```sql
use sakila;

-- GET YOU ALL THE COLUMNS OF THE FILM TABLE
SELECT * FROM FILM;

SELECT 50;

SELECT * FROM ACTOR;

SELECT TITLE , RELEASE_YEAR FROM FILM;

-- ALL THE ROWS IN THE LENGTH COLUMN WILL BE DIVIDED BY 60
SELECT TITLE , LENGTH / 60 FROM FILM;

-- USE ALIAS IF YOU WANT A DIFFERENT NAME OF THE COLUMN IN THE OUTPUT
SELECT TITLE , LENGTH / 60 AS LENGTH_HOURS FROM FILM;

SELECT distinct RELEASE_YEAR FROM FILM;

SELECT distinct RATING FROM FILM;

select distinct rating , release_year from film;

-- if you are using distinct you can not have columns without distinct
--  the below query will not work
select release_year , distinct rating from film;

select distinct * from film;

create table students (
      student_id int Primary key,
    first_name varchar(60)
);

-- inserting from a differnet database

insert into students (student_id , first_name)
select student_id , first_name
from scaler.students;

select * from students;
select * from film where rating = 'PG-13';

-- Name of those movies which were release in the year 2012;

explain select title from film
 where release_year = 2012;

 explain select title from film where film_id = 10;

 -- find all those movies which were released in 2006 and have PG - 13 rating

 select title from film
 where release_year = 2006 and rating = 'PG-13' ;

  select title, release_year from film
 where release_year = 2008 or release_year = 2012;

 select title, release_year from film
 where release_year != 2006;

 select title, release_year from film
 where release_year != 2006;

select title, release_year from film
 where release_year <> 2006;

 select title, release_year from film
 where NOT release_year = 2006;

 -- 2008 , 2012 , 2016

 select title from film
 where release_year = 2008 or release_year = 2012 or release_year = 2016;

 -- where release_year is either of these values
 select title from film
 where release_year IN (2008 , 2012 ,2016);

 select title from film
 where release_year NOT IN (2008 , 2012 ,2016);

 -- sorting the data
 -- sort the data according to the rental rate

 select * from film
 order by rental_rate desc;

 select * from film
 order by rental_rate desc, length desc;

 -- -- FIND DISTINCT RELEASE YEAR AND RATING WHERE LENGTH IS 185 , 181 ,
176
-- ORDER BY RELEASE_YEAR

 select distinct release_year , rating , replacement_cost from film
 where length IN (185 , 181 , 176)
 order by replacement_cost;

-- Error Code: 3065. Expression #1 of ORDER BY clause is not in SELECT
list, references column 'sakila.film.replacement_cost' which is not in
SELECT list; this is incompatible with DISTINCT


select release_year , rating from film
 where length IN (185 , 181 , 176)
 order by replacement_cost;
```