

TRATAMIENTO DE VISTAS EN MYSQL

Vamos a mostrar dos tipos de objetos de uso habitual en cualquier SGBD: las vistas y las tablas temporales.

Mientras que las primeras son objetos persistentes que nos permiten resumir consultas complejas y de uso frecuente, las segundas, como su nombre indica, son objetos de corta vida, durante una sesión o una simple consulta.

VISTAS

Una vista es un objeto que se define a partir de una consulta y que se comporta como una tabla si bien, dependiendo de la consulta en la que se basa, se pueden hacer más o menos cosas: consultar una vista siempre será posible pero insertar o borrar filas en una vista o modificar un valor ya depende de cómo sea esa definición.

El objetivo / utilidad de las vistas es doble. Por un lado seguridad ya que restringe los datos a una vista parcial y por otra de facilitación de consultas ya que son definiciones creadas que no hay que volver a codificar.

Para poder insertar es necesario que la vista dependa de una sola tabla y que no tengamos campos agrupados o calculados en ella. Por supuesto debe incluir todos los campos que sean obligatorios y no tengan un valor por defecto.

Para poder actualizar es preciso que no actualicemos campos calculados ni agrupados ya que esto conlleva un cálculo que la vista no es capaz de hacer.

El caso del borrado de filas es similar, debe ser coherente con la definición que hemos creado.

Más información:

<http://dev.mysql.com/doc/refman/5.0/en/view-updatability.html>

Para crear una vista disponemos de la orden:

CREATE VIEW nombrevista AS consulta

Una vista es un objeto persistente, por lo tanto, para eliminarla del catálogo, hay que ejecutar:

DROP VIEW nombrevista

Lo que se almacena de la vista es su definición, podemos encontrar las vistas creadas y cómo se han creado en **information_schema (diccionario de datos) en la tabla VIEWS**.

También se puede consultar la definición de una vista ya definida (aunque esto ya es particular de MySQL, en otros gestores tienen sus propios métodos):

SHOW CREATE VIEW nombrevista

Toda la información sobre vistas en MySQL se puede consultar en:

<http://dev.mysql.com/doc/refman/5.0/en/views-table.html>

<http://dev.mysql.com/doc/refman/5.0/en/create-view.html>

EJEMPLOS VISTAS DE BD EMPLEADOS2.SQL

1/ -- Crear una vista para listar los empleados de emple que tengan el mayor sueldo

1.b/ Visualizar

1.c/ Consultar la definición de una vista ya definida

1.d/ ¿Podrá actualizarse dicha vista o insertar nuevos elementos en la misma?

a/

```
mysql> create view empMasSueldo as select emp_no,nombre,apellido,salario,dept_no
from emple where salario >=ALL(select max(salario) from emple);
```

b/ `select * from empMasSueldo;`

c/ **SHOW CREATE VIEW** empMasSueldo; (En mysql)

d/ No puede actualizarse, ni insertar nuevos registros.

2/ Crear Vista que nos almacene los empleados de los departamento 10.

```
CREATE VIEW EmpleDep10
AS SELECT * FROM EMPLE WHERE dept_no=10;
```

Una vista se comporta como una tabla y puede consultarse.

SELECT NOMBRE, APELLIDOS, dept_no FROM EMPLEDEP10;

Podemos insertar registros si no hay ninguna de las restricciones que nos lo impida.

```
insert into empledep10 (emp_no,nombre,dept_no) values (7,'Paloma',10);
```

```
select * from empledep10;
```

```
insert into empledep10 (emp_no,nombre,dept_no) values (17,'Pedro',30);
```

```
select * from empledep10;
```

Comprobamos que no se ve con la vista, pero si con la tabla base, ESTO PODRÍAMOS EVITARLO CON LA OPCIÓN WITH CHECK OPTION.

Podemos consultar cómo se ha creado:

SHOW CREATE VIEW EMPLEDEP10; (En mysql)

Vamos a borrar la vista

DROP VIEW EMPLEDEP10;

Una vista se puede generar a partir de cualquier consulta, y tiene la característica añadida de poder restringir el acceso a solo un subconjunto de las filas posibles.

Por tanto puedo crear vistas de consultas muy complejas dándoles la apariencia de una tabla sencilla.

3/ Crear una vista DEPARTNUM que muestre el nombre del departamento y su número de empleados (alias numemple) ordenada por departamento.

```
DROP VIEW IF EXISTS DEPARTNUM;  
CREATE OR REPLACE VIEW DEPARTNUM AS SELECT DNOMBRE,  
D.DEPT_NO, COUNT(EMP_NO) "NUMEMPLE" FROM EMPLE E, DEPART  
D WHERE E. DEPT_NO=D. DEPT_NO GROUP BY DNOMBRE ORDER BY  
1;
```

B/ Consulta la vista creada

```
SELECT * FROM DEPARTNUM;
```

C/ Consulta cómo se ha creado (En mysql)

```
SHOW CREATE VIEW DEPARTNUM;
```

D/ Inserta un nuevo empleado en el departamento 3 en la tabla empleados
Consulta de nuevo la vista

```
INSERT INTO EMPLE  
(EMP_NO,NOMBRE,APELLIDO,SALARIO,FECHA_ALT,DEPT_NO,OFICIO)  
VALUES (18, 'NUEVO','NUEVO',300000,'1990-1-1',30,'JÉFE');  
SELECT * FROM DEPARTNUM;
```

WITH CHECK OPTION

Esta opción al crear la vista restringe la inserción de datos y la actualización sino se cumple el where que hemos puesto

4/ Crear de nuevo la vista EMPLEDEP10 con WITH CHECK OPTION

```
CREATE VIEW EMPLEDEP10  
AS SELECT * FROM EMPLEADOS WHERE ID_DEPARTAMENTO=10 WITH  
CHECK OPTION;
```

Al insertar los datos comprueba que se cumple el where:

```
INSERT INTO EMPLEDEP10  
(emp_no,nombre,apellido,salario,fecha_alt,dept_no, oficio) values (19,  
'Jose','Del Rio',300000,'1995-12-02',30, 'desarrollador');
```

Comprobamos que da el siguiente error: "ERROR 1369 (HY000): CHECK OPTION failed 'empleados2.empledep10' " y además no se inserta en la tabla base.

Consulta la vista creada

```
SELECT * FROM EMPLEDEP10;
```

Consulta cómo se ha creado (En mysql)

```
SHOW CREATE VIEW EMPLEDEP10;
```

Inserta un nuevo empleado en el departamento 10 en la propia vista

```
INSERT INTO EMPLEDEP10  
(emp_no,nombre,apellido,salario,fecha_alt,dept_no, oficio) values (19,  
'Jose','Del Rio',300000,'1995-12-02',10, 'desarrollador');
```

Consulta de nuevo la vista

```
SELECT * FROM EMPLEDEP1;
```

TABLAS TEMPORALES

Se entiende por tabla temporal aquella que se crea y utiliza en un contexto limitado, bien sea una orden concreta de SQL (el caso genérico de las subconsultas) o una sesión o conexión.

Las tablas TEMPORARY, por ejemplo, son objetos que desaparecen automáticamente cuando se cierra la sesión de usuario.

Una subconsulta es accesible solo mientras se ejecuta una determinada orden.

La sintaxis es CREATE TEMPORARY table AS select.

5/ Crear una tabla temporal PUESTOS con el nombre, apellidos y puesto de los empleados ordenados por puesto.

```
CREATE TEMPORARY TABLE TABLA1 AS SELECT NOMBRE, APELLIDO,  
oficio FROM EMPL;
```

Salir de la sesión y al volver a entrar comprobar que no está

EJERCICIOS:

1. Crear una vista VISTA1 con los datos de los empleados que son jefes

```
CREATE OR REPLACE VIEW VISTA1 AS SELECT * FROM emple WHERE  
oficio like '%PRESIDENTE%';
```

-Comprobar si es insertable/actualizable con un ejemplo. - Explicar por qué sí o no

```
Insert into vista1 (emp_no,nombre,apellido, dept_no, oficio) values (20,'Pepe','Rodriguez',30,'Jefe');
```

En este caso al cumplir la condición de creación de la vista se inserta, porque sólo afecta a una tabla y no tiene ninguna de las restricciones que impiden actualizar vistas.

```
Insert into vista1 (emp_no,nombre,apellido, dept_no, oficio) values (21,'Luis','Rodriguez',30,'Ventas');
```

En este caso al no cumplir la condición de creación de la vista no se inserta en la vista, pero si en la tabla porque no está la cláusula WITH CHECK OPTION.

2. Crear una VISTA2 con los datos de los empleados (nombre, apellido, emp_no, dept_no) de los departamentos cuyo nombre contiene Recursos

```
CREATE OR REPLACE VIEW VISTA2 AS SELECT e.nombre,e.apellido,e.dept_no FROM emple e, depart d WHERE e.dept_no=d.dept_no and nombred like "%recursos%";
```

3. Crear una VISTA3 con los datos de los departamentos y su media de sueldo

```
CREATE OR REPLACE VIEW VISTA3 AS SELECT dnombre, avg(sueldo) "mediasueldo" FROM emple e, depart d WHERE e.dept_no=d.dept_no group by d.dept_no;
```

-Comprobar si es insertable/actualizable con un ejemplo

- Explicar por qué sí o no

No, porque está AVG(sueldo) y es una función de grupo
Por ejemplo, da error:

```
delete from vista3 where dnombre like 'cont%';  
ERROR 1288 (HY000): The target table vista3 of the DELETE is not updatable  
update vista3 set nombre="Ventas Nuevas" where dept_no=30;  
ERROR 1288 (HY000): The target table vista3 of the UPDATE is not updatable
```

4. Crear una VISTA4 con los datos de los departamentos que no tienen empleados

```
CREATE VIEW VISTA4 AS SELECT * FROM DEPART WHERE dept_no NOT IN (SELECT dept_no FROM EMPLE);
```

--OTRA--

```
CREATE OR REPLACE VIEW VISTA4B AS SELECT D.* FROM DEPART D LEFT OUTER JOIN EMPLE E ON E.dept_no=D.DEPT_NO WHERE E.DEPT_NO IS NULL;
```

--OTRA--



```
CREATE OR REPLACE VIEW VISTA4C AS SELECT D.dept_no,dnombre,numce  
FROM DEPART D LEFT OUTER JOIN EMPLE E ON E.dept_no=D.DEPT_NO  
GROUP BY D.DEPT_NO,dnombre,numce HAVING COUNT(E.DEPT_NO)=0;
```

-Comprobar si es insertarble/actualizable con un ejemplo, - Explicar por qué sí o no

La primera permite hacer inserciones y actualizaciones:

```
INSERT INTO VISTA4 (DEPT_NO,DNOMBRE,NUMCE,DIREC) VALUES  
(50,'INFORMATICA',5,1);
```

```
update VISTA4 set DIREC=8 where direc is null;
```

En el caso de la primera es actualizable, la segunda no y la tercera no porque tiene un COUNT,

5. Crear una VISTA5 con los datos del departamento que tiene mayor suma de sueldos

```
CREATE OR REPLACE VIEW VISTA5 AS SELECT D.dept_no, dnombre,  
SUM(SALARIO) suma FROM DEPART D, EMPLE E WHERE  
D.DEPT_NO=E. DEPT_NO group by D.dept_no, dnombre HAVING  
SUM(SALARIO) >=all (SELECT sum(SALARIO) FROM DEPART D,  
EMPLE E WHERE D. DEPT_NO=E. DEPT_NO group by D.dept_no);
```

Para comprobar, visualizar la suma de sueldos por departamento:

```
SELECT D.DEPT_NO,SUM(SALARIO) FROM DEPART D, EMPLE E WHERE  
D.DEPT_NO=E.DEPT_NO group by D.Dept_no;
```

-Comprobar si es insertarble/actualizable con un ejemplo. Explicar por qué sí o no

No es insertable, ni actualizable porque tiene un group by

6. Crear una VISTA6 con los datos del departamento que tiene más de un jefe

```
create or replace view vista6 as select dnombre from depart d, emple e  
where e.dept_no=d.dept_no and lower(oficio) like 'presidente' group by  
dnombre having count(*) >1;
```

--OTRA Usando la vista1:

```
CREATE VIEW VISTA6b AS SELECT Dnombre FROM DEPART D, VISTA1  
V WHERE D.DEPT_NO =V.DEPT_NO group by D.dNOMBRE HAVING  
count(*) >1;
```

-Comprobar si es insertarble/actualizable con un ejemplo. Explicar por qué sí o no

NO, PORQUE TIENE UN GROUP BY Y UN COUNT