# Practices for Lesson 15: Managing Schema Objects

Practices for Lesson 15: Overview

Practice Overview

This practice covers the following topics:

Adding and dropping constraints

Deferring constraints

Creating external tables

**Note:** Before starting this practice, execute the **/home/oracle/labs/sql2/code\_ex/**

/cleanup\_scripts/cleanup\_15.sql script.

Practice 15-1: Managing Schema Objects

Overview

In this practice, you add, drop, and defer constraints. You also create external tables.

**Note:** Execute the cleanup\_15.sql script from **/home/oracle/labs/sql2/code\_ex/**

**/cleanup\_scripts/** before performing the following tasks.

Tasks

Create the DEPT2 table based on the following table instance chart. Enter the syntax in the SQL Worksheet. Then execute the statement to create the table. Confirm that the table is created.

Populate the DEPT2 table with data from the DEPARTMENTS table. Include only the columns that you need. Confirm that the rows are inserted.

**…**

Create the EMP2 table based on the following table instance chart. Enter the syntax in the SQL Worksheet. Then execute the statement to create the table. Confirm that the table is created.

Add a table-level PRIMARY KEY constraint to the EMP2 table on the ID column. The constraint should be named at creation. Name the constraint my\_emp\_id\_pk.

Create a PRIMARY KEY constraint to the DEPT2 table by using the ID column. The constraint should be named at creation. Name the constraint my\_dept\_id\_pk.

Add a foreign key reference on the EMP2 table that ensures that the employee is not assigned to a nonexistent department. Name the constraint my\_emp\_dept\_id\_fk.

Modify the EMP2 table. Add a COMMISSION column of the NUMBER data type, precision 2, scale 2. Add a constraint to the COMMISSION column that ensures that a commission value is greater than zero.

Drop the EMP2 and DEPT2 tables so that they cannot be restored.

Create an external table library\_items\_ext. Use the ORACLE\_LOADER access driver.

**Note:** The emp\_dir directory object has to be created at first. Refer to the solution on how to create emp\_dir directory object. Ensure that the external file and the database are on the same machine.

library\_items**.**dat can be found in the labs/sql2/emp\_dir/ folder. library\_items.dat has records in the following format:

Open the lab\_15\_09.sql file. Observe the code snippet to create the library\_items\_ext external table. Then replace **<***TODO1***>**, **<***TODO2***>**, **<***TODO3***>**, and <*TODO4*> as appropriate and save the file as lab\_15\_09\_soln.sql. Run the script to create the external table.

Query the library\_items\_ext table.

The HR department needs a report of the addresses of all departments. Create an external table as dept\_add\_ext by using the ORACLE\_DATAPUMP access driver. The report should show the location ID, street address, city, state or province, and country in the output. Use a NATURAL JOIN to produce the results.

**Note:** The emp\_dir directory is already created for this exercise.

Open the lab\_15\_10.sql file. Observe the code snippet to create the dept\_add\_ext external table. Then replace **<***TODO1***>**, **<***TODO2***>** and **<***TODO3***>** as appropriate and save the script as lab\_15\_10\_soln.sql.

Run the lab\_15\_10\_soln.sql script to create the external table.

Query the dept\_add\_ext table.

**Note:** When you perform the preceding step, two files oraxx\_emp4.exp and

oraxx\_emp5.exp are created in the default directory emp\_dir.

Create the emp\_books table and populate it with data. Set the primary key as deferred and observe what happens at the end of the transaction.

Run the lab\_15\_11\_a.sql file to create the emp\_books table. Observe that the

emp\_books\_pk primary key is not created as deferrable.

Run the lab\_15\_11\_b.sql file to populate data into the emp\_books table. What do you observe?

Set the emp\_books\_pk constraint as deferred. What do you observe?

Drop the emp\_books\_pk constraint.

Modify the emp\_books table definition to add the emp\_books\_pk constraint as deferrable this time.

Set the emp\_books\_pk constraint as deferred.

Run the lab\_15\_11\_g.sql file to populate data into the emp\_books table. What do you observe?

Commit the transaction. What do you observe?

Solution 5-1: Managing Schema Objects

Solution

Create the DEPT2 table based on the following table instance chart. Enter the syntax in the SQL Worksheet. Then execute the statement to create the table. Confirm that the table is created.

Populate the DEPT2 table with data from the DEPARTMENTS table. Include only the columns that you need. Confirm that the rows are inserted.

Create the EMP2 table based on the following table instance chart. Enter the syntax in the SQL Worksheet. Then execute the statement to create the table. Confirm that the table is created.

Add a table-level PRIMARY KEY constraint to the EMP2 table on the ID column. The constraint should be named at creation. Name the constraint my\_emp\_id\_pk.

Create a PRIMARY KEY constraint to the DEPT2 table by using the ID column. The constraint should be named at creation. Name the constraint my\_dept\_id\_pk.

Add a foreign key reference on the EMP2 table that ensures that the employee is not assigned to a nonexistent department. Name the constraint my\_emp\_dept\_id\_fk.

Modify the EMP2 table. Add a COMMISSION column of the NUMBER data type, precision 2, scale 2. Add a constraint to the COMMISSION column that ensures that a commission value is greater than zero.

Drop the EMP2 and DEPT2 tables so that they cannot be restored.

Create an external table library\_items\_ext. Use the ORACLE\_LOADER access driver.

**Note:** The emp\_dir directory object has to be created at first.

library\_items**.**dat can be found in the labs/sql2/emp\_dir/ folder. Ensure that the external file and the database are on the same machine.

library\_items.dat has records in the following format:

Open the lab\_15\_09.sql file. Observe the code snippet to create the

library\_items\_ext external table. Then replace **<*TODO1*>**, **<*TODO2*>**,

**<*TODO3*>**, and <***TODO4***> as appropriate and save the file as lab\_15\_09\_soln.sql. Run the script to create the external table.

Query the library\_items\_ext table.

The HR department needs a report of addresses of all the departments. Create an external table as dept\_add\_ext by using the ORACLE\_DATAPUMP access driver. The report should show the location ID, street address, city, state or province, and country in the output. Use a NATURAL JOIN to produce the results.

**Note:** The emp\_dir directory is already created for this exercise. Ensure that the external file and the database are on the same machine.

Open the lab\_15\_10.sql file. Observe the code snippet to create the dept\_add\_ext external table. Then replace **<***TODO1***>**, **<***TODO2***>** and **<***TODO3***>** as appropriate and save the script as lab\_15\_10\_soln.sql.

**Note:** When you perform the preceding step, two files ora21\_emp4.exp and

ora21\_emp5.exp are created in the default directory emp\_dir.

Run the lab\_15\_10\_soln.sql script to create the external table.

Query the dept\_add\_ext table.

Create the emp\_books table and populate it with data. Set the primary key as deferred and observe what happens at the end of the transaction.

Run the lab\_15\_11\_a.sql script to create the emp\_books table. Observe that the

emp\_books\_pk primary key is not created as deferrable.

Run the lab\_15\_11\_b.sql script to populate data into the emp\_books table. What do you observe?

The first row is inserted. However, you see the ora-00001 error with the second row insertion.

Set the emp\_books\_pk constraint as deferred. What do you observe?

You see the following error: “ORA-02447: Cannot defer a constraint that is not deferrable.”

Drop the emp\_books\_pk constraint.

Modify the emp\_books table definition to add the emp\_books\_pk constraint as deferrable this time.

Set the emp\_books\_pk constraint as deferred.

Run the lab\_15\_11\_g.sql script to populate data into the emp\_books table. What do you observe?

You see that all the rows are inserted.

Commit the transaction. What do you observe?

You see that the transaction is rolled back by the database at this point, because

COMMIT failed due to the constraint violation.