



Bases de datos II

Vistas investigación

Tecnologías de la Información y Telecomunicaciones

Docente: MSI. Raúl Alberto Toledo Piñón

Presenta: ING. Jorge Parra Hidalgo

Matricula: 13104

Fecha: viernes marzo 21, 2025

Indice

Que es una vista en MySQL	3
Ventajas.....	3
Ejercicio de ejemplo.....	4
Explicación del código:	4
Reemplazar o actualizar vista.....	5
Conclusiones de la Investigación sobre Vistas Actualizables en MySQL.....	6
Glosario.....	6

Que es una vista en MySQL

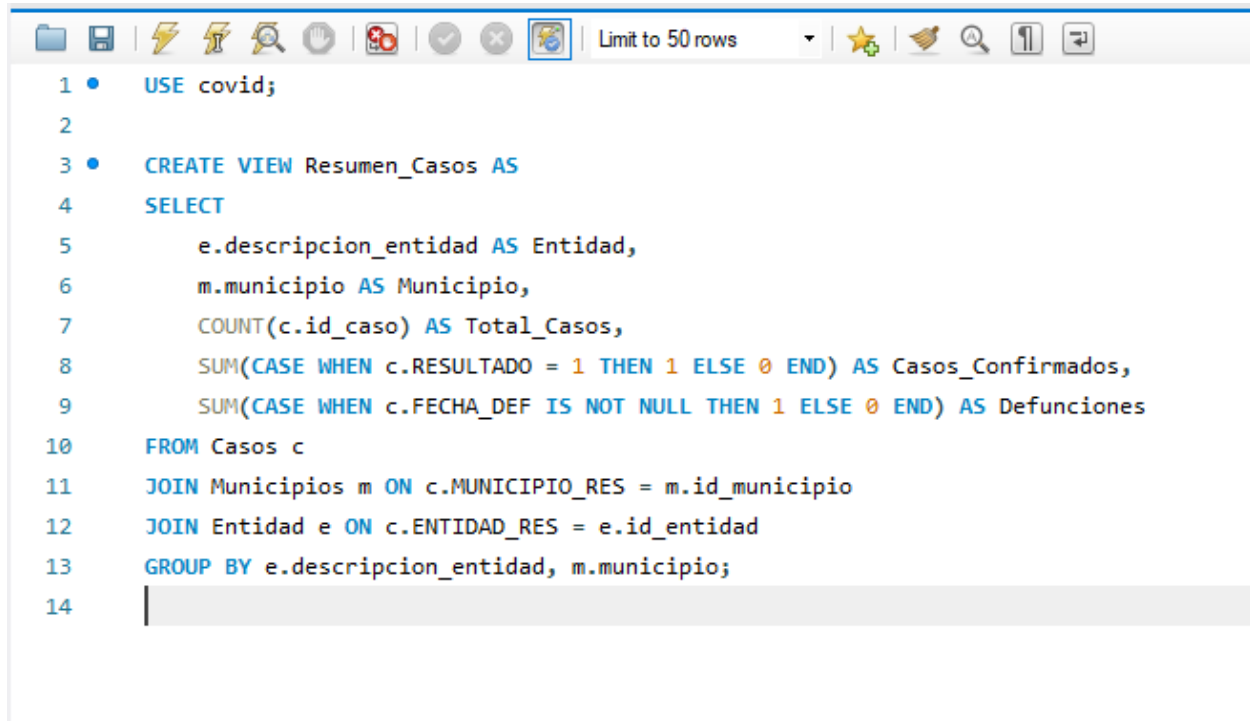
Antes de cualquier cosa hay que definir que es una vista en MySQL y este es una opción que te permite observar los datos de una serie de tablas. La vista parece una tabla virtual que almacena todas las consultas que se realizan sobre las bases de datos. Es por eso que los datos o consultas que se establecen dentro de las vistas no están almacenados como un objeto dentro de las tablas de la base de datos.

De esta manera, cuando haces una consulta sobre una vista, te ejecutará el query que has escrito y te guardará la consulta que vas a realizar en lugar de los valores, como cualquier otra de las tablas.

Ventajas

- Las vistas te permiten ocultar información que no necesites o que no están establecidos dentro de la vista. Estas funcionan como tablas, donde el usuario pueda realizar las modificaciones o cualquier otro ejercicio sobre ellas.
- Otra posibilidad de las vistas es la de dar un mejor control sobre los permisos de los usuarios que ingresan a las bases de datos. De esta manera, se pueden dar permisos a personas para que solo tengan la posibilidad de acceder a ciertos registros, únicamente por medio de las vistas. Así pues, no habrá ningún problema relacionado con las estructuras de las tablas de datos. Por lo tanto, es mucho más seguro.
- Mejora el rendimiento de las bases de datos.

Ejercicio de ejemplo



```
1 • USE covid;
2
3 • CREATE VIEW Resumen_Casos AS
4 SELECT
5     e.descripcion_entidad AS Entidad,
6     m.municipio AS Municipio,
7     COUNT(c.id_caso) AS Total_Casos,
8     SUM(CASE WHEN c.RESULTADO = 1 THEN 1 ELSE 0 END) AS Casos_Confirmados,
9     SUM(CASE WHEN c.FECHA_DEF IS NOT NULL THEN 1 ELSE 0 END) AS Defunciones
10 FROM Casos c
11 JOIN Municipios m ON c.MUNICIPIO_RES = m.id_municipio
12 JOIN Entidad e ON c.ENTIDAD_RES = e.id_entidad
13 GROUP BY e.descripcion_entidad, m.municipio;
14
```

Explicación del código:

1. Se usa JOIN para relacionar la tabla Casos con Municipios y Entidad para obtener los nombres de los municipios y entidades.
2. Se cuentan los casos totales en cada municipio (COUNT(c.id_caso)).
3. Se usa SUM(CASE WHEN c.RESULTADO = 1 THEN 1 ELSE 0 END) para contar los casos confirmados (suponiendo que RESULTADO = 1 indica un caso positivo).
4. Se usa SUM(CASE WHEN c.FECHA_DEF IS NOT NULL THEN 1 ELSE 0 END) para contar las defunciones (si FECHA_DEF no es NULL, significa que el paciente falleció).
5. Se agrupan los resultados por entidad y municipio.

Esta vista te permitirá consultar un resumen rápido sin necesidad de escribir la consulta repetidamente. Para verla, solo se ejecuta:

1 • **SELECT** * **FROM** Resumen_Casos;
2

Result Grid

	Entidad	Municipio	Total_Casos	Casos_Confirmados	Defunciones
▶	BAJA CALIFORNIA	ASIENTOS	3887	2256	233
	BAJA CALIFORNIA	MEXICALI	3887	2256	233
	BAJA CALIFORNIA	MULEGÉ	3887	2256	233
	BAJA CALIFORNIA	CAMPECHE	3887	2256	233
	BAJA CALIFORNIA	ACUÑA	3887	2256	233
	BAJA CALIFORNIA	COLIMA	3887	2256	233
	BAJA CALIFORNIA	ACALA	3887	2256	233
	BAJA CALIFORNIA	ALDAMA	7774	4512	466
	BAJA CALIFORNIA	AZCAPOTZALCO	3887	2256	233
	BAJA CALIFORNIA	CANELAS	3887	2256	233
	BAJA CALIFORNIA	ACÁMBARO	3887	2256	233
	BAJA CALIFORNIA	AHUACUOTZINGO	3887	2256	233
	BAJA CALIFORNIA	ACAXOCHTLÁN	3887	2256	233
	BAJA CALIFORNIA	ACATLÁN DE JU...	3887	2256	233
	BAJA CALIFORNIA	ACOLMAN	3887	2256	233
	BAJA CALIFORNIA	AGUILILLA	3887	2256	233
	BAJA CALIFORNIA	ATLATLAHUCAN	3887	2256	233

Output

#	Time	Action	Message
3	15:32:17	USE covid	0 row(s) affected
4	15:32:17	CREATE VIEW Resumen_Casos AS SELECT e.descripcion_entidad AS Entidad, m.municipio AS Muni...	Error Code: 1050. Table 'Resumen_Casos' already exists
5	15:32:19	USE covid	0 row(s) affected
6	15:32:19	CREATE VIEW Resumen_Casos AS SELECT e.descripcion_entidad AS Entidad, m.municipio AS Muni...	Error Code: 1050. Table 'Resumen_Casos' already exists
7	15:32:55	DROP VIEW IF EXISTS Resumen_Casos	0 row(s) affected
8	15:33:03	USE covid	0 row(s) affected
9	15:33:03	CREATE VIEW Resumen_Casos AS SELECT e.descripcion_entidad AS Entidad, m.municipio AS Muni...	0 row(s) affected
10	15:44:22	SELECT * FROM Resumen_Casos LIMIT 0, 50	50 row(s) returned

Reemplazar o actualizar vista

Si quieres actualizar una vista sin eliminarla, usa **CREATE OR REPLACE VIEW**, así:

```
sql
Copy Edit

CREATE OR REPLACE VIEW Resumen_Casos AS
SELECT
    e.descripcion_entidad AS Entidad,
    m.municipio AS Municipio,
    COUNT(c.id_caso) AS Total_Casos,
    SUM(CASE WHEN c.RESULTADO = 1 THEN 1 ELSE 0 END) AS Casos_Confirmados,
    SUM(CASE WHEN c.FECHA_DEF IS NOT NULL THEN 1 ELSE 0 END) AS Defunciones
FROM Casos c
JOIN Municipios m ON c.MUNICIPIO_RES = m.id_municipio
JOIN Entidad e ON c.ENTIDAD_RES = e.id_entidad
GROUP BY e.descripcion_entidad, m.municipio
```

Conclusiones de la Investigación sobre Vistas Actualizables en MySQL

Condiciones para que una Vista sea Actualizable en MySQL: Para que una vista en MySQL permita realizar inserciones, actualizaciones y eliminaciones que se reflejen en la tabla subyacente, debe cumplir con las siguientes condiciones:

- 1.- Basarse en una sola tabla: No se pueden actualizar vistas que combinan varias tablas mediante JOIN, UNION u otras combinaciones complejas.
- 2.- No usar agregaciones: Funciones como SUM(), COUNT(), AVG() o cláusulas como GROUP BY hacen que la vista sea de solo lectura.
- 3.- Incluir las columnas esenciales: Las columnas involucradas en las operaciones INSERT o UPDATE deben estar presentes en la vista, especialmente las que tienen restricciones como NOT NULL o claves primarias.
- 4.- No usar DISTINCT ni LIMIT: Estas operaciones hacen que la vista no se pueda modificar directamente.
- 5.- Evitar expresiones complejas: No se permiten expresiones calculadas o subconsultas en las columnas a actualizar.

Glosario

Base de Datos: Conjunto organizado de datos que pueden ser almacenados y accesibles electrónicamente mediante un sistema de gestión de bases de datos (DBMS).

Vista (View): En MySQL, una vista es una tabla virtual que muestra los resultados de una consulta almacenada. No guarda los datos directamente, sino que presenta una representación de los datos de otras tablas.

MySQL: Sistema de gestión de bases de datos relacional de código abierto, ampliamente utilizado para administrar bases de datos.

Query (Consulta): Instrucción escrita en lenguaje SQL que permite realizar operaciones como seleccionar, insertar, actualizar o eliminar datos en una base de datos.

Claves Primarias (Primary Key): Identificador único para cada registro en una tabla, que garantiza la unicidad y facilita la referencia entre tablas.

Tabla Virtual: Representación de una tabla que no almacena datos físicamente, sino que se basa en una consulta a una o más tablas reales.

Actualización de Vista: Proceso de modificar una vista existente utilizando el comando `CREATE OR REPLACE VIEW` sin necesidad de eliminarla previamente.

Vista Actualizable: Vista que permite realizar operaciones de inserción, actualización o eliminación directamente sobre los datos de la tabla subyacente, cumpliendo ciertas condiciones.

Subconsulta: Consulta anidada dentro de otra consulta principal, que se utiliza para realizar operaciones complejas en SQL.

Permisos de Usuario: Configuración en MySQL que define el nivel de acceso y las operaciones que un usuario puede realizar en una base de datos.

Seguridad en Bases de Datos: Prácticas y mecanismos implementados para proteger la información almacenada, incluyendo el control de accesos y permisos.

Optimización de Consultas: Proceso de mejorar la eficiencia y velocidad de las consultas SQL para optimizar el rendimiento de una base de datos.