**Lab | SQL Queries - Lesson 2.7 Part 1**

**Instructions**

* Add the new films to the database.
* Update information on rental\_duration, rental\_rate, and replacement\_cost.

################### SQL LAB 2.7 PART 1 #######################

*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;*

*######## Load new data into table #########*

*show variables like 'local\_infile';*

*set global local\_infile = 1;*

*load data local infile 'Users/siljaloik/Desktop/IronHack/Day9/dataV3\_Lesson\_2.7\_lab/files\_for\_part1/films\_2020\_lf.csv'*

*into table films\_2020*

*fields terminated by ',';*

*####### updating new film table ###########*

*update films\_2020*

*set rental\_duration = 3, rental\_rate = 2.99, replacement\_cost = 8.99*

*where description = 2020;*

*select \* from films\_2020;*

**Lab | SQL Queries - Lesson 2.7 Part 2**

################### SQL LAB 2.7 PART 2 #######################

*#Instructions*

*#1) In the table actor, which are the actors whose 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.*

*select COUNT(first\_name) as first\_name\_count, last\_name*

*from sakila.actor*

*group by last\_name*

*having first\_name\_count = 1;*

*#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*

*select COUNT(first\_name) as first\_name\_count, last\_name*

*from sakila.actor*

*group by last\_name*

*having first\_name\_count > 1;*

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

*select staff\_id, count(\*)*

*from sakila.rental*

*group by staff\_id;*

*#4)Using the film table, find out how many films were released each year.*

*select release\_year, count(title) as films\_per\_year*

*from sakila.film*

*group by release\_year*

*order by release\_year;*

*#5) Using the film table, find out for each rating how many films were there.*

*select rating, count(title) as films\_per\_rating*

*from sakila.film*

*group by rating;*

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

*select rating, avg(length) as avgerage\_film\_length*

*from sakila.film*

*group by rating*

*order by avgerage\_film\_length desc;*

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

*select rating, round(avg(length),2) as avgerage\_film\_length*

*from sakila.film*

*group by rating*

*having avgerage\_film\_length > 120*

*order by avgerage\_film\_length desc;*