```
use sakila;

-- for every film, return film_name , language

select film.title , language.name
from film
join language
on film.language_id = language.language_id;

-- writing the complete name is becoming difficult
-- aliases

select f.title , l.name
from film f
join language l
on f.language_id = l.language_id;


-- Write a SQL query to display the first name, last name,
-- and email of all customers who rented a movie.

select distinct c.first_name , c.last_name , c.email
from customer c
join rental r
on c.customer_id = r.customer_id;


-- Retrieve a list of film titles along with their corresponding
-- category name for all movies in the sakila database.


select film.title, fc.film_id , fc.category_id , c.name
from film
join film_category fc
on film.film_id = fc.film_id
join category c
on c.category_id = fc.category_id;

-- Display the staff first name, last name, and the address
-- of the store they are currently working at
-- staff * store
-- store * address

select st.first_name , st.last_name , a.address
```

```
from staff st
join store s
on st.store_id = s.store_id
join address a
on s.address_id = a.address_id;


-- find details of customers with same last name

select c1.customer_id , c1.first_name , c1.last_name , c2.customer_id , c2.first_name ,
c2.last_name
from customer c1
join customer c2
on c1.last_name = c2.last_name and c1.customer_id < c2.customer_id;

-- compound join
-- join in which you have multiple join conditions on different columns

-- customers

-- Mohit sharma.       Mohit sharma
-- Rachit sharma               Rachit sharma
-- Gaurav sharma               Gaurav sharma

-- Mohit * rachit
-- Mohit * gaurav
-- Rachit * gaurav
```