(My notes):

1. Password
2. SHOW databases;
3. USE sakila;
4. SHOW TABLES;
5. Ctrl c *(break out)*
6. \! cls *(clear screen)*
7. DESCRIBE;
8. SELECT last\_name FROM actor;
9. SELECT first\_name, last\_name FROM actor;
10. SELECT \* FROM actor; *(selects all)*
11. SELECT first\_name FROM actor WHERE first\_name = 'JOHN'; *(condition)*
12. SELECT \* FROM actor WHERE first\_name LIKE '%p';
13. SELECT \* FROM actor WHERE first\_name LIKE 'a%';
14. SELECT \*

-> FROM actor

-> ORDER BY first\_name ASC

-> LIMIT 10;

1. SELECT title, release\_year, language\_id

-> FROM film

-> WHERE language\_id = (SELECT language\_id FROM language WHERE name = 'English');

1. SELECT f.title, f.release\_year, f.language\_id

-> FROM film f

-> JOIN language l ON f.language\_id = l.language\_id

-> WHERE l.name = 'English';

1. SELECT c.first\_name, c.last\_name, f.title, p.amount

-> FROM customer c

-> JOIN rental r ON c.customer\_id = r.customer\_id

-> JOIN payment p ON r.rental\_id = p.rental\_id

-> JOIN inventory i ON r.inventory\_id = i.inventory\_id

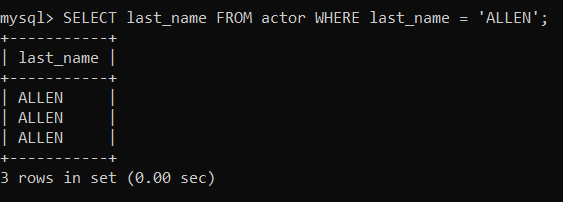
-> JOIN film f ON i.film\_id = f.film\_id

-> ORDER BY p.amount DESC

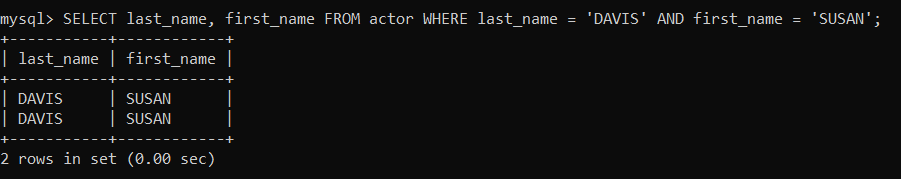
-> LIMIT 5;

**Questions from 1-9.**

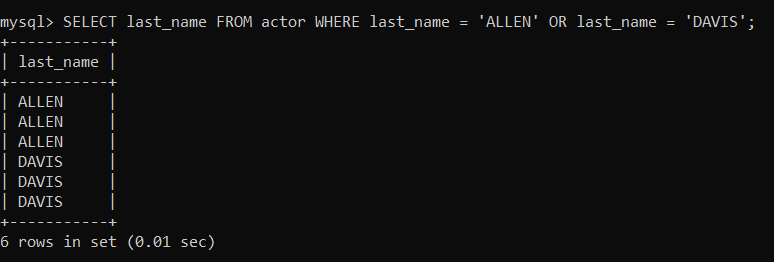
1. Return all actors whose last names are ALLEN from the actor table

****

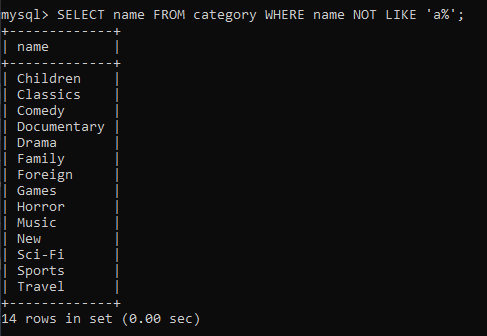
1. Return all the actors who’s last\_name column has a value DAVIS, and the row’s first\_name has a value SUSAN.

****

1. Return all actors where last\_name column has a value ALLEN or DAVIS.

****

1. Return the number of categories which have a name that does not start with the letter A.

****

1. Return all actor names and the name of each film title they were in.

**Text

Description automatically generated**

1. Return the title, release year, and category of all movies.

Text

Description automatically generated with medium confidence

1. Return how much JOHNNY TURPIN has spent on movie rentals.

**Text

Description automatically generated**

1. Return the name of the customer that has spent the most of movie rentals, include the amount they have spent.

Text

Description automatically generated

1. Return all movie title, release year, actor name and category where the actor first\_name is ‘Sean’

**A picture containing graphical user interface

Description automatically generated**

1. 10 challenging SQL statements (Just select, no create, update or delete statements)**.**

**My own exercises:**

1. Find the names of the actors who have appeared in more than 10 films, along with the number of films they have appeared in.

**Graphical user interface

Description automatically generated with medium confidence**

1. Find the number of rentals for each customer, including the customer’s name and email address.

**Graphical user interface, text

Description automatically generated**

1. Find the names of the customers who have rented a movie in the month of February 2006.

**Text

Description automatically generated**

1. Get the total number of rentals for each movie sorted in descending order.

**Text

Description automatically generated with medium confidence**

1. Find the names of all customers who have rented more than 10 movies in total and sort them in descending order.

**Text

Description automatically generated**

1. Find the movies that have been rented by customer with ID 1.

**Text

Description automatically generated**

1. Find the movies that have neem rented more than 30 times.

**Text

Description automatically generated**

1. Find the titles of all movies that have been rented by a customer whose last name starts with ‘S2’.

**Text

Description automatically generated**

1. Find the titles and release years of all films in the "Horror" category.

**A picture containing text

Description automatically generated**

1. Find the names of the customers that have rented a movie in the year 2006.

**Text

Description automatically generated**