Database Programming with PL/SQL

9-6: Using Invoker's Rights Practice

Activities

Vocabulary

Identify the vocabulary word for each definition below:

Definer's Rights	Routines are bound to the schema in which they reside.
Invoker's Rights	Routines let you centralize data retrieval whereas multiple users can manage their own data using a single code base.
Autonomous Transaction	Subprograms that allow your session to have two or more active transactions at the same time

Try It / Solve It

The following questions illustrate how definer's and invoker's rights **WOULD** work if our APEX configuration allowed you to access a different schema.

1. **IMAGINE** the following two procedures have been created in an account called IACAD_SCHEMA, which also contains an EMPS table.

```
CREATE OR REPLACE PROCEDURE show_emps_def (p_emp_id IN NUMBER) IS
v_name          emps.name%TYPE;  v_dept_id      emps.department_id%TYPE;
  v_dept_name  emps.department_name%TYPE;  v_sal
emps.salary%TYPE;
BEGIN
  SELECT name, department_id, department_name, salary
INTO v_name, v_dept_id, v_dept_name, v_sal      FROM
emps
    WHERE employee_id = p_emp_id;
  DBMS_OUTPUT.PUT_LINE('The employee details are: ' || v_name
              || ' ' || v_dept_id || '  '|| v_dept_name || ' ' || v_sal);
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No data found for employee id: ' || p_emp_id
|| '. Sorry, please enter another number and try again.'); END;
```

```
CREATE OR REPLACE PROCEDURE show_emps_inv (p_emp_id IN NUMBER)
AUTHID CURRENT_USER IS   v_name
    emps.name%TYPE;   v_dept_id
    emps.department_id%TYPE;
  v_dept_name  emps.department_name%TYPE;   v_sal
emps.salary%TYPE;
BEGIN
  SELECT name, department_id, department_name, salary
INTO v_name, v_dept_id, v_dept_name, v_sal
   FROM emps
   WHERE employee_id = p_emp_id;
  DBMS_OUTPUT.PUT_LINE('The employee details are: ' || v_name || ' '  ||
          v_dept_id || ' ' || v_dept_name || ' ' || v_sal);
EXCEPTION
 WHEN NO_DATA_FOUND THEN
   DBMS_OUTPUT.PUT_LINE('No data found for employee id: ' || p_emp_id
|| '. Sorry, please enter another number and try again.'); END;
```

A. Write the syntax to DESCRIBE both procedures. If the procedures existed, this would allow you to verify that you can see them in your account. Remember to prefix the procedure name with the schema name, and **remember that the schema/procedures don't exist (we are using our imaginations).**

DESCRIBE show_emps_def;

DESCRIBE show_emps_inv;

B. Write the syntax to execute a SQL statement to try to select directly from the table used in the procedures. **Remember that the schema/procedures don't exist (we are using our imaginations).**

SELECT * FROM EMPS;

C. Explain the differences between the two procedures in regards to definer's rights compared to invoker's rights.

Cuando el procedimiento es usado por el definer, no se necesitan permisos.

Cuando el procedimiento es usado por el invoker, se necesita otorgar permisos en las tablas que use el procedimiento.

2. Write the syntax to execute the first procedure (show_emps_def) with the following actual parameter value: employee_id = 100. Using your imagination, explain what would happen and why if the code executed successfully in your schema?

BEGIN

show_emps_def(100);

END;

3. Write the syntax to execute the first procedure again, this time with employee_id = 10. Using your imagination, and assuming there is no employee with employee_id = 10, explain what would happen and why if the code executed successfully in your schema?

BEGIN

show_emps_def(10);

END;

4. Write the syntax to execute the second procedure (show_emps_inv) with employee_id = 100. Using your imagination, explain what would happen and why if you executed this code in your schema?

Entraria en la excepcion de no_data_found.

5. State two changes that must be made if show_emps_inv is to work correctly and display details of an employee.

Asignar permisos de select en la tabla emp.
Asignar permisos execute a el usuario que invocara el procedimiento.