use sakila;

# check for collation if not you have to use convert the column name to LOWER()

#look a the description column to see if the DB is collated

# 1.a

select first\_name, last\_name from actor;

#1.b

alter table actor

add Actor\_Name varchar(50);

update actor set Actor\_Name= CONCAT(first\_name, ' ' , last\_name);

select \* from actor;

#2.a

select actor\_id, first\_name, last\_name from actor where first\_name="Joe";

#2.b

select last\_name from actor where last\_name like '%GEN%';

#2.c

select first\_name , last\_name from actor where last\_name like '%LI%'

order by first\_name, last\_name;

#2.d

select country\_id , country from country where country IN ('Afghanistan','Bangladesh','China');

#3.a

alter table actor

add middle\_name varchar(50)

after first\_name;

#3.b

alter table actor

modify column middle\_name blob;

#3.c

ALTER TABLE actor

DROP COLUMN middle\_name;

#4.a

select last\_name , count(last\_name) as Count from actor

group by last\_name;

#4.b

select last\_name , count(last\_name) as Count from actor

group by last\_name

having Count > 1;

#4.c

select actor\_id , first\_name , last\_name from actor

where first\_name ='groucho' and last\_name ='williams';

UPDATE actor

SET first\_name='HARPO'

where first\_name ='GROUCHO' AND LAST\_NAME ='WILLIAMS';

#4.d

-- skipping based on Dylan's instructions

#5. a

-- shows the syntax of create table, i am not sure if this is needed or the request was to create a table ?

show create table address;

#'address', 'CREATE TABLE `address` (\n `address\_id` smallint(5) unsigned NOT NULL AUTO\_INCREMENT,\n `address` varchar(50) NOT NULL,\n `address2` varchar(50) DEFAULT NULL,\n `district` varchar(20) NOT NULL,\n `city\_id` smallint(5) unsigned NOT NULL,\n `postal\_code` varchar(10) DEFAULT NULL,\n `phone` varchar(20) NOT NULL,\n `location` geometry NOT NULL,\n `last\_update` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,\n PRIMARY KEY (`address\_id`),\n KEY `idx\_fk\_city\_id` (`city\_id`),\n SPATIAL KEY `idx\_location` (`location`),\n CONSTRAINT `fk\_address\_city` FOREIGN KEY (`city\_id`) REFERENCES `city` (`city\_id`) ON UPDATE CASCADE\n) ENGINE=InnoDB AUTO\_INCREMENT=606 DEFAULT CHARSET=utf8'

-- syntax to create a table

select \* from address;

create table address1( address\_id int auto\_increment NOT NULL , address varchar(100), address2 varchar(100), district varchar(100),

city\_id int(20), postal\_code varchar(50), phone varchar(50), location blob, last\_update timestamp not null default current\_timestamp,

primary key(address\_id));

#6.a

select S.first\_name, S.last\_name , A.address

from staff S left join address A

on S.address\_id = A.address\_id;

#6.b

select S.staff\_id, S.first\_name, S.last\_name , sum(P.amount) as Total\_Amount

from staff S left join payment P

on S.staff\_id=P.staff\_id

where P.payment\_date like '%-08%'

group by staff\_id;

#6.c

select F.title, count(FA.actor\_id) as Number\_Actors

from film F inner join film\_actor FA

on F.film\_id = FA.film\_id

group by F.title ;

#6.d

select count(inventory\_id) as Number\_of\_Copies from inventory

where film\_id in (select film\_id from film where title = 'Hunchback Impossible');

#6.e

select C.customer\_id, C.first\_name ,C.last\_name, sum(P.amount) as Total\_Paid

from customer C left join payment P

on C.customer\_id= P.customer\_id

group by C.customer\_id

order by C.last\_name;

# 7.a

select title from film where

title like 'K%' or title like 'Q%' and

language\_id in (select language\_id from language where name ='English') ;

#7.b

select first\_name , last\_name from actor where actor\_id in

(select actor\_id from film\_actor where film\_id in

(select film\_id from film where title= 'Alone Trip'));

# 7.c

select C.email, A.address\_id , CI.city\_id, CC.country\_id from customer C left outer join address A

on C.customer\_id = A.address\_id

left outer join city CI

on A.city\_id= CI.city\_id

left outer join country CC

on CI.country\_id = CC.country\_id

where CC.country='Canada';

#7.d

select \* from category;

select title from film where film\_id in

(select film\_id from film\_category where category\_id in

(select category\_id from category where name ='Family'));

#7.e -- most rented in the decreasing order

SELECT f.film\_id, f.title, COUNT(r.customer\_id) AS rent\_count

FROM rental r

JOIN inventory i USING (inventory\_id)

JOIN film f USING (film\_id)

GROUP BY f.film\_id

ORDER BY rent\_count DESC;

# 7.f

select sum(amount) from payment where staff\_id in

(select staff\_id from staff where store\_id in

(select store\_id from store))

group by staff\_id;

#7.g

select s.store\_id , c.city, cc.country

from store s join address a

on s.address\_id= a.address\_id

join city c on a.city\_id=c.city\_id

join country cc on cc.country\_id=c.country\_id;

#7h

select c.name , sum(p.amount) as Total

from payment p left outer join rental r on p.rental\_id=r.rental\_id

left outer join inventory i on i.inventory\_id=r.inventory\_id

left outer join film\_category f on i.film\_id=f.film\_id

left outer join category c on f.category\_id = c.category\_id

group by c.name

order by Total DESC;

#8.a

create view gross\_rev\_genre1 as

select c.name as Genre , sum(p.amount) as Total

from payment p left outer join rental r on p.rental\_id=r.rental\_id

left outer join inventory i on i.inventory\_id=r.inventory\_id

left outer join film\_category f on i.film\_id=f.film\_id

left outer join category c on f.category\_id = c.category\_id

group by c.name

order by Total DESC;

#8.b

select \* from gross\_rev\_genre1;

#8.c

drop view gross\_rev\_genre1;