1. Show the title and category of all films, ordered by film title (ascending).

SELECT t.title, a.name AS film\_category\_name

FROM film AS t

INNER JOIN film\_category AS c ON c.film\_id = t.film\_id

INNER JOIN category AS a ON a.category\_id = c.category\_id

ORDER BY t.title asc;

SELECT t.title, a.name AS film\_category\_name

FROM film AS t, film\_category AS c, category AS a

WHERE (c.film\_id = t.film\_id) and (a.category\_id = c.category\_id)

ORDER BY t.title asc;

CHK EVERYTHING WITH OTHER>>>CHK IF U HAV TO SORT BOTH COLUMNS IN ALL QUES

b.Show the number of films for each category (name), sorted by category (ascending).

SELECT c.name, COUNT(c.category\_id) as film\_count

FROM category as c

INNER JOIN film\_category as s on c.category\_id = s.category\_id

INNER JOIN film as t on t.film\_id = s.film\_id

GROUP by c.category\_id

ORDER BY c.name;

SELECT c.name as Category, COUNT(c.category\_id) as film\_count

FROM category c , film\_category s , film t

WHERE (c.category\_id = s.category\_id)

AND ( t.film\_id = s.film\_id)

GROUP by c.name

ORDER BY c.name;

c. Show the number of films for each category (name) that has at least 60 films, sorted by the  number (descending).

SELECT c.name, COUNT(c.category\_id) as Filmcount

FROM category as c

INNER JOIN film\_category as s on c.category\_id = s.category\_id

INNER JOIN film as t on t.film\_id = s.film\_id

GROUP by c.category\_id HAVING Filmcount >= 60

ORDER BY Filmcount desc;

SELECT c.name, COUNT(c.category\_id) as Filmcount

FROM category as c , film\_category as s, film as t

WHERE (c.category\_id = s.category\_id)

AND (t.film\_id = s.film\_id)

GROUP by c.category\_id HAVING Filmcount >= 60

ORDER BY Filmcount desc;

d.Which category (name) has the largest number of film?

SELECT c.name

FROM category as c

INNER JOIN film\_category as s on c.category\_id = s.category\_id

INNER JOIN film as t on t.film\_id = s.film\_id

GROUP by c.category\_id

ORDER BY COUNT(c.category\_id) desc

limit 1;

SELECT c.name

FROM category as c, film\_category as s , film as t

WHERE (c.category\_id = s.category\_id)

AND ( t.film\_id = s.film\_id)

GROUP by c.category\_id

ORDER BY COUNT(c.category\_id) desc

limit 1;

e. Find all (unique) actors (show their names in the form of "first\_name last\_name", e.g., "John  Smith", and in ascending order) who have played in more than one films.

f. Find out how many (active) customers have rented "Action" (category) films.

This should be the count not the names

SELECT COUNT(DISTINCT CONCAT(c.first\_name ," ", c.last\_name)) as customer\_names

FROM customer as c

INNER JOIN rental as r on r.customer\_id = c.customer\_id and c.active>=1

INNER JOIN inventory as i on i.inventory\_id=r.inventory\_id

INNER JOIN film\_category as t on t.film\_id = i.film\_id

INNER JOIN category as cat on (cat.category\_id = t.category\_id) and cat.name = "ACTION"

ORDER BY c.first\_name;

1. Find out how many customers have rented "Action" films but never rented "Horror" films.

SELECT COUNT(DISTINCT (c.customer\_id) ) as Customers\_Action\_butnot\_Horror FROM customer as c

INNER JOIN rental as r on r.customer\_id = c.customer\_id

INNER JOIN inventory as i on i.inventory\_id=r.inventory\_id

INNER JOIN film\_category as t on t.film\_id = i.film\_id

INNER JOIN category as cat on (cat.category\_id = t.category\_id) and cat.name = "Action"

AND r.customer\_id NOT IN

(SELECT c.customer\_id FROM customer as c

INNER JOIN rental as r on r.customer\_id = c.customer\_id

INNER JOIN inventory as i on i.inventory\_id=r.inventory\_id

INNER JOIN film\_category as t on t.film\_id = i.film\_id

INNER JOIN category as cat on (cat.category\_id = t.category\_id) and cat.name = "Horror");

131

* Create a view called “action\_view” that lists distinct customers (show their ids) who have  watched “Action” movies.

CREATE VIEW action\_view as

SELECT DISTINCT c.customer\_id

FROM customer as c

JOIN rental as r on r.customer\_id = c.customer\_id

JOIN inventory as i on i.inventory\_id=r.inventory\_id

JOIN film\_category as t on t.film\_id = i.film\_id

JOIN category as cat on (cat.category\_id = t.category\_id) and cat.name = "Action"

GROUP by c.customer\_id

ORDER BY c.customer\_id;

510

* Create a view called “horror\_view” that lists distinct customers (show their ids) who have  watched “Horrow” movies.

CREATE VIEW horror\_view as

SELECT c.customer\_id

FROM customer as c

JOIN rental as r on r.customer\_id = c.customer\_id

JOIN inventory as i on i.inventory\_id=r.inventory\_id

JOIN film\_category as t on t.film\_id = i.film\_id

JOIN category as cat on (cat.category\_id = t.category\_id) and cat.name = "Horror"

GROUP by c.customer\_id

ORDER BY c.customer\_id;

451

* Use the above two views to answer the question 1.g.

SELECT COUNT(a.customer\_id) as Customers\_Watched\_Action\_Not\_Horror

FROM action\_view a

WHERE a.customer\_id not in(SELECT h.customer\_id FROM horror\_view h );