

Vistas:

dictionary    \_cons\_columns,    \_constraints,    \_role\_privs,    \_tables,    \_sys\_privs,    \_views  
v\$session

**Pregunta 0**

a) Crea el usuarios: “Alba” ponle la contraseña que tu quieras.

Su tablespace por defecto es personal con 1 M de quota en dicho tablespaces.

Dale los privilegio de conectarse y crear tablas en su esquema.

Ejecuta el scripts “TablasAlba” que te doy correspondiente al siguiente esquema relacional:

create user alba identified by alba

default tablespace personal

quota 1M on personal;

grant create table, create session to alba;

CIENTIFICO ( NIE, Nombre, Apellido, Ciudad, DireccionCalle, DireccionNum, Teléfono, FechaNacimiento, Varón)

PROYECTO ( Código, Nombre, Horas)

ASIGNA ( NIF, Codigo)

Asigna.NIF → Científico    Asigna.Codigo → Proyecto

b) Crea el usuarios: “Goya” ponle la contraseña que tu quieras.

Su tablespace por defecto es personal. No tiene cuota.

Dale el privilegio de conectarse

create user goya identified by goya

default tablespace personal;

grant create session to goya;

**Pregunta 1.-**

Con tu usuario administrador, dale todos los permisos necesario para que “Goya” pueda realizar todos los siguientes puntos:

Con “Goya”:

- Crear un rol llamado RolGoya que permita consultar todas las tablas de anteriores de Alba
- Mediante el rol haz que todos los usuarios del sistema puedan consultar esas tablas

Comprueba el apartado anterior con Julio realizando la siguiente consulta:

- Apellido de los científicos que no tienen asignado ningún proyecto

**\*\*Con Administrador**

**GRANT CREATE ROLE TO GOYA;**

**GRANT SELECT ON ALBA.ASIGNA TO GOYA WITH GRANT OPTION;  
GRANT SELECT ON ALBA.CIENTIFICO TO GOYA WITH GRANT OPTION;  
GRANT SELECT ON ALBA.PROYECTO TO GOYA WITH GRANT OPTION;**

**REM \*\* CON GOYA**

**CREATE ROLE ROLGOYA;**

**GRANT SELECT ON ALBA.ASIGNA TO ROLGOYA;  
GRANT SELECT ON ALBA.CIENTIFICO TO ROLGOYA;  
GRANT SELECT ON ALBA.PROYECTO TO ROLGOYA;**

**GRANT ROLGOYA TO PUBLIC;**

**Pregunta 2.- (con Goya)**

Dale todos los derechos necesarios con tu usuario administrador.

```
REM ** SOY ADMINISTRADOR  
GRANT CREATE VIEW to GOYA;  
GRANT CREATE SYNONYM TO GOYA;  
GRANT CREATE ANY TABLE TO GOYA;  
GRANT UPDATE (HORAS) ON ALBA.PROYECTO TO GOYA;  
GRANT DELETE ON ALBA.ASIGNA TO GOYA;
```

NOTA: “Goya” tiene que usar sinónimos para las tablas de alba siempre que pueda

```
CREATE SYNONYM SCIENTIFICO FOR ALBA.CIENTIFICO;  
CREATE SYNONYM SPROYECTO FOR ALBA.PROYECTO;  
CREATE SYNONYM SASIGNA FOR ALBA.ASIGNA
```

Apartado a.-

Crea una vista llamada Vpractica5 que contenga el nif y nombre de los científicos que tienen asignados proyectos de mas de 30 horas.

```
Create view VPRACTICA5 (nif, nombre) as  
select distinct c.nif, c.nombre from Scientifico c, Sasigna a, Sprojeto p  
where c.nif=a.nif and a.codigo=p.codigo and p.horas>30
```

Apartado b.-

Crea una tabla llamado “Tpractica5” en el esquema de “Alba” que muestre el nombre y apellidos del científico y el nº de proyectos que tiene asignado

```
create table alba.tpractica5 (nombre, apellido, proyectos) as  
select nombre, apellido, count(codigo)  
from scientifico c , sasigna a where c.nif=a.nif  
group by nombre, apellido;
```

**Nota: Observar que aunque puede crear la tabla no tiene derecho de consultar sobre ella ya que no se le ha dado derecho de consultar sobre la tabla**

Apartado c.-

Refleja en el esquema que todos proyectos asignados a los científicos que su ciudad es “Sevilla” han disminuido en 3 horas

Nota: “Goya” sólo puede modificar el campo horas

```
update sprojeto set horas=horas-3  
where codigo IN (  
select codigo from sasigna a , scientifico c  
where a.nif=c.nif and upper(Ciudad)='SEVILLA');  
commit;
```

Apartado d.-

Refleja en el esquema que Pepe Perez ya no tiene asignado el “Proyecto1”

```
DELETE FROM sasigna WHERE nif =  
(SELECT nif FROM scientifico where UPPER(NOMBRE)='PEPE' and  
UPPER(APELLIDO)='PEREZ')  
AND CODIGO =  
(SELECT CODIGO FROM sproyecto WHERE UPPER(NOMBRE)='PROYECTO1');  
  
commit;
```

**Pregunta 3.** Con Julia

- Dale los derechos necesarios con tu usuario administrador (Supuestamente no sabes los privilegios que tiene Julia por lo que tendras que averiguarlo..)

Con administrador  
rem Ver privilegios que tiene julia

```
select privilege, admin_option from dba_sys_privs where grantee='JULIA';  
  
grant create sequence to julia
```

- a) Crea la siguiente tabla e introduce 8 filas:

tablita (código, nombre)

- b) Crea una secuencia llamada “SEQJulia” que generes numeros de la siguiente forma (6,8,10, 4, 6, 8,10, 4, 6 ....)

REM Con Julia  
CREATE SEQUENCE SEQJULIA  
START WITH 6  
INCREMENT BY 2  
MINVALUE 4  
MAXVALUE 10  
CYCLE  
CACHE 3;

- c) Modifica la tabla tablita añadiéndole un campo llamado extra que admite valores numericos enteros de 2 cifras

`alter table tablita add extra number (3,0);`

- d) Modifica los datos de la tabla clase para que el campo extra tome los siguientes valores de la secuencia (6, 8, 8, 10, 4, 4, 6 , 8)

```
select * from tablita;

REM fila 1 y 2
update tablita set extra=seqjulia.nextval
where codigo in ('cod_1','cod_2');

REM fila 3
update tablita set extra=seqjulia.currvval
where codigo in ('cod_3');

REM fila 4 y 5
update tablita set extra=seqjulia.nextval
where codigo in ('cod_4','cod_5');

REM fila 6
update tablita set extra=seqjulia.currvval
where codigo in ('cod_6');

rem fila 7 y 8
update tablita set extra=seqjulia.nextval
where codigo in ('cod_7','cod_8');

select * from tablita;
```

**Pregunta 4.-** Con tu usuario administrador, realiza las siguientes consultas:

- a) Muestra los usuarios de Oracle conectados y el número de sesiones por usuario

```
select username Usuario_Oracle, count(username) Numero_Sesiones
from v$session
group by username
```

- b) Muestra todas las tablas que son de tu propiedad y tienen un campo que es clave ajena. En la consulta tiene que salir, el nombre de la tabla, el nombre del campo que es ajeno y a la clave primaria a la que referencia

```
SELECT CON.TABLE_NAME,C.COLUMN_NAME,CON.R_CONSTRAINT_NAME
FROM USER_CONS_COLUMNS C,USER_CONSTRAINTS CON
WHERE C.CONSTRAINT_NAME=CON.CONSTRAINT_NAME
AND CON.CONSTRAINT_TYPE='R'
order by con.r_constraint_name;
```

- c) Muestra el nombre de todos los usuarios que tienen el privilegio crear tabla y crear vistas en cualquier esquema (los dos privilegios)

```
SELECT GRANTEE FROM DBA_SYS_PRIVS WHERE PRIVILEGE = 'CREATE TABLE'
INTERSECT
SELECT GRANTEE FROM DBA_SYS_PRIVS WHERE PRIVILEGE = 'CREATE ANY VIEW';
```