

Capítulo IX

Vistas

En este capítulo aprenderemos a construir vistas, un objeto de la base de datos que almacena una consulta predefinida. Las vistas ocultan la complejidad del modelo de datos y permiten que el usuario se enfoque en la data que es relevante para él, además de simplificar el diseño de consultas complejas y facilitar la administración de los permisos.

En ocasiones, una consulta compleja se puede simplificar si previamente se diseñan vistas que contienen consultas parciales que combinadas permiten llegar al resultado final requerido. Adicionalmente, las vistas mejoran el rendimiento de las consultas ya que la instrucción SELECT asociada se guarda compilada y con su plan de ejecución ya definido.

1. VISTA

Una **vista** es un objeto que almacena una consulta predefinida y que proporciona un modo alternativo de visualización de datos sin tener que redefinir la consulta. Las tablas requeridas en una vista se llaman tablas base. Con algunas excepciones, cualquier declaración SELECT puede nombrarse y guardarse como una vista. Los ejemplos comunes de vistas incluyen:

- Un subconjunto de filas o columnas de una tabla base.
- Una unión de dos o más tablas base.
- Un join de dos o más tablas base.
- Un resumen estadístico de una tabla base.
- Un subconjunto de otra vista, o alguna combinación de vistas y tablas base.

Ejercicio 9.1: Ejemplo de vista

En este ejercicio se crea la vista **v_ListaPrecios** en la base de datos **QhatuPERU**.

```
USE QhatuPERU
go

CREATE VIEW v_ListaPrecios
AS
SELECT LINEA.NomLinea, ARTICULO.CodArticulo,
       ARTICULO.DescripcionArticulo,
       ARTICULO.Presentacion,
       ARTICULO.PrecioProveedor
FROM LINEA INNER JOIN ARTICULO
      ON LINEA.CodLinea = ARTICULO.CodLinea
go

SELECT * FROM v_ListaPrecios
go
```

Resultados		Mensajes			
	NomLinea	CodArticulo	DescripcionArticulo	Presentacion	PrecioProveedor
1	GOLOSINAS	1	CARAMELOS BASTON VIENA ARCOR	PAQUETE 454 GR	1,50
2	GOLOSINAS	2	CARAMELOS SURTIDO DE FRUTAS	PAQUETE 450 GR	1,00
3	GOLOSINAS	3	CARAMELOS FRUTAS SURTIDA ARCOR	PAQUETE 520 GR	1,50
4	GOLOSINAS	4	CARAMELOS FRUTAS MASTICABLES	PAQUETE 454 GR	1,30
5	GOLOSINAS	5	CHUPETES LOLY AMBROSOLI	KILOGRAMO	1,20
6	GOLOSINAS	6	FRUNA SURTIDA DONOFRIO	PAQUETE X 24 UNIDADES	1,80
7	GOLOSINAS	7	CHOCOLATE DOÑA PEPA FIELD	PAQUETE X 6 UNIDADES	2,20
8	GOLOSINAS	8	CHOCOLATE CUA CUA FIELD	PAQUETE X 6 UNIDADES	1,60
9	GOLOSINAS	9	MELLOWS FAMILIAR FIELD	PAQUETE 454 GR	2,10
10	GOLOSINAS	10	WAFER CHOCOLATE FIELD	PAQUETE X 9 UNIDADES	0,70

Observe que para ejecutar la vista se utiliza la instrucción SELECT. Una vista se manipula como si fuera una tabla; es decir, que se le puede aplicar las declaraciones SELECT, INSERT, UPDATE y DELETE. Sin embargo, no debe ver la vista como una tabla, sino pensar en ella como un programa que ejecuta una sola instrucción (la instrucción SELECT).

2. CREACIÓN DE VISTAS – INSTRUCCIÓN CREATE VIEW

Sintáxis

```
CREATE VIEW nombre_vista [ ( lista_columnas ) ]
[ WITH ENCRYPTION, SCHEMABINDING ]
AS
sentencia_select
[ WITH CHECK OPTION ]
```

- WITH ENCRYPTION indica que se debe encriptar la sentencia con la que se define la vista, de modo tal que cuando se utiliza el procedimiento **sp_helptext** no se pueda observar la instrucción que creó la vista.
- WITH SCHEMABINDING enlaza la vista al esquema de las tablas subyacentes. Cuando se crea una vista con enlace de esquema, las tablas subyacentes no se pueden modificar ni eliminar.

- WITH CHECK OPTION indica que si la definición de la vista contiene la cláusula WHERE, y la vista se utiliza para operaciones de mantenimiento de datos, estas operaciones deben respetar la condición definida en el WHERE.

Las siguientes restricciones se aplican a la creación de vistas:

- No se puede definir una vista con ORDER BY o SELECT INTO.
- No pueden hacer referencia a tablas temporales o variables que hacen referencia a tablas.
- No pueden hacer referencia a más de 1024 columnas.
- La sentencia CREATE VIEW no puede combinarse con otras sentencias Transact-SQL en el mismo batch.

2.1. Consideraciones al crear las vistas

Tenga en cuenta lo siguiente al momento de crear una vista:

- Para ejecutar la declaración CREATE VIEW, debe ser miembro de alguno de los siguientes roles: **sysadmin**, **db_owner**, **db_ddladmin**, o debe tener el permiso CREATE VIEW. Además, también debe tener el permiso SELECT en todas las tablas o vistas que son referenciadas dentro de la vista.
- Para evitar situaciones en las que el dueño de una vista y el dueño de las tablas subyacentes son distintos, se recomienda que el usuario **dbo** tenga todos los objetos de una base de datos.
- Si la declaración SELECT de la vista contiene columnas computadas o constantes, éstas deben tener un nombre asignado.

Recomendación: Antes de crear la vista, es importante probar la declaración SELECT que define la vista para asegurar que el servidor devuelve el resultado esperado.

Ejercicio 9.2: Reporte que muestra el balance del almacén de QhatuPERU

Se desea obtener un reporte que muestre el balance entrada/salida de todos los artículos registrados en la base de datos QhatuPERU. El reporte debe mostrar para cada artículo: el código y la descripción del artículo, la cantidad de unidades entrantes en el almacén, y la cantidad de unidades salientes del almacén.

Este problema se resuelve fácilmente si crea una vista para las entradas, y una segunda vista para las salidas. Luego, debe combinar las vistas anteriores para obtener el reporte final.

Creación de la vista para el reporte de entradas

```
USE QhatuPERU
go

-- Vista para el listado de entradas
CREATE VIEW v_UnidadesEntrantes
AS
SELECT ARTICULO.CodArticulo,
       ARTICULO.DescripcionArticulo,
       ISNULL(SUM(
           ORDEN_DETALLE.CantidadRecibida ), 0)
       AS Entradas
FROM ARTICULO LEFT OUTER JOIN ORDEN_DETALLE
ON ARTICULO.CodArticulo =
    ORDEN_DETALLE.CodArticulo
GROUP BY ARTICULO.CodArticulo,
         ARTICULO.DescripcionArticulo
go

-- Ejecución de la vista de entradas
SELECT * FROM v_UnidadesEntrantes
go
```

Resultados

Mensajes

	CodArticulo	DescripcionArticulo	Entradas
1	1	CARAMELOS BASTON VIENA ARCOR	0
2	2	CARAMELOS SURTIDO DE FRUTAS	1500
3	3	CARAMELOS FRUTAS SURTIDA ARCOR	1250
4	4	CARAMELOS FRUTAS MASTICABLES	3000
5	5	CHUPETES LOLY AMBROSOLI	3000
6	6	FRUNA SURTIDA DONOFRIO	3000
7	7	CHOCOLATE DOÑA PEPA FIELD	3000
8	8	CHOCOLATE CUA CUA FIELD	3000
9	9	MELLOWS FAMILIAR FIELD	0
10	10	WAFER CHOCOLATE FIELD	3000

Creación de la vista para el reporte de salidas

```
-- Vista para el listado de salidas
CREATE VIEW v_UnidadesSalientes
AS
SELECT ARTICULO.CodArticulo,
       ARTICULO.DescripcionArticulo,
       ISNULL(SUM(
           GUIA_DETALLE.CantidadEnviada ), 0)
       AS Salidas
FROM ARTICULO LEFT OUTER JOIN GUIA_DETALLE
  ON ARTICULO.CodArticulo =
     GUIA_DETALLE.CodArticulo
GROUP BY ARTICULO.CodArticulo,
         ARTICULO.DescripcionArticulo
go

-- Ejecución de la vista de salidas
SELECT * FROM v_UnidadesSalientes
go
```

Resultados

Mensajes

	CodArticulo	DescripcionArticulo	Salidas
1	1	CARAMELOS BASTON VIENA ARCOR	400
2	2	CARAMELOS SURTIDO DE FRUTAS	400
3	3	CARAMELOS FRUTAS SURTIDA ARCOR	400
4	4	CARAMELOS FRUTAS MASTICABLES	400
5	5	CHUPETES LOLY AMBROSOLI	0
6	6	FRUNA SURTIDA DONOFRIO	600
7	7	CHOCOLATE DOÑA PEPA FIELD	500
8	8	CHOCOLATE CUA CUA FIELD	500
9	9	MELLOWS FAMILIAR FIELD	400
10	10	WAFER CHOCOLATE FIELD	400

Combinación de las vistas para obtener el reporte final

```
-- Combinación de las vistas para el balance
SELECT v_UnidadesEntrantes.CodArticulo,
       v_UnidadesEntrantes.DescripcionArticulo,
       v_UnidadesEntrantes.Entradas,
       v_UnidadesSalientes.Salidas
```

```

FROM v_UnidadesEntrantes
      INNER JOIN v_UnidadesSalientes
      ON v_UnidadesEntrantes.CodArticulo =
         v_UnidadesSalientes.CodArticulo
ORDER BY v_UnidadesEntrantes.CodArticulo
go

```

Resultados		Mensajes			
	CodArticulo	DescripcionArticulo	Entradas	Salidas	
1	1	CARAMELOS BASTON VIENA ARCOR	0	400	
2	2	CARAMELOS SURTIDO DE FRUTAS	1500	400	
3	3	CARAMELOS FRUTAS SURTIDA ARCOR	1250	400	
4	4	CARAMELOS FRUTAS MASTICABLES	3000	400	
5	5	CHUPETES LOLY AMBROSOLI	3000	0	
6	6	FRUNA SURTIDA DONOFRIO	3000	600	
7	7	CHOCOLATE DOÑA PEPA FIELD	3000	500	
8	8	CHOCOLATE CUA CUA FIELD	3000	500	
9	9	MELLOWS FAMILIAR FIELD	0	400	
10	10	WAFER CHOCOLATE FIELD	3000	400	

2.2. Ventajas del uso de las vistas

El uso de las vistas proporciona las siguientes ventajas:

- El usuario accede a la data importante o apropiada para él. Limita el acceso a datos sensibles.
- Oculta la complejidad del modelo de datos. Un join de múltiples tablas se convierte en un simple SELECT para el usuario.
- Desde el punto de vista del usuario, una vista es una "tabla" pues puede ejecutar en ella todas las operaciones de datos: SELECT, INSERT, UPDATE y DELETE.
- Debido a que una vista es un objeto de la base de datos, puede asignarle permisos de usuario. Esto es mucho más eficiente que colocar los mismos permisos sobre columnas individuales en una tabla.
- Los datos pueden exportarse desde una vista por medio de la utilidad **bcp**.

2.3. Averiguar cuál es la consulta que ejecuta una vista

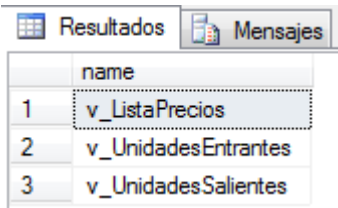
Para averiguar cómo está definida una vista, puede ejecutar los siguientes procedimientos almacenados del sistema para consultar las siguientes tablas del catálogo de una base de datos:

Vista del sistema	Almacena	Procedimiento
sys.sysobjects	Nombre de la vista	<code>sp_help nombre_vista</code>
sys.sysdepends	Nombre de los objetos dependientes de la vista	<code>sp_depends nombre_vista</code>
sys.syscomments	Sentencia que definió la vista	<code>sp_helptext nombre_vista</code>
sys.syscolumns	Columnas definidas en la vista	<code>sp_columns nombre_vista</code>

Ejercicio 9.3: Consultando la metadata de una vista

```
USE QhatuPERU
go

-- Consultado que objetos de tipo Vista contiene
-- la base de datos QhatuPERU
SELECT name FROM sys.sysobjects
      WHERE type = 'V'
go
```



Resultados		Mensajes	
	name		
1	v_ListaPrecios		
2	v_UnidadesEntrantes		
3	v_UnidadesSalientes		

```
-- Obteniendo las propiedades
-- de la vista v_ListaPrecios
sp_help v_ListaPrecios
go
```


Resultados		Mensajes							
	Name	Owner	Type	Created_datetime					
1	v_ListaPrecios	dbo	view	2013-05-16 14:53:37.080					
	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource
1	NomLinea	varchar	no	20			no	no	no
2	CodArticulo	int	no	4	10	0	no	(n/a)	(n/a)
3	DescripcionArticulo	varchar	no	40			no	no	no
4	Presentacion	varchar	no	30			yes	no	yes
5	PrecioProveedor	money	no	8	19	4	yes	(n/a)	(n/a)
Identity		Seed	Increment	Not For Replication					
1	No identity column defined.	NULL	NULL	NULL					
RowGuidCol									
1	No rowguidcol column defined.								

```
-- Mostrando el código que ejecuta la vista
sp_helptext v_ListaPrecios
go
```

Resultados		Mensajes	
	Text		
1			
2	CREATE VIEW v_ListaPrecios		
3	AS		
4	SELECT LINEA.NomLinea, ARTICULO.CodArticulo,		
5	ARTICULO.DescripcionArticulo,		
6	ARTICULO.Presentacion,		
7	ARTICULO.PrecioProveedor		
8	FROM LINEA INNER JOIN ARTICULO		
9	ON LINEA.CodLinea = ARTICULO.CodLinea		

```
-- Mostrando los objetos que dependen de la vista
sp_depends v_ListaPrecios
go
```

Resultados		Mensajes			
	name	type	updated	selected	column
1	dbo.LINEA	usertable	no	yes	CodLinea
2	dbo.LINEA	usertable	no	yes	NomLinea
3	dbo.ARTICULO	usertable	no	yes	CodArticulo
4	dbo.ARTICULO	usertable	no	yes	CodLinea
5	dbo.ARTICULO	usertable	no	yes	DescripcionArticulo
6	dbo.ARTICULO	usertable	no	yes	Presentacion
7	dbo.ARTICULO	usertable	no	yes	PrecioProveedor

3. MODIFICACIÓN Y ELIMINACIÓN DE UNA VISTA

A menudo, en respuesta a las demandas de los usuarios por información adicional, o a los cambios en la definición de la tabla subyacente, es necesario modificar la definición de una vista. Por ejemplo, si la tabla a la que una vista hace referencia se ha eliminado, los usuarios recibirán un mensaje del error cuando ellos intenten utilizar la vista. Se puede modificar una vista eliminándola y luego recreándola, o ejecutando la sentencia ALTER VIEW.

3.1. Modificación de una vista – La instrucción ALTER VIEW

La sentencia ALTER VIEW cambia la definición de una vista permitiéndole retener los permisos para la vista. Esta sentencia está sujeta a las mismas restricciones que la sentencia CREATE VIEW. Si en vez de modificar la vista, la elimina y luego la recrea, se verá obligado a recrear los permisos.

Sintaxis

```
ALTER VIEW nombre_vista [ ( lista_columnas ) ]
[ WITH ENCRYPTION ]
AS
nueva_sentencia_select
[ WITH CHECK OPTION ]
```

3.2. Eliminación de una vista – La instrucción DROP VIEW

La siguiente sentencia se utiliza para eliminar una vista.

Sintáxis

```
DROP VIEW nombre_vista
```

3.3. Ocultando la definición de una vista

El texto de la sentencia que define una vista se almacena en la tabla de sistema **syscomments**. Si al crear la vista utiliza la opción WITH ENCRYPTION, el texto se almacenará en dicha tabla en forma encriptada.

Recomendación

Antes de encriptar una vista, asegúrese que la definición de vista (el script) se guarde en un archivo. Para desencriptar el texto de una vista, debe eliminar y luego recrear o modificar la vista utilizando la sintáxis original.

Ejercicio 9.4: Uso de WITH ENCRYPTION

La siguiente instrucción crea la vista **v_Inventario** cuya definición se guarda encriptada.

```
USE QhatuPERU
go

CREATE VIEW v_Inventario
    WITH ENCRYPTION
AS
SELECT LINEA.NomLinea, ARTICULO.CodArticulo,
    ARTICULO.DescripcionArticulo,
    ARTICULO.Presentacion,
    ARTICULO.stockActual, ARTICULO.stockMinimo
FROM LINEA INNER JOIN ARTICULO
    ON LINEA.CodLinea = ARTICULO.CodLinea
go

-- Consultando la vista
SELECT * FROM v_Inventario
go
```

Resultados		Mensajes					
	NomLinea	CodArticulo	DescripcionArticulo	Presentacion	stockActual	stockMinimo	
1	GOLOSINAS	1	CARAMELOS BASTON VIENA ARCOR	PAQUETE 454 GR	200	50	
2	GOLOSINAS	2	CARAMELOS SURTIDO DE FRUTAS	PAQUETE 450 GR	300	50	
3	GOLOSINAS	3	CARAMELOS FRUTAS SURTIDA ARCOR	PAQUETE 520 GR	250	50	
4	GOLOSINAS	4	CARAMELOS FRUTAS MASTICABLES	PAQUETE 454 GR	250	50	
5	GOLOSINAS	5	CHUPETES LOLY AMBROSOLI	KILOGRAMO	150	100	
6	GOLOSINAS	6	FRUNA SURTIDA DONOFRIO	PAQUETE X 24 UNIDADES	500	100	
7	GOLOSINAS	7	CHOCOLATE DOÑA PEPA FIELD	PAQUETE X 6 UNIDADES	500	100	
8	GOLOSINAS	8	CHOCOLATE CUA CUA FIELD	PAQUETE X 6 UNIDADES	500	100	
9	GOLOSINAS	9	MELLOWS FAMILIAR FIELD	PAQUETE 454 GR	100	50	
10	GOLOSINAS	10	WAFER CHOCOLATE FIELD	PAQUETE X 9 UNIDADES	900	100	

Ahora, trataremos de averiguar cómo se ha definido la vista.

```
-- Revisando la definición de la vista
sp_helptext v_Inventario
go
```

Se obtiene el siguiente mensaje:

```
El texto para el objeto v_inventario está cifrado.
```

3.4. Modificación de datos a través de vistas

Desde el punto de vista del usuario, una vista es una "tabla" ya que él puede ejecutar sobre la vista las sentencias SELECT, INSERT, UPDATE y DELETE. Estas sentencias ejecutadas sobre la vista afectan a las tablas dependientes de ella.

Deben tenerse en cuenta las siguientes consideraciones al momento de crear las vistas que luego se utilizarán para modificar los datos a través de ellas:

- La modificación no puede afectar a más de una de las tablas dependientes.
- Las vistas con columnas computadas deben ser de solo lectura ya que producen error cuando se trata de ejecutar modificaciones a través de ellas.
- Las columnas no nulas que no son referenciadas en la vista pueden producir error cuando se ejecuta una modificación a través de la vista.

- Si el SELECT asociado a la vista ha sido definido con una cláusula WHERE, el uso de la opción WITH CHECK OPTION al momento de crear la vista hará que las modificaciones que se hagan a través de ella respeten el criterio del WHERE.

Teniendo en cuenta todo lo anterior, mi recomendación es que las vistas deben utilizarse solo como medios de consulta (solo para ejecutarles SELECT). Para modificar los datos es mejor utilizar procedimientos almacenados.

4. EJERCICIOS PROPUESTOS

Para los siguientes ejercicios debe utilizar las bases de datos RH y EDUCA. Los scripts para crearlas en su servidor los encontrará en el CD que acompaña al libro.

1. En la base de datos RH, crear una vista que muestre los siguientes datos:
 - Código del empleado
 - Nombre del empleado
 - Departamento
 - Cargo
 - Sueldo
 - Comisión, cero si es nulo
 - Sueldo total (sueldo + comisión)
2. En la base de datos EDUCA, crear una vista que muestre los siguientes datos:
 - Código del curso
 - Nombre del curso
 - Código del alumno
 - Nombre del alumno
 - Precio del curso
 - Pago a cuenta
 - Pago pendiente