

---

## **Unidad. Vistas.**

Apuntes de BD para DAW, DAM y ASIR

José Juan Sánchez Hernández

Curso 2023/2024

# Índice

<b>1</b>	<b>Vistas</b>	<b>1</b>
1.1	¿Qué es una vista?	1
1.2	Crear o modificar una vista	1
1.3	Eliminar una vista	2
1.4	Renombrar una vista	2
1.5	Consultar el listado de vistas disponibles	3
1.6	Consultar la sentencia que se utilizó para crear una vista	3
1.7	Vistas con contenido actualizable con INSERT, UPDATE y DELETE	3
1.8	Ejemplos	3
1.8.1	Base de datos: Sakila	3
1.8.2	Modelo entidad/relación	4
1.8.3	Base de datos para MySQL	4
1.8.4	Vistas utilizadas en la base de datos Sakila	4
1.9	Ejercicios	5
1.9.1	Base de datos: Jardinería	5
<b>2</b>	<b>Referencias</b>	<b>7</b>
<b>3</b>	<b>Licencia</b>	<b>8</b>

## Índice de figuras

## Índice de cuadros

# 1 Vistas

## 1.1 ¿Qué es una vista?

Una vista es una especie de tabla “virtual” que se crea a partir de una consulta SQL y se almacena en la base de datos. Los datos que se muestran en la vista son los resultados de la consulta que la define y están almacenados en las tablas reales de la base de datos.

Algunas de las principales ventajas que nos aporta el uso de vistas son las siguientes:

- Nos permiten convertir una consulta compleja en una tabla “virtual” para que sea más fácil trabajar con ella.
- Nos permiten ocultar columnas de una tabla real que no queremos que estén visibles a ciertos usuarios.

## 1.2 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 de las columnas. En el siguiente ejemplo se muestran los nombres que tendrán las columnas encerrados 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.](#)

### 1.3 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.](#)

### 1.4 Renombrar una vista

```
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;
```

#### Referencias:

- [Documentación oficial de RENAME TABLE en MySQL.](#)

## 1.5 Consultar el listado de vistas disponibles

```
1 SHOW FULL TABLES;
```

```
1 SHOW FULL TABLES
2 WHERE table_type = 'VIEW';
```

### Referencias:

- [Documentación oficial de SHOW FULL TABLES en MySQL.](#)

## 1.6 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.](#)

## 1.7 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.

### Referencias:

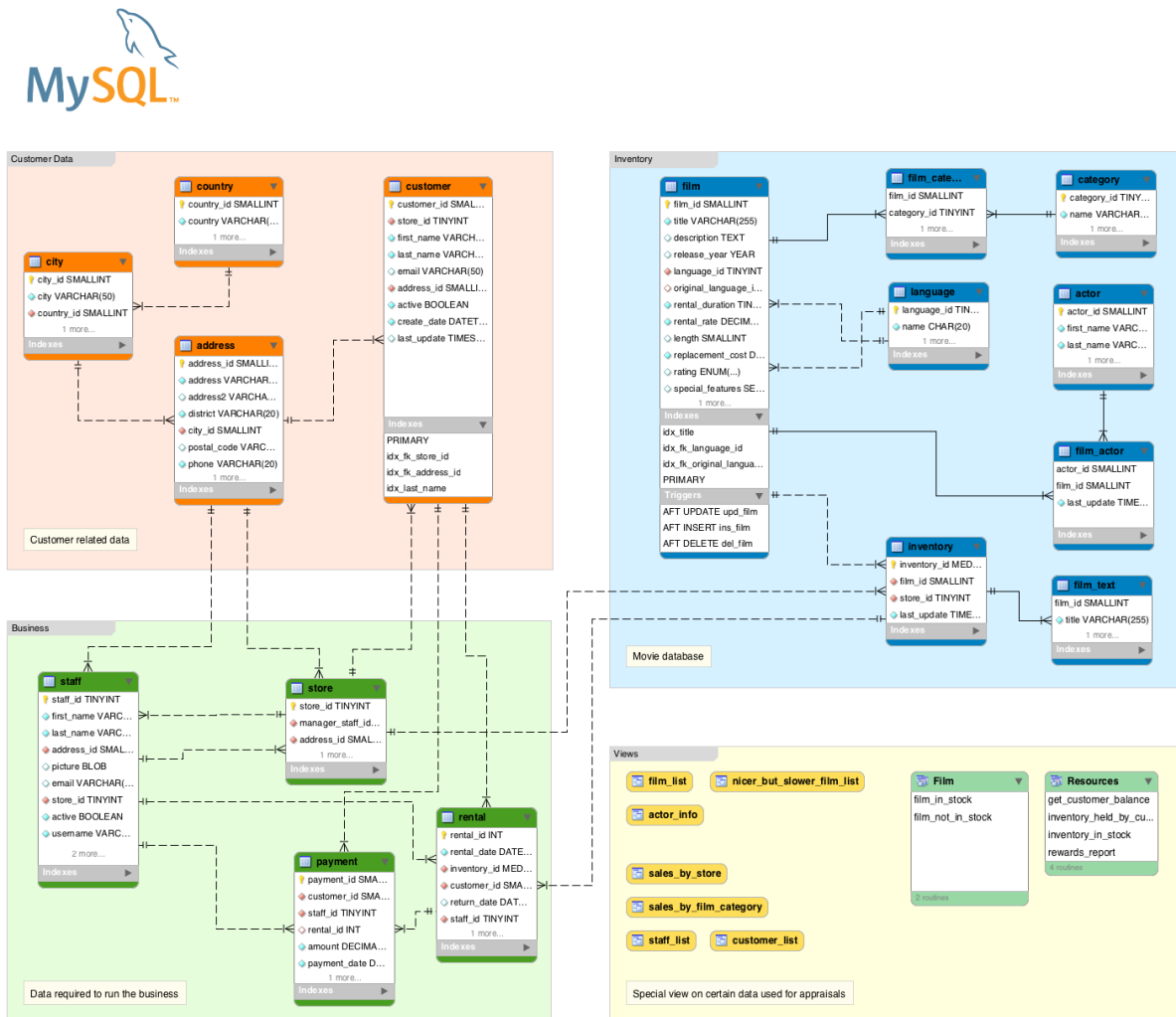
- [Documentación oficial sobre vistas actualizables.](#)

## 1.8 Ejemplos

### 1.8.1 Base de datos: Sakila

La base de datos [Sakila](#) está disponible en la página [web oficial de MySQL](#). Se trata de una base de datos creada por **Mike Hillyers**.

## 1.8.2 Modelo entidad/relación



## 1.8.3 Base de datos para MySQL

La base de datos [está disponible en la web oficial de MySQL](https://dev.mysql.com/doc/relnotes/mysql/8.0/en/mysql-database-for-mysql.html).

## 1.8.4 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  -- View structure for view `customer_list`
3  --
4
5  CREATE VIEW customer_list AS

```



```
6 SELECT
7     cu.customer_id AS ID,
8     CONCAT(cu.first_name, _utf8mb4' ', cu.last_name) AS name,
9     a.address AS address,
10    a.postal_code AS `zip code`,
11    a.phone AS phone,
12    city.city AS city,
13    country.country AS country,
14    IF(cu.active, _utf8mb4'active',_utf8mb4'') AS notes,
15    cu.store_id AS SID
16 FROM
17     customer AS cu JOIN address AS a
18     ON cu.address_id = a.address_id
19     JOIN city
20     ON a.city_id = city.city_id
21     JOIN country
22     ON city.country_id = country.country_id;
```

```
1 --
2 -- View structure for view `film_list`
3 --
4
5 CREATE VIEW film_list AS
6 SELECT
7     film.film_id AS FID,
8     film.title AS title,
9     film.description AS description,
10    category.name AS category,
11    film.rental_rate AS price,
12    film.length AS length,
13    film.rating AS rating,
14    GROUP_CONCAT(CONCAT(actor.first_name, _utf8mb4' ', actor.last_name)
15                  SEPARATOR ', ') AS actors
16 FROM
17     category LEFT JOIN film_category
18     ON category.category_id = film_category.category_id
19     LEFT JOIN film
20     ON film_category.film_id = film.film_id
21     JOIN film_actor
22     ON film.film_id = film_actor.film_id
23     JOIN actor
24     ON film_actor.actor_id = actor.actor_id
25 GROUP BY film.film_id, category.name;
```

## 1.9 Ejercicios

### 1.9.1 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 columnas: nombre y apellidos del cliente concatenados, teléfono, ciudad, país, fecha\_pago, total del pago, id de la transacción

2. Escriba una vista que se llame `listado_pedidos_clientes` que muestre un listado donde aparezcan todos los clientes y los pedidos que ha realizado cada uno de ellos. La vista deberá tener las siguientes columnas: código del cliente, nombre y apellidos del cliente concatenados, teléfono, ciudad, país, código del pedido, fecha 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 la 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.
8. Elimine las vistas que ha creado en los pasos anteriores.

## 2 Referencias

- [MySQL Views](#).
- [SQL Views](#).
- [Documentación oficial de CREATE VIEW en MySQL](#).
- [Documentación oficial de DROP VIEW en MySQL](#).

## 3 Licencia

Esta página forma parte del curso Bases de Datos de José Juan Sánchez Hernández y su contenido se distribuye bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.