* Analizar de qué es la base de datos y para qué problema se necesita.

La base de datos está diseñada para representar una tienda de alquiler de DVD.

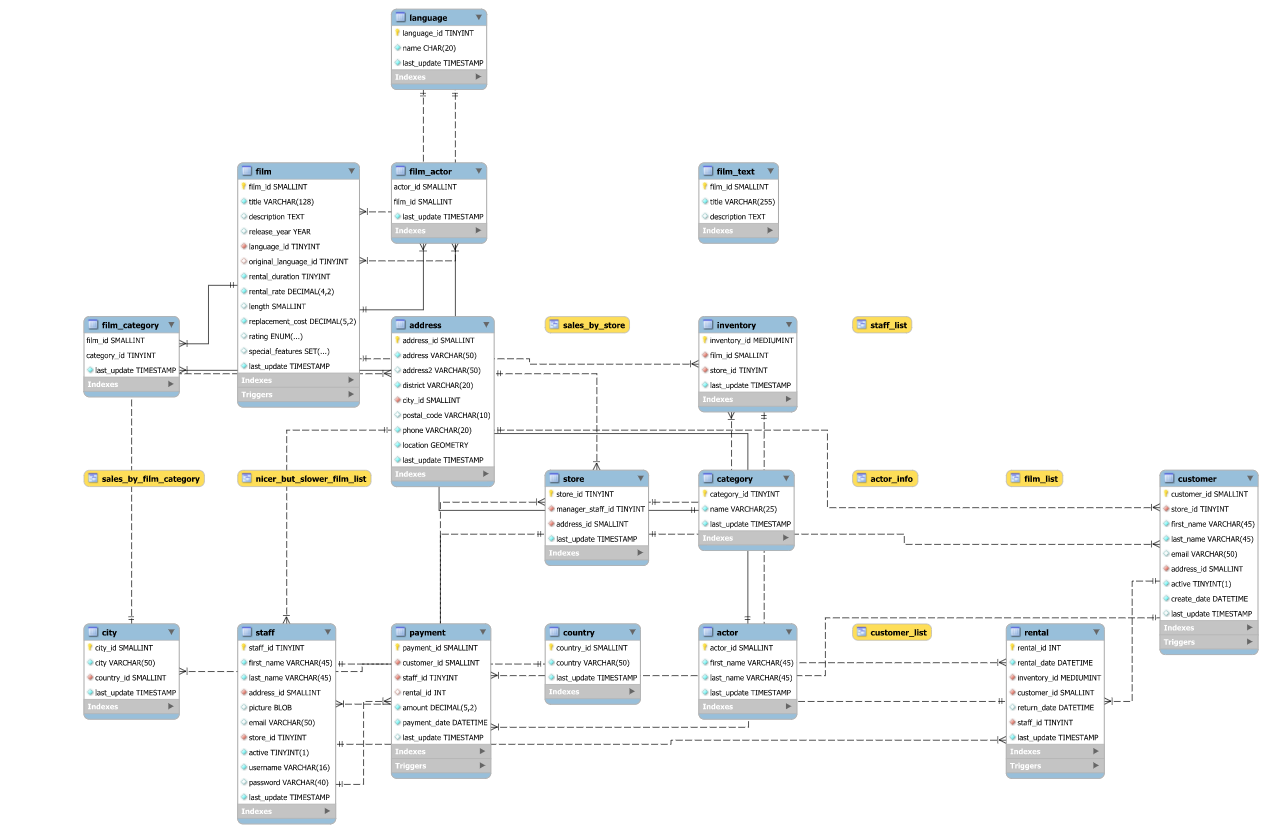
La base de datos tiene por objetivo es proporcionar un entorno de pruebas, ya sea para desarrollar software o probar consultas sql y así poder realizar test con las posibilidades que brinda MySQL. La estructura de Sakila es compleja, posee 1000 registros y múltiples relaciones.  
  
La base de datos [Sakila](http://downloads.mysql.com/docs/sakila-db.zip) también contiene ejemplos de vistas, procedimientos almacenados y triggers o disparadores.  
Esta base de datos supone un tienda de alquiler de películas que puede tener sucursales y vendedores,

El desarrollo de la base de datos de muestra Sakila comenzó a principios de 2005. Los primeros diseños se basaron en base de datos utilizada en el documento técnico de Dell Tres enfoques para aplicaciones MySQL en Dell PowerEdge ervidores

Donde la base de datos de muestra de Dell fue diseñada para representar una tienda de DVD en línea, la muestra de Sakila

El nombre del delfín de MySQL (el logo) es "Sakila", que fué elegido por los fundadores de MySQL AB de una gran lista de nombres sugerida por los usuarios en el concurso "Name the Dolphin" (ponle nombre al delfín). El nombre ganador fue enviado por Ambrose Twebaze, un desarrollador de software Open Source de Swaziland, África. Según Ambrose, el nombre femenino de Sakila tiene sus raíces en SiSwate, el idioma local de Swaziland. Sakila también es el nombre de una ciudad en Arusha, Tanzania, cerca del país de origen de Ambrose, Uganda.

● Definir la estructura de la base de datos (SQL o NoSQL).





* Crear la base de datos y añadir registros muestra (MySQL o MongoDB).

Actor  
actor\_info

address

category

city

country

customer

customer\_list

film

film\_actor

film\_category

film\_list

film\_text

inventory

language

nicer\_but\_slower\_film\_list

payment

rental

sales\_by\_film\_category

sales\_by\_store

staff

staff\_list

store

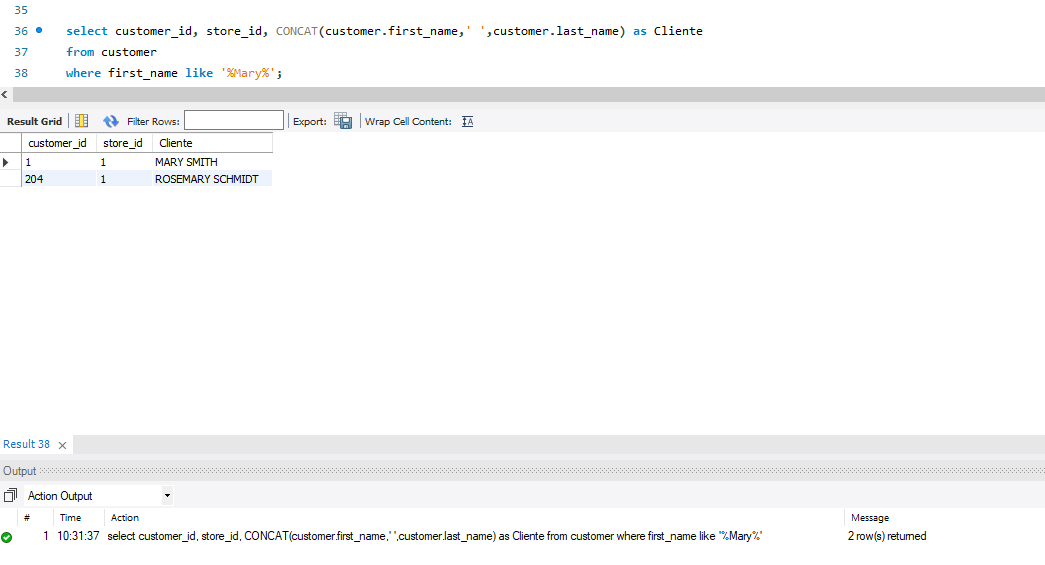
* Escribir preguntas que se puedan contestar a partir de la base de datos que se definió ejemplo: ¿cuántas personas de tal edad hay en la base?
* Definir consultas (SQL o JSON) que permitan responder las preguntas.

El proceso supone que, para alquilar una película, primero deberemos confirmar que la película está disponible o en stock, y luego deberemos consultar si el cliente existe en la base de datos y asignaremos la película o DVD al cliente. También deberemos insertar la venta en la tabla de pagos. Dependiendo de las reglas de negocio, también puede ser necesario para comprobar si el cliente tiene un saldo pendiente.

1. Verificar si está registrado el cliente por el nombre   
     
   select customer.customer\_id, customer.store\_id, CONCAT(customer.first\_name,' ',customer.last\_name) as Cliente

from customer

where customer.first\_name like '%Mary%';

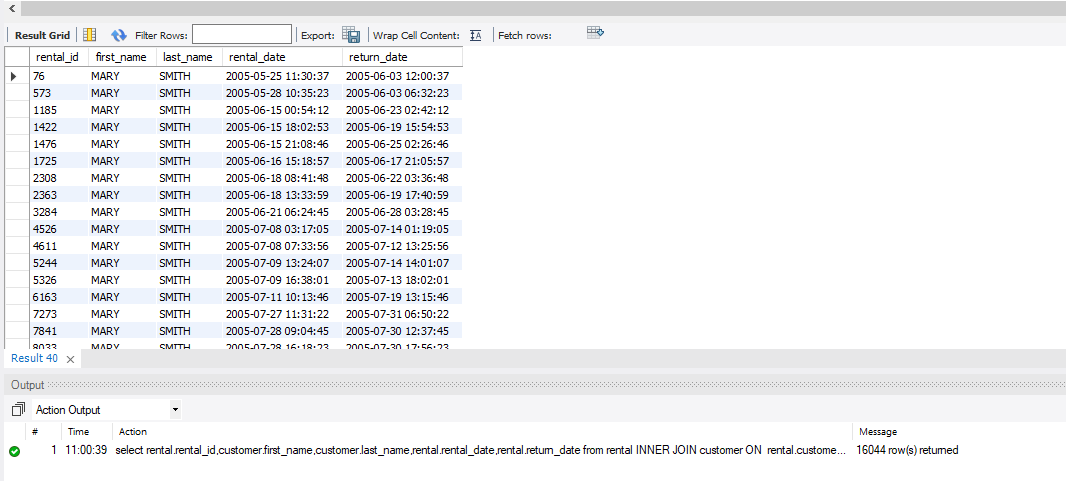


1. Queremos saber que clientes alquilaron películas   
     
   select rental.rental\_id,customer.first\_name,customer.last\_name,rental.rental\_date,rental.return\_date

from rental

INNER JOIN customer  
ON rental.customer\_id=customer.customer\_id

.



1. Tenemos dos tiendas o sucursales que generan una lista diaria de los alquileres atrasados para que los clientes pueden ser contactados y pedirles que devuelvan la película.  
     
   SELECT CONCAT(customer.last\_name, ', ', customer.first\_name) AS customer, address.phone, film.title,customer.email,rental.return\_date, payment.payment\_date, payment.amount

FROM rental

INNER JOIN customer ON rental.customer\_id = customer.customer\_id

INNER JOIN address ON customer.address\_id = address.address\_id

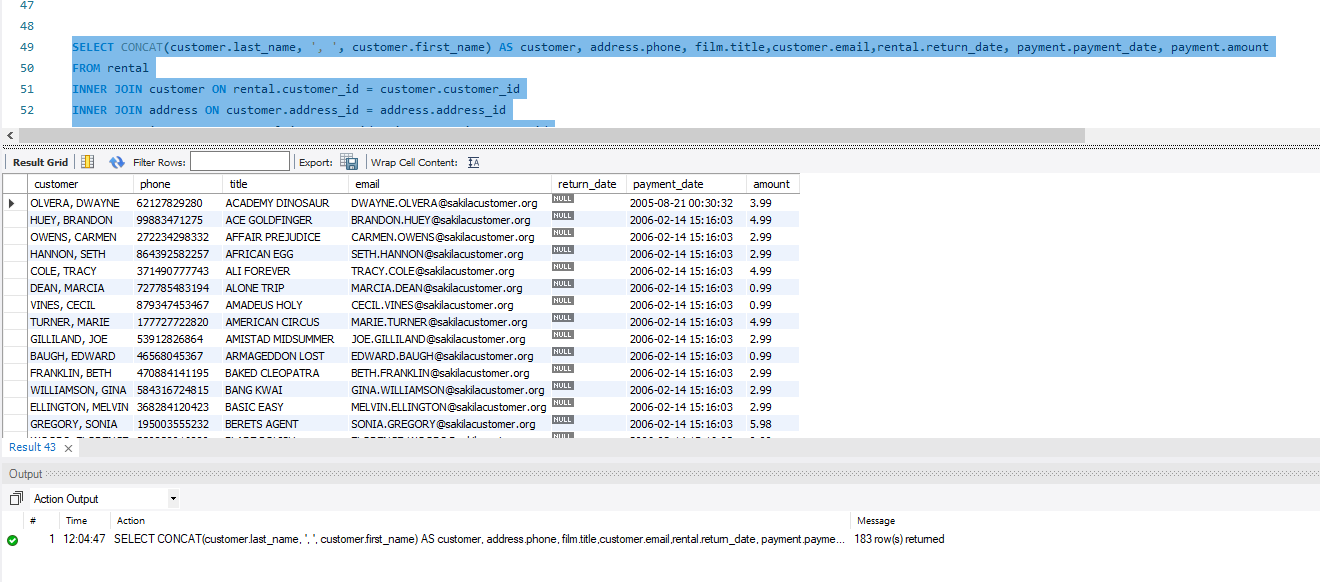
INNER JOIN inventory ON rental.inventory\_id = inventory.inventory\_id

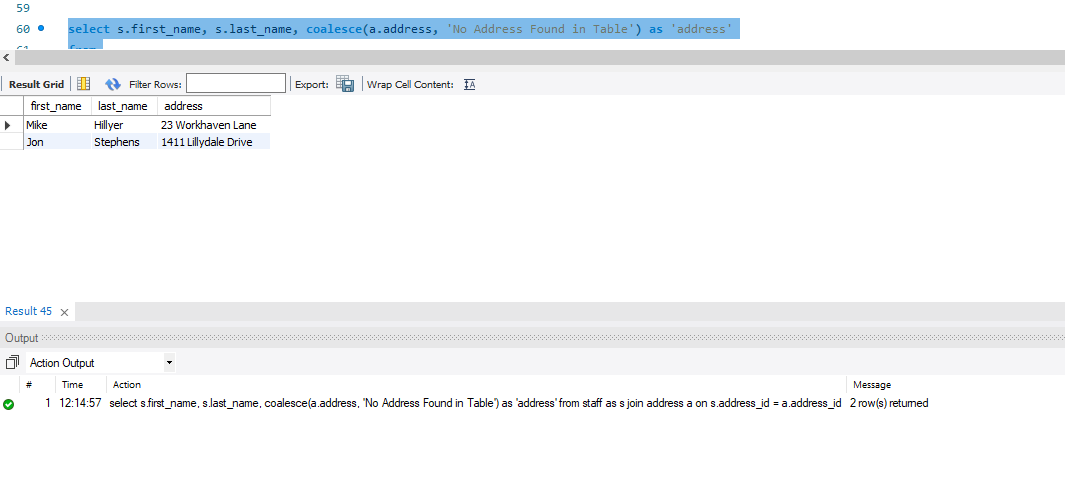
INNER JOIN film ON inventory.film\_id = film.film\_id

INNER JOIN payment ON payment.rental\_id = rental.rental\_id

WHERE rental.return\_date IS NULL

AND rental\_date + INTERVAL film.rental\_duration DAY < CURRENT\_DATE();



1. mostrar el nombre y apellido, así como la dirección, de cada miembro del personal  
     
   
2. Enumerar cada película y el número de actores que figuran en esa película  
   select f.title, count(fa.actor\_id) as 'Number of Actors'

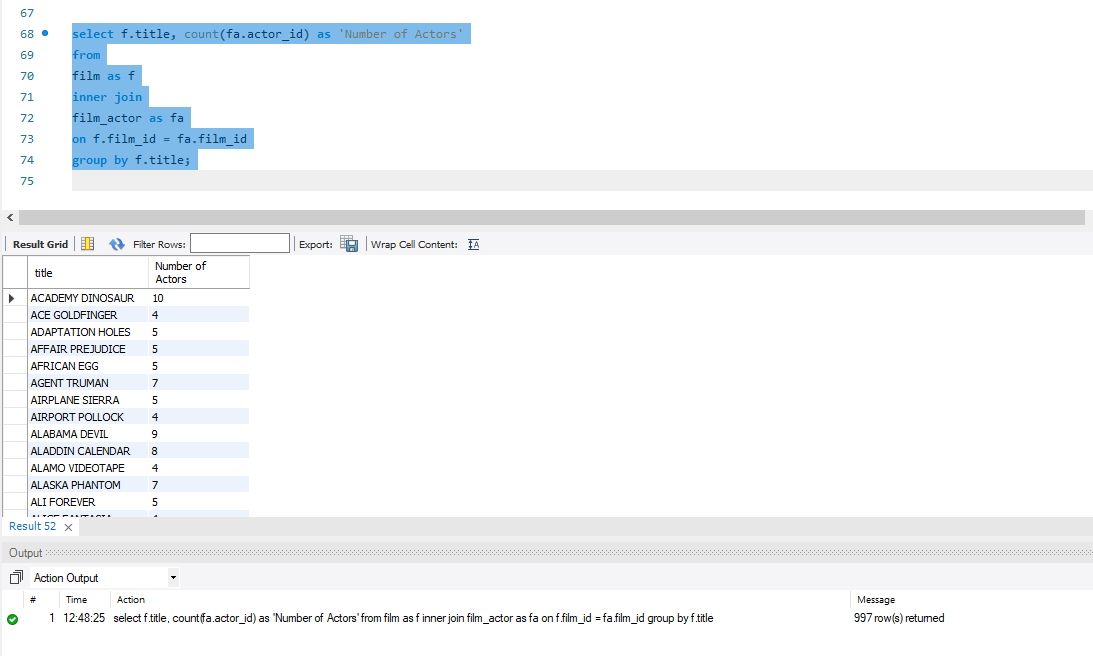
from

film as f

inner join

film\_actor as fa

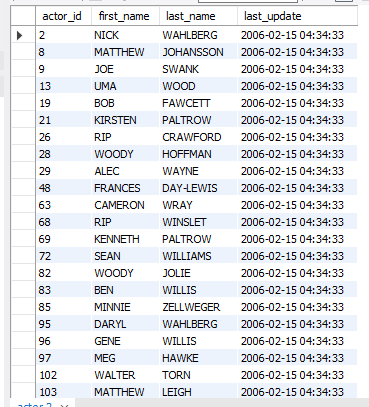
on f.film\_id = fa.film\_id   
group by f.title;.



.

1. A
2. A
3. A
4. A

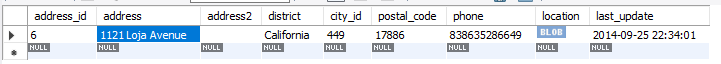
1. .Actores que tengan una w en el nombre o en el apellido

SELECT \* FROM sakila.actor  
where first\_name like '%w%' or last\_name like '%w%';  
  
  


1. Direcciones de Texas que tengan ‘866’ en el número de teléfono

SELECT \* FROM sakila.address

where district='california' and phone like '%866%';

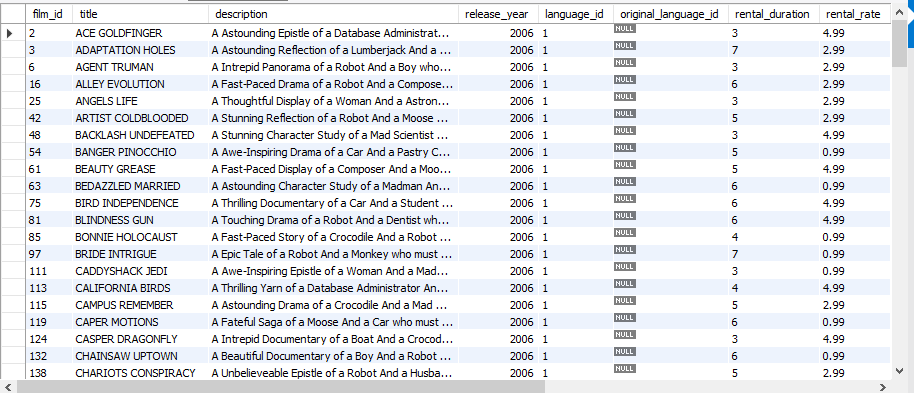


1. Películas en cuya descripción describa Robots o autos que duren más de 200 minutos

SELECT \* FROM sakila.film

where (description like '%robot%' or description like '%car%')

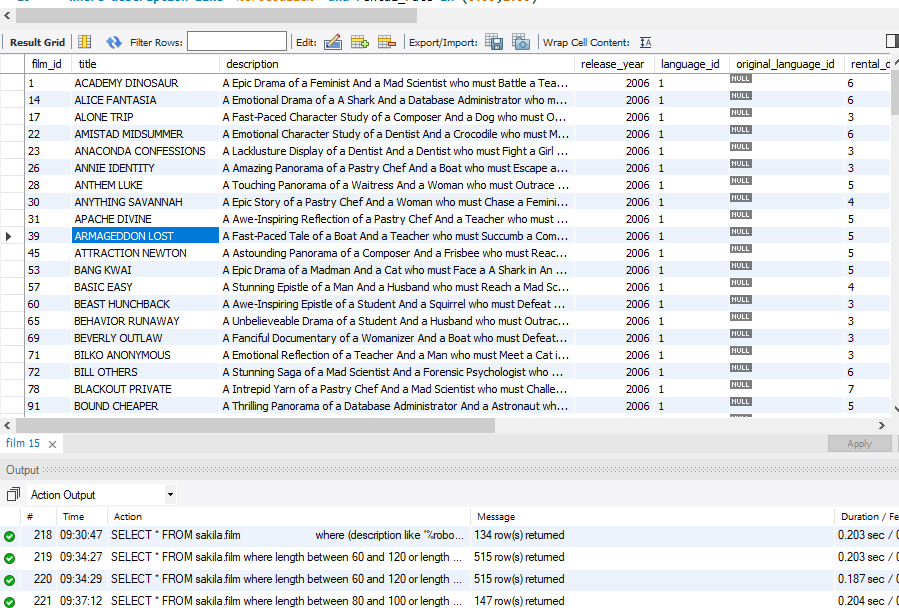
and length>200;



1. Películas que duren entre 80 y 100 minutos o entre 20 y 40 minutos

SELECT \* FROM sakila.film

where length between 80 and 100 or length between 20 and 40;



1. Películas que cuesten 0.50, 2.99 y tengan un rating ‘g’ o ‘r’ y que hablen de  
   héroes.}  
     
   SELECT \* FROM sakila.film;

where description like '%heroes%'

and rental\_rate in (0.50,2.99)

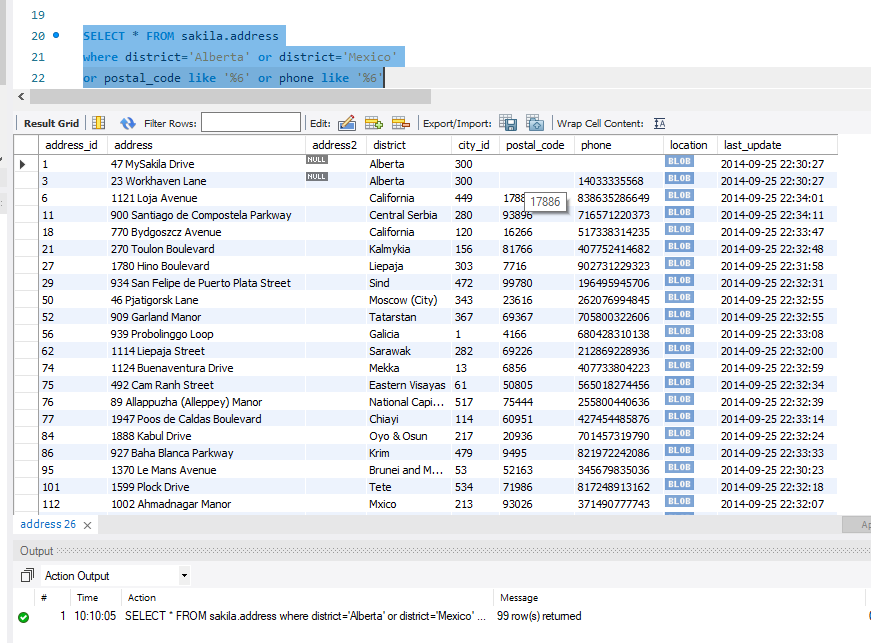
and rating in ('R','G');

1. Direcciones de Alberta o de Mexico o que su código postal acabe en 6 o que  
   teléfono acabe en 6

SELECT \* FROM sakila.address

where district='Alberta' or district='Mexico'

or postal\_code like '%6' or phone like '%6'



1. Encuentra DVD atrasados  
   Muchas tiendas de DVD producen una lista diaria de alquileres atrasados ​​para que los clientes puedan ser contactados y se les solicite  
   devolver sus DVD atrasados.  
    Para crear una lista de este tipo, busque películas en la tabla de alquiler con una fecha de devolución NULA y donde el alquiler  
   la fecha es más antigua que la duración del alquiler especificada en la tabla de películas. Si es así, la película está atrasada y  
   debemos producir el nombre de la película junto con el nombre del cliente y el número de teléfono

SELECT CONCAT(customer.last\_name, ', ', customer.first\_name) AS customer,

address.phone, film.title

FROM rental INNER JOIN customer ON rental.customer\_id = customer.customer\_id

INNER JOIN address ON customer.address\_id = address.address\_id

INNER JOIN inventory ON rental.inventory\_id = inventory.inventory\_id

INNER JOIN film ON inventory.film\_id = film.film\_id

WHERE rental.return\_date IS NULL

AND rental\_date + INTERVAL film.rental\_duration DAY < CURRENT\_DATE()

ORDER BY title;

* Transformar consultas a vistas.