**Lab | SQL Queries 8**

*################# LAB 2.8 Instructions ################*

*#1. Rank films by length (filter out the rows that have nulls or 0s in length column). In your output, only select the columns title, length, and the rank.*

*SELECT title, length, rank() over (order by length desc) AS 'Rank'*

*FROM sakila.film*

*WHERE length is not NULL and length <> 0;*

*#2. Rank films by length within the rating category (filter out the rows that have nulls or 0s in length column). In your output, only select the columns title, length, rating and the rank.*

*SELECT title, length, rating, rank () over (partition by rating order by length desc) AS 'Rank'*

*FROM sakila.film*

*WHERE length is not NULL and length <> 0*

*ORDER BY length;*

*#3. How many films are there for each of the categories in the category table. Use appropriate join to write this query*

*SELECT COUNT(fca.film\_id), ca.name*

*FROM sakila.category ca*

*inner join sakila.film\_category fca on ca.category\_id = fca.category\_id*

*GROUP BY ca.name;*

*#4. Which actor has appeared in the most films?*

*SELECT t1.actor\_id, t1.first\_name, t1.last\_name, COUNT(t2.film\_id) as total\_films*

*FROM actor t1*

*join film\_actor t2 on t2.actor\_id = t1.actor\_id*

*GROUP BY t1.actor\_id*

*ORDER BY total\_films desc;*

*#5. Most active customer (the customer that has rented the most number of films)*

*SELECT customer\_id, COUNT(return\_date), COUNT(rental\_date)*

*FROM sakila.rental*

*GROUP BY customer\_id*

*ORDER BY COUNT(return\_date) DESC;*

*#Bonus: Which is the most rented film? The answer is Bucket Brotherhood This query might require using more than one join statement.*

*SELECT t1.title, COUNT(rental\_id) as rentals*

*FROM sakila.film t1*

*inner join sakila.inventory t2 on t1.film\_id = t2.film\_id*

*inner join sakila.rental t3 on t2.inventory\_id = t3.inventory\_id*

*GROUP BY t1.title*

*ORDER BY rentals DESC*

*limit 1;*