# Practices for Lesson 16: Creating and Managing Tablespaces

Practices for Lesson 16: Overview

Overview

In these practices, you will view information about tablespaces and create new tablespaces.

Practice 16-1: Viewing Tablespace Information

Overview

In this practice, you use SQL\*Plus to query various views to learn about tablespace content in

ORCLPDB1. You also view tablespace information with SQL\*Developer.

Assumptions

You are logged in as the oracle user.

Tasks

Open a new terminal window and source the oraenv script.

Grant DBA to PDBADMIN in ORCLPDB1

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

Grant DBA to PDBADMIN.

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

List the columns in the DBA\_TABLESPACES view by using the DESCRIBE command.

List the tablespaces in ORCLPDB1.

Find out which tablespace contains the HR schema by querying the ALL\_TABLES view.

Query the STATUS, CONTENTS, LOGGING, PLUGGED\_IN, BIGFILE,

EXTENT\_MANAGEMENT, and ALLOCATION\_TYPE columns in the DBA\_TABLESPACES view for the SYSAUX tablespace.

STATUS shows the value ONLINE, indicating the tablespace is available to users.

CONTENTS indicates the PERMANENT tablespace type.

LOGGING shows the value LOGGING, indicating that certain DML operations are logged in the redo log file.

PLUGGED\_IN shows the value NO, indicating that the tablespace is not plugged in.

BIGFILE shows the value NO, indicating that the tablespace is a smallfile tablespace.

EXTENT\_MANAGEMENT shows the value LOCAL, indicating that the tablespace is locally managed (not dictionary managed).

ALLOCATION\_TYPE shows the value SYSTEM, indicating that the extents of the tablespace are managed by the system, and you cannot specify an extent size.

List the columns in the V$TABLESPACE view by using the DESCRIBE command. This view displays tablespace information from the control file.

Query the V$TABLESPACE view for the SYSAUX tablespace.

INCLUDED\_IN\_DATABASE\_BACKUP contains the value YES, indicating that the tablespace is included in full database backups by using the BACKUP DATABASE RMAN command.

BIGFILE contains the value NO, indicating that the tablespace is a smallfile tablespace.

FLASHBACK\_ON contains the value YES, indicating that the tablespace participates in

FLASHBACK DATABASE operations.

ENCRYPT\_IN\_BACKUP contains the value null, indicating that encryption is neither explicitly turned on nor off at the tablespace level.

CON\_ID indicates the container to which the data pertains. In this case, ORCLPDB1 is container ID 3. The container ID will vary, depending on the number of times the PDB has been recreated.

List all the tables in the USERS tablespace owned by the HR account.

List all the indexes in the USERS tablespace owned by the HR account.

List the columns in the DBA\_DATA\_FILES view by using the DESCRIBE command. You can query this view to learn about the data files contained in a tablespace.

List data file information for the SYSAUX tablespace by querying various columns in the

DBA\_DATA\_FILES view.

The results show the following:

AUTOEXTENSIBLE contains the value YES, indicating that the auto extend feature is enabled for a data file. The tablespace size can increase without you having to take any action.

BYTES is the size of the file in bytes.

MAXBYTES is the maximum file size allowed.

USER\_BYTES is the size of the file available for user data.

Find out how many segments there are in the SYSAUX tablespace by querying the

DBA\_SEGMENTS view. This number will vary.

Find out which index in the SYSAUX tablespace takes up the most space by querying the DBA\_SEGMENTS view. This number of bytes may vary. The results indicate that the I\_WRI$\_OPTSTAT\_H\_OBJ#\_ICOL#\_ST index takes up the most space.

Viewing Tablespace Information by SQL\*Developer

Launch SQL\*Developer.

Expand **PDB1-system** in the Connections pane.

In the DBA view, Expand **PDB1-system**.

**Note:** If you do not see the DBA panel on the lower left under Connections & Reports panel, then select **View** > **DBA**

Expand **Storage** and then select **Tablespaces**. A new tab named Tablespaces should appear.

All the tablespaces in the **Space** tab are listed with their size, amount of free space, amount used (MB),%Free, %Used, and Maximum Size setting.

In the **Files** tab, File Type, Tablespace Status, File Status, Used (MB), Free (MB), other properties, and Datafile Name are listed.

In the **Free Space** tab, Pieces, Min(MB), Average (MB), Max (MB) and Total (MB) are listed.

Question: In this example, how much of the SYSAUX tablespace is used?

Answer: 94% of the SYSAUX tablespace has been used. It has 25MB of free space left.

**Note:** The values in your database may differ from what is shown in this example.

Close SQL\*Developer.

Practice 16-2: Creating a Tablespace

Overview

In this practice, you create and populate a tablespace named INVENTORY.

Assumptions

You are logged in as the oracle user.

Tasks

Use SQL\*Plus to Create the INVENTORY Tablespace and Table X

As the PDBADMIN user in SQL\*Plus, execute the CreateINVENTORYTablespace.sql script to create the INVENTORY tablespace. Next, execute a script named CreateTableX.sql to create and populate a table called X in the INVENTORY tablespace. At first, you will get an error trying to populate the table. In the next section, you correct the problem.

Set the environment variable for the ORCLCDB database, then start SQL\*Plus and connect to ORCLPDB1 as the PDBADMIN user. Refer to *Course Practice Environment: Security Credentials* for the password value.

Execute the CreateINVENTORYTablespace.sql script.

Execute the CreateTable\_X.sql script to create and populate the X table. Notice that near the end, you get an error message: unable to extend table PDBADMIN.X by

128 in tablespace INVENTORY. You get this message because the tablespace in which you are trying to create table X is too small. You will remedy this problem in the next section.

Use SQL\*Developer to Increase the Size of the INVENTORY01.DBF Data File

Fix the problem that you encountered in the previous section by increasing the size of the INVENTORY01.dbf data file. Use SQL\*Developer because it provides an easy-to-use interface when working with tablespaces.

Launch SQL\*Developer

In the DBA panel, expand **PDB1-pdbadmin**

Expand **Storage** and then select **Tablespaces**.

Double click the **INVENTORY** tablespace and select the **Datafiles** tab.

Now that you have found the name of **INVENTORY01.DBF** data file.

In the DBA pane, select **Datafiles**. You can see that INVENTORY01.DBF datafile is 100% used.

Double click the INVENTORY01.DBF filename. Click **Actions > Edit** .

In the Edit Datafile box, enter a File Size of **40M**. Don't click Apply just yet.

Click the **SQL** tab to view the SQL command that performs the resize action.

In the dialog box, click **Apply**.

In the Successful dialog box, click **OK**. Data file has been successfully resized.

Verify that the change is reflected in the SQL\*Developer interface. The size for the

INVENTORY tablespace should now be set to 40MB.

Use SQL\*Developer to Add a Data File to the INVENTORY Tablespace

In the DBA pane, Expand **Storage**, and click **Tablespaces**

Double click the **INVENTORY** tablespace.

Expand **Actions** and then select **Add Datafile**.

The "Add Datafiles" dialog box is displayed.

Enter File Name: **INVENTORY02.DBF**

Enter File Directory: **/u01/app/oracle/oradata/ORCLCDB/orclpdb1/**

Enter the File Size: **30M**.

Click **SQL** tab and view the SQL code being generated.

Click **Apply**.

In the Successful window, click **OK**

Refresh the **INVENTORY** tab by clicking on the Datafiles subtab and verify that it now has two data files: **INVENTORY01.DBF** and **INVENTORY02.DBF**

Close the SQL\*Developer window.

Use SQL\*Plus to Create Table X and Populate It

As the PDBADMIN user, run the script named CreateTableX.sql again in SQL\*Plus to create and populate the table called X in the INVENTORY tablespace. This time you shouldn't receive an error because you increased the size of the tablespace.

Return to your terminal window.

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

Run the CreateTable\_X.sql script, located in

/home/oracle/labs/DBMod\_Storage. The script runs without any errors.

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

Verify that table X was created in the INVENTORY tablespace.

Use SQL\*Plus to Drop the INVENTORY Tablespace

Drop the INVENTORY tablespace.

Exit SQL\*Plus.

Close the terminal session.

Practice 16-3: Managing Temporary and Permanent Tablespaces

Overview

In this practice, you will manage the permanent and temporary tablespaces in the CDB root and in the PDBs

Assumptions

The PDB, ORCLPDB1, exists and is open.

Tasks

Then execute the $HOME/labs/storage/glogin\_6. This script sets formatting for all columns selected in queries.

View permanent and temporary tablespaces properties in ORCLCDB.

Create a permanent tablespace CDATA in the CDB root.

Make the CDATA tablespace the default tablespace in the root container.

Create a permanent tablespace, LDATA, in ORCLPDB1. Refer to *Course Practice Environment: Security Credentials* for the password value.

Make the LDATA tablespace the default tablespace in the ORCLPDB1 container.

Create a temporary tablespace in the CDB root. Refer to *Course Practice Environment: Security Credentials* for the password value.

Make TEMP\_ROOT the default temporary tablespace in the CDB root.

Create a temporary tablespace TEMP\_PDB1 in ORCLPDB1. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

Make TEMP\_PDB1 the default temporary tablespace in ORCLPDB1.

Create a temporary tablespace MYTEMP in ORCLPDB1.

Display default tablespaces of another PDB in ORCLCDB. Create a new PDB using the

$HOME/labs/DBMod\_Storage/setup\_newpdb.sql SQL script. This script creates a new PDB, queries it for the default tablespaces, and then drops the PDB.

Manage default permanent and temporary tablespaces of users.

Create a common user C##U. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

View the default tablespace and temporary tablespace assignment for user C##U in all containers.

Create a local user LU in ORCLPDB1. Refer to *Course Practice Environment: Security Credentials* for the ***password*** value.

View the default tablespace and temporary tablespace assignment for user LU.

Change the temporary tablespace assignment for user LU to MY\_TEMP in ORCLPDB2.

View the default temporary tablespace assignment for user LU.

Log out of SQL\*Plus.

Close all terminals.