Database Programming with PL/SQL

9-5: Review of Object Privileges Practice Activities

Vocabulary

Identify the vocabulary word for each definition below:

EXECUTE privilege	Allows the grantee to invoke and execute a PL/SQL subprogram.
INDEX privilege	Allows the grantee to create indexes on the table.
Object privilege	Allows the use of a specific database object, such as a table, a view, or a PL/SQL procedure, by one or more database users.
ALTER privilege	Allows the grantee to ALTER the table.
REFERENCES privilege	Allows the grantee to check for the existence of rows in a table using foreign key constraints.

Try It / Solve It

1. If you wanted user SUSAN to be able to execute SELECT and all DML statements on your COUNTRIES table, what SQL statement would you execute to give her the required privileges?

   GRANT SELECT , UPDATE, INSERT ON COUNTRIES TO SUSAN;

2. User TOM creates a table called TOMTAB, but does not grant you any privileges on it.

   A. If you try to execute the following statement, will it work?

INSERT INTO tom.tomtab (…) VALUES (…);

No funciona, porque TOM no ha dado ningun permiso para usar su tabla.

B. Examine the following code. The INSERT statement has been included in a procedure which you have created. Will it work now?

```
CREATE OR REPLACE PROCEDURE my_ins_proc
  IS
BEGIN
  INSERT INTO tom.tomtab (…)
    VALUES (…);
END;
```

No, porque aún no asigna el permiso de INSERT.

C. TOM now executes the following statement:

```
GRANT INSERT ON tomtab TO JORGE;
```

Will your my_ins_proc procedure work now? Why or why not?

No, porque aún no se asignan permisos de ejecución.

D. TOM now REVOKEs your INSERT privilege on tomtab. TOM then writes the following procedure. Which privilege must TOM grant to you to allow you to execute his tom_ins_proc procedure? With this privilege, will the INSERT work when you invoke TOM's procedure?

```
CREATE OR REPLACE PROCEDURE tom_ins_proc
  IS
```

```
BEGIN
  INSERT into tom.tomtab (…)
    VALUES (…);
END;
```

GRANT EXECUTE ON tom_ins_proc TO JORGE;

3. Create a SELECT statement to display all the procedures and functions which you can execute?

SELECT * FROM user_PROCEDURES WHERE (OBJECT_TYPE = 'FUNCTION' OR OBJECT_TYPE = 'PROCEDURE')

4. In this question, you will need to pair up with one other student. You will then grant access to one of your procedures to the other student, and he or she will do the same to you. Each of you will then execute the other student's procedure and see the effects on the other student's table. For the rest of this question, we call one of you Student A and the other Student B.

   a. (Both students): create a table called NEW_DEPT_TAB using the DEPARTMENT table.
   CREATE TABLE NEW_DEPT_TAB AS SELECT * FROM DEPARTMENTS;

   b. (Both students): create a procedure called ins_new_dept which inserts a record into the new_dept_tab table.
   CREATE OR REPLACE PROCEDURE ins_new_dept(
   p_id IN DEPARTMENTS.department_id%TYPE, p_name IN departments.department_name%TYPE)
   AS
   BEGIN
   INSERT INTO NEW_DEPT_TAB (DEPARTMENT_ID, DEPARTMENT_NAME)
   VALUES(p_id, p_name);
   END INS_NEW_DEPT;

   c. (Both students): grant EXECUTE privilege on your ins_new_emp procedure to the other student.

GRANT EXECUTE ON ins_new_dept to MX_A104_SQL_S49;

d. (Both students): check that you can see the definition of the other student's procedure.
DESCRIBE MX_A104_SQL_S49.ins_new_dept;

e. (Student A): insert an employee record using Student B's ins_new_emp procedure.

f. (Student B): insert an employee record using Student A's ins_new_emp procedure.

BEGIN
MX_A104_SQL_S49.ins_new_dept(32,'ITI');
END;

g. (Both students): Query your own new_dept_tab table and check that you can see the new row inserted by the other student.

h. (Both students): remove the right to execute your procedure from the other student.

REVOKE EXECUTE ON ins_new_dept FROM MX_A104_SQL_S49;

i. (Both students): check that you can no longer see the other student's ins_new_emp procedure.