**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 facilitar 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/ mysql> -- 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/

Create or replace view VISTA1 as select emp\_no, nombre, salario, dept\_no from emple where salario = (select max(salario) from emple) with check option;

select \* from VISTA1;

Query OK, 0 rows affected (0.01 sec)

b/

mysql> select \* from empMasSueldo;

| emp\_no | nombre | apellido | salario | dept\_no |

| 7839 | GERARDO | REY | 4100 | 10 |

1 row in set (0.00 sec)

c/ **SHOW CREATE VIEW** empMasSueldo;

d/

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

**create or replace view VISTA2 as select \* from emple where dept\_no = 10 with check option;**

**select \* from VISTA2;**

**insert into VISTA2 (emp\_no,nombre,salario,dept\_no) values (9998,’Ana’,3599,30);**

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.

mysql> insert into empledep10 (emp\_no,nombre,dept\_no) values (7,'Paloma',10);

Query OK, 1 row affected (0.06 sec)

mysql> select \* from empledep10;

| EMP\_NO | NOMBRE | APELLIDO | OFICIO | FECHA\_ALT | SALARIO | COMISION | DEPT\_NO | EXTEL | FECNA | NUMHI |

| 7 | Paloma | NULL | NULL | NULL | NULL | NULL | 10 | NULL | NULL | NULL |

| 7782 | OLIVIA | CEREZO | ANALISTA | 2007-12-08 | 2885 | NULL | 10 | 120 | 1975-12-10 | 1 |

| 7839 | GERARDO | REY | PRESIDENTE | 1990-10-12 | 4100 | NULL | 10 | 100 | 1945-10-12 | 7 |

3 rows in set (0.00 sec)

mysql>

Query OK, 1 row affected (0.01 sec)

mysql> select \* from empledep10;

Podemos consultar cómo se ha creado

**SHOW CREATE VIEW EMPLEDEP10;**

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**

**create or replace view DEPARTNUM as select d.dnombre, count(e.dept\_no) "numero empleados" from emple e, depart d where d.dept\_no = e.dept\_no group by d.dnombre with check option;**

**CREATE OR REPLACE VIEW DEPARTNUM AS SELECT D.DNOMBRE, COUNT(E.EMP\_NO) NUMEROEMPLEADOS FROM EMPLE E, DEPART D WHERE D.DEPT\_NO = E.DEPT\_NO GROUP BY D.DNOMBRE;**

B/ Consulta la vista creada

**SELECT** \* **FROM** DEPARTNUM;

C/ Consulta cómo se ha creado

**Describe DEPARTNUM;**

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

Consulta de nuevo la vista

**No se puede crear porque la vista contiene algo mas complejo y 2 tablas a la vez, por eso se inserta en empleado**

**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 OR REPLACE VIEW EMPLEDEP10 AS SELECT \* FROM emple where DEPT\_NO = 10 WITH CHECK OPTION;

Al insertar los datos comprueba que se cumple el where:

Consulta la vista creada

SELECT \* from EMPLEDEP10;

Consulta cómo se ha creado

DESCRIBE EMPLEDEP10;

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

INSERT INTO EMPLEDEP10 (EMP\_NO, NOMBRE, APELLIDO, SALARIO, DEPT\_NO) VALUES (3245,’JUAN’,’TENDERO’,456,12);

Consulta de nuevo la vista

SELECT \* from EMPLEDEP10;

**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.**

**Créate global temporary table PUESTOS (nombre,apellido,puesto) as select nombre,apellido,oficio from emple order by 3;**

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 PRESIDENTE

create or replace view VISTA3 as select \* from emple where oficio like '%JEFE%';

select \* from VISTA3;

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

Si es insertable y actualizable porque solo hay una tabla.

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.

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 VISTAS2 as select e.nombre,e.apellido,e.emp\_no,e.dept\_no from emple e, depart d where e.dept\_no=d.dept\_no and d.dnombre like '%RECURSOS%';

select \* from VISTAS2;

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

create or replace view VISTAS3 as select d.dnombre "nombre departamento", avg(e.salario) "pepe" from emple e, depart d where d.dept\_no=e.dept\_no group by d.dnombre;

select \* from VISTAS3;

-Comprobar si es insertarble/actualizable con un ejemplo

- Explicar por qué sí o no

NO SERIA ACTUALIZABLE

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

create or replace view VISTAS4 as select \* from depart where dept\_no NOT IN (select dept\_no from emple);

select \* from VISTAS4;

**--OTRA—**

create or replace view VISTAS5 as select \* from depart d left outer join emple e on d.dept\_no=e.dept\_no and e.dept\_no is null;

create or replace view vista4b as select d.dept\_no,dnombre,numce from depart d, emple e where e.depart\_no(+)=d.depart\_no and e.depart\_no is null;

select \* from VISTAS5;

**--OTRA—**

Create or replace view vista4 as select d.dept\_no, dnombre, numce from depart d left outer join emple e on d.depart\_no = e.depart\_no and e.dept\_no is null group by d.dept\_no, dnombre, numce having count(e.dept\_no\_no) = 0;

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

INSERT INTO EMPLE (EMP\_NO,NOMBRE,DEPT\_NO) VALUES (9990,’ANA’,40);

SI SE PUEDE Y LOS DATOS INSERTADOS EN LAS TABLAS O EN LAS VISTAS SE COMPARTEN Y SE ACTUALIZAN ENTRE ELLOS Y LOS CAMPOS QUE DEJAMOS VACIOS SE QUEDAN A NULL.

INSERT INTO VISTA4 (DEPT\_NO,DENOMBRE) VALUES (90,’INFORMATICA’);

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

CREATE OR REPLACE VIEW VISTA5 AS SELECT \* FROM DEPART D, EMPLE E WHERE sueldo = ( select max(sueldo) from emple where );

Para comprobar, visualizar la suma de sueldos por departamento:

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

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

CREATE OR REPLACE VIEW VISTA6 AS SELECT

--OTRA Usando la vista1:

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

**Con el with check option si hago una vista luego restringe los datos que podemos meter a ese rango que hemos creado solo por ejemplo si fuera la vista con el where dept\_no=10, solo te dejaría crear un insert into con el depart 10!.**