

Laboratorio 7

Usuarios, privilegios y perfiles

7.1 Administración de usuarios.

Crear dos usuarios llamados gaviota y pelicano, con contraseña root, ambos utilizan el tablespace por defecto con 1mb de espacio disponible; Definir a temp como tablespace temporal.

```
CREATE USER gaviota
IDENTIFIED BY root
DEFAULT TABLESPACE system
TEMPORARY TABLESPACE temp
QUOTA 1M ON system;
```

```
CREATE USER pelicano
IDENTIFIED BY root
DEFAULT TABLESPACE system
TEMPORARY TABLESPACE temp
QUOTA 1M ON system;
```

TAREA: Investigar cuales parámetros son obligatorios al momento de crear un usuario.

TAREA: Al crear un usuario ¿Qué hace la opción **PASSWORD EXPIRE**?

7.2 Administración de privilegios.

Otorgar los permisos para iniciar sesión a "gaviota".

```
GRANT CREATE SESSION TO gaviota WITH ADMIN OPTION;  
GRANT CREATE TABLE TO gaviota;  
GRANT SELECT ANY TABLE TO gaviota;
```

TAREA: Investigar que hace la opción **"WITH ADMIN OPTION"**.

Iniciar una conexión con gaviota y desde la conexión de gaviota intentar dar a pelicano los privilegios de iniciar sesión y crear tablas, la conexión debe llamarse gaviota

```
GRANT CREATE SESSION TO pelicano;  
GRANT CREATE TABLE TO pelicano;
```

TAREA: ¿Alguna asignación falló? ¿Por qué?

Iniciar una conexión con pelicano, luego, crear la tabla "nido" desde la conexión de gaviota, la conexión debe llamarse pelicano

```
CREATE TABLE nido (  
    ident INT PRIMARY KEY,  
    lat NUMBER(8,6),  
    lon NUMBER(8,6)  
);
```

Desde la conexión de gaviota verifique quien es el propietario de la tabla "nido"

```
SELECT owner  
    , object_name  
    , object_type  
FROM ALL_OBJECTS  
WHERE object_name = 'NIDO';
```

Desde la conexión de pelicano verifique quien es el propietario de la tabla "nido"

```
SELECT owner  
    , object_name  
    , object_type  
FROM ALL_OBJECTS  
WHERE object_name = 'NIDO';
```

Desde la conexión de gaviota, otorgar a pelicano la posibilidad de realizar consultas sobre la tabla nido:

```
GRANT SELECT ON nido TO pelicano;
```

Desde la conexión de gaviota, inserte 3 registros a la tabla nido

```
INSERT INTO nido VALUES(1, -41.81015, -68.90627);  
INSERT INTO nido VALUES(2, -41.81705, -68.90700);  
INSERT INTO nido VALUES(3, -41.81001, -68.90513);
```

Desde la conexión de pelicano verifique quien es el propietario de la tabla "nido"

```
SELECT owner  
    , object_name  
    , object_type  
FROM ALL_OBJECTS  
WHERE object_name = 'NIDO';
```

Desde la conexión de pelicano, consulte la tabla nido para ver los registros insertados por gaviota

```
SELECT * FROM gaviota.nido;
```

¿Es posible ver los registros?

Desde la conexión de gaviota, realice un commit. Desde la conexión de pelicano, consulte la tabla nido para ver los registros insertados por gaviota

TAREA: Investigar porque los demas usuarios no pueden ver los datos hasta que se realice un commit.

```
SELECT * FROM gaviota.nido;
```

Cerrar sesión con gaviota y pelicano. Desde la conexión de SYS, revoque los privilegios de gaviota y pelicano

```
REVOKE CREATE SESSION FROM gaviota;  
REVOKE CREATE SESSION FROM pelicano;  
REVOKE CREATE TABLE FROM gaviota;  
REVOKE SELECT ON gaviota.nido FROM pelicano;  
REVOKE SELECT ANY TABLE FROM gaviota;
```

7.3 Administración de roles.

Desde la conexión de SYS, crear el rol seabird

```
CREATE ROLE seabird;
```

Desde la conexión de SYS, asignar los siguientes privilegios a seabird

```
GRANT CREATE SESSION TO seabird;  
GRANT CREATE TABLE TO seabird;  
GRANT SELECT ANY TABLE TO seabird;
```

Desde la conexión de SYS, asignar el rol a gaviota y pelicano

```
GRANT seabird TO gaviota;  
GRANT seabird TO pelicano;
```

TAREA: Investigar cual es la diferencia entre un privilegio y un rol de usuario.

Desde la conexión de SYS, consulte la tabla DBA_ROLES y verifique si existe el rol seabird

```
SELECT * FROM DBA_ROLES;
```

Desde la conexión de SYS, verificar los roles y privilegios asignados al usuario SYS

```
select * from USER_ROLE_PRIVS where USERNAME='SYS';  
select * from USER_TAB_PRIVS where Grantee = 'SYS';
```

```
select * from USER_SYS_PRIVS where USERNAME = 'SYS';
```

TAREA: Investigar cual es la diferencia entre las tablas *USER_ROLE_PROVS*, *USER_TAB_PRIVS* y *USER_SYS_PRIVS*.

Desde la conexión de SYS, verificar los roles y privilegios asignados al usuario SYS

```
select * from USER_ROLE_PRIVS where USERNAME='SYS';
select * from USER_SYS_PRIVS where USERNAME = 'SYS';
```

Desde la conexión de GAVIOTA, verificar los roles asignados al usuario GAVIOTA

```
select * from USER_ROLE_PRIVS where USERNAME='GAVIOTA';
select * from USER_TAB_PRIVS where Grantee = 'GAVIOTA';
```

Desde la conexión de SYS, verificar los roles y privilegios asignados al usuario GAVIOTA

```
select * from DBA_ROLE_PRIVS where Grantee='GAVIOTA';
select * from DBA_TAB_PRIVS where owner = 'GAVIOTA';
```

7.4 Administración de perfiles.

Desde la conexión de SYS, crear el siguiente perfil

```
CREATE PROFILE perfilseabird LIMIT
  FAILED_LOGIN_ATTEMPTS 5
  PASSWORD_LIFE_TIME 30
  PASSWORD_REUSE_TIME 60
  PASSWORD_REUSE_MAX 3
  PASSWORD_LOCK_TIME 1
  PASSWORD_GRACE_TIME 10;
```

TAREA: Investigar para que funcionan cada uno de los parámetros configurados en el perfil.

Desde la conexión de SYS, Asignar el perfil recién creado a gaviota

```
ALTER USER gaviota
  PROFILE perfilseabird;
```

Intentar crear una nueva iniciar conexión con el usuario gaviota (Si ya hay una sesión, entonces cerrarla y crear otra), fallando deliberadamente en la contraseña 6 veces y verifique que pasa

Desde la conexión de SYS, verificar el estado de la cuenta gaviota

```
SELECT username, user_id, password, account_status  
FROM dba_users;
```

Desde la conexión de SYS, modificar el estado de la cuenta gaviota y volver a verificar el estado

```
ALTER USER gaviota ACCOUNT UNLOCK;  
SELECT username, user_id, password, account_status  
FROM dba_users;
```

Desde la conexión de SYS, eliminar al usuario gaviota

```
DROP USER gaviota CASCADE;  
DROP USER pelicano;
```

TAREA: Investigar para que funciona el parametro "**CASCADE**" al momento de eliminar un usuario.