Cursos SQL Server 2008 R2

Transact Vistas

Cursos SQL Server 2008 R2

Índice

Vistas

Una vista es una declaración SELECT con nombre, almacenada como un objeto en SQL Server.

Las vistas actúan como tablas virtuales y proporcionan algunos beneficios:

- Una vista proporciona a los desarrolladores un estándar de ejecución de consultas permitiendo escribir algunas consultas comunes como vistas e incluir las vistas en aplicaciones, por lo que todas las aplicaciones utilizan las mismas versiones de las consultas.
- Una vista con índices creados sobre ella, puede mejorar el rendimiento, sobre todo para consultas complejas.
- La mayoría de las vistas solo permiten operaciones de lectura en datos subyacentes pero es posible crear vistas modificables/actualizables que permiten al usuario modificar los datos a través de la vista.
- Vistas

Crear una VISTA

Usaremos el comando Transact-SQL:

CREATE VIEW CREATE VIEW [*schema_name .] view_name [(column [,...n])]*
[WITH <view_attribute> [,...n]]
AS *select_statement [;]*
[WITH CHECK OPTION]
<view_attribute> ::=
{
[ENCRYPTION]
[SCHEMABINDING]
[VIEW_METADATA] }
Es importante dar un nombre identificativo de la vista.

Vistas – Concepto

```
USE Northwind
GO
CREATE VIEW ClientesParis
AS
SELECT *
FROM Customers
WHERE city= 'Paris'
GO
SELECT * FROM ClientesParis
GO

SELECT a.CompanyName, b.OrderDate
FROM ClientesParis a
 INNER JOIN dbo.Orders b
ON a.CustomerID = b.CustomerID;
```

El optimizador encontrará la referencia a la vista y sustituirá la definición de la vista Reescribiendo el código:

```
SELECT a.CompanyName, b. OrderDate
FROM
 (
SELECT *
FROM Customers
WHERE city= 'Paris'
) as a
INNER JOIN dbo.Orders b ON a.CustomerID = b.CustomerID;
```

Vistas - Información

Visualizar información de una vista

USE Northwind

EXEC ***sp_help*** 'ClientesParis'

GO

Presentar la definición de la vista

USE Northwind

EXEC ***sp_helptext*** 'ClientesParis '

GO

Otras formas de obtener información

USE Northwind

SELECT *

FROM INFORMATION_SCHEMA.TABLES

WHERE table_name = 'ClientesParis'

SELECT *

FROM INFORMATION_SCHEMA.VIEWS

WHERE table_name = 'ClientesParis'

GO

Presentar una lista de todos los objetos a los que hace referencia la vista 'ClientesParis'

USE Northwind

sp_depends 'ClientesParis'

GO

La clausula AS se utiliza para especificar la cadena SELECT que define la vista. Se puede referenciar tanto a tablas, vistas, funciones definidas por el usuario(UDF) y funciones del sistema. La única restricción es que la sentencia SELECT **no** puede:

- Utilizar *COMPUTE o COMPUTE BY*
- Utilizar *INTO*
- Utilizar *OPTION*
- Referenciar a tablas temporales o variables de tipo
- Utilizar *ORDER BY* sin la clausula *TOP*
- Vistas

Una vista no puede contener una cláusula ORDER BY. Este es un truco para evitarlo.

```
USE Northwind
GO
CREATE VIEW ClientesPorNombre
AS
SELECT TOP 100 PERCENT *
FROM Customers
ORDER BY CompanyName
GO
```

Las vistas que usan el carácter * en la lista SELECT pueden causar problemas:

```
USE Northwind
ALTER TABLE Customers ADD Movil VARCHAR(12) NULL
GO
sp_depends 'ClientesParis'
GO
```

sp_refreshview actualiza la definición de la vista. ***sp_depends*** visualiza las dependencias de la vista.

```
USE Northwind
sp_refreshview 'ClientesParis‘
sp_depends 'ClientesParis '
GO
```

Vistas – Cláusula WITH

En la clausula WITH se podrán definir tres opciones diferentes a la vista:

- ENCRYPTION, SCHEMABINDING y VIEW_METADATA.
- ENCRYPTION encriptará la definición de la vista. Esta definición no estará visible para nadie, por lo que será importante guardar el código original en algún lugar ya que no se podrá desencriptar la definición.
- SCHEMABINDING. Si especificamos esta opción, no se podrán borrar las tablas vistas o funciones a las que haga referencia la vista sin antes borrar la vista.
- VIEW_METADATA devuelve a las aplicaciones cliente información de la vista, en lugar de información sobre la tabla base.

Vistas - ENCRYPTION

Encriptación del texto de la vista. Una vez encriptada no se puede desencriptar.

```
USE Northwind
GO
CREATE VIEW ClientesLondres
WITH ENCRYPTION
AS
SELECT *
FROM Customers
WHERE country = 'London'
GO
sp_helptext ' ClientesLondres '

GO
```

Vistas SHEMABINDING

Acopla la vista al esquema de la tabla subyacente. La tabla(s) base(s) no pueden ser modificadas. De esta forma se evita que pueda afectar el cambio a la definición de la vista.

```
USE adventureworks
GO
CREATE VIEW Ventas2001
WITH SCHEMABINDING
AS
SELECT SalesOrderID,OrderDate,TotalDue
FROM Sales.SalesOrderHeader
WHERE YEAR(OrderDate)=2001
GO
DROP VIEW Ventas2001
GO
```

Vistas actualizables

Son vistas que permiten la modificación de los datos en la tabla subyacente (INSERT, UPDATE, DELETE) a través de ella.

Limitaciones a la actualización:

- No serán actualizables si existen pseudocolumnas basadas en funciones agregadas.
- No serán actualizables si hay columnas derivadas de otras columnas o de operaciones que implican cálculos, por ejemplo SUBSTRING.
- Los cambios no pueden referenciar a columnas generadas con los operadores UNION, CROSSJOIN e INTERSECT.
- La definición de la vista no podrá contener clausulas GROUP BY, HAVING o DISTINCT.
- No se podrá utilizar TOP al especificar WITH CHECK OPTION.

Sí se puede llevar a cabo la actualización en el caso de que la orden afecte de forma exclusiva a una de las tablas intervinientes.

Si introducimos una clausula WHERE limitamos el rango de filas de la vista.

La cláusula WHERE no limita los cambios que los usuarios pueden realizar.

Para restringir esto cambios se utilizará en la definición de la vista el comando WITH CHECK OPTION.

Recomendación

Aunque es posible utilizar las vistas para la actualización de los datos, normalmente las vista no se utilizan con este propósito. Los procedimientos almacenados son una opción mejor .

Vistas actualizables - Ejemplo

Modificar datos a través de una vista

```
USE Northwind

UPDATE ClientesParis

SET Companyname = ‘Lasa y Asociados',

    Phone= ‘943-076545 '

WHERE customerid = ‘BERGS’'

GO
```

En ocasiones, interesa evitar la actualización de datos que no forman parte del rango de datos pertenecientes a la vista. Para resolver este problema utilizamos la cláusula WITH CHECK OPTION.

```
USE Northwind

GO
```

```
CREATE VIEW ClientesDeLondres

AS

SELECT *

FROM Customers

WHERE country = ‘Londonl'

WITH CHECK OPTION

GO

UPDATE ClientesDeLondres

SET country = 'USA'

WHERE customerid = 'WELLI'

GO
```

Modificación de Vistas

Cuando se modifica una vista, es preciso indicar la definición de la vista y sus opciones, ya que SQL Server sobrescribe las viejas opciones con las nuevas.

```
USE Northwind

GO

ALTER VIEW ClientesDeMexico

WITH ENCRYPTION, SCHEMABINDING

AS

SELECT customerid, companyname, contactname

FROM dbo.Customers

WHERE country = 'Mexico'

GO
```

Eliminación de Vistas

Se eliminan con la instrucción DROP (DDL)

```
USE Northwind

DROP VIEW ClientesDeBrasil

GO
```

Vistas indizadas

Las vistas indizadas funcionan de forma diferente a las vistas que hemos visto hasta ahora, ya que funcionan como una tabla (son similares a las vistas materializadas de Oracle).

Las consultas que se hacen a vistas indizadas devuelven los datos de la propia vista. El optimizador de la consulta no sustituye su definición en la consulta.

Antes de poder crear un índice agrupado en una vista, ésta debe cumplir los requisitos siguientes:

- Las opciones ANSI_NULLS y QUOTED_IDENTIFIER deben establecerse en ON cuando se ejecute la instrucción CREATE VIEW. La función OBJECTPROPERTY notifica esto a las vistas mediante las propiedades **ExecIsAnsiNullsOn** o **ExecIsQuotedIdentOn**.
- La opción ANSI_NULLS debe establecerse en ON para poder ejecutar todas las instrucciones CREATE TABLE, que crean tablas a las que se hace referencia en la vista.
- La vista no debe hacer referencia a ninguna otra vista, sólo a tablas base.
- Todas las tablas base a las que hace referencia la vista, deben estar en la misma base de datos que ésta y pertenecer al mismo propietario.
- Las funciones definidas por el usuario a las que se hace referencia en la vista se deben crear con la opción SCHEMABINDING.
- Es preciso utilizar nombres compuestos de dos partes en la vista para hacer referencia a las tablas y las funciones definidas por el usuario. No se permiten nombres de una, tres y cuatro partes.
- Todas las funciones a las que se hace referencia en las expresiones de la vista deben ser deterministas. La propiedad **IsDeterministic** de la función OBJECTPROPERTY notifica si una función definida por el usuario es determinista.
- Esta vista se debe crear con la opción SCHEMABINDING. El enlace de esquemas asocia la vista al esquema de las tablas base subyacentes.

Más información

C/ Miracruz, 10 (Bº de Gros) 20001 Donostia

Telf.: 943 275819

email: s e i m @ c e n t r o s e i m . c o m

S.E.I.M.