Vistas

Una vista es una alternativa para mostrar datos de varias tablas. Una vista es como una tabla virtual que almacena una consulta. Los datos accesibles a través de la vista no están almacenados en la base de datos como un objeto.

Entonces, una vista almacena una consulta como un objeto para utilizarse posteriormente. Las tablas consultadas en una vista se llaman tablas base. En general, se puede dar un nombre a cualquier consulta y almacenarla como una vista.

Una vista suele llamarse también tabla virtual porque los resultados que retorna y la manera de referenciarlas es la misma que para una tabla.

Las vistas permiten:

- ocultar información: permitiendo el acceso a algunos datos y manteniendo oculto el resto de la información que no se incluye en la vista. El usuario opera con los datos de una vista como si se tratara de una tabla, pudiendo modificar tales datos.

- simplificar la administración de los permisos de usuario: se pueden dar al usuario permisos para que solamente pueda acceder a los datos a través de vistas, en lugar de concederle permisos para acceder a ciertos campos, así se protegen las tablas base de cambios en su estructura.

- mejorar el rendimiento: se puede evitar tipear instrucciones repetidamente almacenando en una vista el resultado de una consulta compleja que incluya información de varias tablas.

Podemos crear vistas con: un subconjunto de registros y campos de una tabla; una unión de varias tablas; una combinación de varias tablas; un resumen estadístico de una tabla; un subconjunto de otra vista, combinación de vistas y tablas.

Una vista se define usando un "select".

La sintaxis básica parcial para crear una vista es la siguiente:

```
create view NOMBREVISTA as
 SENTENCIASSELECT
  from TABLA;
```

El contenido de una vista se muestra con un "select":

```
select *from NOMBREVISTA;
```

En el siguiente ejemplo creamos la vista "vista_empleados", que es resultado de una combinación en la cual se muestran 4 campos:

```
create view vista_empleados as
 select (apellido+' '+e.nombre) as nombre,sexo,
  s.nombre as seccion, cantidadhijos
  from empleados as e
  join secciones as s
  on codigo=seccion
```

Para ver la información contenida en la vista creada anteriormente tipeamos:

```
select *from vista_empleados;
```

Podemos realizar consultas a una vista como si se tratara de una tabla:

```
select seccion,count(*) as cantidad
 from vista_empleados;
```

Los nombres para vistas deben seguir las mismas reglas que cualquier identificador. Para distinguir una tabla de una vista podemos fijar una convención para darle nombres, por ejemplo, colocar el sufijo “vista” y luego el nombre de las tablas consultadas en ellas.

Los campos y expresiones de la consulta que define una vista DEBEN tener un nombre. Se debe colocar nombre de campo cuando es un campo calculado o si hay 2 campos con el mismo nombre. Note que en el ejemplo, al concatenar los campos "apellido" y "nombre" colocamos un alias; si no

lo hubiésemos hecho aparecería un mensaje de error porque dicha expresión DEBE tener un encabezado, SQL Server no lo coloca por defecto.

Los nombres de los campos y expresiones de la consulta que define una vista DEBEN ser únicos (no puede haber dos campos o encabezados con igual nombre). Note que en la vista definida en el ejemplo, al campo "s.nombre" le colocamos un alias porque ya había un encabezado (el alias de la concatenación) llamado "nombre" y no pueden repetirse, si sucediera, aparecería un mensaje de error.

Otra sintaxis es la siguiente:

```
create view NOMBREVISTA (NOMBRESDEENCABEZADOS)
 as
 SENTENCIASSELECT
  from TABLA;
```

Creamos otra vista de "empleados" denominada "vista_empleados_ingreso" que almacena la cantidad de empleados por año:

```
create view vista_empleados_ingreso (fecha,cantidad)
 as
 select datepart(year,fechaingreso),count(*)
  from empleados
  group by datepart(year,fechaingreso)
```

La diferencia es que se colocan entre paréntesis los encabezados de las columnas que aparecerán en la vista. Si no los colocamos y empleamos la sintaxis vista anteriormente, se emplean los nombres de los campos o alias (que en este caso habría que agregar) colocados en el "select" que define la vista. Los nombres que se colocan entre paréntesis deben ser tantos como los campos o expresiones que se definen en la vista.

Las vistas se crean en la base de datos activa.

Al crear una vista, SQL Server verifica que existan las tablas a las que se hacen referencia en ella.

Se aconseja probar la sentencia "select" con la cual definiremos la vista antes de crearla para asegurarnos que el resultado que retorna es el imaginado.

Existen algunas restricciones para el uso de "create view", a saber:

- no puede incluir las cláusulas "compute" ni "compute by" ni la palabra clave "into";

- no se pueden crear vistas temporales ni crear vistas sobre tablas temporales.

- no se pueden asociar reglas ni valores por defecto a las vistas.

- no puede combinarse con otras instrucciones en un mismo lote.

Se pueden construir vistas sobre otras vistas.

Vistas (información)

Las vistas son objetos, así que para obtener información de ellos pueden usarse los siguientes procedimientos almacenados del sistema:

"sp_help" sin parámetros nos muestra todos los objetos de la base de datos seleccionada, incluidas las vistas. En la columna "Object_type" aparece "view" si es una vista. Si le enviamos como argumento el nombre de una vista, obtenemos la fecha de creación, propietario, los campos y demás información.

"sp_helptext" seguido del nombre de una vista nos muestra el texto que la define, excepto si ha sido encriptado.

Ejecutando "sp_depends" seguido del nombre de un objeto, obtenemos 2 resultados:

- nombre, tipo, campos, etc. de los objetos de los cuales depende el objeto nombrado y

- nombre y tipo de los objetos que dependen del objeto nombrado.

Si ejecutamos el procedimiento "sp_depends" seguido del nombre de una vista:

```
sp_depends vista_empleados;
```

aparecen las tablas (y demás objetos) de las cuales depende la vista, es decir, las tablas referenciadas en la misma.

Si ejecutamos el procedimiento seguido del nombre de una tabla:

```
sp_depends empleados;
```

aparecen los objetos que dependen de la tabla, vistas, restricciones, etc.

También se puede consultar la tabla del sistema "sysobjects":

```
select *from sysobjects;
```

Nos muestra nombre y varios datos de todos los objetos de la base de datos actual. La columna "xtype" indica el tipo de objeto, si es una vista, aparece 'V'.

Si queremos ver todas las vistas creadas por nosotros, podemos tipear:

```
select *from sysobjects
 where xtype='V' and-- tipo vista
 name like 'vista%';--búsqueda con comodín
```

Vistas (eliminar)

Para quitar una vista se emplea "drop view":

```
drop view NOMBREVISTA;
```

Si se elimina una tabla a la que hace referencia una vista, la vista no se elimina, hay que eliminarla explícitamente.

Solo el propietario puede eliminar una vista.

Antes de eliminar un objeto, se recomienda ejecutar el procedimiento almacenado de sistema "sp_depends" para averiguar si hay objetos que hagan referencia a él.

Eliminamos la vista denominada "vista_empleados":

```
drop view vista_empleados;
```

Vistas (with check option)

Es posible obligar a todas las instrucciones de modificación de datos que se ejecutan en una vista a cumplir ciertos criterios.

Por ejemplo, creamos la siguiente vista:

```
create view vista_empleados
as
 select apellido, e.nombre, sexo, s.nombre as seccion
 from empleados as e
```

```
join secciones as s
on seccion=codigo
where s.nombre='Administracion'
with check option;
```

La vista definida anteriormente muestra solamente algunos de los datos de los empleados de la sección "Administracion". Además, solamente se permiten modificaciones a los empleados de esa sección.

Podemos actualizar el nombre, apellido y sexo a través de la vista, pero no el campo "seccion" porque está restringuido.

Vistas (modificar datos de una tabla a través de vistas)

Si se modifican los datos de una vista, se modifica la tabla base.

Se puede insertar, actualizar o eliminar datos de una tabla a través de una vista, teniendo en cuenta lo siguiente, las modificaciones que se realizan a las vistas:

- no pueden afectar a más de una tabla consultada. Pueden modificarse datos de una vista que combina varias tablas pero la modificación solamente debe afectar a una sola tabla.

- no se pueden cambiar los campos resultado de un cálculo.

- pueden generar errores si afectan a campos a las que la vista no hace referencia. Por ejemplo, si se ingresa un registro en una vista que consulta una tabla que tiene campos not null que no están incluidos en la vista.

- la opción "with check option" obliga a todas las instrucciones de modificación que se ejecutan en la vista a cumplir ciertos criterios que se especifican al definir la vista.

- para eliminar datos de una vista solamente UNA tabla puede ser listada en el "from" de la definicion de la misma.

Vistas modificar (alter view)

Para modificar una vista puede eliminarla y volver a crearla o emplear "alter view".

Con "alter view" se modifica la definición de una vista sin afectar los procedimientos almacenados y los permisos. Si elimina una vista y vuelve a crearla, debe reasignar los permisos asociados a ella.

Sintaxis básica para alterar una vista:

```
alter view NOMBREVISTA
 with encryption--opcional
as SELECT
```

En el ejemplo siguiente se altera vista_empleados para agregar el campo "domicilio":

```
alter view vista_empleados
 with encryption
as
 select (apellido+' '+e.nombre) as nombre,sexo,
  s.nombre as seccion, cantidadhijos,domicilio
  from empleados as e
  join secciones as s
  on codigo=seccion
```

Si creó la vista con "with encryption" y quiere modificarla manteniendo la encriptación, debe colocarla nuevamente, en caso de no hacerlo, desaparece.

Si crea una vista con "select *" y luego agrega campos a la estructura de las tablas involucradas, los nuevos campos no aparecerán en la vista; esto es porque los campos se seleccionan al ejecutar "create view"; debe alterar la vista.