Practices for Lesson 10: Introducing Stored Procedures and Functions

Chapter 10

Practice 10: Creating and Using Stored Procedures

Note: If you have executed the code examples for this lesson, make sure you execute the following code before starting this practice:

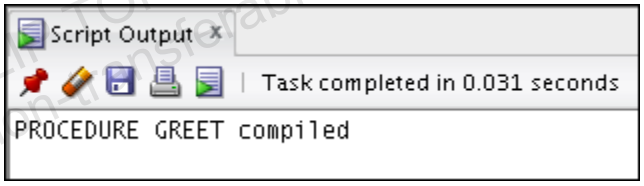
```
DROP table dept;
DROP procedure add_dept;
DROP function check_sal;
```

In this practice, you modify existing scripts to create and use stored procedures.

1. Open `sol_03.sql` script from the `/home/oracle/labs/plsf/soln/` folder. Copy the code under task 4 into a new worksheet.

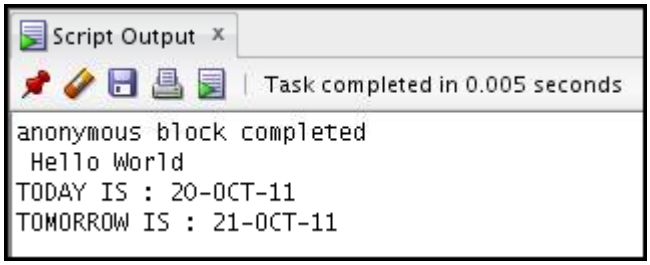
```
SET SERVEROUTPUT ON

DECLARE
   v_today DATE:=SYSDATE;
   v_tomorrow v_today%TYPE;
BEGIN
   v_tomorrow:=v_today +1;
   DBMS_OUTPUT.PUT_LINE(' Hello World ');
   DBMS_OUTPUT.PUT_LINE('TODAY IS : '|| v_today);
   DBMS_OUTPUT.PUT_LINE('TOMORROW IS : '|| v_tomorrow);
END;
```

   a. Modify the script to convert the anonymous block to a procedure called `greet`. (**Hint:** Also remove the `SET SERVEROUTPUT ON` command.)
   b. Execute the script to create the procedure. The output results should be as follows:



   c. Save this script as `lab_10_01_soln.sql`.
   d. Click the Clear button to clear the workspace.
   e. Create and execute an anonymous block to invoke the `greet` procedure. (**Hint:** Ensure that you enable `SERVEROUTPUT` at the beginning of the block.)

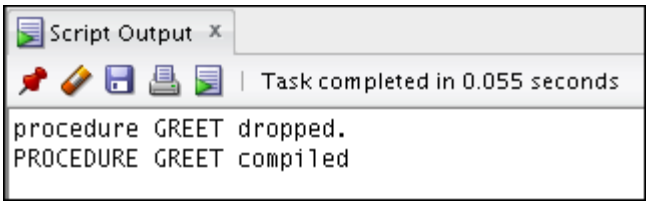
The output should be similar to the following:



2. Modify the `lab_10_01_soln.sql` script as follows:
   a. Drop the `greet` procedure by issuing the following command:

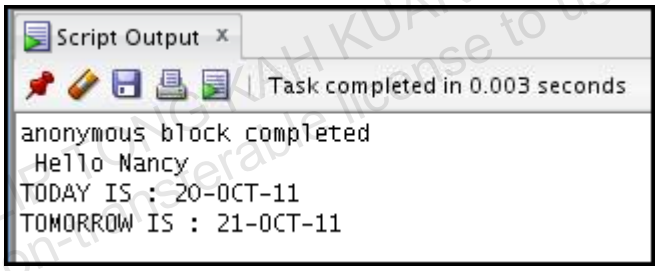
```
DROP PROCEDURE greet;
```

   b. Modify the procedure to accept an argument of type `VARCHAR2`. Call the argument `p_name`.
   c. Print `Hello <name>` (that is, the contents of the argument) instead of printing `Hello World`.
   d. Save your script as `lab_10_02_soln.sql`.
   e. Execute the script to create the procedure. The output results should be as follows:



   f. Create and execute an anonymous block to invoke the `greet` procedure with a parameter value. The block should also produce the output.

The sample output should be similar to the following:



Solution 10: Creating and Using Stored Procedures

In this practice, you modify existing scripts to create and use stored procedures.

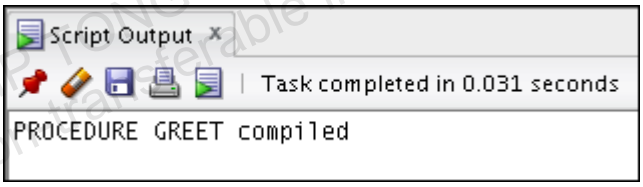
1. Open the `sol_03.sql` script from the `/home/oracle/labs/plsf/soln/` folder. Copy the code under task 4 into a new worksheet.

```
SET SERVEROUTPUT ON

DECLARE
   v_today DATE:=SYSDATE;
   v_tomorrow v_today%TYPE;
BEGIN
   v_tomorrow:=v_today +1;
   DBMS_OUTPUT.PUT_LINE(' Hello World ');
   DBMS_OUTPUT.PUT_LINE('TODAY IS : '|| v_today);
   DBMS_OUTPUT.PUT_LINE('TOMORROW IS : ' || v_tomorrow);
END;
```

   a. Modify the script to convert the anonymous block to a procedure called `greet`. (**Hint:** Also remove the `SET SERVEROUTPUT ON` command.)

```
CREATE PROCEDURE greet IS
   V_today DATE:=SYSDATE;
   V_tomorrow today%TYPE;
...
```

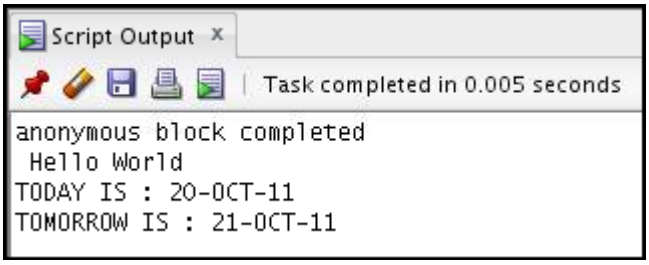
   b. Execute the script to create the procedure. The output results should be as follows:



   c. Save this script as `lab_10_01_soln.sql`.
   d. Click the Clear button to clear the workspace.
   e. Create and execute an anonymous block to invoke the `greet` procedure. (**Hint:** Ensure that you enable `SERVEROUTPUT` at the beginning of the block.)

```
SET SERVEROUTPUT ON

BEGIN
 greet;
END;
```

The output should be similar to the following:



2. Modify the `lab_10_01_soln.sql` script as follows:
   a. Drop the `greet` procedure by issuing the following command:

```
DROP PROCEDURE greet;
```

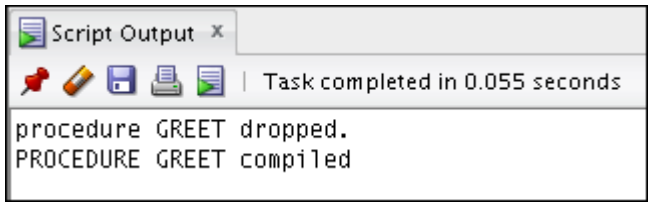
   b. Modify the procedure to accept an argument of type `VARCHAR2`. Call the argument `p_name`.

```
CREATE PROCEDURE greet(p_name VARCHAR2) IS
   V_today DATE:=SYSDATE;
   V_tomorrow today%TYPE;
```

   c. Print `Hello` *<name>* instead of printing `Hello World`.

```
BEGIN
   V_tomorrow:=v_today +1;
   DBMS_OUTPUT.PUT_LINE(' Hello '|| p_name);
...
```

   d. Save your script as `lab_10_02_soln.sql`.
   e. Execute the script to create the procedure. The output results should be as follows:



   f. Create and execute an anonymous block to invoke the `greet` procedure with a parameter value. The block should also produce the output.

```
SET SERVEROUTPUT ON;
BEGIN
 greet('Nancy');
END;
```

The sample output should be similar to the following: