

VI. VISTAS EN SQL 2008

VISTAS

Una vista es una tabla virtual cuyo contenido está definido por una consulta. Al igual que una tabla real, una vista consta de un conjunto de columnas y filas de datos con un nombre. Sin embargo, a menos que esté indexada, una vista no existe como conjunto de valores de datos almacenados en una base de datos. Las filas y las columnas de datos proceden de tablas a las que se hace referencia en la consulta que define la vista y se producen de forma dinámica cuando se hace referencia a la vista.

Una vista actúa como filtro de las tablas subyacentes a las que se hace referencia en ella.

La consulta que define la vista puede provenir de una o de varias tablas, o bien de otras vistas de la base de datos actual u otras bases de datos. Asimismo, es posible utilizar las consultas distribuidas para definir vistas que utilicen datos de orígenes heterogéneos. Esto puede resultar de utilidad, por ejemplo, si desea combinar datos de estructura similar que proceden de distintos servidores, cada uno de los cuales almacena los datos para una región distinta de la organización.

No existe ninguna restricción a la hora de consultar vistas y muy pocas restricciones a la hora de modificar los datos de éstas.

En esta ilustración se muestra una vista basada en dos tablas.

Tabla **EmployeeMaster**

EmployeeID	FirstName	AddressID	ShiftID	LastName	MiddleName	SSN	...
1	Sheri	1	1	Nowmer	E	245797967	...
2	Derrick	2	1	Whelply	R	509647174	...
3	Michael	3	1	Spence	C	42487730	...
4	Maya	4	1	Gutierrez	Y	56920285	...
5	Roberta	5	1	Damstra	B	695256908	...

Ver

FirstName	LastName	Description
Sheri	Nowmer	Engineering
Derrick	Whelply	Engineering
Michael	Spence	Engineering

Tabla **Department**

DepartmentID	Description	rowguid
1	Engineering	3FFD2603-EB6E-43B2-A8EF-C4F5C3064026
2	Tool Design	AE948718-D4BF-40E0-8ECD-2D9F4A0B211E
3	Sales	702C0EE3-03E6-4F95-9AB8-99F4F25921F3
4	Marketing	3E3C4476-B9EC-43CB-AA12-1E7A140A71A4
5	Purchasing	D6C63691-93B5-4F43-AD88-34B6B9A3C4A3

Las principales razones por las que podemos crear vistas son:

- **Seguridad**, nos pueden interesar que los usuarios tengan acceso a una parte de la información que hay en una tabla, pero no a toda la tabla.
- **Comodidad**, como hemos dicho el modelo relacional no es el más cómodo para visualizar los datos, lo que nos puede llevar a tener que escribir complejas sentencias SQL, tener una vista nos simplifica esta tarea.

Las vistas no tienen una copia física de los datos, son consultas a los datos que hay en las tablas, por lo que si actualizamos los datos de una vista, estamos actualizando realmente la tabla, y si actualizamos la tabla estos cambios serán visibles desde la vista.

Nota: No siempre podremos actualizar los datos de una vista, dependerá de la complejidad de la misma (dependerá de si el conjunto de resultados tiene acceso a la clave principal de la tabla o no).

Antes de crear una vista, considere las siguientes indicaciones:

- Sólo puede crear vistas en la base de datos actual. Sin embargo, las tablas y las vistas a las que se haga referencia desde la nueva vista pueden encontrarse en otras bases de datos e, incluso, en otros servidores, si la vista se define mediante consultas distribuidas.
- Los nombres de las vistas deben seguir las reglas que se aplican a los identificadores y ser únicos para cada esquema. Además, el nombre debe ser distinto del de las tablas incluidas en ese esquema.
- Es posible generar vistas dentro de otras vistas. Microsoft SQL Server permite anidar vistas. El anidamiento no debe superar los 32 niveles. Es posible que el límite real del anidamiento de vistas sea inferior en función de la complejidad de la vista y de la memoria disponible.
- No puede asociar con las vistas reglas ni definiciones DEFAULT.
- Los desencadenadores AFTER no se pueden asociar con las vistas; sólo se pueden asociar los desencadenadores INSTEAD OF.

- La consulta que define la vista no puede incluir las cláusulas COMPUTE ni COMPUTE BY, y tampoco puede incluir la palabra clave INTO.
- La consulta que define la vista no puede incluir la cláusula ORDER BY, a menos que también haya una cláusula TOP en la lista de selección de la instrucción SELECT.
- La consulta que define la vista no puede incluir la cláusula OPTION que especifica una sugerencia de consulta.
- La consulta que define la vista no puede incluir la cláusula TABLESAMPLE.
- No se pueden definir definiciones de índice de texto completo en las vistas.
- No se pueden crear vistas temporales, ni vistas dentro de tablas temporales.
- Las vistas, las tablas o las funciones que participan en una vista creada con la cláusula SCHEMABINDING no se pueden quitar, a menos que se quite o cambie esa vista de forma que deje de tener un enlace de esquema. Además, las instrucciones ALTER TABLE sobre tablas que participan en vistas que tienen enlaces de esquemas provocarán un error si estas instrucciones afectan a la definición de la vista.
- Si una vista no se crea con la cláusula SCHEMABINDING, debe ejecutarse sp_refreshview cuando se realicen cambios en los objetos subyacentes de la vista que afecten a la definición de ésta. De lo contrario, la vista puede generar resultados inesperados cuando se realiza una consulta.
- No puede emitir consultas de texto completo en una vista, aunque una definición de vista puede incluir una consulta de texto completo si ésta hace referencia a una tabla configurada para la indización de texto completo.
- Debe especificar el nombre de todas las columnas de la vista en el caso de que:
 - I. Alguna de las columnas de la vista derive de una expresión aritmética, una función integrada o una constante.
 - II. Dos o más columnas de la vista tuviesen, en caso contrario, el mismo nombre (normalmente, debido a que la definición de la vista incluye una combinación y las columnas de dos o más tablas diferentes tienen el mismo nombre).
 - III. Desea darle a una columna de la vista un nombre distinto del de la columna de la que deriva. (También puede cambiar el nombre de las columnas en la vista). Una columna de una vista hereda los tipos de datos de la columna de la que deriva, aunque no cambie su nombre.

CREACIÓN DE VISTAS.

Para crear una vista debemos utilizar la sentencia CREATE VIEW, debiendo proporcionar un nombre a la vista y una sentencia SQL SELECT válida.

```
CREATE VIEW <nombre_vista>
```

```
AS
```

```
(<sentencia_select>);
```

Ejemplo: Crear una vista sobre la tabla alquileres, en la que se nos muestre el nombre y apellidos del cliente en lugar de su código.

```
CREATE VIEW vAlquileres
```

```
AS
```

```
(
```

```
SELECT nombre,
```

```
    apellidos,
```

```
    matricula
```

```
FROM tAlquileres,
```

```
    tClientes
```

```
WHERE ( tAlquileres.codigo_cliente = tClientes.codigo )
```

```
)
```

Si queremos, modificar la definición de nuestra vista podemos utilizar la sentencia ALTER VIEW, de forma muy parecida de cómo se realiza con las tablas. En este caso queremos añadir los campos fx_alquiler y fx_devolucion a la vista.

```
ALTER VIEW vAlquileres
```

```
AS
```

```
(
```

```
SELECT nombre,
```

```
    apellidos,
```

```
    matricula,
```

```
    fx_alquiler,
```

```
    fx_devolucion
```

```
FROM tAlquileres,  
    tClientes  
WHERE ( tAlquileres.codigo_cliente = tClientes.codigo )  
)
```

Por último podemos eliminar la vista a través de la sentencia DROP VIEW. Para eliminar la vista que hemos creado anteriormente se utilizaría:

```
DROP VIEW vAlquileres;
```