

AMPLIACIÓN SQL

Objetivos

- Vistas y su función en la BD
- Sinónimos
- Roles

Vistas

CREACION DE UNA VISTA. USO DE UNA VISTA

- **VISTA:** es una tabla lógica que permite acceder a la información de una o varias tablas
- No contiene información por sí misma, sino que su información está basada en la que contienen otras tablas
- Tiene por tanto la misma estructura que una tabla y se trata de igual forma
- Una vista es una sentencia SQL
- Al borrar una tabla la vista asociada deja de funcionar

Vistas

CREACION DE UNA VISTA. USO DE UNA VISTA

Formato:

```
CREATE [OR REPLACE] VIEW nombre_vista (nom_col [,nom_col])  
AS consulta_sql;
```

Nombre_vista:	nombre de la vista
Nom_col:	nombre de las columnas de la vista, si no se especifican se adoptarán los mismos que los de la consulta
Consulta:	determina las columnas y las tablas que aparecerán en la vista
OR REPLACE:	crea de nuevo la vista si ya existía.

Vistas

CREACION DE UNA VISTA. USO DE UNA VISTA

Ejemplo: Aunque nos adelantamos en el concepto de consulta SQL (*select*), nos centramos en las consultas más básicas para entender lo que es una vista. Ya en la unidad 4 al meternos en las consultas veremos la verdadera utilidad de las vistas.

Create view DEP30 as

```
select APELLIDO, OFICIO, SALARIO , DEPT_NO from  
EMPLE where DEPT_NO=30;
```

} consulta

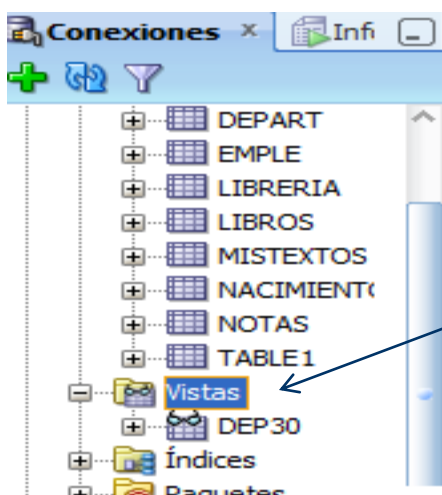
Create or replace view DEP30 (APE, OFI, SAL, DEP) as

```
select APELLIDO, OFICIO, SALARIO , DEPT_NO  
from EMPL where DEPT_NO=30; 4
```

Vistas

CREACION DE UNA VISTA. USO DE UNA VISTA

Si creamos una VISTA a partir de una consulta solo de los empleados del departamento 30, tendremos un nuevo objeto (en VISTAS/VIEW)



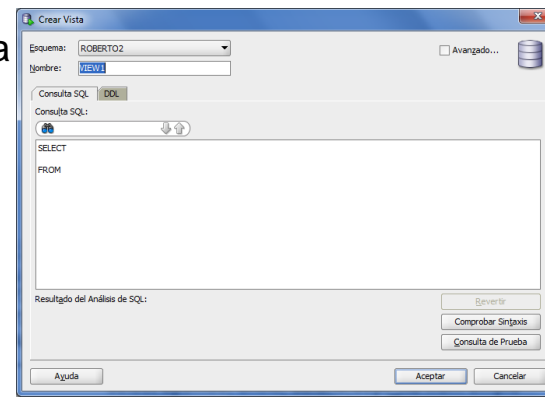
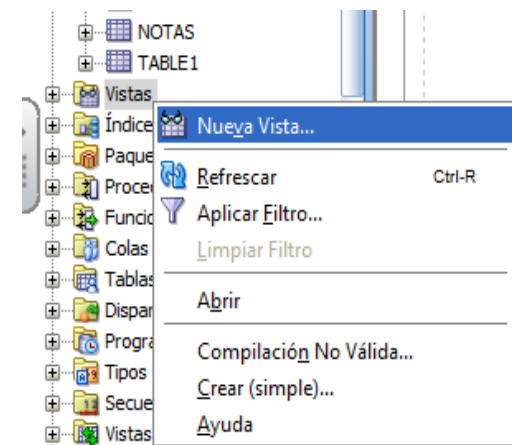
nueva vista DEP30

Vistas

CREACION DE UNA VISTA. SQLDEVELOPER

Si creamos una VISTA utilizando el SQLdeveloper

- Seleccionar el objeto VISTA del esquema
- Pulsar botón derecho del ratón
- Seleccionar NUEVA VISTA
- En la ventana que se abre:
 - indicar NOMBRE VISTA
 - Sentencia SELECT que generará la vista

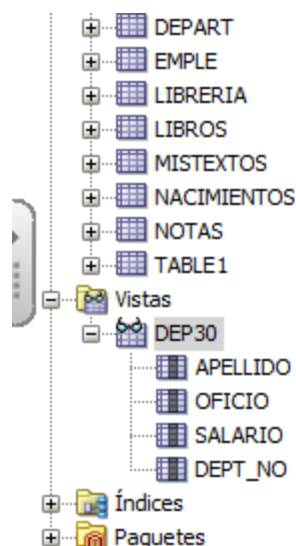


Vistas

CREACION DE UNA VISTA. USO DE UNA VISTA

A partir de este momento si pinchamos en el objeto vista, se mostrará la información de la sentencia SELECT asociada a dicha vista, pero nosotros haríamos

SELECT * FROM DEP30;



	APELLIDO	OFICIO	SALARIO	DEPT_NO
1	ARROYO	VENDEDOR	2080	30
2	SALA	VENDEDOR	1625	30
3	MARTIN	VENDEDOR	1625	30
4	NEGRO	DIRECTOR	3705	30
5	TOVAR	VENDEDOR	1950	30
6	JIMENO	EMPLEADO	1235	30

Vistas

CREACION DE UNA VISTA. USO DE UNA VISTA

Para consultar las vistas creadas se dispone de la **vista** USER_VIEWS:

```
SELECT * FROM USER_VIEWS;
```

Si queremos seleccionar solo el nombre de la vista y los campos que la forman, deberemos visualizar solo las columnas de la vista:

```
select VIEW_NAME, TEXT from USER_VIEWS
```

VIEW_NAME: muestra el nombre de la vista

TEXT: muestra la sentencia que permite obtener la vista

Vistas

CREACION DE UNA VISTA. USO DE UNA VISTA

Recordar:

- Un usuario solo podrá crear vistas si tiene el privilegio concedido
- La vista no es una tabla, sino que contiene la consulta que se ejecutara siempre que se invoque dicha vista.

Vistas

CREACION DE UNA VISTA. USO DE UNA VISTA

Ejemplo:

```
/* CREAMOS LA VISTA*/
```

```
create view DEP30 as select apellido,salario,oficio from EMPL  
where dept_no=30;
```

```
/* Visualizamos la vista y la consulta de la que se obtiene*/
```

```
SELECT VIEW_NAME,TEXT FROM USER_VIEWS;
```

VIEW_NAME	TEXT
DEP30	select apellido, oficio, salario,dept_no from emple where dept_no=30

Vistas

BORRADO DE UNA VISTA

NOTA:.. Recordar que si se elimina una tabla utilizada para obtener una vista, esta última seguirá existiendo pero quedará inutilizada

Formato:

`DROP VIEW nombre_vista;`

Ejemplo: `DROP VIEW dep30;`

Sinónimos

Creación de Sinónimos

Es un nombre que se puede dar a una tabla o una vista, permitiendo de esta forma abreviar a la hora de escribir el acceso a la misma.

Utilidad: Acceso a tablas de otro usuario

Formato:

```
CREATE [public] SYNONYM nom_sinonimo  
FOR [usuario.]nombre_tabla;
```

Public: hace que el sinónimo esté disponible para todos los usuarios. Solo puede ser usado por el DBA o usuarios con el privilegio CREATE PUBLIC SYNONYM

Sinónimos

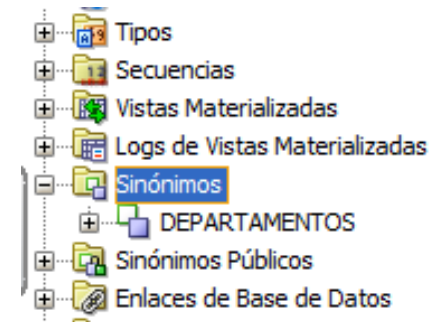
Creación de Sinónimos

Ejemplo:

Create synonym DEPARTAMENTOS for DEPART;

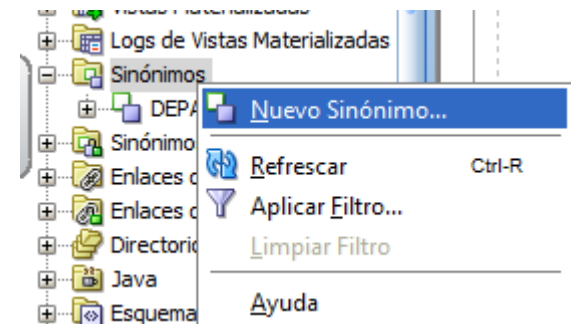
Create synonym DEPART for U3.DEPART;

Create public synonym EMPLEADOS for U3.EMPLE;



SQLDEVELOPER:

Para crear un sinónimo basta con situarse en la carpeta SINONIMOS del esquema del usuario, /botón derecho / crear



Sinónimos

Creación de Sinónimos

Ejemplo:

Si tengo acceso a la tabla DEPART del usuario U3 para realizar una acceso a dicha tabla tendré que indicar U3.DEPART

p.e. `Select * from U3.DEPART;`

Otra posibilidad es: crear un sinónimo

`Create synonym DEPARTAMENTOS for U3.DEPART;`

También el administrador podría crear el siguiente sinónimo público:

`Create public synonym EMPLEADOS for U3.EMPLE;`

Sinónimos

Borrado de Sinónimos

Formato:

```
DROP [public] SYNONYM [usuario].nom_sinonimo ;
```

Al igual que antes, solo el DBA y los usuarios con el privilegio DROP PUBLIC SYNONYM pueden suprimir sinónimos públicos.

Los usuarios con el privilegio DROP ANY SYNONYM pueden borrar los sinónimos de otros usuarios

```
DROP SYNONYM DEPARTAMENTOS;
```

```
DROP PUBLIC SYNONYM EMPLEADOS; (solo DBA o usr con priv.)
```

Sinónimos

Consulta de Sinónimos del esquema de un usuario

Para ver los sinónimos, existe una vista USER_SYNONYMS que permite ver los sinónimos que son propiedad del usuario.

DESC SYS.USER_SYNONYMS;

```
SELECT      SYNONYM_NAME "Nombre Sinónimo",  
            TABLE_OWNER "Propietario",  
            TABLA_NAME "Tabla"  
            from USER_SYNONYMS;
```


Renombrar Vistas, Sinónimos

RENAME

Permite cambiar además del nombre de una tabla el de una vista o un sinónimo.

Oracle invalida todos los objetos que dependan del objeto renombrado, como las vistas, los sinónimos que hacen referencia a la tabla renombrada.

Formato:

RENAME nombre_old TO nombre_new;

Ejemplo:

create synonym ALUM from ALUMNOS;
rename ALUMNOS to TALUMNOS;

//sinónimo para tabla ALUMNOS
// al renombrar la tabla, el sinónimo
ALUM deja de funcionar

GESTION DE SEGURIDAD

ROLES

- Conjunto de privilegios que recibe un nombre común para facilitar la tarea de asignación a los usuarios o a otros roles
- Los privilegios de un rol pueden ser de sistema y a nivel de objeto.

Formato:

CREATE ROLE nombre_rol [IDENTIFIED BY contraseña];

GESTION DE SEGURIDAD

ROLES: Creación de un objeto ROL

Ejemplo:

//-como administrador-, Creamos el rol llamado acceso

CREATE ROLE acceso;

//asignamos privilegios -como administrador-

GRANT SELECT, INSERT ON EMPLE TO acceso;

GRANT INSERT ON DEPART TO acceso;

GRANT CREATE SESSION TO acceso;

//Podemos ahora crear un usuario y a continuación asignarle este rol – como administrador-

CREATE USER MIGUEL IDENTIFIED BY MIGUEL ;

GRANT acceso TO MIGUEL; // automáticamente le estamos asignando todos los privilegios anteriores a la vez

GESTION DE SEGURIDAD

ROLES

- ***Supresión de privilegios en los roles***

REVOKE privilegios [ON nom_tabla] FROM nombre_rol;

- ***Supresión de un rol***

DROP ROLE nombre_rol;

- ***Establecer rol por defecto***

ALTER USER nombre_usuario DEFAULT {ROLE nombre_rol} | [none]};

GESTION DE SEGURIDAD

VISTAS SOBRE ROLES EN EL DICCIONARIO

- **USER_TAB_PRIVS**: concesiones sobre objetos que son propiedad del usuario, concedidos o recibidos por éste
- **ROLE_SYS_PRIVS**: privilegios del sistema asignados a roles
- **ROLE_TAB_PRIVS**: privilegios sobre tabas aplicados a roles
- **ROLE_ROLE_PRIVS**: roles asignados a otros roles

GESTION DE SEGURIDAD

VISTAS SOBRE ROLES EN EL DICCIONARIO

- **SESSION_ROLES**: roles activos para el usuario
- **USER_ROLE_PRIVS**: roles asignados al usuarios
- **DBA_SYS_PRIVS**: privilegios del sistema asignados a los usuarios o roles
- **DBA_ROLE_PRIVS**: privilegios asignados a todos los usuarios y roles
- **DBA_ROLES**: todos los roles

GESTION DE SEGURIDAD

PERFILES

Limita el uso de los recursos de la base de datos a los usuarios.
Si no se especifica el perfil, se asigna el perfil por defecto.

Ejemplo:

- Tiempo de conexión
- Conexiones concurrentes de un mismo usuario

NOTA: para activar el uso de perfiles en el sistema, el administrador ha de ejecutar la orden:

ALTER SYSTEM SET RESOURCE_LIMIT = true
(false desactiva la utilización de perfiles)

GESTION DE SEGURIDAD

PERFILES

```

CREATE PROFILE nombre_perfil LIMIT
    {SESSIONS_PER_USER | CPU_PER_SESSION |
      CPU_PER_CALL | CONNECT_TIME |
      IDLE_TIME | LOGICAL_READS_PER_SESSIONS |
      LOGICAL_READS_PER_CALL | PRIVATE_SGA |
      COMPOSITE_LIMIT}
    { entero [K | M] | UNLIMITED | DEFAULT }
    [ {SESSIONS_PER_USER | CPU_PER_SESSION |
      CPU_PER_CALL | CONNECT_TIME |
      IDLE_TIME | LOGICAL_READS_PER_SESSIONS |
      LOGICAL_READS_PER_CALL | PRIVATE_SGA |
      COMPOSITE_LIMIT}
      { entero [K | M] | UNLIMITED | DEFAULT }
      .....]
    ...
  
```


GESTION DE SEGURIDAD

PERFILES

Sessions_per_user	Nº sesiones múltiples concurrentes permitidas por usuario
Connect_time	Minutos que puede estar una sesión conectada
Idle_time	Minutos que puede estar una sesión conectada sin ser utilizada
CPU_per_session	Tiempo max. CPU por sesión (centésimas de segundo)
CPU_per_call	Tiempo max. CPU por llamada
Failed_login_attempts	Número de intentos de acceso sin éxito consecutivos para bloquear la cuenta
Password_life_time	Nº días para uso de una contraseña antes de que caduque
Password_reuse_time	Nº días que deben de pasar antes de que se pueda usar la pwd
Password_reuse_max	Nº de veces que debe cambiarse la pwd antes de poder usarla
Password_lock_time	Nº de días que quedará bloqueada la cuenta si se sobrepasa el failed_login_attempts
Password_grace_time	Período de gracia durante el cual la pwd puede cambiarse cuando ha alcanzado se ppassword_life_time

GESTION DE SEGURIDAD

PERFILES

Para activar un perfil, es necesario ejecutar previamente:

```
ALTER SYSTEM SET RESOURCE_LIMIT = TRUE;
```

- ***Supresión de un perfil***

```
DROP PROFILE nombre_perfil [CASCADE];
```

Ejemplo

```
CREATE PROFILE perfil1 LIMIT SESSIONS_PER_USER 1 CONNECT_TIME 2;  
CREATE USER prueba IDENTIFIED BY PRUEBA PROFILE perfil1;  
DROP PROFILE perfil1;
```

OTROS OBJETOS

SECUENCIAS

- Objeto de base de datos que sirve para generar enteros únicos de forma secuencial
- Útil para generar automáticamente valores para claves primarias
- Es necesario tener el privilegio `CREATE ANY SEQUENCE` para poder crear secuencias.

OTROS OBJETOS

SECUENCIAS

```
CREATE SEQUENCE nombre_secuencia  
    [ INCREMENT BY entero]  
    [START WITH entero]  
    [MAXVALUE entero | NOMAXVALUE]  
    [MINVALUE entero | NOMINVALUE]  
    [CYCLE | NOCYCLE]  
    [ORDER | NOORDER]  
    [CACHE entero | NOCACHE];
```

OTROS OBJETOS

SECUENCIAS

INCREMENT BY entero : incremento de la secuencia de números

START WITH entero : valor inicial de la secuencia

MAXVALUE entero | NOMAXVALUE: valor máximo/para no definir máximo de 10^{27} .

MINVALUE entero | NOMINVALUE: valor mínimo/para no definir mínimo de 10^{27}

CYCLE | NOCYCLE: comienza desde el principio al alcanzar MAXVALUE/ no ciclico

ORDER | NOORDER: ORDER garantiza que los números sean generados en el orden solicitado. NOORDER no garantiza ese orden

CACHE entero | NOCACHE : CACHE permite que los número de la secuencia para ser asignados se almacenen en la memoria, haciendo más rápido el acceso. El valor mínimo es 2. NOCACHE impide la asignación previa de números a la secuencia.

OTROS OBJETOS

SECUENCIAS

Para acceder a los valores actuales de la secuencia

`Nombresecuencia.CURRVAL` → devuelve el valor actual de la secuencia

`Nombresecuencia.NEXTVAL` → devuelve el siguiente valor de la secuencia

Ejemplo:

```
CREATE TABLE frutas (  
  codigo          NUMBER(2) NOT NULL PRIMARY KEY,  
  nombre          VARCHAR2(15)  
);
```

```
CREATE SEQUENCE codigo START WITH 1 INCREMENT BY 1 MAXVALUE 99;  
INSERT INTO frutas VALUES (codigo.nextval, 'MANZANAS');
```

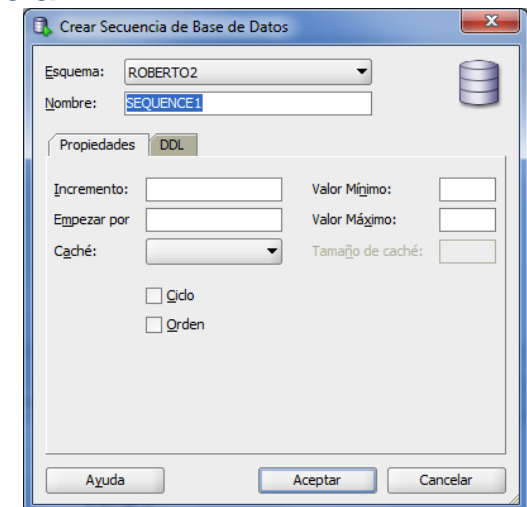
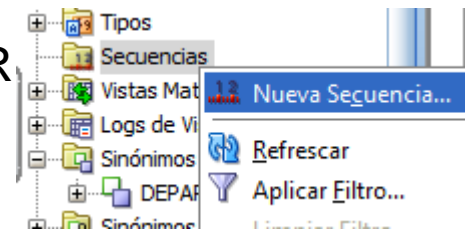
Hacemos referencia
a la secuencia

OTROS OBJETOS

SECUENCIAS

Para crear secuencias desde el SQLDEVELOPER

- Nos situamos en el objeto secuencias del esquema del usuario
- Pulsamos el botón derecho del ratón/nueva secuencia
- Asignamos el nombre, incremento, valor min, max
- Si queremos que sea ciclico o no.



OTROS OBJETOS

SECUENCIAS. EJERCICIOS

- Diseñar la tabla USUARIOS con los siguientes campos:
num_usr → number(1); clave primaria
Nom_usr → varchar(20);
- Crear una secuencia SECUENCIA_USR, que empiece en el número 1 y su último valor sea 5. El incremento de la secuencia será de dos. En el TOAD pinchar la pestaña SEQUENCE y comprobar la secuencia
- Insertar el nombre USUARIO1 y asignarle como clave primaria el número generado por la secuencia
- Insertar el nombre USUARIO2 y asignarle como clave primaria el siguiente número generado por la secuencia
- Comprobar, utilizando la tabla DUAL, el valor actual que tiene la secuencia
- Ir insertando usuarios sucesivamente USUARIO3,..... y comprobar que sucede cuando se inserta el USUARIO6