

Vistas de la Base de datos

A veces para obtener datos de varias tablas hemos de construir sentencias SELECT complejas y si en otro momento necesitamos realizar la misma consulta y no la hemos guardado, tenemos que hacerlas de nuevo. Las vistas solucionan este problema: mediante una consulta simple de una vista obtenemos los datos de una consulta compleja.

Una vista también ayuda a limitar el acceso a los datos a determinados usuarios ya que puede que no se quiera que un subconjunto de datos importantes puedan ser consultados por todos sino por algunos de ellos.

Una vista es en realidad una tabla lógica que permite acceder a la información de una o de varias tablas. La vista no contiene información en si misma, almacena simplemente una sentencia SELECT. Si se suprime una tabla, la vista en la cual se haga referencia a dicha tabla queda invalidada. Si se modifica una tabla, la vista deberá ser modificada también siempre y cuando las columnas modificadas intervengan en la misma.

Aunque se usan menos, las vistas pueden estar basadas en otras vistas.

Creación de vistas en MySQL

```
1 CREATE
2   [ALGORITHM = {MERGE | TEMPTABLE | UNDEFINED}]
3 VIEW [database_name]. [view_name]
4 AS
5   SELECT statement;
```

El algoritmo permite controlar qué mecanismo se utiliza al crear una vista. Por omisión es UNDEFINED que permite a MySQL tomar una decisión sobre si se debe utilizar MERGE o TEMPTABLE. La diferencia entre estas dos opciones radica en si la vista usará una tabla temporal o no.

En la declaración SELECT se pueden consultar las tablas o las vistas disponibles en la base de datos pero tiene varias reglas que debe cumplir:

- Las variables locales y las de sesión no se puede utilizar en la instrucción SELECT.
- Una Vista no se puede asociar con disparadores.

Ejemplos de Creación de vistas

Vamos a crear una vista que con los importes totales por pedido.

```

1 CREATE VIEW ventasporpedido
2   AS
3   SELECT numpedido,
4     SUM (importe) total
5   FROM lineaspedidos
6   GROUP by numpedido
7   ORDER BY total DESC;

```

Fijarse que en las vistas las columnas de expresiones deben tener obligatoriamente un alias para podernos referir a ellas.

Cada vez que queramos conocer el total de los importes por pedido, sólo se tiene que ejecutar una consulta simple de esta vista:

```

SELECT * FROM ventasporpedido WHERE numpedido=2345;
SELECT * FROM ventasporpedido;

```

El siguiente es un ejemplo de creación de una vista con join. La vista devuelve el número de pedido, el número del cliente y el importe total por pedido.

```

CREATE VIEW ventasporpedido2
AS
SELECT l.numpedido,numcliente
SUM (importe) total
FROM lineaspedidos l, pedidos p
WHERE l.numpedido=p.numpedido
GROUP by l.numpedido,numcliente
ORDER BY total DESC;

```

Creación de vistas actualizables. Operaciones sobre vistas.

Las operaciones que se pueden realizar sobre vistas son las mismas que las que se llevan a cabo sobre tablas: SELECT, INSERT, UPDATE y DELETE aunque con bastantes restricciones.

Las vistas no sólo son de lectura, sino que también se pueden actualizar, pero para ello la instrucción SELECT que define la vista tiene que seguir varias reglas:

- ☐ La [instrucción SELECT](#) sólo debe referirse a una tabla de la [base de datos](#).
- ☐ La sentencia SELECT no debe usar [GROUP BY](#) ni [HAVING](#).
- ☐ La sentencia SELECT no debe utilizar [DISTINCT](#) en la lista de columnas de la cláusula SELECT.

- ❑ La sentencia SELECT no debe hacer referencia a vistas de sólo lectura.
- ❑ La sentencia SELECT no puede contener expresión, columnas calculadas ni funciones de grupo.

```
1 CREATE VIEW vistaoficinas
2 AS
3   SELECT oficina, ciudad, region
4   FROM oficinas;
```

```
1 SELECT * FROM vistaoficinas;
```

```
1 UPDATE vistasoficinas
2 SET ciudad = 'ALGECIRAS'
3 WHERE oficina=4;
```

Para mostrar la definición de las vistas:

```
1 SHOW CREATE VIEW [database_name].[view_name];
```

```
1 SHOW CREATE VIEW organization
```

Para modificar una vista:

```
ALTER
[ALGORITHM = {MERGE | TEMPTABLE | UNDEFINED}]
VIEW [database_name].[view_name]
AS
  SELECT statement;
```

Para borrar una vista:

```
1 DROP VIEW [IF EXISTS] [database_name].[view_name];
```

El IF EXISTS es opcional y permite comprobar si la vista existe o no. Ayuda a evitar la eliminación de una vista no existente.

```
1 DROP VIEW IF EXISTS organization;
```

Cada vez que se modifica o se borra una vista, MySQL hace una copia de seguridad del archivo de definición de vistas en la carpeta `/database_name/arc/`. Así en caso de que accidentalmente se altere o elimine una vista, puede obtener una copia de seguridad de allí.