# Practices for Lesson 24: Loading Data

Practices for Lesson 24: Overview

Overview

In these practices, you will use SQL\*Loader to load data.

Practice 24-1: Loading Data into a PDB from an External File

Overview

In this practice, you use SQL\*Loader to perform the following load operations:

Load data into the SH.PRODUCTS table in ORCLPDB1 by using SQL\*Loader in express mode. Data and control files are provided.

Load data into the SH.INVENTORIES table in ORCLPDB1 by using SQL\*Loader in conventional mode.

Load data into the SH.INVENTORIES table in ORCLPDB1 by using SQL\*Loader in direct mode.

Assumptions

You are logged in as the oracle user.

Tasks

Load Data by Using SQL\*Loader in Express Mode

As the SH user, use SQL\*Loader in Express Mode to load data from the

$HOME/labs/DBMod\_LoadTrans/products.dat data file into the SH.PRODUCTS table in

ORCLPDB1.

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

database. Use the dbstart.sh script to start the database and listener.

Execute the $HOME/labs/DBMod\_LoadTrans/DP\_setup.sh shell script. Note: this script take take apx 2 minutes to run.

View the products.dat file to learn about its structure.

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

Count the number of rows in the SH.PRODUCTS table. The results indicate that there are seven rows in the table and, therefore, seven products.

Exit SQL\*Plus.

Change to the /home/oracle/labs/DBMod\_LoadTrans directory & verify.

Start SQL\*Loader, connect to ORCLPDB1 as the SH user, and load the records from the products.dat file into the SH.PRODUCTS table in ORCLPDB1. The results show that eight rows were successfully loaded. Refer to “Course Practice Environment: Security Credentials” for the ***password*** value.

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

Verify that the table is loaded with the eight records from the products.dat file. The results show that the records were loaded.

Exit SQL\*Plus.

View the products.log file.

$ cat products.log

SQL\*Loader: Release 19.0.0.0.0 - Production on Wed Oct 21 23:24:22 2020

Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.

Express Mode Load, Table: PRODUCTS Data File: products.dat

Bad File: products\_%p.bad Discard File: none specified

(Allow all discards)

Number to load: ALL Number to skip: 0 Errors allowed: 50

Continuation: none specified Path used: External Table

Table PRODUCTS, loaded from every logical record. Insert option in effect for this table: APPEND

Column Name Position Len Term Encl Datatype

Generated control file for possible reuse: OPTIONS(EXTERNAL\_TABLE=EXECUTE, TRIM=LRTRIM) LOAD DATA

INFILE 'products' APPEND

INTO TABLE PRODUCTS FIELDS TERMINATED BY "," (

PRODUCT\_ID,

COUNTRY, LABEL, DETAILED\_LABEL

)

End of generated control file for possible reuse.

created temporary directory object SYS\_SQLLDR\_XT\_TMPDIR\_00000 for path /home/oracle/labs/DBMod\_LoadTrans

enable parallel DML: ALTER SESSION ENABLE PARALLEL DML

creating external table "SYS\_SQLLDR\_X\_EXT\_PRODUCTS" CREATE TABLE "SYS\_SQLLDR\_X\_EXT\_PRODUCTS"

(

"PRODUCT\_ID" NUMBER(6), "COUNTRY" CHAR(3), "LABEL" VARCHAR2(10),

"DETAILED\_LABEL" VARCHAR2(20)

)

ORGANIZATION external (

TYPE oracle\_loader

DEFAULT DIRECTORY SYS\_SQLLDR\_XT\_TMPDIR\_00000 ACCESS PARAMETERS

(

RECORDS DELIMITED BY NEWLINE CHARACTERSET US7ASCII

BADFILE 'SYS\_SQLLDR\_XT\_TMPDIR\_00000':'products\_%p.bad' LOGFILE 'products\_%p.log\_xt'

READSIZE 1048576

FIELDS TERMINATED BY "," LRTRIM REJECT ROWS WITH ALL NULL FIELDS (

"PRODUCT\_ID" CHAR(255), "COUNTRY" CHAR(255), "LABEL" CHAR(255), "DETAILED\_LABEL" CHAR(255)

)

)

location (

'products.dat'

**Question:** Which operations did SQL\*Loader execute in express mode?

**Answer:** SQL\*Loader first created a temporary external table, used the external table to load the content of the external data file into the table, and finally dropped the temporary external table.

In the /home/oracle/labs/DBMod\_LoadTrans directory where you are working, find the file named products\_nnnn.log\_xt that you just created and display its contents. The date in the file listing will distinguish the right file from the others.

Load Data by Using SQL\*Loader in Conventional Mode

In this section, you will load data into the SH.INVENTORIES table in ORCLPDB1 by using SQL\*Loader in conventional mode. Currently, there are 476 rows in the SH.INVENTORIES table.

Make sure that your current directory is /home/oracle/labs/DBMod\_LoadTrans.

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

Determine the number of rows in the SH.INVENTORIES table. The result shows 476 rows.

Exit from SQL\*Plus.

Start SQL\*Loader, connect to ORCLPDB1 as the SH user, and load the SH.INVENTORIES table from the $HOME/labs/DBMod\_LoadTrans/DP\_inventories.dat data file in conventional mode. Refer to “Course Practice Environment: Security Credentials” for the ***password*** value. The result shows that 83 rows were successfully loaded in the SH.INVENTORIES table.

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

Hint: use the Linux command line buffer recall using the up arrow.

Determine the number of rows in the SH.INVENTORIES table. The result shows 559 rows.

**Question:** Did SQL\*Loader append new rows or replace rows in the SH.INVENTORIES

table?

**Answer:** Originally, there were 476 rows in this table. Now there are 559 rows, which means 83 new rows were added, or "appended," by SQL\*Loader.

Exit SQL\*Plus.

View the inventories.log file. Notice that the insert option in effect for the

SH.INVENTORIES table is APPEND.

View the end of the control file named DP\_inventories.ctl in the cat command. Notice the APPEND command.

Using the vi editor, change APPEND to TRUNCATE so that the control file truncates the table. Save the file and quit the vi editor. Hint: to write and quite, use :wq

Start SQL\*Loader, connect to ORCLPDB1 as the SH user, and re-execute the load operation with the ROWS parameter set to 10. Refer to “Course Practice Environment: Security Credentials” for the ***password*** value.

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

Verify the number of rows in the INVENTORIES table. The table now has 83 rows. The

TRUNCATE option cleared out the original rows in the table and inserted 83 new rows.

Re-enable the Check Constraint

Suppose a DBA discovers that the CHECK constraint was disabled on the WAREHOUSE\_ID column in the SH.INVENTORIES table at the time of the load. This disabled constraint allowed only values within a certain range. Use the $HOME/labs/DBMod\_LoadTrans/DP\_check.sql SQL script to empty the table and re-enable the check constraint. Then reload the table.

Execute the $HOME/labs/DBMod\_LoadTrans/DP\_check.sql script.

Start SQL\*Loader, connect to ORCLPDB1 as the SH user, and reload the table. Refer to “Course Practice Environment: Security Credentials” for the ***password*** value. The results indicate that 20 rows were successfully loaded into the SH.INVENTORIES table.

View the inventories.log file. The log file says that 20 rows were successfully loaded into the SH.INVENTORIES table; however, 51 rows were not loaded due to errors.

0 Rows not loaded because all WHEN clauses were failed.

0 Rows not loaded because all fields were null.

…

Record 70: Rejected - Error on table SH.INVENTORIES.

ORA-02290: check constraint (SH.CK\_WAREHOUSE\_ID) violated

Record 71: Rejected - Error on table SH.INVENTORIES.

ORA-02290: check constraint (SH.CK\_WAREHOUSE\_ID) violated

MAXIMUM ERROR COUNT EXCEEDED - Above statistics reflect partial run.

Table SH.INVENTORIES:

20 Rows successfully loaded.

51 Rows not loaded due to data errors.

0 Rows not loaded because all WHEN clauses were failed.

0 Rows not loaded because all fields were null.

Space allocated for bind array: 7740 bytes(10 rows)

Read buffer bytes: 1048576

Total logical records skipped: 0

Total logical records read: 80

Total logical records rejected: 51

Total logical records discarded: 0

Run began on Thu Oct 22 00:36:20 2020

**Question:** Did SQL\*Loader try to load all rows?

**Answer:** No. After 20 rows successfully loaded, 51 rows did not load due to a constraint violation error. The load stopped at this point. The default number of errors tolerated is 50. When the number was exceeded, SQL\*Loader stopped.

Load Data by Using SQL\*Loader in Direct Mode

Observe how SQL\*Loader behaves when loading the SH.INVENTORIES table in direct mode.

Start SQL\*Loader, connect to ORCLPDB1 as the SH user, and load the SH.INVENTORIES

table in direct mode. The results indicate that the load completed and the record count is

83. Refer to “Course Practice Environment: Security Credentials” for the ***password*** value.

**Question:** Does the direct load use the SQL INSERT statement? How does the direct path commit the rows inserted?

**Answer:** The direct load loads records into the blocks, writing the data blocks directly to the database files. You can observe that there is no COMMIT instruction, but SAVE instead.

During a data save, only full database blocks are written to the database.

**Question:** Did it enforce the CHECK constraint?

**Answer:** No, it did not. This is the reason all rows were loaded, regardless of the

WAREHOUSE\_ID value to be inserted.

**Question:** Does SQL\*Loader in direct mode ignore all constraints?

**Answer:** No, it does not. It enforces PRIMARY KEY, UNIQUE, and NOT NULL constraints.

Close the terminal window.