ORACLE
Academy

Database Programming with PL/SQL

9-4: Managing Procedures and Functions

Practice Activities

Vocabulary

Identify the vocabulary word for each definition below:

	The dictionary table that contains source code for all of the subprograms that you own.
	The dictionary table that contains the names and types of procedures and functions that you own.
	The dictionary table that contains source code for subprograms that are owned by others who have granted you the EXECUTE privilege.

Try It / Solve It

1. Complete the steps below to see how exceptions are propagated.

   A. Execute the following two SQL statements to create a duplicate of the DEPARTMENTS table, with department_id as the primary key.

   ```
   CREATE TABLE my_depts AS SELECT * FROM departments;

   ALTER TABLE my_depts
     ADD CONSTRAINT my_dept_id_pk PRIMARY KEY (department_id);
   ```

   B. Examine the following code and create the procedure. Save your work (you will need to modify the procedure code later).

```
CREATE OR REPLACE PROCEDURE add_my_dept
  (p_dept_id IN VARCHAR2, p_dept_name IN VARCHAR2) IS
BEGIN
  INSERT INTO my_depts (department_id, department_name)
    VALUES (p_dept_id, p_dept_name);
END add_my_dept;
```

C. What do you think would happen if you execute this procedure to insert department_id 10 (which already exists)? Write and execute an anonymous block to test your theory.

D. Modify your procedure to handle the exception in a generic WHEN OTHERS exception handler.

E. Now what do you think would happen if you execute this procedure for department_id 10 (which already exists)? Test it again as in step C.

F. Modify the procedure code to leave out the exception section again. Run the code.

```
CREATE OR REPLACE PROCEDURE add_my_dept
  (p_dept_id IN VARCHAR2, p_dept_name IN VARCHAR2) IS
BEGIN
  INSERT INTO my_depts (department_id,department_name)
    VALUES (p_dept_id, p_dept_name);
END add_my_dept;
```

G. Execute the following code to create a new procedure called outer_proc which calls add_my_dept, passing department_id 10 to it:

```
CREATE OR REPLACE PROCEDURE outer_proc IS
    v_dept      NUMBER(2)          := 10;
    v_dname  VARCHAR2(30)       := 'Admin';
BEGIN
```

```
  add_my_dept(v_dept, v_dname);
EXCEPTION
      WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Exception was propagated to outer_proc');
END;
```

H. Execute outer_proc from an anonymous block. What happens and why?

2. Write and execute a SELECT statement to list the names of all the procedures you have created so far.

3. Delete the last procedure you created: outer_proc.

4. Write and execute a SELECT statement to list the source code of your add_my_dept procedure. Make sure your SELECT statement list the lines of code in the correct order.