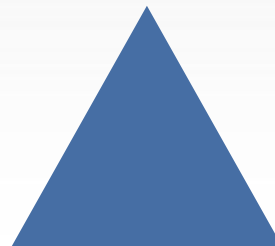


TEMA 7. VISTAS



DEFINICIÓN

- Las vistas se pueden definir como **tablas virtuales** basadas en una o más tablas o vistas y cuyos contenidos vienen definidos por una consulta sobre las mismas.
- Esta tabla virtual o consulta se le asigna un nombre y se almacena permanentemente en la BD, generando al igual que en las tablas una entrada en el diccionario de datos.
- Las vistas **NO tienen una copia física** de los datos.
- Si **actualizamos** los datos de una vista, estamos actualizando realmente la tabla, y si actualizamos la tabla estos cambios serán visibles desde la vista.

VENTAJAS DE UTILIZAR VISTAS

- Se almacenan en el servidor con lo que el consumo de recursos y eficacia siempre serán más óptimos.
- Las vistas se pueden utilizar para realizar de particiones de datos y mejorar el rendimiento cuando se copian, se importan y se exportan datos.
- Las vistas suelen utilizarse para simplificar y personalizar la percepción de la base de datos para cada usuario.

RAZONES PARA CREAR VISTAS

- Como mecanismo de **Seguridad**, que permiten a los usuarios obtener acceso a los datos por medio de la vista, pero no les conceden el permiso de obtener acceso directo a las tablas.
- Por **Comodidad**. Una vista es una forma simple para guardar complejas consultas de selección en nuestra base de datos. Tener una vista de determinados datos nos facilita no tener que escribir complejas sentencias SQL.

CREACION DE VISTAS

- Las vistas se crean con la orden CREATE VIEW.
- Su sintaxis es:

```
CREATE VIEW nombre_vista [nombre_columna]  
AS  
(<sentencia_select>);
```

- *nombre_vista*: es el nombre de la tabla virtual o vista que se va a crear.
- *nombre_columna*: son los nombres de los atributos.
 - **Es opcional**: se puede asignar un nombre a cada columna de la vista; si no, se creará la vista con los mismos nombres que las columnas de las tablas.
 - **Es obligatorio** cuando hay campos que tienen el mismo nombre o cuando la sentencia select es una función de agregado o una operación aritmética.

CREACIÓN DE VISTAS

- Las tablas y las vistas comparten el mismo espacio de nombres en la base de datos, por lo tanto, una base de datos no puede contener una tabla y una vista con el mismo nombre.
- Las vistas se crean en la base de datos activa.
- **Es aconsejable probar la sentencia "select" con la cual definiremos la vista antes de crearla para asegurarnos que el resultado que retorna es el que esperamos.**
- Se pueden construir vistas sobre otras vistas.

EJEMPLO

- Crear una vista con los datos personales de los proveedores (tabla proveedores)

CREATE VIEW DATOS_PROVEEDORES

AS

SELECT codigo, nombre, localidad, telefono

FROM Proveedores

Más Ejemplos

```
CREATE VIEW APELLIDOS_EMPLEADOS  
AS  
SELECT * FROM EMPLEADOS  
WHERE apellido LIKE 'A%'
```

¿Qué hace?

Más Ejemplos

```
CREATE VIEW APELLIDOS_EMPLEADOS  
AS  
SELECT * FROM EMPLEADOS  
WHERE apellido LIKE 'A%'
```

En este ejemplo se crea una vista que llamamos APELLIDOS_EMPLEADOS que contiene los EMPLEADOS cuyo apellido comienza con la letra A.

Más Ejemplos

Crear una vista que muestre todos los datos de la tabla PEDIDOS y todos los datos de la tabla PROVEEDORES.

```
CREATE VIEW DATOS_PEDIDOS
```

```
AS
```

```
SELECT Pedidos.numped, Pedidos.fechaped,  
Proveedores.nombrpro, Proveedores.localpro, Proveedores.telefpro,  
Proveedores.faxpro, FROM Proveedores INNER JOIN  
Pedidos ON Proveedores.codigpro = Pedidos.codigpro
```

EN ESTE EJEMPLO, SE HA ANTEPUESTO A CADA CAMPO EL NOMBRE DE LA TABLA PARA EVITAR POSIBLES AMBIGÜEDADES CON LOS NOMBRES DE CAMPOS QUE PUDIERAN ESTAR REPETIDOS EN AMBAS TABLAS.

CLAÚSULA WITH CHECK OPTION

- Puede especificarse cuando se crea una vista.
- La cláusula CHECK OPTION **evita que sean insertados o modificados registros** que no sean posteriormente incluidos en la vista mediante la sentencia WHERE.

**CREATE VIEW nombre_vista AS
sentencia_select [WITH CHECK OPTION]**

CLÁUSULA WITH CHECK OPTION-

Ejemplo de uso

- Creamos la tabla LOCALIDADES, que guarda el nombre de la población y el número de habitantes:

```
CREATE TABLE localidades  
(poblacion CHAR (30) NOT NULL,  
habitantes INT NOT NULL);
```

Y creamos la vista HABITANTES, donde solo guardaremos aquellas localidades con más de 20000 habitantes.

```
CREATE VIEW habitantes  
AS SELECT * FROM entidad  
WHERE habitantes<20000 WITH CHECK OPTION;
```

- Si ahora intentáramos insertar en la vista una población con más de 20.000 habitantes, nos daría **error**, al haber creado la vista con la opción **with check option**.

CLAÚSULA WITH CHECK OPTION-

Ejemplo de uso

- Si ahora intentáramos insertar en la vista una población con más de 20000 habitantes, nos daría error, al haber creado la vista con la opción with check option.

```
insert into entidades  
values ('madrid',450000)
```

```
ERROR:  new row violates check option for view "entidades"  
DETAIL:  La fila que falla contiene (madrid                , 450000).  
  
***** Error *****
```

VER EL CONTENIDO DE UNA VISTA

- Una vez creada la vista, se puede utilizar de la misma manera que una tabla creada con la orden CREATE TABLE.
- Para consultar el contenido de la vista:

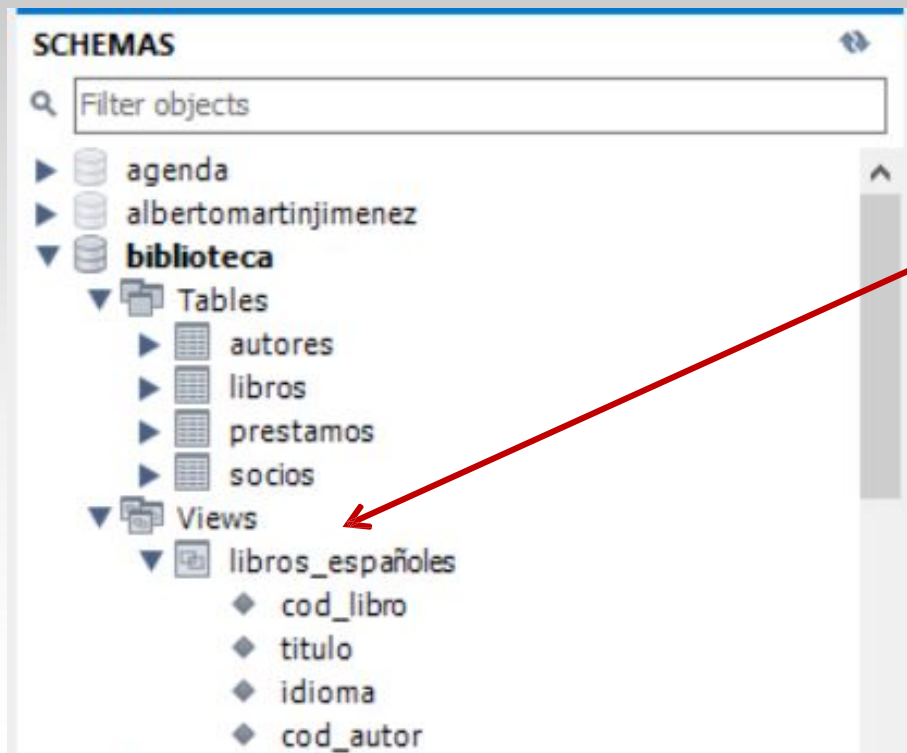
```
SELECT * FROM nombre_vista  
[where condición]
```

Por ejemplo:

```
SELECT *  
FROM APELLIDOS_EMPLEADOS;
```

```
CREATE VIEW LIBROS_ESPAÑOL  
AS  
SELECT * FROM LIBROS  
WHERE IDIOMA='ESPAÑOL'
```

- En el **entorno gráfico** podemos comprobar que se ha creado nuestra vista:



MODIFICAR VISTAS

CREATE OR REPLACE Nombre_vista
AS

<nueva consulta select>

Si la vista fue creada anteriormente, será sustituida por una nueva, resultado de la ejecución de la nueva consulta.

También es válida la instrucción:

ALTER VIEW Nombre_vista
AS

<nueva consulta select>

EJEMPLO

- **Modifica** la vista **DATOS_PROVEEDORES** para que solo contenga los datos personales de los proveedores que son de Madrid.

ALTER VIEW DATOS_PROVEEDORES

AS

SELECT codigo, nombre, localidad, telefono

FROM Proveedores

WHERE localidad='MADRID'

OTRO EJEMPLO

La vista que creamos antes APELLIDOS_EMPLEADOS contenía los siguientes datos:

1	ALVAREZ	ANALISTA	CALLE EL PE	1990-01-28	1500.00	0.25
2	ANTUNEZ	ANALISTA	CALLE EL SC	1990-01-28	1500.00	0.25
3	ALVAREZ	PROGRAMADOR	CALLE LUNA	1995-01-02	1350.00	
10	ALVAREZ	COMERCIAL	CALLE LA OC	1990-05-05	1500.00	0.25

Modificamos para que sólo salgan los que se llaman ALVAREZ

**CREATE OR REPLACE VIEW APELLIDOS_EMPLEADOS
AS**

SELECT * FROM EMPLEADOS WHERE APELLIDO='ALVAREZ'

1	ALVAREZ	ANALISTA	CALLE EL PE	1990-01-28	1500.00	0.25
3	ALVAREZ	PROGRAMADOR	CALLE LUNA	1995-01-02	1350.00	
10	ALVAREZ	COMERCIAL	CALLE LA OC	1990-05-05	1500.00	0.25

BORRAR VISTAS

- Las vistas pueden borrarse como cualquier otro objeto de la BD.
- Cuando se elimina una vista, las tablas y los datos en los que está basada **no se ven afectados**. (NO SE BORRAN)
- Se pueden producir errores al ejecutar las consultas, en el caso de que éstas utilicen objetos que dependen de la vista eliminada y a menos que se cree otra vista con el mismo nombre.

EJEMPLO

- Borrar las vistas DATOS_PROVEEDORES y APELLIDOS_EMPLEADOS que habíamos creado antes.

```
DROP VIEW DATOS_PROVEEDORES;  
DROP VIEW APELLIDOS_EMPLEADOS;
```

EN RESUMEN, UNA VISTA :

- Es una **estructura lógica** que permite visualizar un grupo de datos que provienen de una o varias tablas u otras vistas.
- No **contiene datos propios**, estos provienen de otras tablas con datos reales.
- Su **utilización es como la de una tabla**, podemos usar cualquier sentencia de tipo SELECT sobre ellas.