Practices for Lesson 2: Creating Procedures

Chapter 2

Practices for Lesson 2: Overview

Overview

In this practice, you create, compile, and invoke procedures that issue DML and query commands. You also learn how to handle exceptions in procedures.

Note:

1. Before starting this practice, execute the `/home/oracle/labs/plpu/code_ex/cleanup_scripts/cleanup_02.sql` script.
2. If you missed a step in a practice, please run the appropriate solution script for that practice step before proceeding to the next step or the next practice.

Practice 2-1: Creating, Compiling, and Calling Procedures

Overview

In this practice, you create and invoke the ADD_JOB procedure and review the results. You also create and invoke a procedure called UPD_JOB to modify a job in the JOBS table and create and invoke a procedure called DEL_JOB to delete a job from the JOBS table. Finally, you create a procedure called GET_EMPLOYEE to query the EMPLOYEES table, retrieving the salary and job ID for an employee when provided with the employee ID.

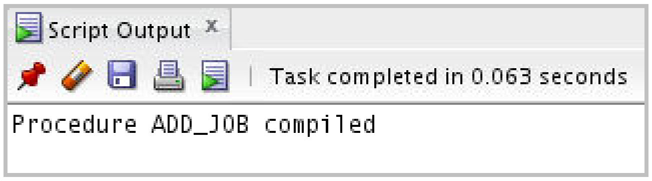
Note: Execute cleanup_02.sql from /home/oracle/labs/plpu/code_ex/cleanup_scripts/ before performing the following task.

Task

1. Create, compile, and invoke the ADD_JOB procedure and review the results.
   a. Create a procedure called ADD_JOB to insert a new job into the JOBS table. Provide the ID and job title using two parameters.

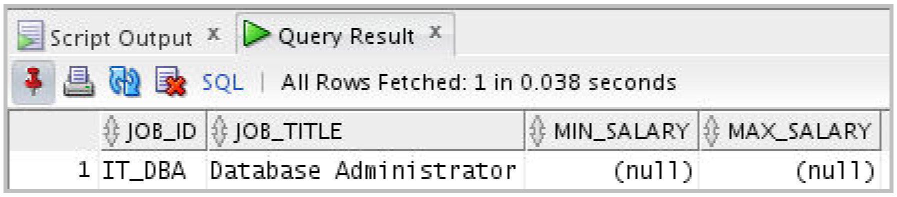
      Note: You can create the procedure (and other objects) by entering the code in the SQL Worksheet area, and then click the Run Script (F5) icon. This creates and compiles the procedure. To find out whether or not the procedure has any errors, click the procedure name in the procedure node, and then select Compile from the pop-up menu.

   b. Invoke the procedure with IT_DBA as the job ID and Database Administrator as the job title. Query the JOBS table and view the results.



Task completed in 0.063 seconds

```
Procedure ADD_JOB compiled
```

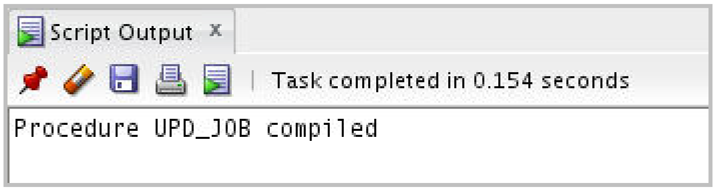


All Rows Fetched: 1 in 0.038 seconds

	JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
1	IT_DBA	Database Administrator	(null)	(null)

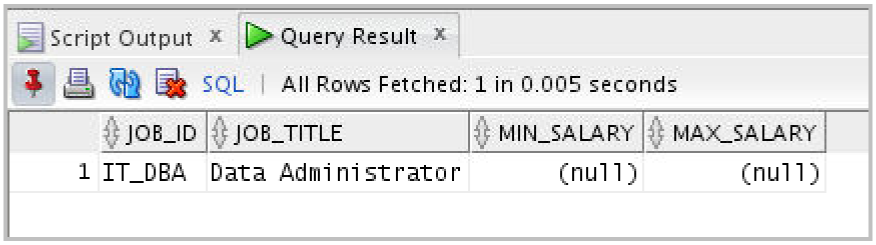
   c. Invoke your procedure again, passing a job ID of ST_MAN and a job title of Stock Manager. What happens and why?

2. Create a procedure called UPD_JOB to modify a job in the JOBS table.
   a. Create a procedure called UPD_JOB to update the job title. Provide the job ID and a new title using two parameters. Include the necessary exception handling if no update occurs.
   b. Invoke the procedure to change the job title of the job ID IT_DBA to Data Administrator. Query the JOBS table and view the results.



Task completed in 0.154 seconds

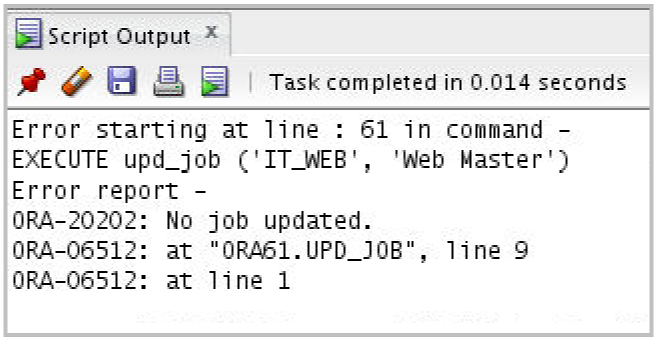
```
Procedure UPD_JOB compiled
```



All Rows Fetched: 1 in 0.005 seconds

	JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
1	IT_DBA	Data Administrator	(null)	(null)

c. Test the exception-handling section of the procedure by trying to update a job that does not exist. You can use the job ID `IT_WEB` and the job title `Web Master`.

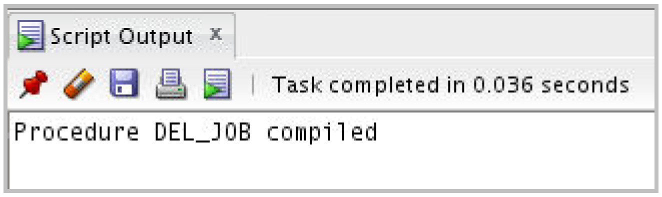


Task completed in 0.014 seconds

```
Error starting at line : 61 in command -
EXECUTE upd_job ('IT_WEB', 'Web Master')
Error report -
ORA-20202: No job updated.
ORA-06512: at "ORA61.UPD_JOB", line 9
ORA-06512: at line 1
```

3. Create a procedure called `DEL_JOB` to delete a job from the `JOBS` table.

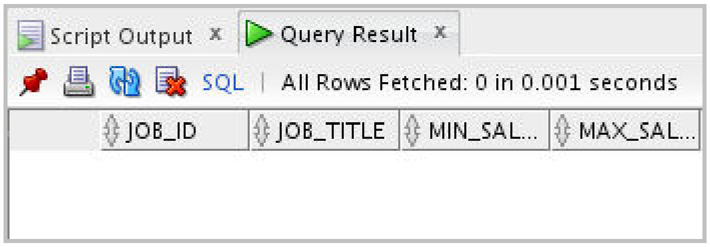
a. Create a procedure called `DEL_JOB` to delete a job. Include the necessary exception-handling code if no job is deleted.



Task completed in 0.036 seconds

```
Procedure DEL_JOB compiled
```

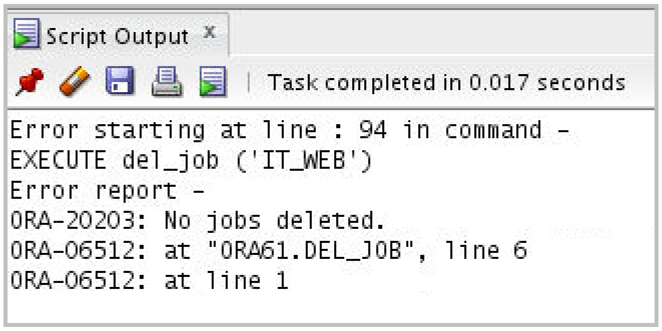
b. Invoke the procedure using the job ID `IT_DBA`. Query the `JOBS` table and view the results.



All Rows Fetched: 0 in 0.001 seconds

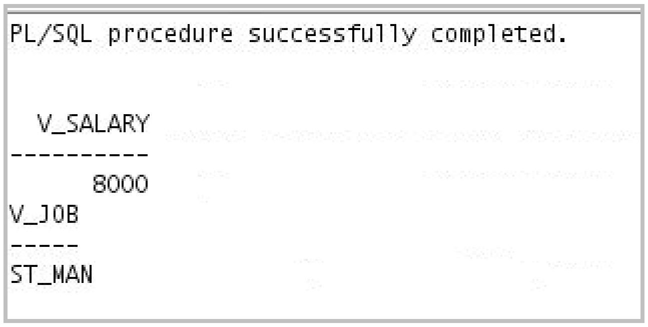
	JOB_ID	JOB_TITLE	MIN_SAL...	MAX_SAL...

c. Test the exception-handling section of the procedure by trying to delete a job that does not exist. Use `IT_WEB` as the job ID. You should get the message that you included in the exception-handling section of the procedure as the output.



```
Error starting at line : 94 in command -
EXECUTE del_job ('IT_WEB')
Error report -
ORA-20203: No jobs deleted.
ORA-06512: at "ORA61.DEL_JOB", line 6
ORA-06512: at line 1
```

4. Create a procedure called `GET_EMPLOYEE` to query the `EMPLOYEES` table, retrieving the salary and job ID for an employee when provided with the employee ID.
   a. Create a procedure that returns a value from the `SALARY` and `JOB_ID` columns for a specified employee ID. Remove syntax errors, if any, and then recompile the code.
   b. Execute the procedure using host variables for the two `OUT` parameters—one for the salary and the other for the job ID. Display the salary and job ID for employee ID 120.



```
PL/SQL procedure successfully completed.


  V_SALARY
----------
      8000
V_JOB
-----
ST_MAN
```

   c. Invoke the procedure again, passing an `EMPLOYEE_ID` of `300`. What happens and why?