**Unidad. Vistas.**

Blanca Calderón

Curso 2023/2024

**Índice**

1. [Vistas](#_bookmark0) 1
   1. [Crear o modificar una vista](#_bookmark1) 1
   2. [Eliminar una vista](#_bookmark2) 2
   3. [Renombrar una vista](#_bookmark3) 2
   4. [Consultar el listado de vistas disponibles](#_bookmark4) 2
   5. [Consultar la sentencia que se utilizó para crear una vista](#_bookmark5) 3
   6. [Vistas con contenido actualizable con INSERT, UPDATE y DELETE](#_bookmark6) 3
   7. [Ejemplos](#_bookmark7) 3
      1. [Base de datos: Sakila](#_bookmark8) 3
      2. [Modelo entidad/relación](#_bookmark9) 4
      3. [Base de datos para MySQL](#_bookmark10) 4
      4. [Vistas utilizadas en la base de datos Sakila](#_bookmark11) 4
   8. [Ejercicios](#_bookmark12) 5
      1. [Base de datos: Jardinería](#_bookmark13) 5
2. [Referencias](#_bookmark14) 7

¿Qué es una vista en SQL?

Una vista actúa como filtro de las tablas subyacentes a las que se hace referencia en ella. La consulta que define la vista puede provenir de una o de varias tablas, o bien de otras vistas de la base de datos actual u otras bases de datos.

# Crear o modificar una vista

1. **CREATE**
2. [**OR** REPLACE]
3. [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
4. [DEFINER = user]
5. [SQL SECURITY { DEFINER | INVOKER }]
6. **VIEW** view\_name [(column\_list)]
7. **AS** select\_statement
8. [WITH [**CASCADED** | **LOCAL**] **CHECK** OPTION]

### Ejemplo:

Crea una vista que muestre para cada uno de los pedidos, el código del pedido, la fecha, el nombre del cliente que realizó el pedido y el importe total del pedido.

1. **CREATE OR** REPLACE **VIEW** resumen\_pedidos **AS**
2. **SELECT**
3. pedido.codigo\_pedido,
4. pedido.fecha\_pedido,
5. cliente.nombre\_cliente,
6. **SUM**(detalle\_pedido.cantidad \* detalle\_pedido.precio\_unidad) **AS** total
7. **FROM**
8. cliente **INNER JOIN** pedido
9. **ON** cliente.codigo\_cliente = pedido.codigo\_cliente
10. **INNER JOIN** detalle\_pedido
11. **ON** pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido
12. **GROUP BY** pedido.codigo\_pedido

Cuando creamos una vista, es posible crear un alias para cada una las columnas. En el siguiente ejemplo se muestran los nombres que tendrán las columnas encerradas entre paréntesis.

1. **CREATE OR** REPLACE **VIEW** resumen\_pedidos (codigo\_pedido, fecha\_pedido, nombre\_cliente, total) **AS**
2. **SELECT**
3. pedido.codigo\_pedido,
4. pedido.fecha\_pedido,
5. cliente.nombre\_cliente,
6. **SUM**(detalle\_pedido.cantidad \* detalle\_pedido.precio\_unidad)
7. **FROM**
8. cliente **INNER JOIN** pedido
9. **ON** cliente.codigo\_cliente = pedido.codigo\_cliente
10. **INNER JOIN** detalle\_pedido
11. **ON** pedido.codigo\_pedido = detalle\_pedido.codigo\_pedido

12 **GROUP BY** pedido.codigo\_pedido

### Referencias:

* [Documentación oficial de CREATE VIEW en MySQL](https://dev.mysql.com/doc/refman/8.0/en/create-view.html).

# Eliminar una vista

1. **DROP VIEW** [IF **EXISTS**]
2. view\_name [, view\_name] ...
3. [**RESTRICT** | **CASCADE**]

### Ejemplo:

1 **DROP VIEW** resumen\_pedidos;

### Referencias:

* [Documentación oficial de DROP VIEW en MySQL](https://dev.mysql.com/doc/refman/8.0/en/drop-view.html).

# Renombrar una vista

# Alter????

1. RENAME **TABLE**
2. tbl\_name TO new\_tbl\_name
3. [, tbl\_name2 TO new\_tbl\_name2] ...

### Ejemplo:

1. RENAME **TABLE** old\_table TO new\_table;

||

1. ALTER VIEW old\_view RENAME TO new\_view;

### Referencias:

* [Documentación oficial de RENAME TABLE en MySQL](https://dev.mysql.com/doc/refman/8.0/en/rename-table.html).

# Consultar el listado de vistas disponibles

# SHOW FULL TABLES IN tu\_base\_de\_datos WHERE TABLE\_TYPE LIKE 'VIEW';

1 SHOW **FULL** TABLES;

1. SHOW **FULL** TABLES
2. **WHERE** table\_type = 'VIEW';

### Referencias:

* [Documentación oficial de SHOW FULL TABLES en MySQL](https://dev.mysql.com/doc/refman/8.0/en/show-tables.html).

# Consultar la sentencia que se utilizó para crear una vista

1 SHOW **CREATE VIEW** view\_name

### Referencias:

* [Documentación oficial de SHOW CREATE VIEW en MySQL](https://dev.mysql.com/doc/refman/8.0/en/show-create-view.html).

# Vistas con contenido actualizable con INSERT, UPDATE y DELETE

En algunos casos es posible actualizar el contenido de las tablas que se utilizan para crear una vista.

Para que una vista sea actualizable, es necesario que todas las columnas obligatorias de la tabla subyacente estén presentes en la vista.

*Cuando hablamos de "vistas con contenido actualizable con INSERT, UPDATE y DELETE", nos referimos a* *vistas que permiten realizar operaciones de inserción, actualización y eliminación de datos a través de la vista. En otras palabras, puedes usar la vista como si fuera una tabla normal en la que puedes modificar los datos.*

*Sin embargo, no todas las vistas son actualizables. Para que una vista sea actualizable, es necesario que se cumplan ciertas condiciones. Una de esas condiciones es que todas las columnas que son obligatorias (no nulas) en la tabla subyacente también estén presentes en la vista. Esto significa que si intentas actualizar una fila a través de la vista, todas las columnas que no permiten valores nulos deben estar presentes en la vista para que la operación sea posible.*

*Por ejemplo, supongamos que tienes una tabla llamada "Clientes" con las columnas "ID\_Cliente" (clave primaria), "Nombre" y "Email". Si creas una vista de esta tabla pero omites la columna "Email" en la vista, la vista no será actualizable, ya que "Email" es una columna obligatoria (no nula) en la tabla original.*

*En resumen, para que una vista sea actualizable, todas las columnas que no permiten valores nulos en la tabla subyacente deben estar presentes en la vista. Esto garantiza que las operaciones de inserción, actualización y eliminación puedan realizarse correctamente a través de la vista.*

*/\*suele ser mejor realizar actualizaciones directamente sobre la tabla y no sobre la vista\*/*

### Referencias:

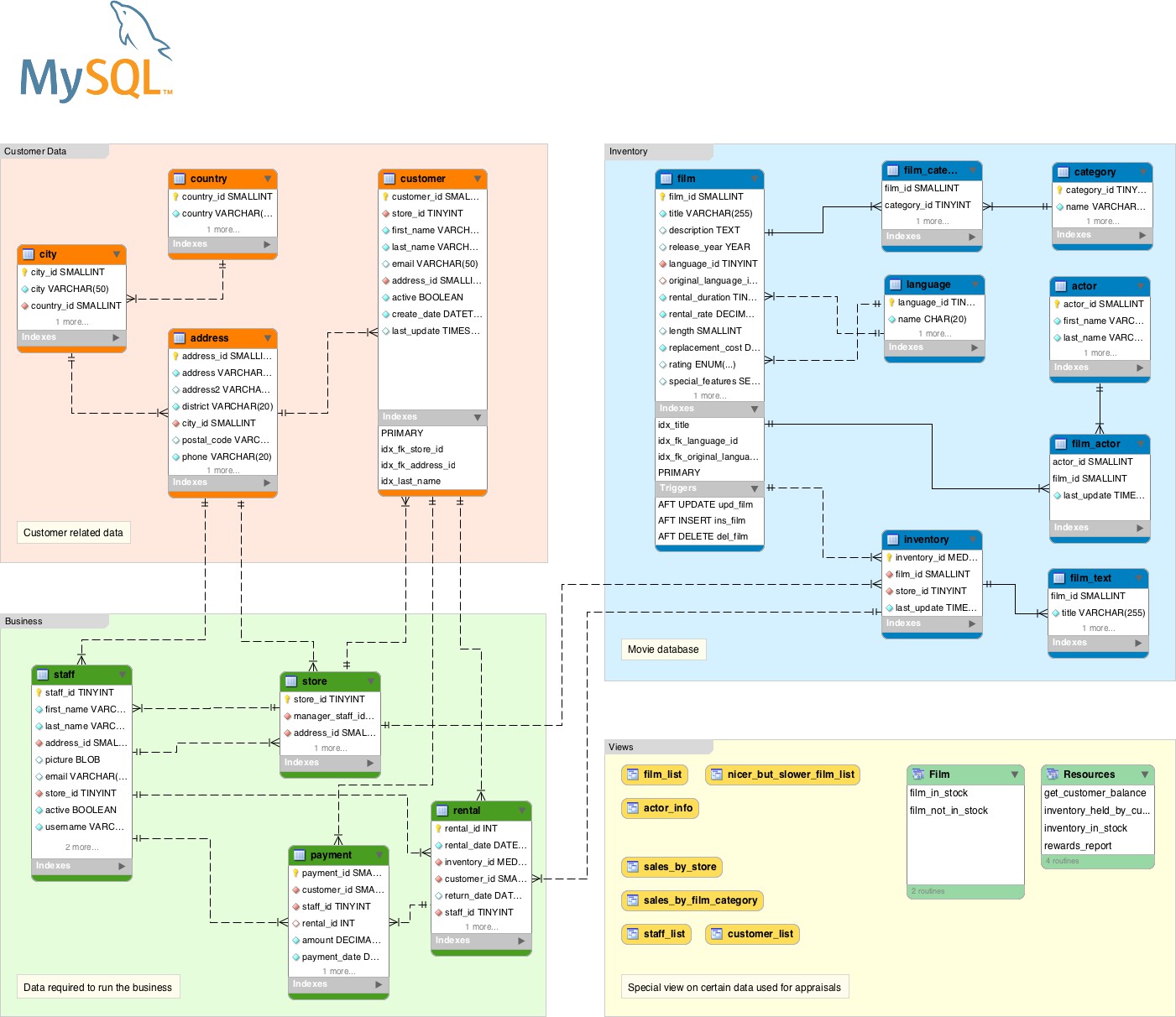
* [Documentación oficial sobre vistas actualizables](https://dev.mysql.com/doc/refman/8.0/en/view-updatability.html).

# Ejemplos

## Base de datos: Sakila

La base de datos Sakila está disponible en la página [web oficial de MySQL](https://dev.mysql.com/doc/sakila/en/). Se trata de una base de datos creada por **Mike Hillyers**.

## Modelo entidad/relación



## Base de datos para MySQL

La base de datos [está disponible en la web oficial de MySQL](https://dev.mysql.com/doc/index-other.html).

## Vistas utilizadas en la base de datos Sakila

A continuación, se muestran algunas de las vistas que se han utilizado en la base de datos Sakila.

1

2

3

4

5

--

-- View structure for view `customer\_list`

--

**CREATE VIEW** customer\_list **AS**

1. **SELECT**
2. cu.customer\_id **AS** ID,
3. CONCAT(cu.first\_name, \_utf8mb4' ', cu.last\_name) **AS** name,
4. a.address **AS** address,
5. a.postal\_code **AS** `zip code`,
6. a.phone **AS** phone,
7. city.city **AS** city,
8. country.country **AS** country,
9. IF(cu.active, \_utf8mb4'active',\_utf8mb4'') **AS** notes,
10. cu.store\_id **AS** SID
11. **FROM**
12. customer **AS** cu **JOIN** address **AS** a
13. **ON** cu.address\_id = a.address\_id
14. **JOIN** city
15. **ON** a.city\_id = city.city\_id
16. **JOIN** country
17. **ON** city.country\_id = country.country\_id;

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

--

-- View structure for view `film\_list`

--

**CREATE VIEW** film\_list **AS SELECT**

film.film\_id **AS** FID, film.title **AS** title,

film.description **AS** description, category.name **AS** category, film.rental\_rate **AS** price,

film.length **AS** length, film.rating **AS** rating,

GROUP\_CONCAT(CONCAT(actor.first\_name, \_utf8mb4' ', actor.last\_name) SEPARATOR ', ') **AS** actors

**FROM**

category **LEFT JOIN** film\_category

**ON** category.category\_id = film\_category.category\_id

**LEFT JOIN** film

**ON** film\_category.film\_id = film.film\_id

**JOIN** film\_actor

**ON** film.film\_id = film\_actor.film\_id

**JOIN** actor

**ON** film\_actor.actor\_id = actor.actor\_id

**GROUP BY** film.film\_id, category.name;

# Ejercicios

## Base de datos: Jardinería

* + - 1. Escriba una vista que se llame listado\_pagos\_clientes que muestre un listado donde aparezcan todos los clientes y los pagos que ha realizado cada uno de ellos. La vista deberá tener las siguientes co‑ lumnas: nombre y apellidos del cliente concatenados, teléfono, ciudad, pais, fecha\_pago, total del pago, id de la transacción
      2. Escriba una vista que se llame listado\_pedidos\_clientes que muestre un listado donde aparez‑ can todos los clientes y los pedidos que ha realizado cada uno de ellos. La vista deberá tener las siguientes columnas: nombre y apellidos del cliente concatendados, teléfono, ciudad, pais, código del pedido, fe‑ cha del pedido, fecha esperada, fecha de entrega y la cantidad total del pedido, que será la suma del producto de todas las cantidades por el precio de cada unidad, que aparecen en cada línea de pedido.
      3. Utilice las vistas que ha creado en los pasos anteriores para devolver un listado de los clientes de la ciudad de Madrid que han realizado pagos.
      4. Utilice las vistas que ha creado en los pasos anteriores para devolver un listado de los clientes que todavía no han recibido su pedido.
      5. Utilice las vistas que ha creado en los pasos anteriores para calcular el número de pedidos que se ha realizado cada uno de los clientes.
      6. Utilice las vistas que ha creado en los pasos anteriores para calcular el valor del pedido máximo y mínimo que ha realizado cada cliente.
      7. Modifique el nombre de las vista listado\_pagos\_clientes y asígnele el nombre listado\_de\_pagos

. Una vez que haya modificado el nombre de la vista ejecute una consulta utilizando el nuevo nombre de la vista para comprobar que sigue funcionando correctamente.

* + - 1. Elimine las vistas que ha creado en los pasos anteriores.

* + - * + [MySQL Views](https://www.mysqltutorial.org/mysql-views-tutorial.aspx).
        + [SQL Views](https://www.w3schools.com/sql/sql_view.asp).
        + [Documentación oficial de CREATE VIEW en MySQL](https://dev.mysql.com/doc/refman/8.0/en/create-view.html).
        + [Documentación oficial de DROP VIEW en MySQL](https://dev.mysql.com/doc/refman/8.0/en/drop-view.html).