**Data Analysis Using MySQL**

use sakila;

-- **Task 1: Display the full names of actors available in database**

select concat(first\_name,' ',last\_name) as full\_name from actor;

-- **Task 2.a: Display the number of times each first names appears in the database**

select first\_name,count(\*) from actor group by first\_name;

-- **Task 2.b: What is the count of actors that have unique first names in db? display the first name of all** these actor.

select first\_name,count(\*) as count from actor group by first\_name order by count asc;

select first\_name,count(\*) as count from actor group by first\_name having count=1;

-- **Task 3.a: Display the number of times each last name appears in the db**

select last\_name,count(\*) as count from actor group by last\_name;

-- **Task 3.b: Display all unique last names in the db**

select last\_name,count(\*) as count from actor group by last\_name order by count asc;

select last\_name,count(\*) as count from actor group by last\_name having count=1;

-- **Task 4.a:Display the list of records from movies with the rating "R"**

select \* from film where rating='R';

-- **Task 4.b:Display the list of records from movies that are not rating "R"**

select \* from film where rating<>'R';

-- **Task 4.c:Display the list of records for the movies that are suitable for audience below 13 years of age.**

select \* from film where rating in('G','PG');

-- **Task 5.a:Display the list of records for the movies where the replacement cost is up to $11.**

select \* from film where replacement\_cost>11;

-- **Task 5.b:Display the list of records for the movies where the replacement cost is between $11 and $20**

select \* from film where replacement\_cost between 11 and 20;

-- **Task 5.c:Display the list of records for the movies in descending order of their replacement costs.**

select \* from film order by replacement\_cost desc;

-- **Task 6: Display the names of the top 3 movies with the gratest number of actors.**

select f.title,count(aa.actor\_id) as actor\_count from film f join film\_actor a on f.film\_id=a.film\_id

join actor aa on aa.actor\_id=a.actor\_id

group by f.title order by actor\_count desc limit 3;

-- **Task 7: Display the titles of the the movies atarting with letter 'K' and 'Q'**

select title from film where title like 'k%' or 'Q%';

-- **Task 8: The film 'Agent Truman' has been a great success. display the names of all actors who appeared** in this film.

select \* from film;

select \* from film\_actor where film\_id=6;

select \* from actor where actor\_id in(21,23,62,108,137,169,197);

select a.actor\_id,a.first\_name,a.last\_name from film f join film\_actor fa on f.film\_id=fa.film\_id join actor a on a.actor\_id=fa.actor\_id where f.title='agent truman';

-- **Task 9: identify and display the names of the movies in the family category**

select \* from film;

select \* from film\_category;

select \* from category;

select f.title,c.name from film f join film\_category fc on f.film\_id=fc.film\_id join category c on fc.category\_id=c.category\_id where c.name='Family';

-- **Task 10.a: Display the maximum,minimum and average rental rates of movies based on their rating. the output must be sorted in descending order of the average rental rates**.

select rating,max(rental\_rate) as maximum, min(rental\_rate) as minimum, avg(rental\_rate) as average from film group by rating order by average desc;

**-- Task 10.b:display the names of the most frequently rented movies in descending order, so that the management can maintain more copies of such movies.**

select \* from film;

select \* from inventory;

select f.title,count(i.inventory\_id) as most\_frequently\_rented\_movie from film f join inventory i on f.film\_id=i.film\_id group by f.title order by most\_frequently\_rented\_movie desc;

-- **Task 11 - calculate and display the number of movie categories where the average between the movie replacement**

-- cost and the rental rate is greater than $15

select \* from film;

desc film;

select \* from film\_category;

desc film\_category;

select \* from category;

desc category;

SELECT c.name,COUNT(\*) AS category\_count

FROM film f

JOIN film\_category fc ON fc.film\_id = f.film\_id

JOIN category c ON c.category\_id = fc.category\_id

WHERE (f.replacement\_cost + f.rental\_rate) / 2 > 15 group by c.name;

-- **Task 12: Display the film categories in which the number of movies is grater than 70.**

select \* from category;

select \* from film;

select \* from film\_category;

desc category;

desc film;

desc film\_category;

SELECT c.name AS category\_name, COUNT(fc.film\_id) AS movie\_count

FROM category c

JOIN film\_category fc ON c.category\_id = fc.category\_id

GROUP BY c.name

HAVING COUNT(fc.film\_id) > 70;