scripts will be provided for the

rule AVOIDDROPRECREATE.Please check the report for more details.

-- No scripts will be provided for the rule AVOIDOUTOFRANGE.Please

check the report for more details.

-- No scripts will be provided for the rule AVOIDANALYZETABLE.Please check the report for more

details.

-- No scripts will be provided for the rule USEAUTOJOB.Please check the report for more details.

-- No scripts will be

provided for the rule COMPLETEAUTOJOB.Please check the report for more details.

-- No scripts will be provided for the rule

MAINTAINSTATSHISTORY.Please check the report for more details.

-- No scripts will be provided for the rule

TURNONSQLPLANDIRECTIVE.Please check the report for more details.

-- No scripts will be provided for the rule AVOIDSETPROCEDURES.Please check the report for more details.

-- No scripts will be provided for the rule USEDEFAULTPARAMS.Please

check the report for more details.

-- No scripts will be provided for the rule USEGATHERSCHEMASTATS.Please check the report for more

details.

-- No scripts will be provided for the rule AVOIDINEFFICIENTSTATSOPRSEQ.Please check the report for more details.

-- No

scripts will be provided for the rule AVOIDUNNECESSARYSTATSCOLLECTION.Please check the report for more details.

-- No scripts will

be provided for the rule GATHERSTATSAFTERBULKDML.Please check the report for more details.

-- No scripts will be provided for the

rule AVOIDDROPRECREATE.Please check the report for more details.

-- No scripts will be provided for the rule AVOIDOUTOFRANGE.Please

check the report for more details.

-- No scripts will be provided for the rule AVOIDANALYZETABLE.Please check the report for more

details.

-- Scripts for rule USECONCURRENT

-- Rule Description: Use Concurrent preference for Statistics Collection

-- No

scripts will be provided for the rule USEAUTOJOB.Please check the report for more details.

-- No scripts will be provided for the

rule COMPLETEAUTOJOB.Please check the report for more details.

-- No scripts will be provided for the rule MAINTAINSTATSHISTORY.Please check the report for more details.

-- No scripts will be provided for the rule

TURNONSQLPLANDIRECTIVE.Please check the report for more details.

-- No scripts will be provided for the rule AVOIDSETPROCEDURES.Please check the report for more details.

-- No scripts will be provided for the rule USEDEFAULTPARAMS.Please

check the report for more details.

-- No scripts will be provided for the rule USEGATHERSCHEMASTATS.Please check the report for more

details.

-- No scripts will be provided for the rule AVOIDINEFFICIENTSTATSOPRSEQ.Please check the report for more details.

-- No

scripts will be provided for the rule AVOIDUNNECESSARYSTATSCOLLECTION.Please check the report for more details.

-- No scripts will

be provided for the rule GATHERSTATSAFTERBULKDML.Please check the report for more details.

-- No scripts will be provided for the

rule AVOIDDROPRECREATE.Please check the report for more details.

-- No scripts will be provided for the rule AVOIDOUTOFRANGE.Please

check the report for more details.

-- No scripts will be provided for the rule AVOIDANALYZETABLE.Please check the report for more

details.

-- Scripts for rule USEDEFAULTPREFERENCE

-- Rule Description: Use Default Preference for Stats Collection

-- Set global

preferenes to default values.

-- Scripts for rule UNLOCKNONVOLATILETABLE

-- Rule Description: Statistics for objects with non-volatile should not be locked

-- Unlock statistics for objects that are not volatile.

-- Scripts for rule USEAUTODEGREE

--

Rule Description: Use Auto Degree for statistics collection

-- Turn on auto degree for those objects for which using auto degree is

helpful.

-- Scripts for rule LOCKVOLATILETABLE

-- Rule Description: Statistics for objects with volatile data should be locked

--

Lock statistics for volatile objects.

-- Scripts for rule NOTUSEINCREMENTAL

-- Rule Description: Statistics should not be maintained incrementally when it is not beneficial

-- Turn off incremental option for those objects for which using incremental is

not helpful.

-- Scripts for rule USEINCREMENTAL

-- Rule Description: Statistics should be maintained incrementally when it is

beneficial

-- Turn on the incremental option for those objects for which using incremental is helpful.

-- Scripts for rule USEDEFAULTOBJECTPREFERENCE

-- Rule Description: Use Default Object Preference for statistics collection

-- Setting object-level preferences to default values

-- setting CASCADE to default value for object level preference

-- setting ESTIMATE\_PERCENT to default value for object level preference

-- setting METHOD\_OPT to default value for object level preference

-- setting GRANULARITY to

default value for object level preference

-- setting NO\_INVALIDATE to default value for object level preference

-- Scripts for

rule AVOIDSTALESTATS

-- Rule Description: Avoid objects with stale or no statistics

-- Gather statistics for those objcts that are missing or have no statistics.

-- Scripts for rule MAINTAINSTATSCONSISTENCY

-- Rule Description: Statistics of dependent objects should be consistent

-- Gather statistics for those objcts that are missing or have no statistics.

declare obj\_filter\_list

dbms\_stats.ObjectTab;

obj\_filter dbms\_stats.ObjectElem; obj\_cnt number := 0;

begin

obj\_filter\_list := dbms\_stats.ObjectTab();

obj\_filter.ownname := 'OE'; obj\_filter.objtype := 'TABLE'; obj\_filter.objname := 'CUSTOMER';

obj\_filter\_list.extend(); obj\_cnt := obj\_cnt + 1;

obj\_filter\_list(obj\_cnt) := obj\_filter; obj\_filter.ownname := 'OE';

obj\_filter.objtype := 'TABLE'; obj\_filter.objname := 'CUSTOMERS'; obj\_filter\_list.extend();

obj\_cnt := obj\_cnt + 1;

obj\_filter\_list(obj\_cnt) := obj\_filter; obj\_filter.ownname := 'OE'; obj\_filter.objtype := 'TABLE'; obj\_filter.objname :=

'DATE\_DIM';

obj\_filter\_list.extend();

obj\_cnt := obj\_cnt + 1; obj\_filter\_list(obj\_cnt) := obj\_filter; obj\_filter.ownname :=

'OE';

obj\_filter.objtype := 'TABLE'; obj\_filter.objname := 'INVENTORIES'; obj\_filter\_list.extend();

obj\_cnt := obj\_cnt + 1;

obj\_filter\_list(obj\_cnt) := obj\_filter; obj\_filter.ownname := 'OE'; obj\_filter.objtype := 'TABLE'; obj\_filter.objname :=

'LINEORDER';

obj\_filter\_list.extend(); obj\_cnt := obj\_cnt + 1;

obj\_filter\_list(obj\_cnt) := obj\_filter; obj\_filter.ownname :=

'OE';

obj\_filter.objtype := 'TABLE'; obj\_filter.objname := 'ORDERS'; obj\_filter\_list.extend(); obj\_cnt := obj\_cnt + 1;

obj\_filter\_list(obj\_cnt) := obj\_filter; obj\_filter.ownname := 'OE'; obj\_filter.objtype := 'TABLE'; obj\_filter.objname :=

'ORDER\_ITEMS';

obj\_filter\_list.extend(); obj\_cnt := obj\_cnt + 1;

obj\_filter\_list(obj\_cnt) := obj\_filter; obj\_filter.ownname

:= 'OE';

obj\_filter.objtype := 'TABLE'; obj\_filter.objname := 'PART'; obj\_filter\_list.extend(); obj\_cnt := obj\_cnt + 1;

obj\_filter\_list(obj\_cnt) := obj\_filter; obj\_filter.ownname := 'OE'; obj\_filter.objtype := 'TABLE';

Question: What does obj\_filter\_list indicate?

Answer: The object filter list contains the names of the ten objects with no statistics.

Question: What would you do if you agree with the recommendations? Answer: You could either execute the generated SQL script or use the

DBMS\_STATS.IMPLEMENT\_ADVISOR\_TASK procedure.

Execute the $HOME/labs/DBMod\_MonTune/OPTADV\_8.sql script to invoke the DBMS\_STATS.IMPLEMENT\_ADVISOR\_TASK PL/SQL procedure. This procedure implements the actions recommended by the advisor based on results from a specified Optimizer Statistics Advisor execution.

VARIABLE b\_ret CLOB DECLARE

v\_tname VARCHAR2(32767); BEGIN

v\_tname := 'my\_task';

:b\_ret := DBMS\_STATS.IMPLEMENT\_ADVISOR\_TASK(v\_tname); END;

/

Check that the statistics are collected for the ten objects. Columns such as NUM\_ROWS, EMPTY\_BLOCKS, BLOCKS, and AVG\_ROW\_LEN have null values until the statistics are collected.

Drop the advisor task.

Exit SQL\*Plus and close Window 2.

Window 1: Stop the Workload

In Window 1, press Ctrl+c to stop the activity and then close the window.