


### 7.3 Vistas (tablas virtuales) en SQL

#### 7.3.1 Concepto de una vista en SQL

Una vista en la terminología de SQL es una tabla única que se deriva de otras tablas. Estas otras tablas pueden ser tablas base o vistas previamente definidas. Una vista no existe necesariamente en forma física; se considera una tabla virtual, a diferencia de las tablas base, cuyas tuplas siempre se almacenan físicamente en la base de datos. Esto limita las posibles operaciones de actualización que se pueden aplicar a las vistas, pero no proporciona ninguna limitación para consultar una vista. Podemos pensar en una vista como una forma de especificar una tabla a la que necesitamos hacer referencia con frecuencia, aunque no exista físicamente.

#### 7.3.2 Especificación de vistas en SQL

En SQL, el comando para especificar una vista es **CREATE VIEW**. La vista recibe un nombre de tabla (virtual) (o nombre de vista), una lista de nombres de atributos y una consulta para especificar el contenido de la vista. Si ninguno de los atributos de vista resulta de la aplicación de funciones u operaciones aritméticas, no tenemos que especificar nuevos nombres de atributo para la vista, ya que serían los mismos que los nombres de los atributos de las tablas de definición en el caso predeterminado.

V1:	<b>CREATE VIEW</b>	<b>WORKS_ON1</b>
	<b>AS SELECT</b>	Fname, Lname, Pname, Hours
	<b>FROM</b>	EMPLOYEE, PROJECT, WORKS_ON
	<b>WHERE</b>	Ssn = Essn <b>AND</b> Pno = Pnumber;
V2:	<b>CREATE VIEW</b>	<b>DEPT_INFO(Dept_name, No_of_ems, Total_sal)</b>
	<b>AS SELECT</b>	Dname, <b>COUNT (*)</b> , <b>SUM (Salary)</b>
	<b>FROM</b>	DEPARTMENT, EMPLOYEE
	<b>WHERE</b>	Dnumber = Dno
	<b>GROUP BY</b>	Dname;

### WORKS\_ON1

Fname	Lname	Pname	Hours
-------	-------	-------	-------

### DEPT\_INFO

Dept_name	No_of_emps	Total_sal
-----------	------------	-----------

En V1, no especificamos ningún nombre de atributo nuevo para la vista WORKS\_ON1 (aunque podríamos haberlo hecho); en este caso, WORKS\_ON1 hereda los nombres de los atributos de vista de las tablas de definición EMPLOYEE, PROJECT y WORKS\_ON. View V2 especifica explícitamente nuevos nombres de atributos para la vista DEPT\_INFO, utilizando una correspondencia uno a uno entre los atributos especificados en la cláusula CREATE VIEW y los especificados en la cláusula SELECT de la consulta que define la vista. Ahora podemos especificar consultas SQL en una vista, o tabla virtual, de la misma manera que especificamos consultas que involucran tablas base. Por ejemplo, para recuperar el apellido y el nombre de todos los empleados que trabajan en el proyecto "ProductX", podemos utilizar la vista WORKS\_ON1 y especificar la consulta como en QV1:

```
QV1:      SELECT      Fname, Lname
           FROM        WORKS_ON1
           WHERE        Pname = 'ProductX';
```

La misma consulta requeriría la especificación de dos combinaciones si se especifica en las relaciones base directamente; una de las principales ventajas de una vista es simplificar la especificación de determinadas consultas. Las vistas también se utilizan como mecanismo de seguridad y autorización.

Se supone que una vista debe estar siempre actualizada; si modificamos las tuplas en las tablas base en las que se define la vista, la vista debe reflejar automáticamente estos cambios. Por lo tanto, la vista no tiene que realizarse o materializarse en el momento de la definición de la vista, sino en el momento en que especificamos una consulta en la vista. Es responsabilidad del DBMS y no del usuario asegurarse de que la vista se mantenga actualizada. Si ya no necesitamos una vista, podemos usar el comando DROP VIEW para deshacernos de ella.

### 7.3.3 Implementación de vista, actualización de vista y vistas en línea

El problema de cómo un DBMS puede implementar de manera eficiente una vista para realizar consultas de manera eficiente es complejo. Se han sugerido dos enfoques principales. Una estrategia, llamada modificación de consulta, implica modificar o transformar la consulta de vista (enviada por el usuario) en una consulta en las tablas base subyacentes. Por ejemplo, el DBMS modificaría automáticamente la consulta QV1 a la siguiente consulta:

```
SELECT      Fname, Lname
FROM        EMPLOYEE, PROJECT, WORKS_ON
WHERE       Ssn = Essn AND Pno = Pnumber
           AND Pname = 'ProductX';
```

La desventaja de este enfoque es que es ineficaz para las vistas definidas a través de consultas complejas que requieren mucho tiempo de ejecución.

La segunda estrategia, denominada materialización de la vista, implica la creación física de una tabla de vista temporal o permanente cuando la vista se consulta o crea por primera vez y mantiene esa tabla en el supuesto de que seguirán otras consultas en la vista. En este caso, se debe desarrollar una estrategia eficiente para actualizar automáticamente la tabla de la vista cuando se actualizan las tablas base para mantener la vista actualizada.

Se han desarrollado técnicas que utilizan el concepto de actualización incremental para este propósito, donde el DBMS puede determinar qué nuevas tuplas deben insertarse, eliminarse o modificarse en una tabla de vista materializada cuando se aplica una actualización de base de datos a una de las tablas base definitorias. La vista generalmente se mantiene como una tabla materializada (almacenada físicamente) mientras se esté consultando. Si no se consulta la vista durante un cierto período de tiempo, el sistema puede eliminar automáticamente la tabla física y volver a calcularla desde cero cuando las consultas futuras hagan referencia a la vista. Son posibles diferentes estrategias en cuanto a cuándo se actualiza una vista materializada.

La estrategia de actualización inmediata actualiza una vista tan pronto como se cambian las tablas base; la estrategia de actualización diferida actualiza la vista cuando lo necesita una consulta de vista; y la estrategia de actualización periódica actualiza la vista periódicamente (en la última estrategia, una consulta de vista puede obtener un resultado que no está actualizado). Un usuario siempre puede emitir una consulta de recuperación en cualquier vista. Sin embargo, en muchos casos **no es posible emitir un comando INSERT, DELETE o UPDATE en una tabla de vista**. En general, una actualización en una vista definida en una sola tabla sin ninguna función agregada se puede asignar a una actualización en la tabla base subyacente bajo ciertas condiciones.

Para una vista que involucra combinaciones, una operación de actualización se puede mapear para actualizar operaciones en las relaciones bases subyacentes de múltiples formas. Por lo tanto, a menudo no es posible que el DBMS determine cuál de las actualizaciones está prevista. Para ilustrar posibles problemas con la actualización de una vista definida en varias tablas, considere la vista WORKS\_ON1 y suponga que emitimos el comando para actualizar el atributo PNAME de "John Smith" de "ProductX" a "ProductY". Esta actualización de vista se muestra en UV1.

```
UV1:  UPDATE WORKS_ON1
      SET      Pname = 'ProductY'
      WHERE    Lname = 'Smith' AND Fname = 'John'
            AND Pname = 'ProductX';
```

Esta consulta se puede mapear en varias actualizaciones en las relaciones base para dar el efecto de actualización deseado en la vista. Además, algunas de estas actualizaciones crearán efectos secundarios adicionales que afectarán el resultado de otras consultas. Por ejemplo, aquí hay dos posibles actualizaciones, (a) y (b), en las relaciones base correspondientes a la operación de actualización de vista en UV1:

```
(a):  UPDATE WORKS_ON
      SET      Pno = ( SELECT Pnumber
                      FROM    PROJECT
                      WHERE    Pname = 'ProductY' )
      WHERE    Essn IN ( SELECT Ssn
                        FROM    EMPLOYEE
                        WHERE    Lname = 'Smith' AND Fname = 'John' )
            AND Pno = ( SELECT Pnumber
                      FROM    PROJECT
                      WHERE    Pname = 'ProductX' );

(b):  UPDATE PROJECT  SET      Pname = 'ProductY'
      WHERE    Pname = 'ProductX';
```

La actualización (a) relaciona "John Smith" con la tupla "ProductY" PROJECT en lugar de la tupla "ProductX" PROJECT y es la actualización deseada más probable. Sin embargo, (b) también daría el efecto de actualización deseado en la vista, pero lo logra cambiando el nombre de la tupla "ProductX" en la relación PROJECT a "ProductY". Es muy poco probable que el usuario que especificó la actualización de vista UV1 quiera que la actualización sea interpretado como en (b), ya que también tiene el efecto secundario de cambiar todas las tuplas de vista con Pname = "ProductX". Es posible que algunas actualizaciones de vista no tengan mucho sentido; por ejemplo, modificar el

atributo Total\_sal de la vista DEPT\_INFO no tiene sentido porque Total\_sal se define como la suma de los salarios de los empleados individuales. Esta solicitud incorrecta se muestra como UV2:

```
UV2:      UPDATE    DEPT_INFO
          SET       Total_sal = 100000
          WHERE     Dname = 'Research';
```

Generalmente, una actualización de la vista es factible cuando solo una posible actualización de las relaciones base puede realizar la operación de actualización deseada en la vista. Siempre que una actualización de la vista se pueda asignar a más de una actualización de las relaciones base subyacentes, normalmente no se permite. Algunos investigadores han desarrollado métodos para elegir la actualización más probable, mientras que otros investigadores prefieren que el usuario elija el mapeo de actualización deseado durante la definición de la vista. Pero estas opciones generalmente no están disponibles en la mayoría de los DBMS comerciales. En resumen, podemos hacer las siguientes observaciones:

- Una vista con una única tabla de definición es actualizable si los atributos de la vista contienen la clave principal de la relación base, así como todos los atributos con la restricción NOT NULL que no tienen valores predeterminados especificados.
- Las vistas definidas en varias tablas que utilizan combinaciones generalmente no son actualizables.
- Las vistas definidas mediante funciones de agrupación y agregación no se pueden actualizar.

En SQL, la cláusula WITH CHECK OPTION debe agregarse al final de la definición de vista si una vista se va a actualizar mediante instrucciones INSERT, DELETE o UPDATE. Esto permite que el sistema rechace operaciones que violen las reglas de SQL para ver actualizaciones.

También es posible definir una tabla de vista en la cláusula FROM de una consulta SQL. Esto se conoce como vista en línea. En este caso, la vista se define dentro de la propia consulta.

#### **7.3.4 Vistas como mecanismos de autorización**

Usar las vistas para ocultar ciertos atributos o tuplas de usuarios no autorizados. Supongamos que cierto el usuario solo puede ver la información de los empleados que trabajan para el departamento 5; luego podemos crear la siguiente vista DEPT5EMP y otorgarle al usuario el privilegio de consultar la vista pero no la tabla base EMPLOYEE en sí. Este usuario solo podrá recuperar la información de los

empleados para las tuplas de empleados cuyo Dno = 5, y no podrá ver otras tuplas de empleados cuando se consulte la vista.

```
CREATE VIEW      DEPT5EMP      AS  
SELECT          *  
FROM            EMPLOYEE  
WHERE           Dno = 5;
```

De manera similar, una vista puede restringir a un usuario para que solo vea ciertas columnas; por ejemplo, solo el nombre, apellido y dirección de un empleado pueden ser visibles como sigue:

```
CREATE VIEW      BASIC_EMP_DATA  AS  
SELECT          Fname, Lname, Address  
FROM            EMPLOYEE;
```

Por lo tanto, al crear una vista adecuada y otorgar a ciertos usuarios acceso a la vista y no a las tablas base, estarían restringidos a recuperar solo los datos especificados en la vista.0020