

---

## **Unidad. Vistas.**

JOSÉ JUAN SÁNCHEZ HERNÁNDEZ

# Índice general

<b>1</b>	<b>Vistas</b>	<b>2</b>
1.1	Crear o modificar una vista . . . . .	2
1.2	Eliminar una vista . . . . .	3
1.3	Renombrar una vista . . . . .	3
1.4	Consultar el listado de vistas disponibles . . . . .	4
1.5	Consultar la sentencia que se utilizó para crear una vista . . . . .	4
1.6	Ejemplos . . . . .	4
1.6.1	Base de datos: Sakila . . . . .	4
1.6.2	Modelo entidad/relación . . . . .	5
1.6.3	Base de datos para MySQL . . . . .	5
1.6.4	Vistas utilizadas en la base de datos Sakila . . . . .	5
<b>2</b>	<b>Referencias</b>	<b>7</b>
<b>3</b>	<b>Licencia</b>	<b>8</b>

# Capítulo 1

## Vistas

### 1.1 Crear o modificar una vista

```
CREATE
  [OR REPLACE]
  [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
  [DEFINER = user]
  [SQL SECURITY { DEFINER | INVOKER }]
  VIEW view_name [(column_list)]
  AS select_statement
  [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.

```
CREATE VIEW resumen_pedidos AS
SELECT
  pedido.codigo_pedido,
  pedido.fecha_pedido,
  cliente.nombre_cliente,
  SUM(detalle_pedido.cantidad * detalle_pedido.precio_unidad) AS total
FROM
  cliente INNER JOIN pedido
    ON cliente.codigo_cliente = pedido.codigo_cliente
  INNER JOIN detalle_pedido
    ON pedido.codigo_pedido = detalle_pedido.codigo_pedido
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.

```
CREATE VIEW resumen_pedidos (codigo_pedido, fecha_pedido, nombre_cliente, total)
AS
SELECT
    pedido.codigo_pedido,
    pedido.fecha_pedido,
    cliente.nombre_cliente,
    SUM(detalle_pedido.cantidad * detalle_pedido.precio_unidad)
FROM
    cliente INNER JOIN pedido
        ON cliente.codigo_cliente = pedido.codigo_cliente
    INNER JOIN detalle_pedido
        ON pedido.codigo_pedido = detalle_pedido.codigo_pedido
GROUP BY pedido.codigo_pedido
```

#### Referencias:

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

## 1.2 Eliminar una vista

```
DROP VIEW [IF EXISTS]
    view_name [, view_name] ...
[RESTRICT | CASCADE]
```

#### Ejemplo:

```
DROP VIEW resumen_pedidos;
```

#### Referencias:

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

## 1.3 Renombrar una vista

```
RENAME TABLE
    tbl_name TO new_tbl_name
    [, tbl_name2 TO new_tbl_name2] ...
```

**Ejemplo:**

```
RENAME TABLE old_table TO new_table;
```

**Referencias:**

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

## 1.4 Consultar el listado de vistas disponibles

```
SHOW FULL TABLES;
```

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

**Referencias:**

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

## 1.5 Consultar la sentencia que se utilizó para crear una vista

```
SHOW CREATE VIEW view_name
```

**Referencias:**

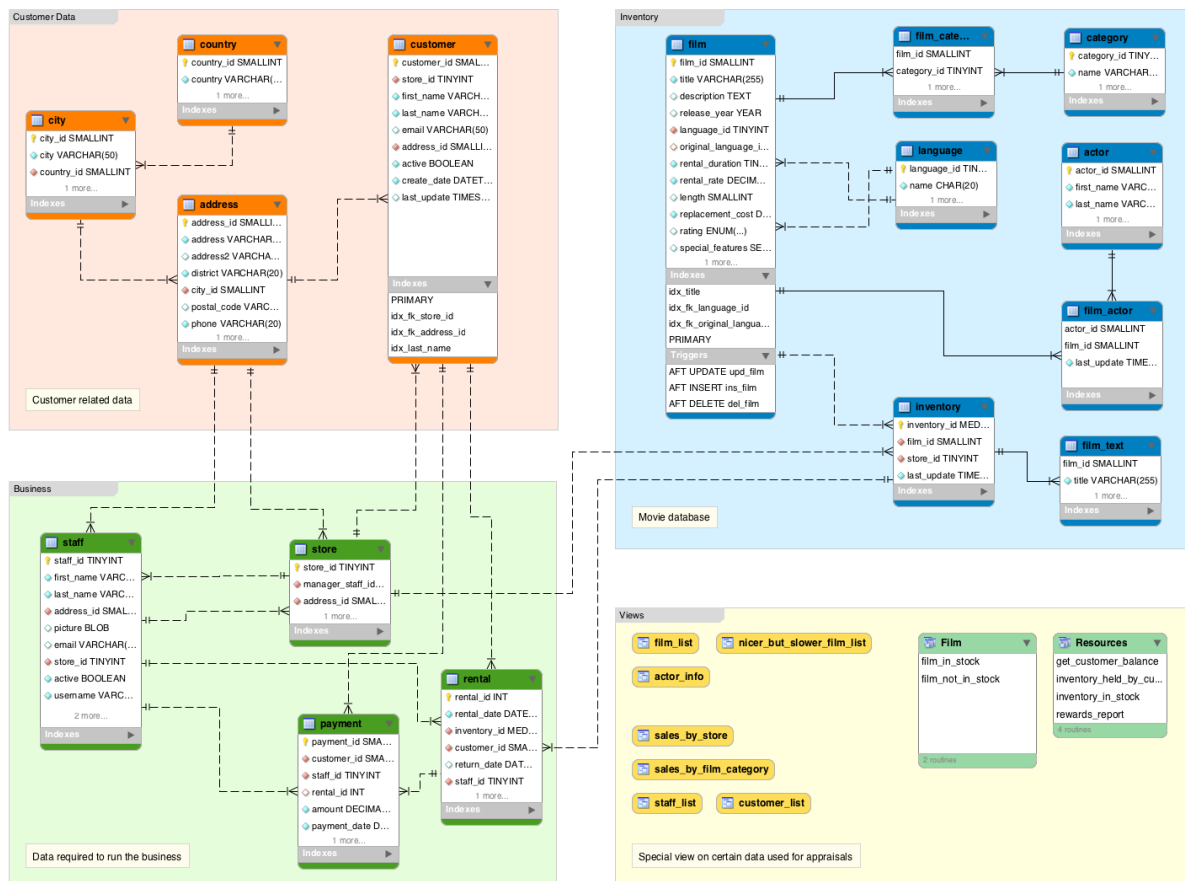
- [Documentación oficial de SHOW CREATE VIEW en MySQL.](#)

## 1.6 Ejemplos

### 1.6.1 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**.

## 1.6.2 Modelo entidad/relación



## 1.6.3 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-sakila.html).

## 1.6.4 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.

```
--
-- View structure for view `customer_list`
--
```

```
CREATE VIEW customer_list AS
SELECT
  cu.customer_id AS ID,
  CONCAT(cu.first_name, _utf8mb4' ', cu.last_name) AS name,
  a.address AS address,
  a.postal_code AS `zip code`,
  a.phone AS phone,
  city.city AS city,
  country.country AS country,
  IF(cu.active, _utf8mb4'active',_utf8mb4'') AS notes,
  cu.store_id AS SID
FROM
  customer AS cu JOIN address AS a
    ON cu.address_id = a.address_id
    JOIN city
    ON a.city_id = city.city_id
  JOIN country
    ON city.country_id = country.country_id;
```

```
--
-- 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;
```

## Capítulo 2

# Referencias

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



## Capítulo 3

# Licencia

El contenido de esta web está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.