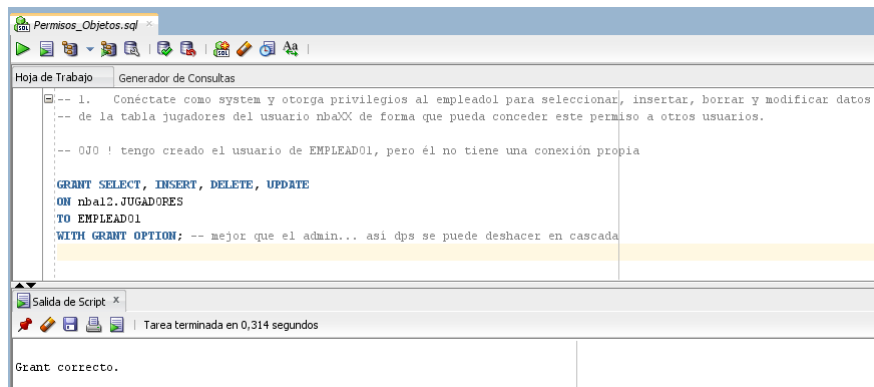


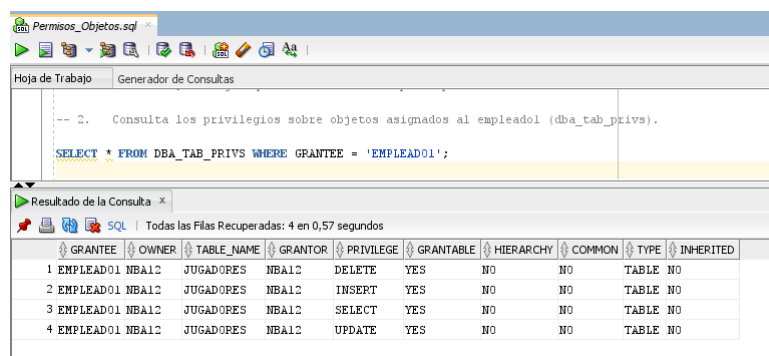
Gestión de permisos (II)

1. Conéctate como **system** y otorga privilegios al **empleado1** para seleccionar, insertar, borrar y modificar datos de la tabla jugadores del usuario **nbaXX** de forma que pueda conceder este permiso a otros usuarios.

GRANT SELECT, INSERT, DELETE, UPDATE
ON nba12.JUGADORES
TO EMPLEADO1
WITH GRANT OPTION;



2. Consulta los privilegios sobre objetos asignados al **empleado1** (dba_tab_privs).
- SELECT * FROM DBA_TAB_PRIVS WHERE GRANTEE = 'EMPLEADO1';**
- SELECT * FROM DBA_TAB_PRIVS WHERE GRANTEE = 'EMPLEADO1';**



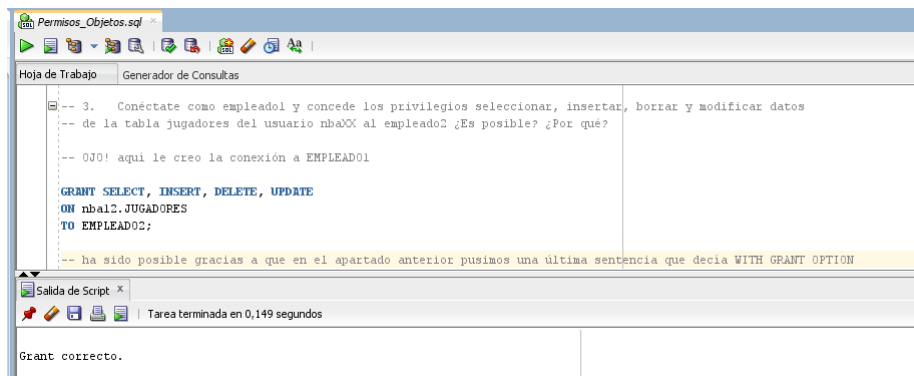
3. Conéctate como **empleado1** y concede los privilegios seleccionar, insertar, borrar y modificar datos de la tabla jugadores del usuario **nbaXX** al **empleado2**
¿Es posible? ¿Por qué?

GRANT SELECT, INSERT, DELETE, UPDATE

ON nba12.JUGADORES

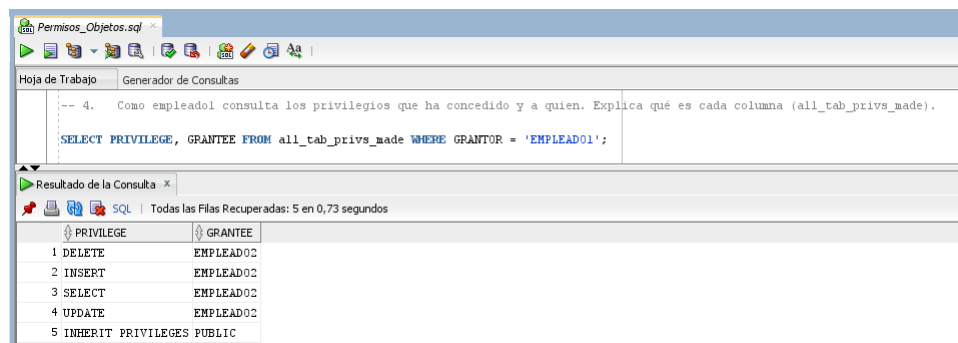
TO EMPLEADO2;

Ha sido posible gracias a que en el apartado anterior pusimos una última sentencia que decía **WITH GRANT OPTION**



4. Como **empleado1** consulta los privilegios que ha concedido y a quien. Explica qué es cada columna (**all_tab_privs_made**).

SELECT PRIVILEGE, GRANTEE **FROM** all_tab_privs_made **WHERE** GRANTOR = 'EMPLEADO1';



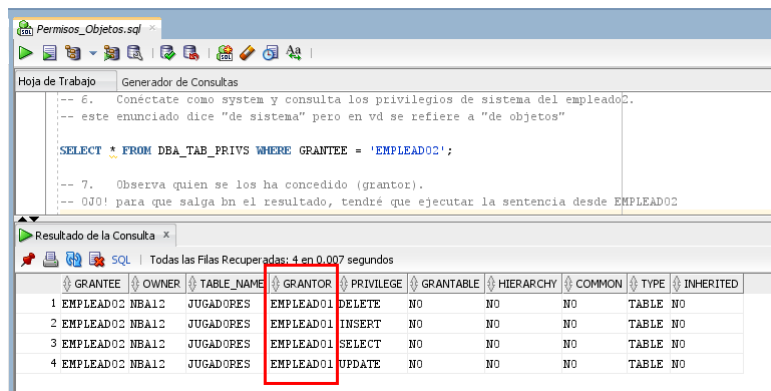
5. Como **empleado1** consulta los privilegios que tiene concedidos (recibidos) y quién se los ha concedido. Explica qué es cada columna ().

SELECT PRIVILEGE, GRANTOR FROM all_tab_privs_recd WHERE GRANTEE = 'EMPLEADO1';

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2 (30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2 (30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2 (30)	NOT NULL	Name of the object
GRANTOR	VARCHAR2 (30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2 (40)	NOT NULL	Privilege on the object
GRANTABLE	VARCHAR2 (3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)
HIERARCHY	VARCHAR2 (3)		Indicates whether the privilege was granted with the HIERARCHY OPTION (YES) or not (NO)

6. Conéctate como **system** y consulta los privilegios de sistema del **empleado2**.

SELECT * FROM DBA_TAB_PRIVS WHERE GRANTEE = 'EMPLEADO2';



Hoja de Trabajo Generador de Consultas

```
-- 6. Conéctate como system y consulta los privilegios de sistema del empleado2.
-- este enunciado dice "de sistema" pero en vd se refiere a "de objetos"

SELECT * FROM DBA_TAB_PRIVS WHERE GRANTEE = 'EMPLEADO2';

-- 7. Observa quien se los ha concedido (grantor).
-- OJO! para que salga bn el resultado, tendré que ejecutar la sentencia desde EMPLEADO2
```

Resultado de la Consulta

Todas las Filas Recuperadas: 4 en 0.007 segundos

	GRANTEE	OWNER	TABLE_NAME	GRANTOR	PRIVILEGE	GRANTABLE	HIERARCHY	COMMON	TYPE	INHERITED
1	EMPLEADO2	NBA12	JUGADORES	EMPLEADO1	DELETE	NO	NO	NO	TABLE	NO
2	EMPLEADO2	NBA12	JUGADORES	EMPLEADO1	INSERT	NO	NO	NO	TABLE	NO
3	EMPLEADO2	NBA12	JUGADORES	EMPLEADO1	SELECT	NO	NO	NO	TABLE	NO
4	EMPLEADO2	NBA12	JUGADORES	EMPLEADO1	UPDATE	NO	NO	NO	TABLE	NO

7. Observa quien se los ha concedido (grantor).

```
SELECT PRIVILEGE, GRANTOR FROM all_tab_privs_recd WHERE GRANTEE = 'EMPLEADO2';
```

8. Consulta los privilegios de sistema que ha concedido el empleado1.

-- este enunciado dice "de sistema" pero en vd se refiere a "de objetos"

-- ejecutar la sentencia desde el usuario "EMPLEADO1"

```
SELECT PRIVILEGE, GRANTEE FROM all_tab_privs_made WHERE GRANTOR = 'EMPLEADO1';
```

Hoja de Trabajo | Generador de Consultas

-- 8. Consulta los privilegios de sistema que ha concedido el empleado1.
 -- este enunciado dice "de sistema" pero en vd se refiere a "de objetos"
 -- ejecutar la sentencia desde el usuario "EMPLEADO1"

```
SELECT PRIVILEGE, GRANTEE FROM all_tab_privs_made WHERE GRANTOR = 'EMPLEADO1';
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 5 en 0,016 segundos

	PRIVILEGE	GRANTEE
1	DELETE	EMPLEADO2
2	INSERT	EMPLEADO2
3	SELECT	EMPLEADO2
4	UPDATE	EMPLEADO2
5	INHERIT PRIVILEGES PUBLIC	

9. Como empleado2 consulta los privilegios que ha concedido y a quien.

```
SELECT PRIVILEGE, GRANTEE FROM all_tab_privs_made WHERE GRANTOR = 'EMPLEADO2';
```

Hoja de Trabajo | Generador de Consultas

-- 9. Como empleado2 consulta los privilegios que ha concedido y a quien.

```
SELECT PRIVILEGE, GRANTEE FROM all_tab_privs_made WHERE GRANTOR = 'EMPLEADO2';
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0,013 segundos

	PRIVILEGE	GRANTEE
1	INHERIT PRIVILEGES PUBLIC	

10. Como **empleado2** consulta los privilegios que tiene concedidos (recibidos) y quien se los ha concedido.

SELECT PRIVILEGE, GRANTOR FROM all_tab_privs_recd WHERE GRANTEE = 'EMPLEADO2';

PRIVILEGE	GRANTOR
1 DELETE	EMPLEADO1
2 INSERT	EMPLEADO1
3 SELECT	EMPLEADO1
4 UPDATE	EMPLEADO1

11. Desde **system** revoca al **empleado1** los privilegios para seleccionar, insertar, borrar y modificar datos de la tabla coches del usuario **taller**. ¿Se le ha revocado el permiso de también al **empleado2**? ¿Por qué?

REVOKE SELECT, INSERT, DELETE, UPDATE

ON nba12.JUGADORES

FROM EMPLEADO1;

```
-- 11. Desde system revoca al empleado1 los privilegios para seleccionar, insertar, borrar y modificar datos
-- de la tabla coches del usuario taller.
-- (no es taller, es el ejercicio de la nba)
-- ¿Se le ha revocado el permiso de también al empleado2? ¿Por qué?

REVOKE SELECT, INSERT, DELETE, UPDATE
ON nba12.JUGADORES
FROM EMPLEADO1;
```

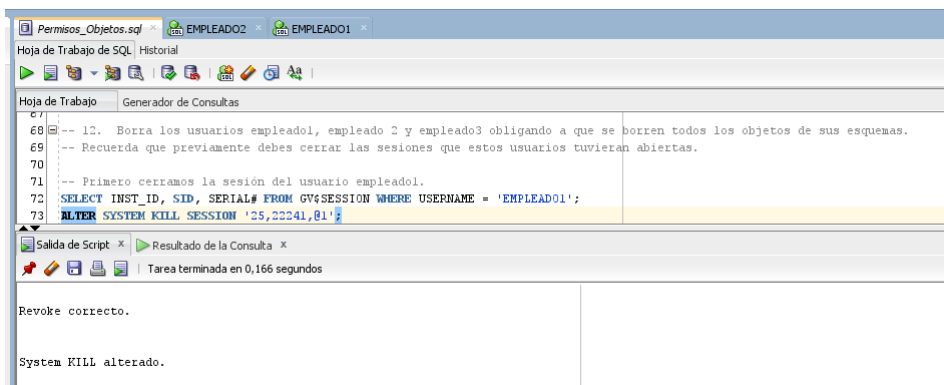
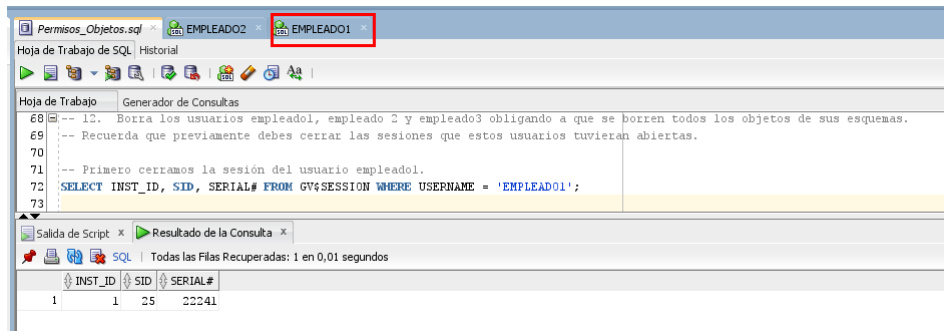
Revoke correcto.

12. Borra los usuarios **empleado1**, **empleado 2** y **empleado3** obligando a que se borren todos los objetos de sus esquemas. Recuerda que previamente debes cerrar las sesiones que estos usuarios tuvieran abiertas.

-- Primero cerramos la sesión del usuario empleado1.

SELECT INST_ID, SID, SERIAL# FROM GV\$SESSION WHERE USERNAME = 'EMPLEADO1';

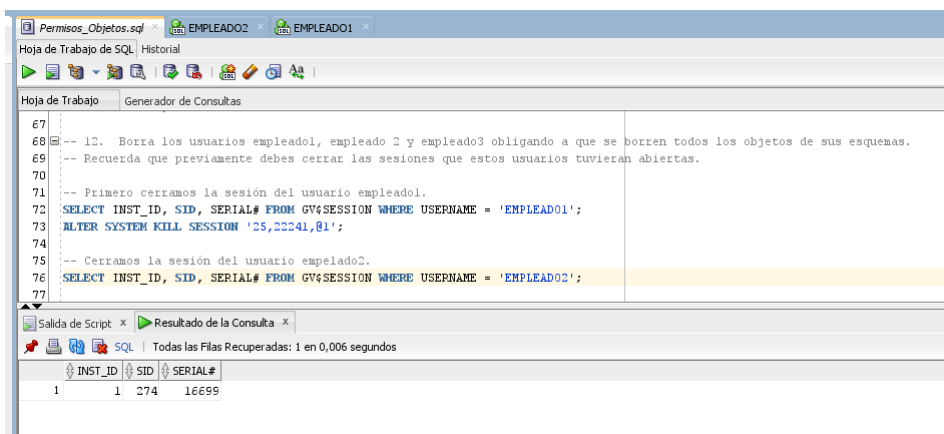
ALTER SYSTEM KILL SESSION '25,22241,@1';



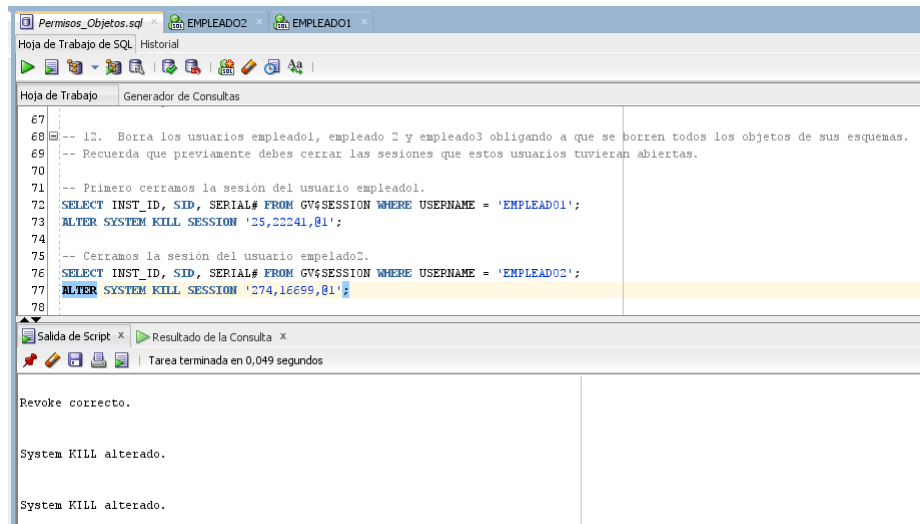
-- Cerramos la sesión del usuario empleado2.

SELECT INST_ID, SID, SERIAL# FROM GV\$SESSION WHERE USERNAME = 'EMPLEADO2';

ALTER SYSTEM KILL SESSION '274,16699,@1';



1ºDAM – Bases de datos



The screenshot shows the SQL Developer interface with a script named 'Permisos_Objetos.sql'. The script contains the following SQL commands:

```
67
68 -- 12. Borra los usuarios empleado1, empleado 2 y empleado3 obligando a que se borren todos los objetos de sus esquemas.
69 -- Recuerda que previamente debes cerrar las sesiones que estos usuarios tuvieran abiertas.
70
71 -- Primero cerramos la sesión del usuario empleado1.
72 SELECT INST_ID, SID, SERIAL# FROM GV$SESSION WHERE USERNAME = 'EMPLEADO1';
73 ALTER SYSTEM KILL SESSION '25,22241,81';
74
75 -- Cerramos la sesión del usuario empleado2.
76 SELECT INST_ID, SID, SERIAL# FROM GV$SESSION WHERE USERNAME = 'EMPLEADO2';
77 ALTER SYSTEM KILL SESSION '274,16699,81';
78
```

The output pane shows the following messages:

```
Revoke correcto.

System KILL alterado.

System KILL alterado.
```

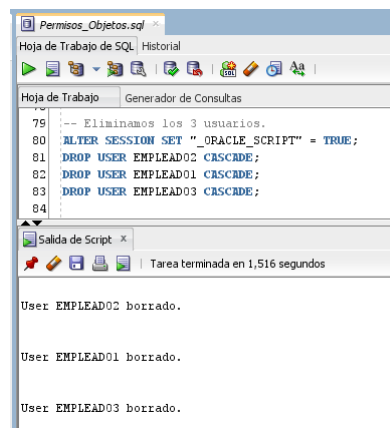
-- Eliminamos los 3 usuarios.

ALTER SESSION SET "_ORACLE_SCRIPT" = TRUE;

DROP USER EMPLEADO2 CASCADE;

DROP USER EMPLEADO1 CASCADE;

DROP USER EMPLEADO3 CASCADE;



The screenshot shows the SQL Developer interface with a script named 'Permisos_Objetos.sql'. The script contains the following SQL commands:

```
79 -- Eliminamos los 3 usuarios.
80 ALTER SESSION SET "_ORACLE_SCRIPT" = TRUE;
81 DROP USER EMPLEADO2 CASCADE;
82 DROP USER EMPLEADO1 CASCADE;
83 DROP USER EMPLEADO3 CASCADE;
84
```

The output pane shows the following messages:

```
User EMPLEADO2 borrado.

User EMPLEADO1 borrado.

User EMPLEADO3 borrado.
```