



PRÁCTICAS DE SISTEMAS DE BASES DE DATOS

GRADO EN INGENIERÍA INFORMÁTICA
2º CURSO

*SESIÓN 3: Creación /eliminación de
índices, vistas y sinónimos. Privilegios de
acceso. Modificación definición de tablas.*

Ana Belén Gil González
Ana de Luis Reboredo

Departamento de Informática y Automática
Universidad de Salamanca



Ana Belén Gil González, Ana de Luis Reboredo

3. INTRODUCCIÓN A LA CREACIÓN/ELIMINACIÓN DE ÍNDICES, VISTAS Y SINÓNIMOS. PRIVILEGIOS

3.1. Creación de índices *INDEX*

Sintaxis:

```
CREATE [ UNIQUE] INDEX nombre_índice  
ON tabla ( columna {, columna } )  
[ WITH cláusula ];
```

Los índices son estructuras que proporcionan un acceso rápido a las filas de una tabla en base a los valores de una o más columnas. Todos los datos están ordenados para tardar menos en buscarlos. Son transparentes al usuario. La principal ventaja es que se tarda menos tiempo en buscar datos. El principal inconveniente es el tiempo perdido en actualizar índices cada vez que se añaden/editan/eliminan registros.

3.2. Creación de una vista simple *VIEW*

Sintaxis:

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW nombre_vista  
AS subconsulta  
[WITH CHECK OPTION]  
[WITH READ ONLY];
```

Explicación:

- OR REPLACE. Se utiliza por si la vista ya estuviera creada anteriormente. De esta forma una vista podrá ser modificada.
- FORCE. Crea la vista independientemente de si la tabla base existe.
- NOFORCE. Crea la vista sólo si la tabla base existe. Esta opción es la que está por defecto.
- Subconsulta. Es una sentencia SELECT
- [WITH CHECK OPTION]. no se permitirá que mediante un *insert* o *update* se cree una fila que no cumpla las condiciones de la vista.
- WITH READ ONLY. Asegura que no podrán ejecutarse operaciones DML sobre la vista
- Restricciones: En la definición de la vista no se puede utilizar el ORDER BY aunque si podrá utilizarse cuando se extraigan datos de la vista.

3.3. Creación de sinónimos *SYNONYM*:

Sintaxis:

```
CREATE SYNONYM nombre_sinónimo  
FOR tabla / vista / índice ;
```

Explicación:

Crea un sinónimo para una tabla, vista o índice

3.4. Creación de secuencias SEQUENCE:

Sintaxis:

```
CREATE SEQUENCE nombre_secuencia  
[INCREMENT BY valor] [START WITH inicio];
```

Explicación:

Crea una secuencia que se puede utilizar para obtener valores para una clave primaria. Se dispone de las pseudocolumnas *nombre_secuencia.NEXTVAL* y *nombre_secuencia.CURRVAL* que pueden ser usados en sentencias SQL.

3.5. Eliminación de objetos

Sintaxis:

```
DROP [ TABLE | VIEW | INDEX | SYNONYM | SEQUENCE ]  
nombre_objeto ;
```

3.6. Concesión de privilegios de acceso a tablas y vistas

Sintaxis:

```
GRANT ALL [ PRIVILEGES ] | privilegio { , privilegio }  
ON nombre  
TO PUBLIC | usuario { , usuario }  
[ WITH GRANT OPTION ];
```

3.7. Revocación de privilegios de acceso a tablas y vistas

Sintaxis:

```
REVOKE { SELECT | DELETE | USAGE | ALL [PRIVILEGES] }  
[, ...] ON objeto FROM { PUBLIC | nombre_usuario [, ...] }  
[ FORCE | CASCADE CONSTRAINTS ]  
  
REVOKE { INSERT | UPDATE | REFERENCES } [, ...] [ ( columna  
[, ...] ) ] ON objeto FROM { PUBLIC | nombre_usuario  
[, ...] } [ FORCE | CASCADE CONSTRAINTS ]
```

3.8. Modificación en la definición de alguna tabla

El comando **ALTER TABLE** se utiliza para insertar, borrar o modificar columnas o condiciones de integridad de una tabla ya creada. Su sintaxis es la siguiente:

Sintaxis:

```
ALTER TABLE tabla
ADD [CONSTRAINT nombre] restricción_tabla
| DROP CONSTRAINT nombre
| ADD especificación_columna
| DROP COLUMN columna [ CASCADE CONSTRAINT]
```

Explicación:

- Permite añadir o eliminar columnas y condiciones de integridad a tablas ya creadas
- Al eliminar una columna se puede especificar si se eliminará el elemento indicado y todos los que dependan de él:
 - **CASCADE CONSTRAINT**

EJERCICIOS PROPUESTOS

1. Se ha observado que una parte importante de las consultas a la base de datos que requieren mejorar su eficiencia acceden a los datos de la tabla LECTOR según el valor de la PROVINCIA. ¿Qué podemos hacer para mejorar los tiempos de respuesta de dichas consultas?
2. Crear una vista para seleccionar los códigos de los préstamos activos (libros no devueltos) junto a los códigos de los lectores.
3. Crear una vista que liste los libros que se encuentran en la actualidad prestados, incluyendo el ISBN y título. Generar dicha esta vista eliminando las filas duplicadas.
4. Crear una vista para el fondo de préstamo de la sucursal 3, indicando el ISBN, título del libro y el número de ejemplares asignados y disponibles para dicha sucursal.
5. Crear una vista que liste todas las columnas de la tabla PRESTAMO para aquellos prestamos finalizados en la sucursal 1.
6. Usar la vista anterior para insertar una nueva tupla en la tabla PRESTAMO correspondiente a un préstamo de la sucursal 4. Comprobar la diferencia de comportamiento si la vista está creada con la cláusula WITH CHECK OPTION o no.
7. Modificar la vista anterior de forma que no pueda realizarse ninguna modificación sobre ella. Intentar borrar con esa vista los préstamos finalizados hace más de 5 años. ¿Cuál es la salida?
8. Examínese la diferencia entre tener un privilegio sobre una tabla y tenerlo sobre una vista definida sobre esa tabla. En especial, la manera en que un usuario puede tener un privilegio (por ejemplo SELECT) sobre una vista sin tenerlo también sobre todas las tablas subyacentes.
9. Crear un sinónimo para la tabla dispone y hacer uso de él para consultar un listado por sucursal de los ISBN que tienen a su disposición.
10. Un análisis de la base de datos muestra que es necesario añadir un campo más a la tabla sucursal, para almacenar el nombre de la sucursal. Haga una copia de la tabla sucursal y posteriormente, realice en esa tabla las operaciones necesarias para incluir el nuevo dato.
11. Se desea disponer de una nueva tabla AUTORESP que contenga información de los autores de nacionalidad española. En esa tabla, cada autor tendrá un nuevo atributo que llamaremos CodAutorEsp que será la clave primaria de esa tabla. El valor del atributo CodAutorEsp no tiene por qué coincidir con el código que el autor tenga en la tabla AUTOR. El valor de este código se generará de manera automática mediante una secuencia.
 - a. Crear la secuencia necesaria.
 - b. Crear la tabla que contenga los siguientes atributos: CodAutorEsp, Nombre, Apellido.
 - c. Rellenar la nueva tabla con los datos de los escritores españoles que se obtengan de la tabla AUTOR.
12. Crear una relación ANUNCIO que permita que los distintos usuarios de la base de datos inserten anuncios de cualquier tipo. El esquema de la relación será:
ANUNCIO (Codigo, autor, texto). El Código deberá ser único y creado automáticamente mediante una secuencia. El atributo autor se rellenará por defecto con el user de quien realice la inserción. Se darán permisos para que cualquier

usuario pueda hacer insercciones y consultas en la tabla. Probar a insertar alguna tupla en nuestra tabla y también en la creada por algún compañero.

13. Crear una vista MISANUNCIOS que recupere los datos de los anuncios cuyo autor coincida con el usuario que está consultando la vista. Dar los permisos adecuados a dicha vista. Realizar las pruebas del funcionamiento de esta vista cooperando con un compañero. Hay que recordar que varios usuarios pueden crear objetos con el mismo nombre y que se puede acceder a los objetos creados por otros usuarios mediante *esquema.objeto*, siendo *esquema* el usuario propietario del objeto.
14. Eliminar todos los índices, vistas, tablas, sinónimos y secuencias creados en los ejercicios anteriores.