# Practices for Lesson 17: Improving Space Usage

Practices for Lesson 17: Overview

Overview

In these practices, you will use the Segment Advisor to manage space in your database. You will also use the Compression Advisor. Finally, you enable the Resumable Space Allocation feature.

Time Estimate:

It is estimated that this practice can be completed in 40 minutes.

Practice 17-1: Managing Space in Tablespaces

Overview

In this practice, you will set a warning threshold and a critical threshold on a tablespace and then test those thresholds. You then create a Segment Advisor task to get recommendations about the current space situation.

Tip

For problems that cannot be resolved automatically and require DBAs to be notified, such as running out of space, the Oracle Database server provides server-generated alerts. Two alert thresholds are defined by default:

The warning threshold is the limit at which space is beginning to run low.

The critical threshold is a serious limit that warrants your immediate attention. The database issues alert at both thresholds. The alerts notify you and often provide recommendations on how to resolve the reported problem.

Tasks

Set a Warning Threshold

Source the oraenv script.

Connect to **ORCLPDB1** as the **SYSTEM** user. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

Execute the DBMS\_SERVER\_ALERT.SET\_THRESHOLD procedure to reset the database-wide threshold values for the Tablespace Space Usage metric.

**Note:** The following command must either be all on one line or each line must be ended with a '-' character with no spaces following it.

Check the database-wide threshold values for the Tablespace Space Usage metric.

Connect to the root container.

Query the WARNING\_VALUE and the CRITICAL\_VALUE columns in the DBA\_THRESHOLDS view. The results show that the warning threshold value is 85 and the critical threshold value is 97.

In ORCLPDB1, create a new tablespace called TBSALERT with a 120MB file called tbsalert.dbf. Make sure that this tablespace is locally managed and uses Automatic Segment Space Management. Do not make it auto-extensible and do not specify any thresholds for this tablespace.

Connect to ORCLPDB1

Create the TBSALERT tablespace by executing the Create\_TBSALERT\_TS.sql

script.

Query how much free space the TBSALERT tablespace holds by executing the

$HOME/labs/TBSALERT\_free\_space.sql script.

Modify the thresholds values for the Tablespace Space Usage metric for the TBSALERT

tablespace. Set the Warning Threshold to 55 and the Critical Threshold to 70.

Verify that the thresholds are set correctly. The query returns a warning value of 55 and a critical value of 70, which indicates that the thresholds are set correctly.

Query the REASON and RESOLUTION columns from the DBA\_ALERT\_HISTORY view for the

TBSALERT tablespace.

Exit SQL\*Plus.

Execute the $HOME/labs/seg\_advsr\_setup.sh shell script to create and populate new tables in the TBSALERT tablespace.

Check the fullness level of the TBSALERT tablespace to see if the warning level has been reached.

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

Query the size of the TBSALERT tablespace. The results show that the tablespace is 60% full.

Query the number of free bytes that are left in the TBSALERT tablespace by executing the $HOME/labs/DBMod\_Storage/TBSALERT\_free\_space.sql script. Recall that you created the tablespace with 120MB (125829120 bytes) of space. The query result shows that there are 125829120 bytes free and the tablespace is 39% free.

Wait a few minutes, then query the DBA\_OUTSTANDING\_ALERTS view to see if there are any new messages. The REASON column is updated with a message stating that the tablespace is 60 percent full. This message is there because the warning level for the tablespace has been reached. If your result is “no rows selected,” wait a little longer and repeat the query. *You may have to wait as much as 10 minutes.*

Set a Critical Threshold

In this section, you add more data to the TBSALERT tablespace and check the tablespace fullness threshold again.

Execute and commit the following INSERT statements.

Wait a few minutes.

Query the fullness of the tablespace. The result shows that the tablespace is 75% full.

Query the outstanding alerts. The REASON column is updated with a message that states the tablespace is 75 percent full.

If your result still displays 60, wait a little longer and repeat the query. *It may take as long as 10 minutes to display 75 percent*.

Delete rows from three tables in the HR schema to try to reduce the space used in the tablespace.

Check if there is some reclaimed space after these tables were deleted. The query result indicates that this is not the case. The tablespace is still 75 percent full. Deleting rows frees space in blocks, but it does not return blocks to the tablespace.

Create a Segment Advisor Task.

Create a Segment Advisor task to get recommendations about the current space situation by executing the **$HOME/labs/seg\_advsr\_task.sql** script.

Execute the task.

Query the DBA\_ADVISOR\_TASKS view for recommendations. The recommendation is to get shrink advice for segments stored in the tablespace.

Execute the $HOME/labs/DBMod\_Storage/segments\_to\_shrink.sql script to find out which segments should be shrunk to reclaim space. The result shows that the first three segments should be shrunk.

SQL> @/home/oracle/labs/DBMod\_Storage/segments\_to\_shrink.sql

SQL> col attr1 format a5 SQL> col attr2 format a12 SQL> col message format a55 SQL> set echo on

SQL> SELECT attr1, attr2, message FROM dba\_advisor\_findings f, dba\_advisor\_objects o WHERE f.task\_name = o.task\_name AND f.object\_id = o.object\_id AND f.task\_name = 'my\_seg\_task';

ATTR1 ATTR2 MESSAGE

HR EMPLOYEES2 Perform shrink, estimated savings is 18873242 bytes.

HR EMPLOYEES1 Perform shrink, estimated savings is 18873242 bytes.

HR EMPLOYEES3 Perform shrink, estimated savings is 18873242 bytes.

Proceed with the SHRINK operation on the HR.EMPLOYEES1, HR.EMPLOYEES2, and

HR.EMPLOYEES3 tables.

Check if the SHRINK operations reclaimed unused space by running the following query. The result shows that the tablespace did reclaim unused space. It went down to 30% full from 75% full.

Drop the TBSALERT tablespace.

Exit SQL\*Plus.

Close all terminals.

Practice 17-2: Using Compression

Overview

In this practice, you will use Advanced Index Compression to reduce the storage for indexes. You use the Compression Advisor, provided by the DBMS\_COMPRESSION package, to get detailed space information about compressing the index with different compression levels.

Assumptions

You are logged in to the compute node as the oracle user.

Tasks

Open a terminal and set the environment variable to ORCLCDB database.

Execute the $HOME/labs/DBMod\_Storage/setup\_index.sh shell script to create an index with low compression on the HR.TEST table in ORCLPDB1.

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

Query the compression level of the index created on the **HR.TEST** table. The result indicates that compression is disabled, and therefore, the index is not compressed.

Query the space used by the index created on the **HR.TEST** table. The result indicates that 1152 blocks are used.

Exit SQL\*Plus, but keep the terminal window open.

View the different compression levels that exist in your Oracle Database version. To do this, use the cat command to review the predefined SQL script that creates the DBMS\_COMPRESSION package.

Rem

Rem DESCRIPTION

Rem Contains package specification for the wrapper dbms\_compression

Rem package and internal prvt\_compression package. We integrate these

Rem packages with the advisor framework. Rem

…

create or replace package dbms\_compression authid current\_user is

COMP\_NOCOMPRESS CONSTANT NUMBER := 1;

COMP\_ADVANCED CONSTANT NUMBER := 2;

COMP\_QUERY\_HIGH CONSTANT NUMBER := 4;

COMP\_QUERY\_LOW CONSTANT NUMBER := 8;

COMP\_ARCHIVE\_HIGH CONSTANT NUMBER := 16;

COMP\_ARCHIVE\_LOW CONSTANT NUMBER := 32;

COMP\_BLOCK CONSTANT NUMBER := 64;

COMP\_LOB\_HIGH CONSTANT NUMBER := 128;

COMP\_LOB\_MEDIUM CONSTANT NUMBER := 256;

COMP\_LOB\_LOW CONSTANT NUMBER := 512; COMP\_INDEX\_ADVANCED\_HIGH CONSTANT NUMBER := 1024; COMP\_INDEX\_ADVANCED\_LOW CONSTANT NUMBER := 2048; COMP\_BASIC CONSTANT NUMBER := 4096; COMP\_INMEMORY\_NOCOMPRESS CONSTANT NUMBER := 8192; COMP\_INMEMORY\_DML CONSTANT NUMBER := 16384; COMP\_INMEMORY\_QUERY\_LOW CONSTANT NUMBER := 32768; COMP\_INMEMORY\_QUERY\_HIGH CONSTANT NUMBER := 65536; COMP\_INMEMORY\_CAPACITY\_LOW CONSTANT NUMBER := 131072; COMP\_INMEMORY\_CAPACITY\_HIGH CONSTANT NUMBER := 262144;

COMP\_RATIO\_MINROWS CONSTANT NUMBER := 1000000;

COMP\_RATIO\_ALLROWS CONSTANT NUMBER := -1; COMP\_RATIO\_LOB\_MINROWS CONSTANT NUMBER := 1000; COMP\_RATIO\_LOB\_MAXROWS CONSTANT NUMBER := 5000; COMP\_RATIO\_INDEX\_MINROWS CONSTANT NUMBER := 100000;

OBJTYPE\_TABLE CONSTANT NUMBER := 1;

OBJTYPE\_INDEX CONSTANT NUMBER := 2;

OBJTYPE\_PART CONSTANT NUMBER := 3;

OBJTYPE\_SUBPART CONSTANT NUMBER := 4;

...

grant execute on dbms\_compression to public

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

Hint: use the up arrow key several times to recall the command from the OS command-line buffer.

Use the Compression Advisor to get recommendations about the space you would save by compressing the index with the COMP\_INDEX\_ADVANCED\_LOW compression level by executing the $HOME/labs/DBMod\_Storage/Compression\_index\_low.sql script. The result indicates that the space used by the index would be reduced down to 809 blocks. The Advanced Low Compression ratio equals 1.

Use the Compression Advisor again to get recommendations about the space you would save by compressing the index with the COMP\_INDEX\_ADVANCED\_HIGH compression level by executing the $HOME/labs/DNMod\_Storage/Compression\_index\_high.sql script. The result indicates that the space used by the index would be reduced down to 130 blocks. The Advanced High Compression ratio is equal to 8.

comptype => dbms\_compression.COMP\_INDEX\_ADVANCED\_HIGH, blkcnt\_cmp => blkcnt\_cmp,

blkcnt\_uncmp => blkcnt\_uncmp, row\_cmp => row\_cmp,

row\_uncmp => row\_uncmp, cmp\_ratio => cmp\_ratio, comptype\_str => comptype\_str,

subset\_numrows => dbms\_compression.COMP\_RATIO\_MINROWS, objtype => dbms\_compression.OBJTYPE\_INDEX

);

DBMS\_OUTPUT.PUT\_LINE('Block used by compressed index = ' || blkcnt\_cmp);

DBMS\_OUTPUT.PUT\_LINE('Block used by uncompressed index = ' || blkcnt\_uncmp);

DBMS\_OUTPUT.PUT\_LINE('Compression type = ' || comptype\_str); DBMS\_OUTPUT.PUT\_LINE('Compression ratio org = '||cmp\_ratio); END;

/

Block used by compressed index = 130 Block used by uncompressed index = 1029

Compression type = "Compress Advanced High" Compression ratio org = 8

PL/SQL procedure successfully completed.

SQL>

Question: Based on the previous steps, which compression ratio is the best—the

COMP\_INDEX\_ADVANCED\_LOW or COMP\_INDEX\_ADVANCED\_HIGH compression level? Answer: The Advanced High Compression ratio (8) is much better than the Advanced Low Compression ratio (1). Therefore, you would be inclined to rebuild the index with Advanced

High Compression.

Rebuild the index with Advanced High Compression.

Query the compression level of the index created on the HR.TEST table. The result shows that the compression level is ADVANCED HIGH.

Query the space used by the index created on the **HR.TEST** table. The space is now 256 blocks.

*Question*: Is it possible to revert back to the initial compression level?

*Answer*: Yes.

Revert back to the initial compression level.

Query the space used by the index created on the HR.TEST table. The space used is 1152 blocks again.

Exit SQL\*Plus.

Practice 17-3: Enabling the Resumable Space Allocation Feature

Overview

In this practice, you enable the Resumable Space Allocation feature to avoid situations where a tablespace runs out of space and causes operations to fail; for example, rows cannot be loaded into a table. You will work in two terminal windows (window 1 and window 2).

With the Resumable Space Allocation feature:

Some operations are resumable, but not all. These operations are called resumable statements. INSERT, INSERT INTO SELECT, UPDATE, and DELETE statements are candidates.

Some errors are correctable, but not all; for example: out of space condition (ORA-01653, ORA-01654), maximum extents reached condition (ORA-01631, ORA-01632), space quota exceeded condition (ORA-01536).

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 window number.

Click **OK**.

Assumptions

You are logged in to the compute node as the oracle user.

Tasks

Window 1: Enable the Resumable Space Allocation Feature

In your open terminal window, start SQL\*Plus and connect to ORCLPDB1 as the SYSTEM

user. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

Grant the PDBADMIN usee DBA role.

Connect to ORCLPDB1 as the PDBADMIN user. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

Execute the $HOME/labs/DBMod\_Storage/CreateINVENTORYTablespace.sql script to create an unpopulated tablespace named INVENTORY.

Execute the $HOME/labs/DBMod\_Storage/CreateTable\_X.sql script to create and populate a table named X in the INVENTORY tablespace. As the script runs, notice that rows are being inserted into the table. Part way through the script, you get an error telling you that there is not enough space in the INVENTORY tablespace to insert the remaining rows.

Imagine that the operation in the previous step had lasted 5 hours and that the load had nearly reached its end and other operations were depending on its success.

Question: Are the rows that were inserted into the table lost or definitely inserted? Answer: 2048 rows were inserted.

Question: How could this situation be avoided when you do not know how much space is required for a table to load all its rows?

Answer: In the case of heavy load operations, you can use a corrective action rather than a reactive action after an error is raised. For example, you can use the Resumable Space Allocation feature.

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

Enable resumable mode.

Re-execute the CreateTable\_X script. The script is suspended.

Question: Why is the script suspended?

Answer: Enabling the resumable mode for your session suspends the failing statement during 7200 seconds (2 hours), by default.

Question: Is there any warning message to tell you the load is suspended? Answer: No. If the script does not execute any further, check the alert log file or the

DBA\_RESUMABLE view. An operation-suspended alert is issued on the object that needs

allocation of resource for the operation to complete.

Window 2: Resolve a Suspended Script

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

Source the oraenv script.

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

Query the DBA\_RESUMABLE view for information about the suspended script. The DBA\_RESUMABLE view lists all resumable statements executed in the system. Your times and session information will be different from those shown below.

Exit SQL\*Plus, but keep the terminal window open.

Check the alert log file for information about the suspended script. The log states that the suspension occurred because the table could not be extended.

Proceed with the appropriate corrective action. Because the INVENTORY tablespace is not autoextensible, you can configure it as autoextensible with a size limit.

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

Query the DBA\_DATA\_FILES view to verify whether the INVENTORY tablespace is autoextensible. The result shows that the tablespace is not.

Enable autoextend for the INVENTORY01.DBF data file.

Query the DBA\_DATA\_FILES view again to verify whether the INVENTORY tablespace is autoextensible. The result shows that it is.

Window 1: Check the Suspended Session

Return to Window 1. Notice that the session is no longer suspended. The results show that 2048 rows were created, and the transaction was committed. After the resource had been allocated, the operation completed, and the operation-suspended alert cleared.

Close the terminal window.

Window 2: Verify that there are no Suspended Sessions.

Return to Window 2. Verify that there are no suspended sessions in the system by querying the DBA\_RESUMABLE view again.

Exit from SQL\*Plus and close the terminal window.