4 **Explain what's the purpose of the PUBLIC role, which privileges has it and which users have this role? Explain the difference between these views: DBA\_SYS\_PRIVS, DBA\_TAB\_PRIVS and DBA\_ROLE\_PRIVS; When we talk about privileges in Oracle, we find three main categories of privileges, which are: SYSTEM privileges, OBJECT privileges and Privilege hierarchy, define what is the purpose of each category and provide some examples of privileges which belong to each one of them.**

Tanto SYS como SYSTEM son usuarios predeterminados, creados con la creación de la base de datos. Aunque tienen mucho poder, ya que se les otorga el rol de DBA, siguen siendo usuarios comunes. Debido a que SYS posee el diccionario de datos, se lo considera un poco más especial que SYSTEM.

Pero SYS tiene el privilegio SYSDBA que SYSTEM no tiene. Esto hace posible que SYS se convierta en un usuario muy poderoso. Este es el caso cuando se conecta como sys / contraseña como SYSDBA o / como sysdba . La frase as sysdba es una solicitud para adquirir los privilegios asociados con los privilegios del sistema SYSDBA.

La diferencia se vuelve clara si intenta cerrar la base de datos como SYS ordinario: como resultado, obtiene privilegios insuficientes. Sin embargo, si está conectado como SYSDBA, es posible.

Tenga en cuenta que SYSDBA no es un rol, es un privilegio. Lo encontrará en system\_privilege\_map , no en dba\_roles .

En cualquier momento, alguien se conecta como SYSDBA, resulta que está siendo SYS. Es decir, si SYSDBA se otorga a JOHN y John se conecta como SYSDBA y selecciona un usuario de dual, revela que en realidad es SYS.

SYS también es especial porque no es posible crear un activador en el esquema sys. Además, un disparador de inicio de sesión no se ejecuta cuando sys se conecta a la base de datos.

**SYS**

SYS es el propietario de la base de datos y el propietario del diccionario de datos. Nunca cree objetos en el esquema SYS. Los objetos que pertenecen a SYS no se pueden exportar.

**SISTEMA**

SYSTEM es un usuario de administración privilegiado, y normalmente posee tablas proporcionadas por Oracle que no son el diccionario. No cree sus propios objetos en SYSTEM.

**PÚBLICO**

El script sql.bsq, que se ejecuta cuando se crea una base de datos , crea la función pública: crear rol público

Sin embargo, este rol no es visible en dba\_roles porque está oculto en esta vista ( where ... name not in ('PUBLIC', '\_NEXT\_USER').

Cualquier privilegio otorgado al público automáticamente se convierte en un privilegio para otros usuarios también.

**Notas**

Las siguientes notas se aplican a los sinónimos públicos:

Si crea un sinónimo público y posteriormente tiene tablas dependientes o tipos de objeto definidos por el usuario válidos dependientes, entonces no puede crear otro objeto de base de datos con el mismo nombre que el sinónimo en el mismo esquema que los objetos dependientes.

Tenga cuidado de no crear un sinónimo público con el mismo nombre que un esquema existente. Si lo hace, todas las unidades PL / SQL que usen ese nombre serán invalidadas.

**ESQUEMA**

Especifique el esquema para contener el sinónimo. Si omite schema, Oracle Database crea el sinónimo en su propio esquema. No puede especificar un esquema para el sinónimo si lo ha especificado PUBLIC.

**INTERNO**

INTERNAL es un usuario especial obsoleto (a partir de 8i) al que se le permite acceder a la base de datos incluso cuando la base de datos está en estado NOMOUNT o MOUNT. Este usuario generalmente se usa para el mantenimiento físico de la base de datos. El usuario interno no se mantiene en el datadictionary sino en el archivo de contraseña de Oracle. El mecanismo interno ha sido reemplazado por el privilegio SYSDBA y SYSOPER en Oracle 8 y más allá.

**USUARIOS PRIVS Y ROLES**

La Seguridad es una de las tareas importantes que tiene un DBA y esta se realiza siguiendo una serie de estándares que incluyen estándares legales y de negocio.

La seguridad de los datos es uno de los aspectos que todo negocio debe de reconocer como algo importante; además, deberá desarrollar políticas acordes que les permitan proteger los datos de forma adecuada. Una de las formas de proteger los datos es por medio de privilegios que se conceden a los usuarios y que en el momento que el usuario realiza una conexión con la base de datos, estos privilegios concedidos se van a habilitar para permitirle realizar aquellas acciones que el usuario requiera.

**La base de datos tiene dos tipos de privilegios:**

**De sistema:** estos privilegios le permiten al usuario realizar acciones específicas sobre la base de datos.

**De objetos:** estos privilegios le permiten al usuario acceder y manipular objetos específicos.

Una de las preocupaciones de muchos administradores de base de datos es la posibilidad de que usuarios de aplicación y de base de datos tengan privilegios excesivos; más aún, que algunos de ellos no se estén utilizando; pero que en algún momento podrían ser aprovechados para ejecutar sentencias SQL que puedan dañar la base de datos o, por el contrario, extraer información para usos no deseados.

**LISTAS DE CONTROL DE ACCESO**

DBA\_NETWORK\_ACLS: Muestra la lista de control de acceso a las asignaciones de los hosts de la red. El privilegio SELECT en esta vista se otorga únicamente a la función SELECT\_CATALOG\_ROLE.

DBA\_NETWORK\_ACL\_PRIVILEGES: muestra los privilegios de red definidos en todas las listas de control de acceso que están asignadas actualmente a los hosts de la red. El privilegio SELECT en esta vista se otorga únicamente a la función SELECT\_CATALOG\_ROLE.

USER\_NETWORK\_ACL\_PRIVILEGES: muestra el estado de los privilegios de red para que el usuario actual acceda a los hosts de la red. El privilegio SELECT en la vista se otorga a PUBLIC.

**PRIVILEGIOS Y ROLES DEL USUARIO**

**ALL\_COL\_PRIVS:** describe todas las concesiones de objetos de columna para las cuales el usuario actual o PUBLIC es el propietario, otorgante o concesionario del objeto

**ALL\_COL\_PRIVS\_MADE :** enumera las concesiones de objetos de columna para las cuales el usuario actual es propietario o otorgante de objetos.

**ALL\_COL\_PRIVS\_RECD :** describe las concesiones de objetos de columna para las cuales el usuario actual o PUBLIC es el concesionario

**ALL\_TAB\_PRIVS :** enumera las concesiones en objetos donde el usuario o PUBLIC es el concesionario

**ALL\_TAB\_PRIVS\_MADE :** enumera todas las concesiones de objetos realizadas por el usuario actual o realizadas en los objetos propiedad de El usuario actual.

**ALL\_TAB\_PRIVS\_RECD :** enumera las concesiones de objetos para las cuales el usuario o PUBLIC es el concesionario

**DBA\_COL\_PRIVS :** describe todas las concesiones de objetos de columna en la base de datos

**DBA\_TAB\_PRIVS :** enumera todas las concesiones de todos los objetos en la base de datos

**DBA\_ROLES :** esta vista enumera todos los roles que existen en la base de datos, incluido el seguro Roles de aplicación. Tenga en cuenta que no incluye la función PUBLIC.

**DBA\_ROLE\_PRIVS :** Muestra los papeles otorgados a los usuarios y roles

**DBA\_SYS\_PRIVS :** privilegios del sistema Listas otorgados a los usuarios y roles

**SESSION\_ROLES :** enumera todas las funciones que están habilitadas para el usuario actual. Tenga en cuenta que no incluye la función PUBLIC.

**ROLE\_ROLE\_PRIVS:** Esta vista describe los roles otorgados a otros roles. La información se proporciona solo sobre los roles a los que el usuario tiene acceso.

**ROLE\_SYS\_PRIVS :** esta vista contiene información sobre los privilegios del sistema otorgados a los roles. La información se proporciona solo sobre los roles a los que el usuario tiene acceso.

**ROLE\_TAB\_PRIVS :** esta vista contiene información sobre los privilegios de objeto otorgados a los roles. La información se proporciona solo sobre los roles a los que el usuario tiene acceso.

**USER\_COL\_PRIVS :** describe las concesiones de objetos de columna para las cuales el usuario actual es el propietario del objeto, el otorgante o el concesionario

**USER\_COL\_PRIVS\_MADE :** describe las concesiones de objetos de columna para las cuales el usuario actual es el otorgante

**USER\_COL\_PRIVS\_RECD:** Describe las concesiones de objetos de columna para las cuales el usuario actual es el concesionario

**USER\_ROLE\_PRIVS :** enumera los roles otorgados al usuario actual

**USER\_TAB\_PRIVS**: enumera las concesiones en todos los objetos donde el usuario actual es el concesionario

**USER\_SYS\_PRIVS :** enumera los privilegios del sistema otorgados al usuario actual

**USER\_TAB\_PRIVS\_MADE:** enumera las concesiones en todos los objetos propiedad del usuario actual

**USER\_TAB\_PRIVS\_RECD:** listas de objetos subvenciones para el cual el usuario actual es el concesionario

**SESSION\_PRIVS:** Lista los privilegios que actualmente están habilitados para los usuarios

**SESSION\_ROLES:** Se enumeran las funciones que actualmente están habilitados para el usuario.

**EJEMPLO # 1 – PRIVILEGIOS ASIGNADOS DIRECTAMENTE**

**CREACIÓN DE USUARIOS**

Conectado con un usuario con privilegios de DBA, procedemos a crear dos usuarios que tendrán diferentes privilegios de sistema.

DROP USER test1 CASCADE;

CREATE USER test1 IDENTIFIED BY "oracle\_12C"

DEFAULT TABLESPACE USERS

TEMPORARY TABLESPACE TEMP;

GRANT CREATE SESSION,

CREATE TABLE,

CREATE VIEW,

CREATE PROCEDURE,

CREATE SEQUENCE,

CREATE TABLESPACE,

CREATE SYNONYM,

UNLIMITED TABLESPACE TO test1;

DROP USER test2 CASCADE;

CREATE USER test2 IDENTIFIED BY "oracle\_12C"

DEFAULT TABLESPACE USERS

TEMPORARY TABLESPACE TEMP;

GRANT CREATE SESSION,

CREATE TABLE,

CREATE VIEW,

CREATE PROCEDURE,

CREATE SEQUENCE,

CREATE ANY DIRECTORY,

CREATE USER,

CREATE SYNONYM,

UNLIMITED TABLESPACE TO test2;

**CREAR LA POLÍTICA DE ANÁLISIS**

Utilizando el paquete DBMS\_PRIVILEGE\_CAPTURE, procedemos a crear una política de análisis que nos permita analizar todos los privilegios a nivel de la base de datos.

BEGIN

SYS.DBMS\_PRIVILEGE\_CAPTURE.CREATE\_CAPTURE(

name => 'Analisis\_BD',

description => 'Captura todos los privilegios a nivel de BD',

type => dbms\_privilege\_capture.g\_database);

END;

/

**HABILITAR LA POLÍTICA DE ANÁLISIS**

Utilizando el paquete DBMS\_PRIVILEGE\_CAPTURE, procedemos a habilitar la política de análisis. Es importante indicar que al crear la política esta NO queda habilitada de forma inmediata.

BEGIN

SYS.DBMS\_PRIVILEGE\_CAPTURE.ENABLE\_CAPTURE(name => 'Analisis\_BD');

END;

/

**USO DE PRIVILEGIOS CON USUARIO TEST1**

Utilizando SQL Developer, creamos una conexión con el usuario TEST1 y procedemos con la ejecución de algunas instrucciones SQL que están relacionadas con los privilegios concedidos a este usuario:

* Creación de la tabla TABLA1.
* Creación de la secuencia SECUENCIA1;
* Creación de la vista VISTA1;
* Se conceden privilegios de SELECT y UPDATE sobre la tabla al usuario TEST2.
* Se concede privilegio de SELECT sobre la vista al usuario TEST2.

CREATE TABLE tabla1

(campo1      VARCHAR2(10) PRIMARY KEY,

campo2       DATE,

campo3       NUMBER);

CREATE SEQUENCE secuencia1

START WITH 1;

INSERT INTO tabla1 VALUES ('001',SYSDATE,1);

INSERT INTO tabla1 VALUES ('002',SYSDATE,2);

INSERT INTO tabla1 VALUES ('003',SYSDATE,3);

INSERT INTO tabla1 VALUES ('004',SYSDATE,4);

CREATE VIEW vista1 AS SELECT campo1, campo2 FROM tabla1;

GRANT SELECT, UPDATE ON tabla1 TO test2;

GRANT SELECT ON vista1 TO test2;

**Uso de privilegios con usuario TEST2**

Utilizando SQL Developer, creamos una conexión con el usuario TEST2 y procedemos con la ejecución de algunas instrucciones SQL que están relacionadas con los privilegios concedidos a este usuario:

* Creación sinónimos para la tabla y la vista.
* Ejecución de SELECT sobre la tabla y sobre la vista de forma directa; y luego por medio de los sinónimos.
* Creación de usuario TEST3.

CREATE SYNONYM test1\_tabla1 FOR test1.tabla1;

CREATE SYNONYM test1\_vista1 FOR test1.vista1;

SELECT \* FROM test1.tabla1;

CAMPO1     CAMPO2                  CAMPO3

---------- ------------------- ----------

001        12/11/2016:17:56:28          1

002        12/11/2016:17:56:34          2

003        12/11/2016:17:56:40          3

004        12/11/2016:17:56:44          4

SELECT \* FROM test1.vista1;

CAMPO1     CAMPO2

---------- -------------------

001        12/11/2016:17:56:28

002        12/11/2016:17:56:34

003        12/11/2016:17:56:40

004        12/11/2016:17:56:44

SELECT \* FROM test1\_tabla1;

CAMPO1     CAMPO2                  CAMPO3

---------- ------------------- ----------

001        12/11/2016:17:56:28          1

002        12/11/2016:17:56:34          2

003        12/11/2016:17:56:40          3

004        12/11/2016:17:56:44          4

SELECT \* FROM test1\_vista1;

CAMPO1     CAMPO2

---------- -------------------

001        12/11/2016:17:56:28

002        12/11/2016:17:56:34

003        12/11/2016:17:56:40

004        12/11/2016:17:56:44

CREATE USER test3 IDENTIFIED BY "oracle\_12C"

DEFAULT TABLESPACE USERS

TEMPORARY TABLESPACE TEMP;

DROP USER test3 CASCADE;

Error starting at line : 17 in command -

DROP USER test3 CASCADE

Error report -

ORA-01031: insufficient privileges

01031. 00000 -  "insufficient privileges"

\*Cause:   An attempt was made to perform a database operation without

the necessary privileges.

\*Action:  Ask your database administrator or designated security

administrator to grant you the necessary privileges

**Inhabilitar la Política de Análisis**

Utilizando el paquete DBMS\_PRIVILEGE\_CAPTURE, procedemos a inhabilitar la política de análisis.

BEGIN

SYS.DBMS\_PRIVILEGE\_CAPTURE.DISABLE\_CAPTURE(name => 'Analisis\_BD');

END;

/

**Generar resultados del análisis**

Utilizando el paquete DBMS\_PRIVILEGE\_CAPTURE, procedemos a generar los resultados del análisis. Al ejecutar este procedimiento se llenan vistas del diccionario de datos, tales como:

* DBA\_USED\_SYSPRIVS
* DBA\_USED\_OBJPRIVS
* DBA\_USED\_SYSPRIVS\_PATH
* DBA\_USED\_OBJPRIVS\_PATH
* DBA\_UNUSED\_PRIVS

BEGIN

SYS.DBMS\_PRIVILEGE\_CAPTURE.GENERATE\_RESULT(name => 'Analisis\_BD');

END;

/

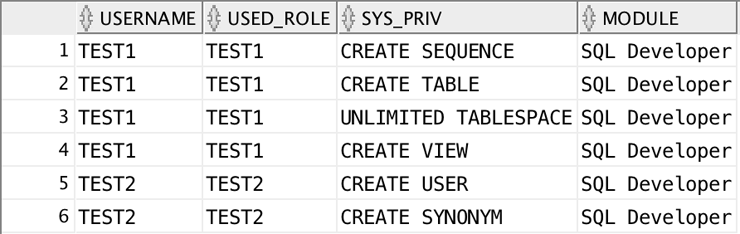
**Privilegios de Sistema Utilizados**

Por medio de una consulta a la vista DBA\_USED\_SYSPRIVS nos damos cuenta cuales privilegios de sistema han sido utilizados por los usuarios TEST1 y TEST2 durante el periodo de análisis.

SELECT username, used\_role, sys\_priv, module

FROM   dba\_used\_sysprivs

ORDER BY username;



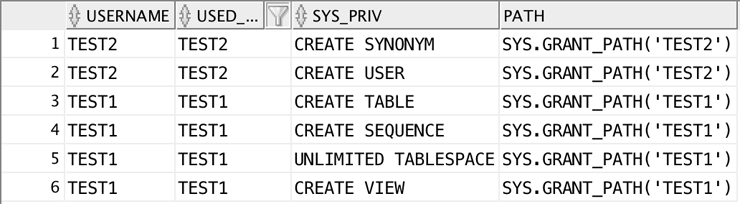
La consulta nos muestra que el usuario TEST1 utilizó los privilegios de sistema CREATE SEQUENCE, CREATE TABLE, UNLIMITED TABLESPACE (en el momento que realiza las inserciones de los registros) y CREATE VIEW. En el caso del usuario TEST2 nos muestra que utilizó los privilegios de sistema CREATE USER y CREATE SYNONYM.

La vista DBA\_USED\_SYSPRIVS\_PATH permite determinar el medio por el cual se utilizó el privilegio de sistema. La columna PATH en dicha vista nos dice si el privilegio se utilizó de forma directa o por medio de un rol.

SELECT username, used\_role, sys\_priv, path

FROM   DBA\_USED\_SYSPRIVS\_PATH

WHERE  username IN ('TEST1','TEST2');



La consulta nos muestra que tanto para el usuario TEST1 como para el usuario TEST2, los privilegios utilizados, todos ellos, fueron utilizados porque ellos fueron asignados de forma directa a los usuarios.

**Privilegios de Objetos Utilizados**

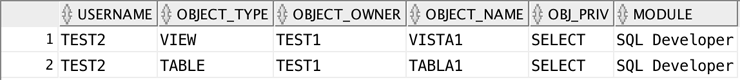
La vista DBA\_USED\_OBJPRIVS permite determinar cuáles privilegios sobre objetos fueron utilizados por los usuarios durante el periodo de análisis.

SELECT username, object\_type, object\_owner, object\_name, obj\_priv, module

FROM   DBA\_USED\_OBJPRIVS

WHERE  username = 'TEST2'

AND  object\_owner NOT IN ('SYS','SYSTEM');



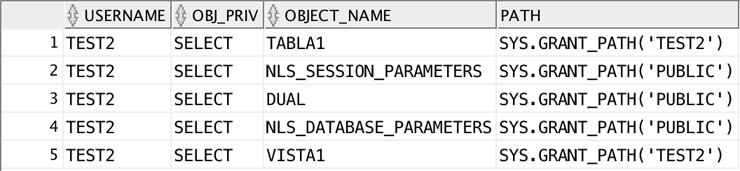
La consulta nos muestra que el usuario TEST2 utilizó el privilegio de objetos SELECT para acceder a la tabla y a la vista.

La vista DBA\_USED\_OBJPRIVS\_PATH permite determinar el medio por el cual se utilizó el privilegio de objeto. La columna PATH en dicha vista nos dice si el privilegio se utilizó de forma directa o por medio de un rol.

SELECT username, obj\_priv, object\_name, path

FROM   DBA\_USED\_OBJPRIVS\_PATH

WHERE  username IN ('TEST1','TEST2');



La consulta nos muestra que el usuario TEST2 utilizó el privilegio de SELECT sobre la vista y la tabla por medio de la concesión directa.

**Privilegios NO Utilizados**

Por medio de la vista DBA\_UNUSED\_PRIVS se pueden ver todos los privilegios de sistema y de objetos no utilizados por los usuarios. A partir de la comparación entre los privilegios utilizados y los no utilizados el DBA puede identificar cuales privilegios puede revocar, en el caso de que lo considere necesario.

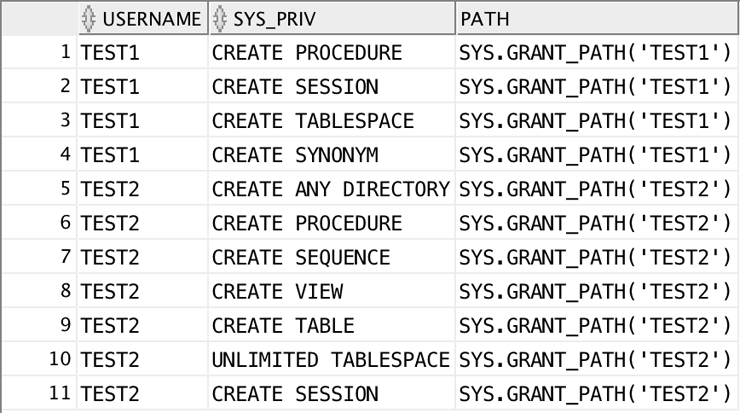
SELECT username, sys\_priv, path

FROM   DBA\_UNUSED\_PRIVS

WHERE  username IN ('TEST1','TEST2')

AND  sys\_priv IS NOT NULL

ORDER BY username;



La consulta nos muestra que el usuario TEST1 no utilizó alguno de los privilegios que le fueron concedidos; mientras que el usuario TEST2 no utilizó la mayoría de los privilegios que le fueron concedidos, de hecho, solamente dos de los privilegios de sistema fueron utilizados por este último usuario.

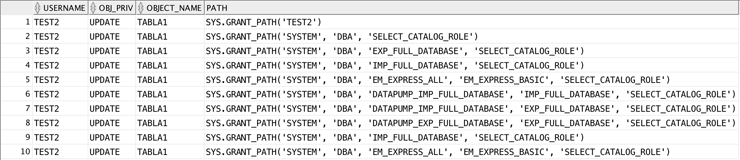
SELECT username, obj\_priv, object\_name, path

FROM   DBA\_UNUSED\_PRIVS

WHERE  username IN ('TEST1','TEST2')

AND  obj\_priv IS NOT NULL

ORDER BY username;



La consulta nos muestra los privilegios de objeto que el usuario TEST2 no utilizó. Como se observó en su momento, al usuario TEST2 se le concedió el privilegio de SELECT y UPDATE sobre la tabla; y el privilegio de SELECT sobre la vista.

**Eliminar la Política de Análisis**

Utilizando el paquete DBMS\_PRIVILEGE\_CAPTURE, procedemos a eliminar la política de análisis. Al eliminar la captura se elimina toda la información relacionada con la misma; es por esa razón que se debe tener cuidado al eliminar una captura.

BEGIN

SYS.DBMS\_PRIVILEGE\_CAPTURE.DROP\_CAPTURE(name => 'Analisis\_BD');

END;

/

**EJEMPLO # 2 – PRIVILEGIOS ASIGNADOS POR ROLES**

**CREACIÓN DE USUARIOS**

Conectado con un usuario con privilegios de DBA, procedemos a realizar lo siguiente:

* Crear dos roles: TEST\_ROLE1 y TEST\_ROLE2.
* Eliminar los usuarios TEST1 y TEST2.
* Crear los usuarios TEST1 y TEST2.
* Conceder privilegios de sistema al rol TEST\_ROLE1 y TEST\_ROLE2
* Conceder el rol TEST\_ROLE1 al usuario TEST1; y el rol TEST\_ROLE2 al usuario TEST2.

CREATE ROLE test\_role1;

CREATE ROLE test\_role2;

DROP USER test1 CASCADE;

DROP USER test2 CASCADE;

CREATE USER test1 IDENTIFIED BY "oracle\_12C"

DEFAULT TABLESPACE USERS

TEMPORARY TABLESPACE TEMP;

GRANT CREATE SESSION,

CREATE TABLE,

CREATE VIEW,

CREATE PROCEDURE,

CREATE SEQUENCE,

CREATE TABLESPACE,

CREATE SYNONYM TO test\_role1;

GRANT test\_role1, UNLIMITED TABLESPACE TO test1;

CREATE USER test2 IDENTIFIED BY "oracle\_12C"

DEFAULT TABLESPACE USERS

TEMPORARY TABLESPACE TEMP;

GRANT CREATE SESSION,

CREATE TABLE,

CREATE VIEW,

CREATE PROCEDURE,

CREATE SEQUENCE,

CREATE ANY DIRECTORY,

CREATE USER,

CREATE SYNONYM TO test\_role2;

GRANT test\_role2 TO test2;

**CREAR LA POLÍTICA DE ANÁLISIS**

Utilizando el paquete DBMS\_PRIVILEGE\_CAPTURE, procedemos a crear una política de análisis que nos permita analizar todos los privilegios que se han concedido por medio de los roles. El argumento TYPE se fija en “dbms\_privilege\_capture.g\_role” y el argumento “roles” indica la lista de roles que se desean analizar.

BEGIN

SYS.DBMS\_PRIVILEGE\_CAPTURE.CREATE\_CAPTURE(

name => 'Analisis\_roles',

description => 'Captura los privilegios relacionados con roles: TEST\_ROLE1, TEST\_ROLE2',

type => dbms\_privilege\_capture.g\_role,

roles => role\_name\_list('TEST\_ROLE1','TEST\_ROLE2'));

END;

/

**HABILITAR LA POLÍTICA DE ANÁLISIS**

Utilizando el paquete DBMS\_PRIVILEGE\_CAPTURE, procedemos a habilitar la política de análisis. Es importante indicar que al crear la política esta NO queda habilitada de forma inmediata.

BEGIN

SYS.DBMS\_PRIVILEGE\_CAPTURE.ENABLE\_CAPTURE(name => 'Analisis\_roles');

END;

/

**NOTA:** es importante indicar que no pueden habilitarse dos capturas que correspondan al mismo tipo.

**USO DE PRIVILEGIOS CON USUARIO TEST1**

Utilizando SQL Developer, creamos una conexión con el usuario TEST1 y procedemos con la ejecución de algunas instrucciones SQL que están relacionadas con los privilegios concedidos a este usuario:

* Creación de la tabla TABLA1.
* Creación de la secuencia SECUENCIA1;
* Creación de la vista VISTA1;
* Se conceden privilegios de SELECT y UPDATE sobre la tabla al usuario TEST2.
* Se concede privilegio de SELECT sobre la vista al usuario TEST2.

CREATE TABLE tabla1

( campo1       VARCHAR2(10) PRIMARY KEY,

campo2       DATE,

campo3       NUMBER );

CREATE SEQUENCE secuencia1

START WITH 1;

INSERT INTO tabla1 VALUES ('001',SYSDATE,1);

INSERT INTO tabla1 VALUES ('002',SYSDATE,2);

INSERT INTO tabla1 VALUES ('003',SYSDATE,3);

INSERT INTO tabla1 VALUES ('004',SYSDATE,4);

CREATE VIEW vista1 AS SELECT campo1, campo2 FROM tabla1;

GRANT SELECT, UPDATE ON tabla1 TO test2;

GRANT SELECT ON vista1 TO test2;

**USO DE PRIVILEGIOS CON USUARIO TEST2**

Utilizando SQL Developer, creamos una conexión con el usuario TEST2 y procedemos con la ejecución de algunas instrucciones SQL que están relacionadas con los privilegios concedidos a este usuario:

* Creación sinónimos para la tabla y la vista.
* Ejecución de SELECT sobre la tabla y sobre la vista de forma directa; y luego por medio de los sinónimos.
* Creación de usuario TEST3.

CREATE SYNONYM test1\_tabla1 FOR test1.tabla1;

CREATE SYNONYM test1\_vista1 FOR test1.vista1;

SELECT \* FROM test1.tabla1;

CAMPO1     CAMPO2                  CAMPO3

---------- ------------------- ----------

001        12/11/2016:17:56:28          1

002        12/11/2016:17:56:34          2

003        12/11/2016:17:56:40          3

004        12/11/2016:17:56:44          4

SELECT \* FROM test1.vista1;

CAMPO1     CAMPO2

---------- -------------------

001        12/11/2016:17:56:28

002        12/11/2016:17:56:34

003        12/11/2016:17:56:40

004        12/11/2016:17:56:44

SELECT \* FROM test1\_tabla1;

CAMPO1     CAMPO2                  CAMPO3

---------- ------------------- ----------

001        12/11/2016:17:56:28          1

002        12/11/2016:17:56:34          2

003        12/11/2016:17:56:40          3

004        12/11/2016:17:56:44          4

SELECT \* FROM test1\_vista1;

CAMPO1     CAMPO2

---------- -------------------

001        12/11/2016:17:56:28

002        12/11/2016:17:56:34

003        12/11/2016:17:56:40

004        12/11/2016:17:56:44

CREATE USER test3 IDENTIFIED BY "oracle\_12C"

DEFAULT TABLESPACE USERS

TEMPORARY TABLESPACE TEMP;

**INHABILITAR Y GENERAR RESULTADOS DE ANÁLISIS**

Utilizando el paquete DBMS\_PRIVILEGE\_CAPTURE, procedemos a inhabilitar la política de análisis.

BEGIN

SYS.DBMS\_PRIVILEGE\_CAPTURE.DISABLE\_CAPTURE(name => 'Analisis\_roles');

SYS.DBMS\_PRIVILEGE\_CAPTURE.GENERATE\_RESULT(name => 'Analisis\_roles');

END;

/

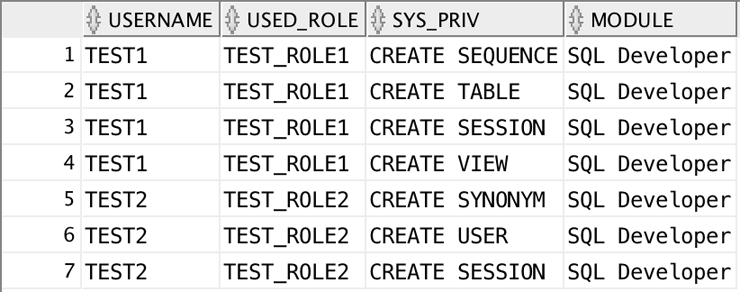
**PRIVILEGIOS DE SISTEMA UTILIZADOS**

Por medio de una consulta a la vista DBA\_USED\_SYSPRIVS nos damos cuenta cuales privilegios de sistema han sido utilizados por los usuarios TEST1 y TEST2 durante el periodo de análisis.

SELECT username, used\_role, sys\_priv, module

FROM   dba\_used\_sysprivs

ORDER BY username;



La consulta nos muestra que el usuario TEST1 utilizó los privilegios de sistema CREATE SEQUENCE, CREATE TABLE y CREATE VIEW. En el caso del usuario TEST2 nos muestra que utilizó los privilegios de sistema CREATE SESSION, CREATE USER y CREATE SYNONYM. La columna USED\_ROLE nos muestra el rol utilizado por el usuario, que es por medio del cual se concede el privilegio de sistema.

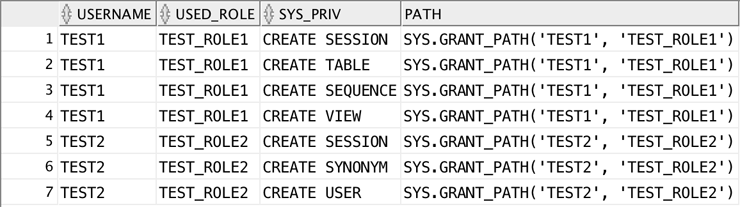
Una mención especial merece el privilegio UNLIMITED TABLESPACE. Este privilegio no aparece en la lista debido a que este privilegio se asignó de forma directa a ambos usuarios y el análisis incluye solo a los privilegios que se concedieron por medio de los roles.

La vista DBA\_USED\_SYSPRIVS\_PATH permite determinar el medio por el cual se utilizó el privilegio de sistema. La columna PATH en dicha vista nos dice si el privilegio se utilizó de forma directa o por medio de un rol.

SELECT username, used\_role, sys\_priv, path

FROM   DBA\_USED\_SYSPRIVS\_PATH

WHERE  username IN ('TEST1','TEST2');



La consulta nos muestra que el usuario TEST1 y TEST2 utilizaron los privilegios de sistema por medio de los privilegios concedidos a los roles.

**PRIVILEGIOS DE OBJETOS UTILIZADOS**

La vista DBA\_USED\_OBJPRIVS permite determinar cuáles privilegios sobre objetos fueron utilizados por los usuarios durante el periodo de análisis.

SELECT username, object\_type, object\_owner, object\_name, obj\_priv, module

FROM   DBA\_USED\_OBJPRIVS

WHERE  username = 'TEST2'

AND  object\_owner NOT IN ('SYS','SYSTEM');

Esta consulta no muestra resultado debido a que los privilegios sobre objetos fueron concedidos de forma directa al usuario TEST2 y no por medio de un rol.

La vista DBA\_USED\_OBJPRIVS\_PATH permite determinar el medio por el cual se utilizó el privilegio de objeto. La columna PATH en dicha vista nos dice si el privilegio se utilizó de forma directa o por medio de un rol.

SELECT username, obj\_priv, object\_name, path

FROM   DBA\_USED\_OBJPRIVS\_PATH

WHERE  username IN ('TEST1','TEST2');

Sucede lo mismo que con la consulta anterior.

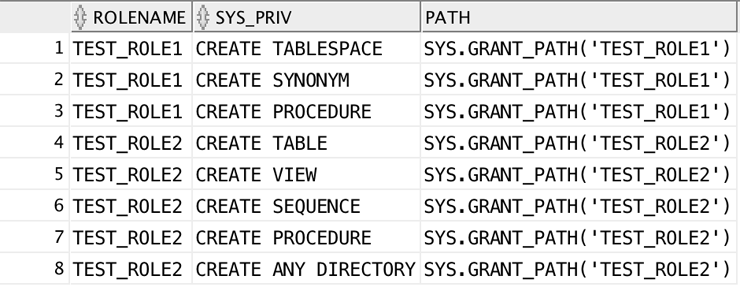
**PRIVILEGIOS NO UTILIZADOS**

Por medio de la vista DBA\_UNUSED\_PRIVS se pueden ver todos los privilegios de sistema que se asignaron a los roles y que no fueron utilizados durante el periodo de análisis. A partir de la comparación entre los privilegios utilizados y los no utilizados el DBA puede identificar cuales privilegios puede revocar.

SELECT rolename, sys\_priv, path

FROM   DBA\_UNUSED\_PRIVS

ORDER BY username;



La consulta nos muestra que del rol TEST\_ROLE1 no se utilizaron algunos de los privilegios que le fueron concedidos; mientras que del rol TEST\_ROLE2 no se utilizaron la mayoría de los privilegios que le fueron concedidos.

**ELIMINAR LA POLÍTICA DE ANÁLISIS**

Utilizando el paquete DBMS\_PRIVILEGE\_CAPTURE, procedemos a eliminar la política de análisis. Al eliminar la captura se elimina toda la información relacionada con la misma; es por esa razón que se debe tener cuidado al eliminar una captura.

BEGIN

SYS.DBMS\_PRIVILEGE\_CAPTURE.DROP\_CAPTURE(name => 'Analisis\_roles');

END;

/