



## CAPÍTULO 11º: VISTAS

### 1 Concepto de Vista. -

Una vista es una "tabla virtual" en una Base de Datos, cuyos contenidos están definidos por una consulta. Para el usuario la vista aparece como una tabla real formada por columnas designadas y filas de datos. Sin embargo, a diferencia de la tabla real, la vista no existe en la base de datos como un conjunto almacenado de valores, esto es, los datos y columnas visibles en la vista son los resultados de la consulta que define la vista. SQL suministra a la vista un nombre semejante al de una tabla y almacena la definición de la vista en la base de datos.

Las vistas permiten:

- ◆ *Acomodar* el aspecto de la base de datos de manera que diferentes usuarios vean la base desde distintas perspectivas.
- ◆ *Restringir* el acceso a los datos, de modo que diferentes usuarios únicamente vean ciertas filas o ciertas columnas de una tabla.
- ◆ *Simplificar* el acceso a la base de datos permitiendo una presentación de la estructura de los datos almacenados de modo que sea más natural a cada usuario.

Cuando el SGBD encuentra una referencia a una vista en una sentencia SQL determina la definición de la vista almacenada en la base de datos, traduce la petición de consulta indicada en la vista a una petición equivalente con las tablas fuente de la vista y ejecuta la sentencia SQL. De este modo el SGBD mantiene la integridad de las tablas fuente.

### 2 Ventajas y desventajas de las Vistas. -

La utilización de vistas suministra importantes **beneficios**:

- ⊗ *Seguridad.*- Cada usuario puede obtener permiso para acceder a la base de datos mediante determinadas vistas que contienen los datos específicos que dicho usuario está autorizado a ver.
- ⊗ *Simplicidad de Consulta.*- La vista puede extraer datos de varias tablas diferentes y presentarlos como una única tabla, haciendo que consultas multitabla pasen a ser consultas sobre una sola vista.
- ⊗ *Simplicidad estructurada.*- Pueden dar al usuario una visión personalizada de la base de datos ofreciéndole a éste un conjunto de tablas virtuales que son precisamente las que tienen sentido para dicho usuario.



- ⊗ *Integridad de los datos.*- Si se accede a los datos y se introducen a través de una vista, el SGBD puede comprobar automáticamente los datos para asegurarse que satisfacen restricciones de integridad especificadas.

Aunque las vistas presentan ventajas sustanciales hay que tener en cuenta dos **desventajas** importantes de cara a utilizar una vista en vez de una tabla real:

- *Rendimiento.*- Las vistas crean una apariencia de una tabla, pero el SGBD debe traducir las consultas definidas sobre una vista en consultas sobre las tablas fuente subyacentes a la vista. Ello hace que si la vista se define mediante una consulta multitabla compleja, aunque la consulta realizada sobre la vista sea sencilla, se convierte en una composición complicada para el SGBD y su realización puede tardar mucho tiempo en completarse.
- *Restricciones de actualización.*- Al tratar de actualizar filas en una vista, el SGBD debe traducir la petición a una actualización en las tablas fuente origen de la vista. No hay problemas si la vista es sencilla, pero vistas complejas pueden no ser actualizadas y convertirse en de "sólo lectura".

### **3 Creación de una vista.** -

La sentencia `CREATE VIEW` permite crear una vista. Asigna a la vista un nombre y especifica la consulta que define la vista. La creación de una vista implica el tener permiso de acceso a todas las tablas referenciadas en la consulta. Además puede asignar un nombre a cada columna de la vista recién creada.

Aunque todas las vistas se crean del mismo modo, en la práctica se utilizan tipos distintos de vista para propósitos diferentes:

#### **3.1 Vistas Horizontales**

Tiene por objeto restringir al usuario el acceso a determinadas filas de una tabla. Se crea, por tanto mediante una consulta sobre todas las columnas de la tabla y cuya condición de búsqueda limita las filas únicamente a las deseadas. Son ejemplos de vistas horizontales:

```
CREATE VIEW Vista AS

SELECT *

FROM Tabla

WHERE Campo1 IN (Valor1, Valor2, ... , ValorN)

----
```



```
CREATE VIEW Vista AS
```

```
SELECT *
```

```
FROM Tabla
```

```
WHERE Campo1 = 'Texto' OR Campo2= Valor
```

### 3.2 *Vistas Verticales*

Tiene por objeto restringir al usuario el acceso a determinadas columnas de una tabla. Su sintaxis general más elemental sería:

```
CREATE VIEW Vista AS
```

```
SELECT Campo1, Campo2, ... , CampoN
```

```
FROM Tabla
```

Donde Campo1, Campo2, ... , CampoN son las columnas que se quieren contemplar en la vista.

### 3.3 *Vistas con subconjuntos fila/columna*

Son aquel tipo de vistas en las que se restringe tanto el acceso a determinadas columnas de una tabla como a ciertas filas que no cumplen una determinada condición de búsqueda:

```
CREATE VIEW Vista AS
```

```
SELECT Campo1, Campo2, ... , CampoN
```

```
FROM Tabla
```

```
WHERE Campo1 = Valor AND Campo2 = 'Texto'
```

Donde Campo1, Campo2, ... , CampoN son las columnas que se quieren contemplar en la vista.

Y únicamente se verán aquellas filas en las que el valor de Campo1 sea Valor y el contenido de Campo2 sea Texto.

### 3.4 *Vistas Agrupadas*

Son aquellas en las que los datos visualizados proceden de una consulta agrupada. Agrupa filas relacionadas de datos y producen una fila de resultados de consulta para cada grupo resumiendo los datos de ese grupo.



```
CREATE VIEW Vista (Vcampo1, Vcampo2, Vcampo3, Vcampo4, Vcampo5, Vcampo6) AS  
  
SELECT Campo1, COUNT(*), SUM(Campo2), MIN(Campo3), MAX(Campo4), AVG(Campo5)  
  
FROM Tabla  
  
GROUP BY Campo1
```

La vista creada agrupa, en función de Campo1 de la Tabla las operaciones de columna (COUNT(\*), SUM(), MIN(), MAX(), AVG()) respectivamente, de los campos Campo2, Campo3, Campo4 y Campo5 y las asigna respectivamente a las columnas de la consulta Vcampo1, Vcampo2, Vcampo3, Vcampo4, Vcampo5 y Vcampo6.

### 3.5 *Vistas compuestas*

Una de las principales razones para la utilización de vistas es simplificar las consultas multitabla. Especificando una consulta de dos o tres tablas en la definición de la vista, se puede crear una vista compuesta que extrae sus datos de dos o tres tablas diferentes y presenta los resultados como una tabla virtual.

Una vez definida la vista, pueden realizarse consultas simples sobre dicha vista con peticiones que, en caso contrario requerirían una composición de dos o tres tablas.

```
CREATE VIEW Vista Vcampo1, Vcampo2, Vcampo3 AS  
  
SELECT CampoA1, CampoB1, CampoC1  
  
FROM TablaA, TablaB, TablaC  
  
WHERE CampoA2 = CampoB2 AND CampoB3 = CampoC3
```

### 4 *Actualización de una vista.* -

Determinadas vistas permiten insertar datos, modificarlos o suprimir filas de la vista, estas operaciones son obviamente traducidas a operaciones equivalentes respecto de las tablas fuente de la vista. Esto es, una actualización en los datos de una vista comporta la actualización de los datos (eliminación, modificación o inserción) de la o las tablas subyacentes a la vista.

El estándar ANSI/ISO especifica las vistas que pueden ser actualizables. Según este estándar, una vista puede ser actualizada si la consulta que la define satisface todas las restricciones siguientes:

- No debe especificarse la cláusula DISTINCT esto es, las filas duplicadas no deben ser eliminadas de los resultados de la consulta .



- La cláusula FROM de la consulta subyacente debe especificar únicamente una tabla actualizable, esto es debe tener una única tabla fuente para la cual el usuario tiene las autorizaciones y privilegios requeridos. Si la tabla fuente es así mismo una vista, ésta debe satisfacer estos criterios.
- Cada elemento de selección debe ser una referencia de columna simple. La lista de selección no puede contener expresiones, columnas calculadas o funciones de columna.
- La cláusula WHERE no debe incluir una subconsulta, sólo pueden aparecer condiciones de búsqueda simples fila a fila.
- La consulta subyacente no debe incluir una cláusula GROUP BY o HAVING

El resumen de estas restricciones es el siguiente: Para que una vista sea actualizable, el SGBD debe ser capaz de relacionar cualquier fila de la vista con su respectiva fila fuente de la tabla fuente. Análogamente debe ser capaz de relacionar cada columna individual de la vista con la columna individual a actualizar de la tabla fuente.

En general, las actualizaciones de vistas en los productos comerciales basados en SQL son bastante menos restrictivas que el estándar fijado por el ANSI/ISO.

#### 5 Eliminación de una vista .-

La eliminación de una vista según el estándar ANSI/ISO para SQL2 se realiza mediante la sentencia DROP VIEW seguida del nombre de la vista y del método de eliminación (CASCADE, RESTRICT, etc.). Según:

```
DROP VIEW NombreVista CASCADE
```

Aunque la mayoría de los productos comerciales soporta la versión de la sentencia DROP VIEW sin establecer explícitamente el método de eliminación.