# Practices for Lesson 26: Using External Tables to Load and Transport Data

Practices for Lesson 26: Overview

Overview

In these practices, you will query and unload external tables.

Practice 26-1: Querying External Tables

Overview

In this practice, you query partitioned external tables.

Suppose you received new external files containing records about sales. The sales records are dispatched in two files according to the sales year:

/home/oracle/labs/DBMod\_LoadTrans/DP\_sales\_1998.dat

/home/oracle/labs/DBMod\_LoadTrans/DP2\_sales\_1999.dat

You don't want to load or insert the records into a table in ORCLPDB1, rather you want to be able to read the sales data from the external files.

Assumptions

N/A

Tasks

Open a new terminal window and use oraenv to set the environment variables for

orclcdb database.

Execute the $HOME/labs/DBMod\_loadTrans/DP\_glogin.sh shell script to set formatting for all columns selected in queries and to place both .dat files in DP and DP2 subdirectories.

**Note:** You can ignore the error message about not being able to remove the orders.dmp

file.

Start SQL\*Plus and connect to ORCLPDB1 as the SYSTEM user. See the “Course Practice Environment: Security Credentials” document in your Activity Guide for the ***password***.

In ORCLPDB1, create the SH.SALES\_EXT\_RANGE external table.

Create two directories in the database that point to where the external files are stored.

Create an SH schema for the sales data. See Product-Specific Credentials for the ***password***. Grant the SH user CREATE SESSION and CREATE TABLE privileges. Also grant the SH user READ WRITE privileges on the directories you just created (ext\_dir and ext\_dir2).

In case it already exists, drop the SH.SALES\_EXT\_RANGE table. You should get an error stating that the table does not exist.

View the script $HOME/labs/DBMod\_LoadTrans/external\_table.sql

Execute the following code to create the structure of the external table SH.SALES\_EXT\_RANGE. The code partitions the table on the TIME\_ID column. You can copy the code from $HOME/labs/DBMod\_LoadTrans/external\_table.sql and paste it into SQL\*Plus.

**Question:** Based on the code in the previous step, which directories does the external table use?

**Answer:** The partitions of the external table use two directories. The default directory for any partition created is ext\_dir. The last partition uses another directory, ext\_dir2, which corresponds to the active files for the current sales.

Verify that the locations are correctly set for the partitions by querying the

DBA\_XTERNAL\_LOC\_PARTITIONS view.

Determine the number of rows in the external table based on specific criteria.

Determine the number of rows for sales in 1998.

Determine the number of rows for sales in 1999.

Determine the number of rows for sales in both 1998 and 1999.

Exit SQL\*Plus.

Issue the following commands to find out whether the number of rows read is equivalent to the number of records that exist in the two external files. The results show that the number of records in the DP\_sales\_1998.dat file is 357675 and the number of records in the DP2\_sales\_1999.dat file is 495899. Together, the number of records equals 853574. This value is higher than the number of rows read, which you found to equal 853558 in the previous step.

Check the log files to determine the reason for the discrepancy.

List the log files. Note: your log file names will be different.

View the content of all log files. According to the log files, there were 16 records that could not be "inserted" into the external table structure because some fields in the external files contained NULL value whereas the column in the table is set to NOT NULL.

$ more /home/oracle/labs/DBMod\_LoadTrans/DP/\*.log

...

error processing column TIME\_ID in row 50000 for datafile

/home/oracle/labs/DBMod\_LoadTrans/DP/DP\_sales\_1998.dat

ORA-01400: cannot insert NULL into (TIME\_ID)error processing column TIME\_ID

in row 100000 for datafile

/home/oracle/labs/DBMod\_LoadTrans/DP/DP\_sales\_1998.dat ORA-01400: cannot insert NULL into (TIME\_ID)

LOG file opened at 11/24/16 16:32:39

...

error processing column TIME\_ID in a row for datafile

/home/oracle/labs/DBMod\_LoadTrans/DP/DP\_sales\_1998.dat ORA-01400: cannot insert NULL into (TIME\_ID)

error processing column TIME\_ID in a row for datafile

/home/oracle/labs/DBMod\_LoadTrans/DP/ DP\_sales\_1998.dat

ORA-01400: cannot insert NULL into (TIME\_ID)

error processing column TIME\_ID in a row for datafile

/home/oracle/labs/DBMod\_LoadTrans/DP/ DP\_sales\_1998.dat

ORA-01400: cannot insert NULL into (TIME\_ID)

error processing column TIME\_ID in a row for datafile

/home/oracle/labs/DBMod\_LoadTrans/DP/ DP\_sales\_1998.dat

…

error processing column TIME\_ID in a row for datafile

/home/oracle/labs/DBMod\_LoadTrans/DP2/DP2\_sales\_1999.dat

Start SQL\*Plus and connect to ORCLPDB1 as the SYSTEM user. See Product-Specific Credentials for the ***password***.

Attempt to create an index on the partition key of the external table to get better query performance. The resulting error indicates that you cannot create an index on an external organized table.

Suppose that a new file with sales for year 2000 has arrived. Add a new partition called

year2000 to the table.

Count the number of sales rows in the SH.SALES\_EXT\_RANGE table for year 2000. The number of rows read equals 235893.

Count the actual number of rows in the DP2\_sales\_2000.dat file. Again, the result indicates that the number of rows read (235893) is less than the number of rows in the data file (235898). The discrepancy may or may not be due to null rows getting discarded, as

you observed in preceding steps.

Perform another check on the data. Query the number of rows that have a TIME\_ID value that falls within the year 2000. The results show that the database read only one row.

Exit SQL\*Plus.

View the contents of the DP2\_sales\_2000.dat file. Notice that most records do not contain sales for year 2000. You must ensure that the records satisfy the partitioning conditions. If you were to remedy this situation, you would need to create two distinct files: one for 2000 sales and another one for 2001 sales, and then add another partition for 2001 sales.

Close the terminal window.

Practice 26-2: Unloading External Tables

Overview

In this practice, you will write the OE.ORDERS table data to a dump file using the external tables.

Assumptions

You completed Practice 4-1 Querying External Tables (only steps 1, 2, and 3 are necessary).

Tasks

Open a terminal window and use oraenv to set the environment variables for the orclcdb

database.

Start SQL\*Plus and connect to ORCLPDB1 as the SYSTEM user. See Product-Specific Credentials for the ***password***.

In ORCLPDB1, create an external table called OE.ORDERS\_EXT that unloads the rows from OE.ORDERS to an external file called orders.dmp. Later, that file will be read from an external table in ORCLPDB2.

Verify that the external file (orders.dmp) is listed in the

/home/oracle/labs/DBMod\_LoadTrans/DP directory. The result indicates that it is listed. Your date will be different than the one shown below.

Determine the number of rows in the OE.ORDERS\_EXT table

Connect to ORCLPDB2 as the SYSTEM user. See Product-Specific Credentials for the

***password***.

Create a user named OE. If the user exists, drop it first. See “Course Practice Environment: Security Credentials” for the ***password***.

Create an external directory named ext\_dir.

Create an external table named OE.ORDERS\_EXT that loads orders.dmp.

Try to query the entire OE.ORDERS\_EXT table. You get an error.

**Question:** Which type of access driver was used to unload the data from the table into an external file?

**Answer:** The access driver was ORACLE\_DATAPUMP. The binary file (orders.dmp) created has the same format as the files used by the Data Pump Import and Export utilities and can be interchanged with them. During the loading (reading from the external table), the same access driver must be used.

Re-create the external table with the appropriate access driver.

Drop the OE.ORDERS\_EXT table that you just created.

Create the OE.ORDERS\_EXT table again, and this time, specify TYPE ORACLE\_DATAPUMP (see line 7 below) instead of what you used before, which was TYPE ORACLE\_LOADER.

Query the entire OE.ORDERS\_EXT table again. This time, the query returns 105 rows.

Exit SQL\*Plus and close the terminal window.