

Vistas

Las vistas (*views*) hacen posible el definir representaciones especiales de una tabla. Una vista se comporta de manera casi idéntica a una tabla. Se pueden consultar con SELECT y modificar con INSERT, UPDATE y DELETE. Las vistas se pueden usar en MySQL desde la versión 5.0. Hay dos razones básicas para usar vistas:

- **Seguridad.** Hay ocasiones en las que querríamos que un usuario pudiera acceder a una tabla, pero no a todos sus registros. Un ejemplo es una tabla con los datos de los empleados de una empresa. En general, queremos que todos los empleados accedan a esa tabla, pero que no puedan acceder a todas las columnas de cada registro donde probablemente hay información confidencial.

La solución es crear una vista de la tabla de empleados en la que solo aparezcan las columnas con información pública, y permitir el acceso a la vista y no a la tabla original.

- **Eficiencia.** En muchas ocasiones **las aplicaciones** ejecutan muchas veces la misma consulta para seleccionar un mismo conjunto de registros de una tabla. En lugar de dejar que cada usuario o programador repita una y otra vez la misma consulta, podemos crear una vista con dicha consulta y hacerla visible a los usuarios o programadores.

Preparamos una vista adecuada a la aplicación, esta no accede a la tabla sino a la vista.

Las vistas son **como tablas virtuales** cuyo contenido es el resultado de un SELECT. Por ejemplo, si queremos que **algunos usuarios** de la base de datos biblioteca tengan acceso a la tabla títulos, pero solo a las columnas títuloID, título y subtítulo ordenadas alfabéticamente haremos lo siguiente:

```
CREATE VIEW v1 AS
SELECT títuloID, título, subtítulo FROM títulos
ORDER BY título, subtitle
```

Pero la definición puede ser tan compleja como nuestras necesidades. Otro ejemplo: si queremos que determinados usuarios de biblioteca sólo puedan acceder a la lista de libros en español, y de cada uno solo saber la editorial y la categoría crearemos la siguiente vista:

```
CREATE VIEW v2 AS
SELECT título, nombreEditorial, nombreCat FROM títulos, editoriales,
categorías
WHERE títulos.editID = editoriales.editID
AND títulos.catID = categorías.catID
AND langID = 5
```

ALGUNAS VISTAS SON ACTUALIZABLES: ACTUALIZANDO LA VISTA, ESTAMOS ACTUALIZANDO LA TABLA, TABLAS SUBYACENTES.

Para poder modificar una vista con INSERT, UPDATE y DELETE la única restricción que se aplica es sobre el comando SELECT que se ha usado para crearla. Para ello se aplican las siguientes reglas:

- El comando SELECT no debe contener:
GROUP BY, DISTINCT, LIMIT, UNION o HAVING.
- Las vistas que se forman con la consulta de dos o más **tablas son casi siempre imposibles de modificar**. (SÓLO INNER JOIN...Y NO PERMITE DELETE)
- Las vistas deberían contener todas las columnas para las que hay claves primarias o índices únicos o claves foráneas. Si no lo hacen, la opción `updatable_views_with_limit` de MySQL define qué cambios deben producir un aviso (comportamiento por defecto) o dar un error.

Al crear una vista hay una serie de atributos que podemos usar. La sintaxis de CREATE VIEW es la siguiente:

```
CREATE [OR REPLACE] [ALGORITHM = UNDEFINED | MERGE | TEMPTABLE]
VIEW name [(columnlist)] AS select command
[WITH [CASCADED | LOCAL] CHECK OPTION]
```

El significado de las opciones es:

- OR REPLACE significa que si existía una vista con el mismo nombre debe de ser reemplazada con la nueva definición sin dar ningún mensaje de error.
- ALGORITHM nos dice como se representa la vista internamente. Esta opción esta sin documentar hasta el momento.
- WITH CHECK OPTION significa que está permitido modificar los registros sólo si las condiciones WHERE del SELECT se continuarán cumpliendo después de la modificación. Esta opción solo tiene sentido utilizarla si la vista que queremos crear va a ser modificada posteriormente.
 - La variante LOCAL afecta a las vistas que se derivan a su vez de otras vistas. Esto significa que solo se consideran las condiciones WHERE de la primera vista, no las de las siguientes.
 - El efecto de la variante CASCADE es el contrario. Se consideran todas las condiciones WHERE.

