

Managing Procedures and Functions

What Will I Learn?

In this lesson, you will learn to:

- Describe how exceptions are propagated
- Remove a function and a procedure
- Use Data Dictionary views to identify and manage stored programs

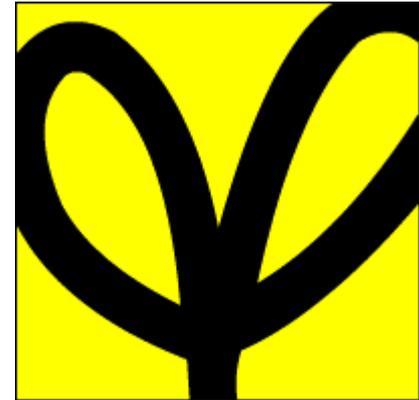




Why Learn It?

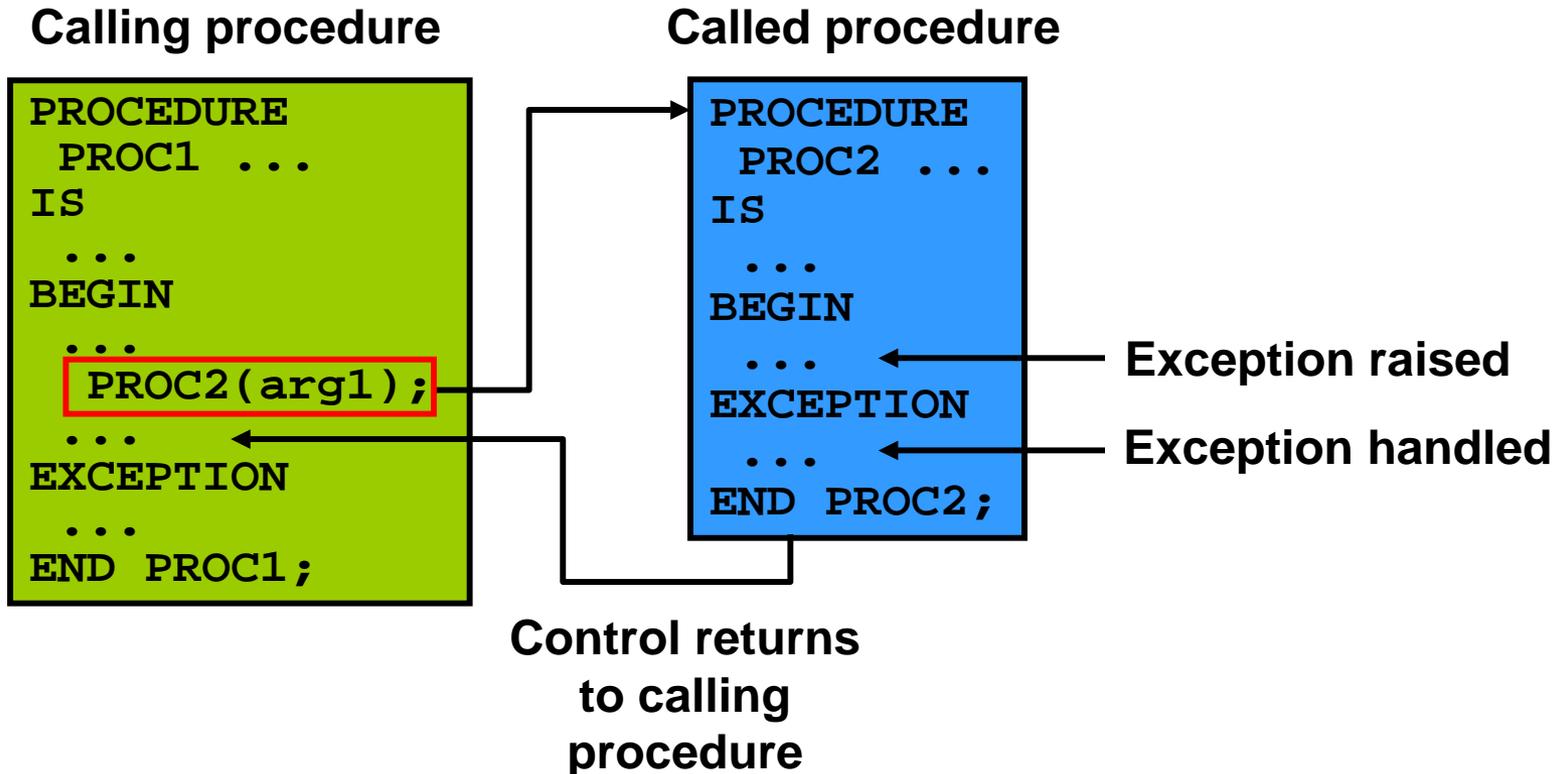
In this lesson, you learn to manage procedures and functions.

To make your programs robust, you should always manage exception conditions by using the exception-handling features of PL/SQL.



Tell Me / Show Me

Handled Exceptions



This and the following slides use procedures as examples, but the same rules apply to functions.



Tell Me / Show Me

Handled Exceptions: Example

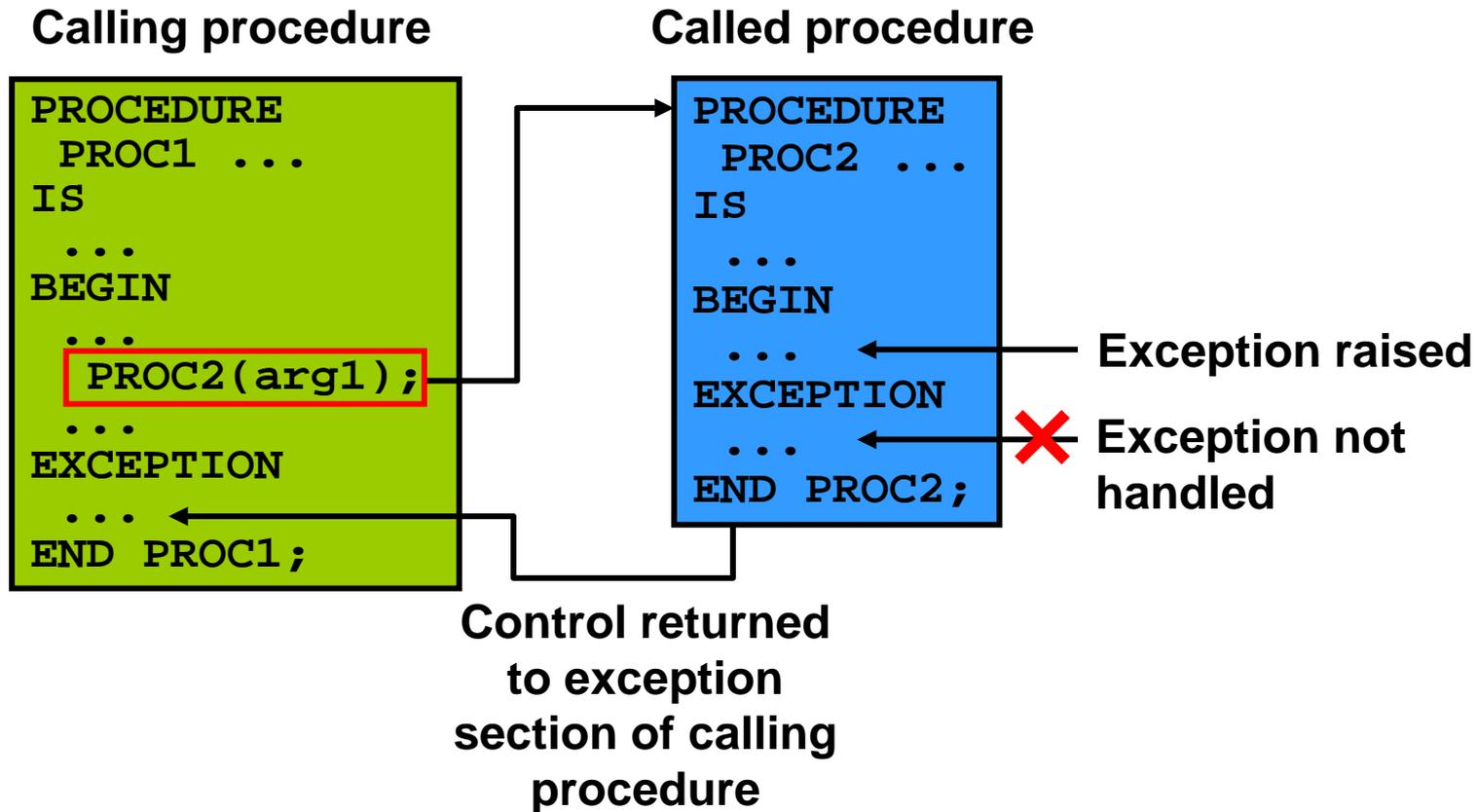
```
CREATE OR REPLACE PROCEDURE add_department(  
    p_name VARCHAR2, p_mgr NUMBER, p_loc NUMBER) IS  
BEGIN  
    INSERT INTO DEPARTMENTS (department_id,  
        department_name, manager_id, location_id)  
VALUES (DEPARTMENTS_SEQ.NEXTVAL, p_name, p_mgr, p_loc);  
    DBMS_OUTPUT.PUT_LINE('Added Dept: ' || p_name);  
EXCEPTION  
    WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('Error adding dept: ' || p_name);  
END;
```

```
BEGIN  
    add_department('Media', 100, 1800);  
    add_department('Editing', 99, 1800);  
    add_department('Advertising', 101, 1800);  
END;
```



Tell Me / Show Me

Exceptions Not Handled





Tell Me / Show Me

Exceptions Not Handled: Example

```
CREATE OR REPLACE PROCEDURE add_department_noex(  
    p_name VARCHAR2, p_mgr NUMBER, p_loc NUMBER) IS  
BEGIN  
    INSERT INTO DEPARTMENTS (department_id,  
        department_name, manager_id, location_id)  
    VALUES (DEPARTMENTS_SEQ.NEXTVAL, p_name, p_mgr, p_loc);  
    DBMS_OUTPUT.PUT_LINE('Added Dept: ' || p_name);  
END;
```

```
BEGIN  
    add_department_noex('Media', 100, 1800);  
    add_department_noex('Editing', 99, 1800);  
    add_department_noex('Advertising', 101, 1800);  
END;
```





Tell Me / Show Me

Removing Procedures and Functions

You can remove a procedure or function that is stored in the database.

- Syntax:

```
DROP {PROCEDURE procedure_name | FUNCTION function_name}
```

- Examples:

```
DROP PROCEDURE raise_salary;
```

```
DROP FUNCTION get_sal;
```

Tell Me / Show Me

Viewing Subprograms in the Data Dictionary

The source code for PL/SQL subprograms is stored in the Data Dictionary tables. The source code is stored in the database even when the PL/SQL subprogram did not compile successfully.

- The `USER_OBJECTS` table contains the names and types of procedures and functions.
- The `USER_SOURCE` table contains source code for all of the subprograms that you own.
- The `ALL_SOURCE` table contains source code for all the subprograms that you have privileges to invoke.

Tell Me / Show Me

Viewing Subprogram Names in the USER_OBJECTS Table

This example lists the names of all the PL/SQL functions that you own:

```
SELECT object_name
FROM   USER_OBJECTS
WHERE  object_type = 'FUNCTION';
```

OBJECT_NAME
TAX
DML_CALL_SQL

Tell Me / Show Me

Viewing PL/SQL Source Code in the USER_SOURCE Table

This example shows the source code of the TAX function, which you own. Make sure you include ORDER BY line to see the lines of code in the correct sequence!

```
SELECT text
FROM   USER_SOURCE
WHERE  type = 'FUNCTION' AND name = 'TAX'
ORDER BY line;
```

TEXT
FUNCTION tax(value IN NUMBER)
RETURN NUMBER IS
BEGIN
RETURN (value * 0.08);
END tax;

Tell Me / Show Me

Viewing Object Names and Source Code in Application Express

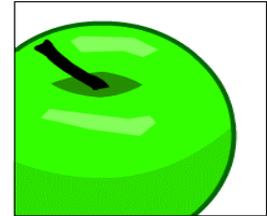
You can easily view subprogram information in Application Express:

- From SQL Workshop, click Object Browser, then Browse, and choose either Procedures or Functions as required. A list of subprograms appears.
- Click the required subprogram name. The source code of the subprogram appears.
- From here, you can edit and recompile it, or drop it if you want.

Tell Me / Show Me

Terminology

Key terms used in this lesson include:



USER_OBJECTS

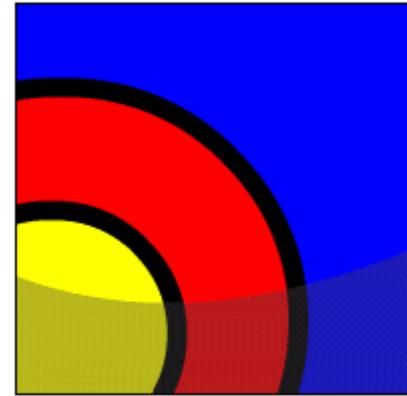
USER_SOURCE

ALL_SOURCE

Summary

In this lesson, you learned to:

- Describe how exceptions are propagated
- Remove a function and a procedure
- Use Data Dictionary views to identify and manage stored programs



Try It / Solve It

The exercises in this lesson cover the following topics:

- Describing how exceptions are propagated
- Removing a function and a procedure
- Using Data Dictionary views to identify and manage procedures and functions

