Lab | SQL Queries - Lesson 2.7 Part 1

In this activity we are going to do some database maintenance on the sakila database. In the current database we only have information on movies for the year 2006. Now we have received the film catalog for 2020 as well. For this new data we will create another table and bulk insert all the data there. Code to create a new table has been provided below.

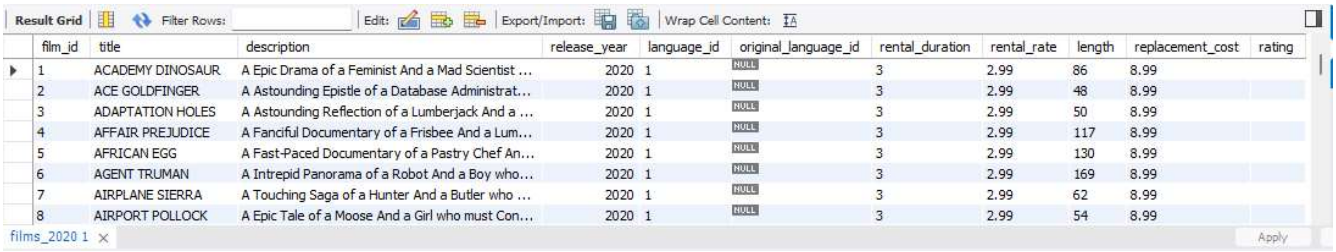
```
USE sakila;

drop table if exists films_2020;
CREATE TABLE `films_2020` (
  `film_id` smallint(5) unsigned NOT NULL AUTO_INCREMENT,
  `title` varchar(255) NOT NULL,
  `description` text,
  `release_year` year(4) DEFAULT NULL,
  `language_id` tinyint(3) unsigned NOT NULL,
  `original_language_id` tinyint(3) unsigned DEFAULT NULL,
  `rental_duration` int(6),
  `rental_rate` decimal(4,2),
  `length` smallint(5) unsigned DEFAULT NULL,
  `replacement_cost` decimal(5,2) DEFAULT NULL,
  `rating` enum('G','PG','PG-13','R','NC-17') DEFAULT NULL,
  PRIMARY KEY (`film_id`),
  CONSTRAINT FOREIGN KEY (`original_language_id`) REFERENCES `language` (`language_id`)
ON DELETE RESTRICT ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1003 DEFAULT CHARSET=utf8;

#We have just one item for each film, and all will be placed in the new table.
# For 2020, the rental duration will be 3 days, with an offer price of `2.99€` and a replacement cost of `8.99€`
# (these are all fixed values for all movies for this year).
# The catalog is in a CSV file named **films_2020.csv** that can be found at `files_for_lab` folder.
```

Instructions

```
# - Add the new films to the database.
# - Update information on `rental_duration`, `rental_rate`, and `replacement_cost`.
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating
1	ACADEMY DINOSAUR	A Epic Drama of a Feminist And a Mad Scientist ...	2020	1	NULL	3	2.99	86	8.99	
2	ACE GOLDFINGER	A Astounding Epistle of a Database Administrat...	2020	1	NULL	3	2.99	48	8.99	
3	ADAPTATION HOLES	A Astounding Reflection of a Lumberjack And a ...	2020	1	NULL	3	2.99	50	8.99	
4	AFFAIR PREJUDICE	A Fanciful Documentary of a Frisbee And a Lum...	2020	1	NULL	3	2.99	117	8.99	
5	AFRICAN EGG	A Fast-Paced Documentary of a Pastry Chef An...	2020	1	NULL	3	2.99	130	8.99	
6	AGENT TRUMAN	A Intrepid Panorama of a Robot And a Boy who...	2020	1	NULL	3	2.99	169	8.99	
7	AIRPLANE SIERRA	A Touching Saga of a Hunter And a Butler who ...	2020	1	NULL	3	2.99	62	8.99	
8	AIRPORT POLLOCK	A Epic Tale of a Moose And a Girl who must Con...	2020	1	NULL	3	2.99	54	8.99	

films_2020 1 | Apply

Lab | SQL Queries - Lesson 2.7 Part 2

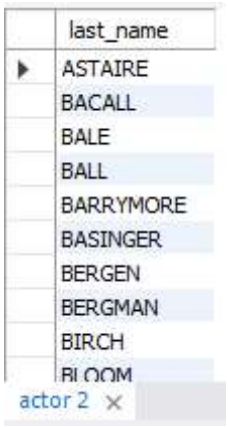
In this lab, you will be using the [Sakila](https://dev.mysql.com/doc/sakila/en/) database of movie rentals. You have been using this database for a couple labs already, but if you need to get the data again, refer to the official [installation link](https://dev.mysql.com/doc/sakila/en/sakila-installation.html).

The database is structured as follows:
![DB schema](https://education-team-2020.s3-eu-west-1.amazonaws.com/data-analytics/database-sakila-schema.png)

Instructions

#1. In the table actor, what last names are not repeated? For example if you would sort the data in the table actor by last_name,
#you would see that there is Christian Arkoyd, Kirsten Arkoyd, and Debbie Arkoyd.
#These three actors have the same last name. So we do not want to include this last name in our output.

#Last name "Astaire" is present only one time with actor "Angelina Astaire", hence we would want this in our output list.



#2. Which last names appear more than once? We would use the same logic as in the previous question but this time we want to include
the last names of the actors where the last name was present more than once

	last_name
▶	AKROYD
	ALLEN
	BAILEY
	BENING
	BERRY
	BOLGER
	BRODY
	CAGE
	CHASE
	CRAWFORD

actor 3 ×

#3. Using the rental table, find out how many rentals were processed by each employee.

	staff_id	count(*)
▶	1	8041
	2	8004

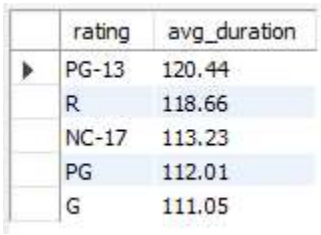
#4. EDITED QUESTION: Using the film table, find out how many films have a replacement cost over 25.

	replacement_cost	count(*)
▶	26.99	46
	28.99	41
	27.99	53
	29.99	53
	25.99	43

#5. Using the film table, find out how many films there are of each rating.

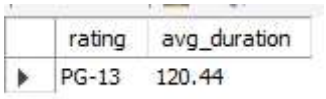
	rating	num_films
▶	PG	194
	G	178
	NC-17	210
	PG-13	223
	R	195

#6. What is the mean length of the film for each rating type. Round off the average lengths to two decimal places



rating	avg_duration
PG-13	120.44
R	118.66
NC-17	113.23
PG	112.01
G	111.05

#7. Which kind of movies (rating) have a mean duration of more than two hours?



rating	avg_duration
PG-13	120.44