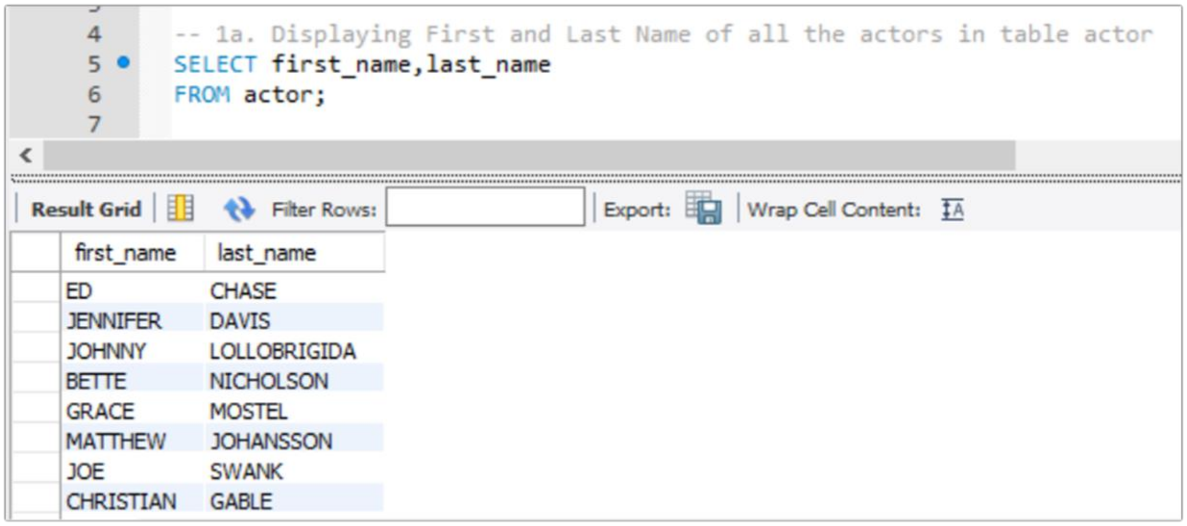
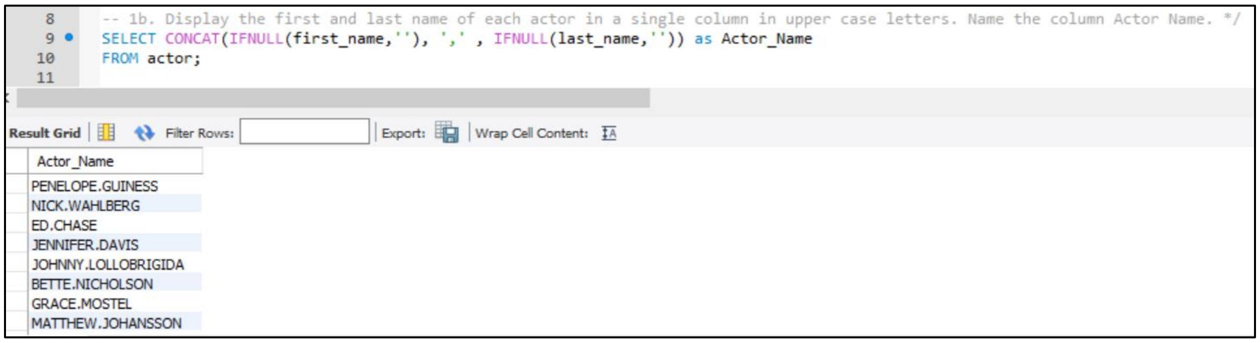
Week 10 – MySQL Home work



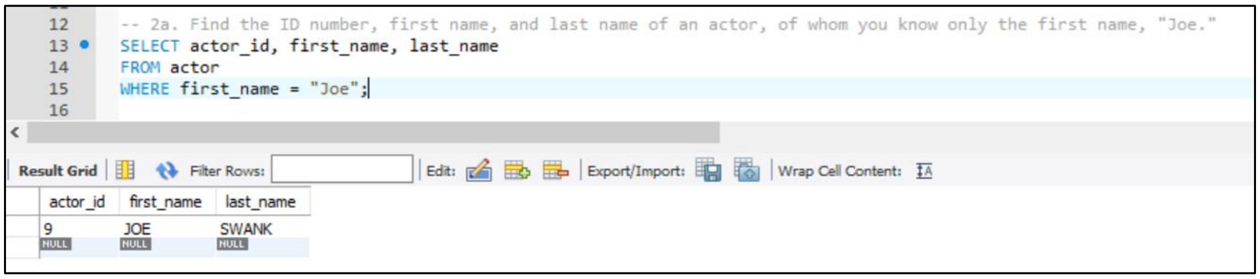
```sql
-- 1a. Displaying First and Last Name of all the actors in table actor
SELECT first_name,last_name
FROM actor;
```

first_name	last_name
ED	CHASE
JENNIFER	DAVIS
JOHNNY	LOLLOBRIGIDA
BETTE	NICHOLSON
GRACE	MOSTEL
MATTHEW	JOHANSSON
JOE	SWANK
CHRISTIAN	GABLE



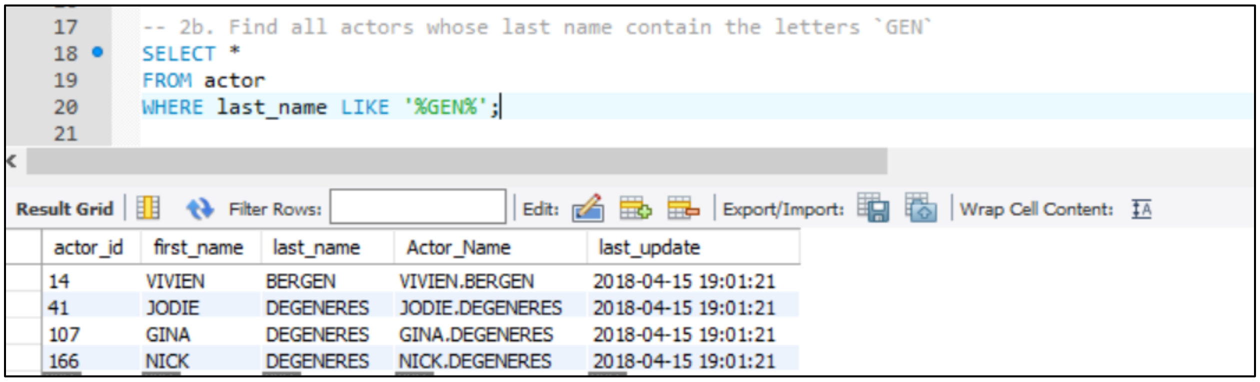
```sql
-- 1b. Display the first and last name of each actor in a single column in upper case letters. Name the column Actor Name. */
SELECT CONCAT(IFNULL(first_name,''), ',' , IFNULL(last_name,'')) as Actor_Name
FROM actor;
```

Actor_Name
PENELOPE.GUINESS
NICK.WAHLBERG
ED.CHASE
JENNIFER.DAVIS
JOHNNY.LOLLOBRIGIDA
BETTE.NICHOLSON
GRACE.MOSTEL
MATTHEW.JOHANSSON



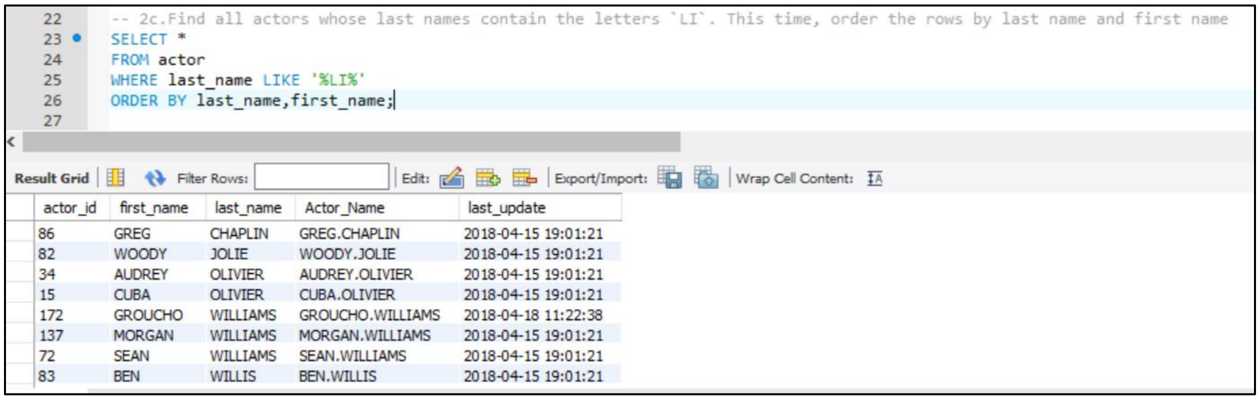
```sql
-- 2a. Find the ID number, first name, and last name of an actor, of whom you know only the first name, "Joe."
SELECT actor_id, first_name, last_name
FROM actor
WHERE first_name = "Joe";
```

actor_id	first_name	last_name
9	JOE	SWANK
NULL	NULL	NULL



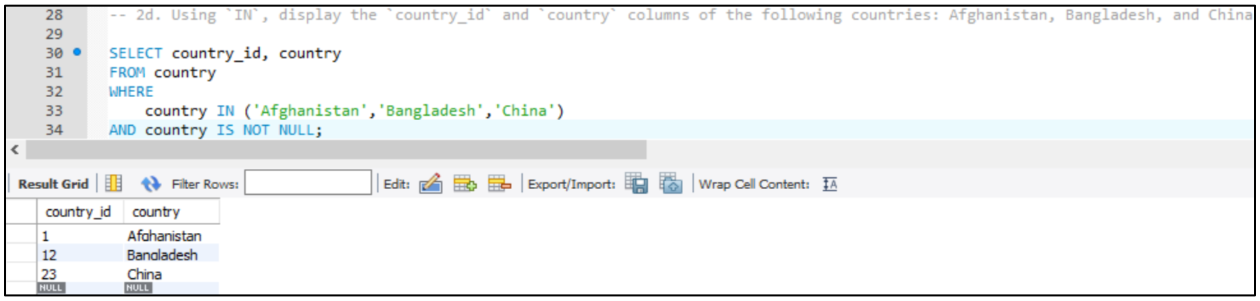
```sql
-- 2b. Find all actors whose last name contain the letters `GEN`
SELECT *
FROM actor
WHERE last_name LIKE '%GEN%';
```

actor_id	first_name	last_name	Actor_Name	last_update
14	VIVIEN	BERGEN	VIVIEN.BERGEN	2018-04-15 19:01:21
41	JODIE	DEGENERES	JODIE.DEGENERES	2018-04-15 19:01:21
107	GINA	DEGENERES	GINA.DEGENERES	2018-04-15 19:01:21
166	NICK	DEGENERES	NICK.DEGENERES	2018-04-15 19:01:21



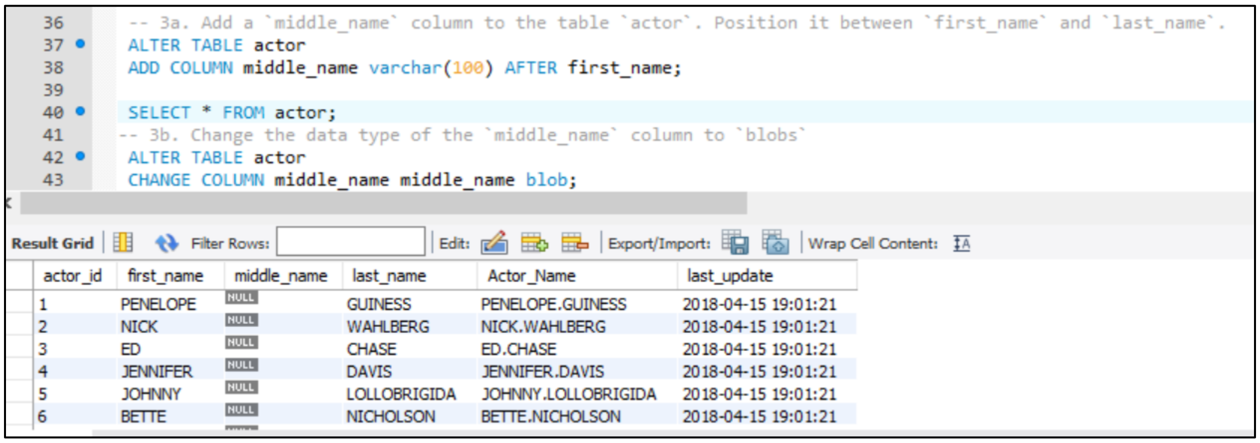
```sql
-- 2c.Find all actors whose last names contain the letters `LI`. This time, order the rows by last name and first name
SELECT *
FROM actor
WHERE last_name LIKE '%LI%'
ORDER BY last_name,first_name;
```

actor_id	first_name	last_name	Actor_Name	last_update
86	GREG	CHAPLIN	GREG.CHAPLIN	2018-04-15 19:01:21
82	WOODY	JOLIE	WOODY.JOLIE	2018-04-15 19:01:21
34	AUDREY	OLIVIER	AUDREY.OLIVIER	2018-04-15 19:01:21
15	CUBA	OLIVIER	CUBA.OLIVIER	2018-04-15 19:01:21
172	GROUCHO	WILLIAMS	GROUCHO.WILLIAMS	2018-04-18 11:22:38
137	MORGAN	WILLIAMS	MORGAN.WILLIAMS	2018-04-15 19:01:21
72	SEAN	WILLIAMS	SEAN.WILLIAMS	2018-04-15 19:01:21
83	BEN	WILLIS	BEN.WILLIS	2018-04-15 19:01:21



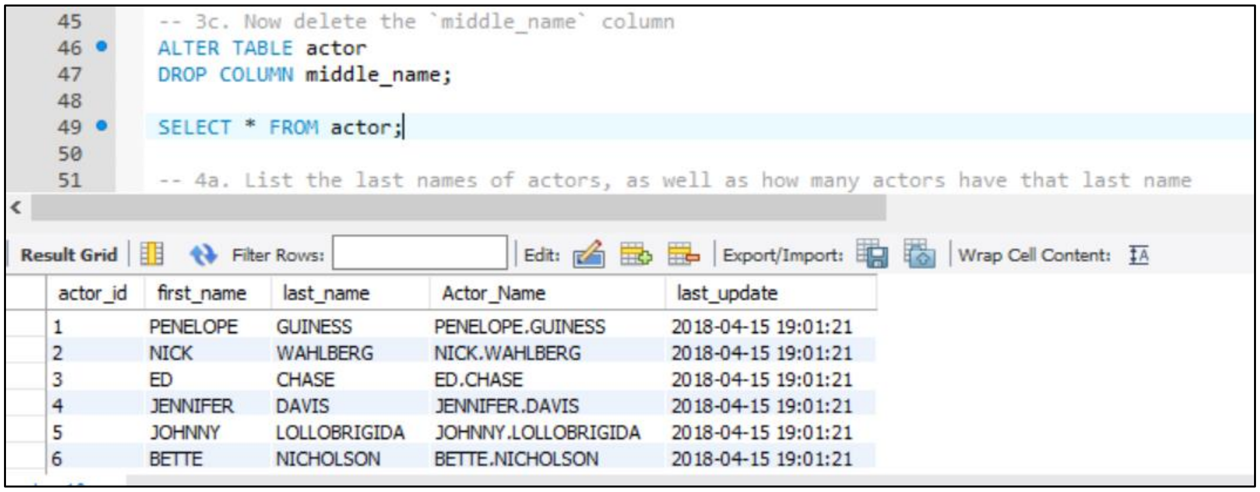
```sql
-- 2d. Using `IN`, display the `country_id` and `country` columns of the following countries: Afghanistan, Bangladesh, and China

SELECT country_id, country
FROM country
WHERE
    country IN ('Afghanistan','Bangladesh','China')
AND country IS NOT NULL;
```

country_id	country
1	Afghanistan
12	Bangladesh
23	China
NULL	NULL



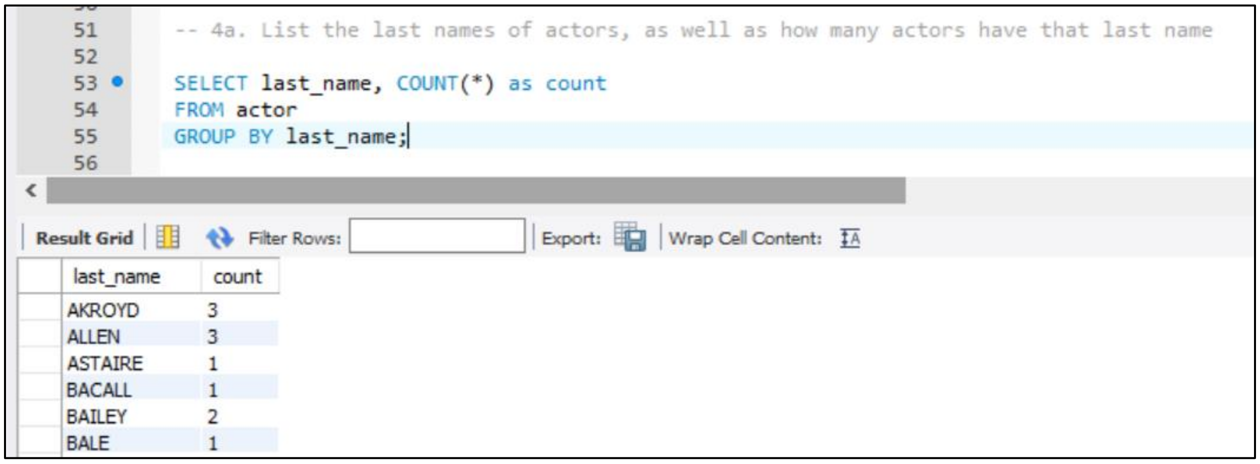
```sql
-- 3a. Add a `middle_name` column to the table `actor`. Position it between `first_name` and `last_name`.
ALTER TABLE actor
ADD COLUMN middle_name varchar(100) AFTER first_name;

SELECT * FROM actor;
-- 3b. Change the data type of the `middle_name` column to `blobs`
ALTER TABLE actor
CHANGE COLUMN middle_name middle_name blob;
```

actor_id	first_name	middle_name	last_name	Actor_Name	last_update
1	PENELOPE	NULL	GUINESS	PENELOPE.GUINESS	2018-04-15 19:01:21
2	NICK	NULL	WAHLBERG	NICK.WAHLBERG	2018-04-15 19:01:21
3	ED	NULL	CHASE	ED.CHASE	2018-04-15 19:01:21
4	JENNIFER	NULL	DAVIS	JENNIFER.DAVIS	2018-04-15 19:01:21
5	JOHNNY	NULL	LOLLOBRIGIDA	JOHNNY.LOLLOBRIGIDA	2018-04-15 19:01:21
6	BETTE	NULL	NICHOLSON	BETTE.NICHOLSON	2018-04-15 19:01:21



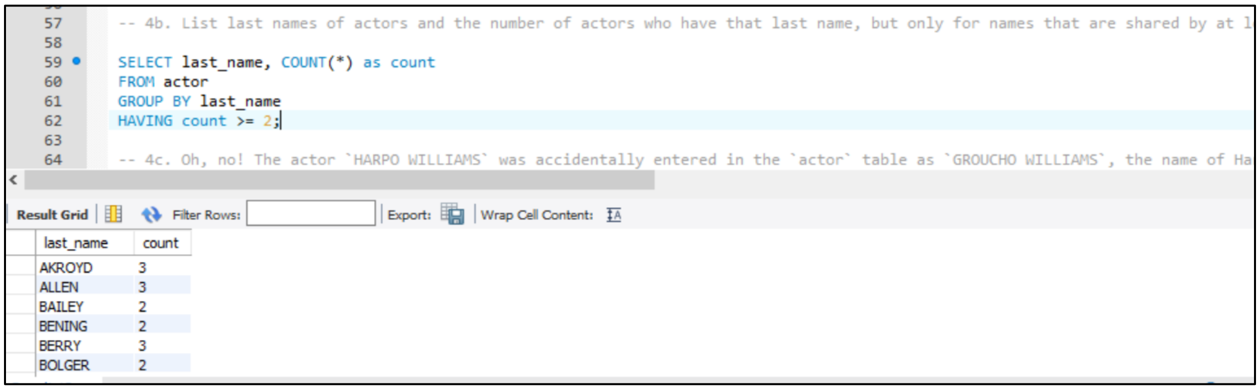
```sql
-- 3c. Now delete the `middle_name` column
ALTER TABLE actor
DROP COLUMN middle_name;

SELECT * FROM actor;

-- 4a. List the last names of actors, as well as how many actors have that last name
```

actor_id	first_name	last_name	Actor_Name	last_update
1	PENELOPE	GUINESS	PENELOPE.GUINESS	2018-04-15 19:01:21
2	NICK	WAHLBERG	NICK.WAHLBERG	2018-04-15 19:01:21
3	ED	CHASE	ED.CHASE	2018-04-15 19:01:21
4	JENNIFER	DAVIS	JENNIFER.DAVIS	2018-04-15 19:01:21
5	JOHNNY	LOLLOBRIGIDA	JOHNNY.LOLLOBRIGIDA	2018-04-15 19:01:21
6	BETTE	NICHOLSON	BETTE.NICHOLSON	2018-04-15 19:01:21



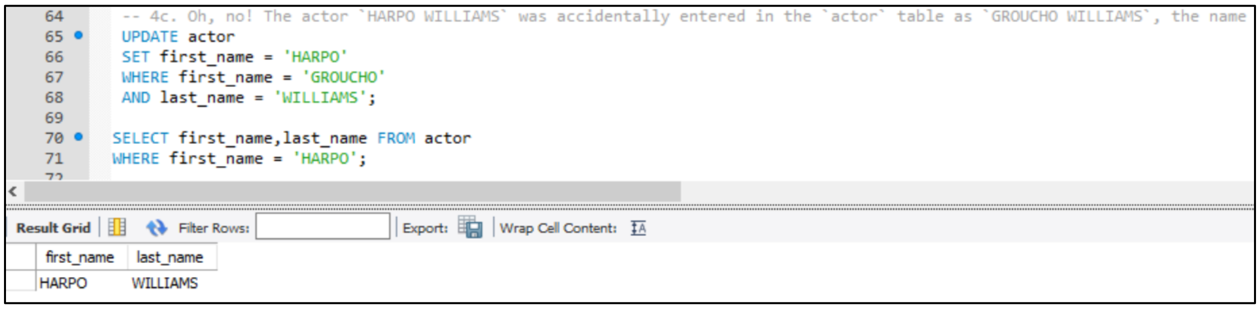
```sql
-- 4a. List the last names of actors, as well as how many actors have that last name

SELECT last_name, COUNT(*) as count
FROM actor
GROUP BY last_name;
```

last_name	count
AKROYD	3
ALLEN	3
ASTAIRE	1
BACALL	1
BAILEY	2
BALE	1



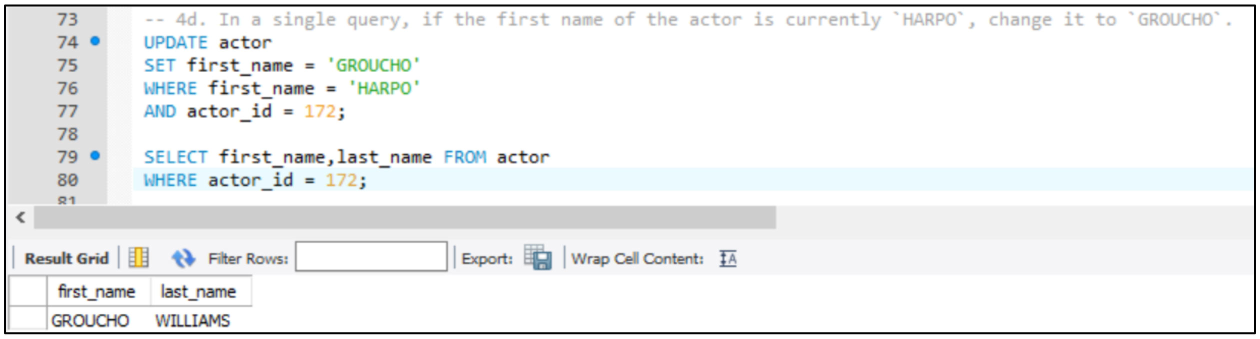
```sql
-- 4b. List last names of actors and the number of actors who have that last name, but only for names that are shared by at l

SELECT last_name, COUNT(*) as count
FROM actor
GROUP BY last_name
HAVING count >= 2;

-- 4c. Oh, no! The actor `HARPO WILLIAMS` was accidentally entered in the `actor` table as `GROUCHO WILLIAMS`, the name of Ha
```

last_name	count
AKROYD	3
ALLEN	3
BAILEY	2
BENING	2
BERRY	3
BOLGER	2



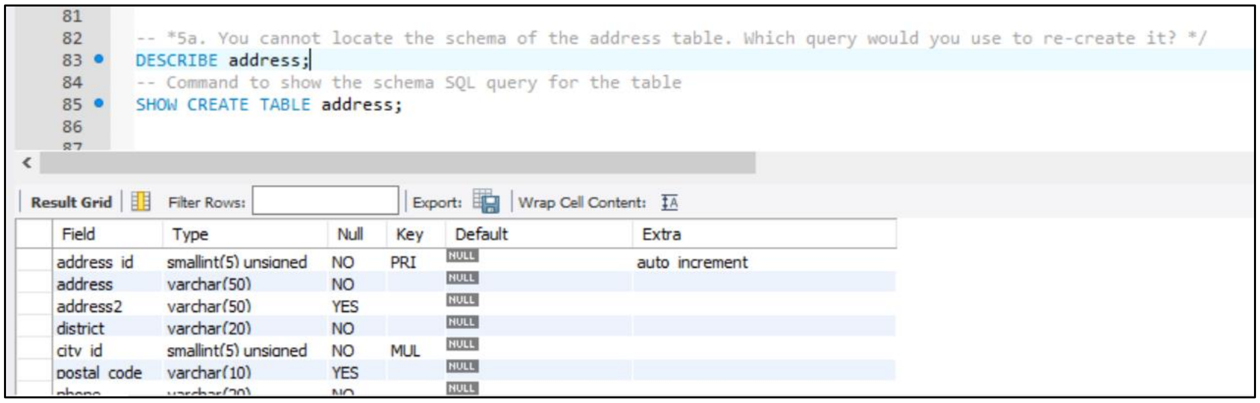
```sql
 -- 4c. Oh, no! The actor `HARPO WILLIAMS` was accidentally entered in the `actor` table as `GROUCHO WILLIAMS`, the name
 UPDATE actor
 SET first_name = 'HARPO'
 WHERE first_name = 'GROUCHO'
 AND last_name = 'WILLIAMS';

SELECT first_name,last_name FROM actor
WHERE first_name = 'HARPO';
```

first_name	last_name
HARPO	WILLIAMS



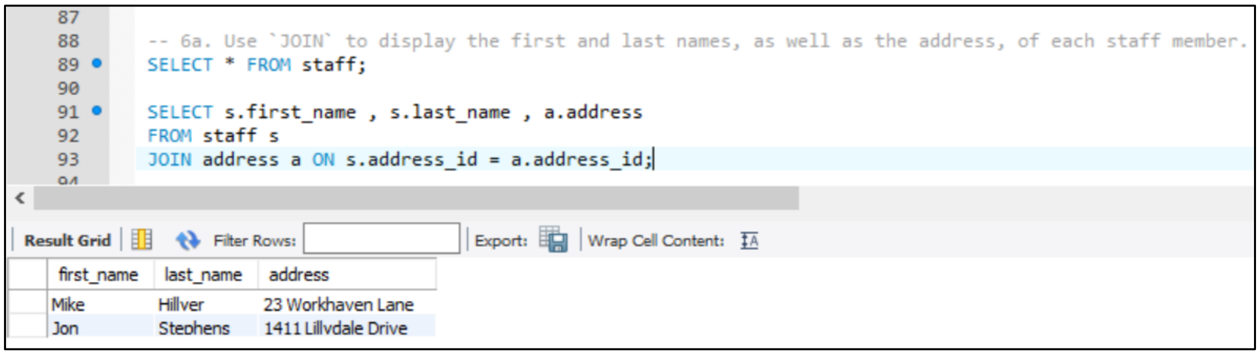
```sql
-- 4d. In a single query, if the first name of the actor is currently `HARPO`, change it to `GROUCHO`.
UPDATE actor
SET first_name = 'GROUCHO'
WHERE first_name = 'HARPO'
AND actor_id = 172;

SELECT first_name,last_name FROM actor
WHERE actor_id = 172;
```

first_name	last_name
GROUCHO	WILLIAMS



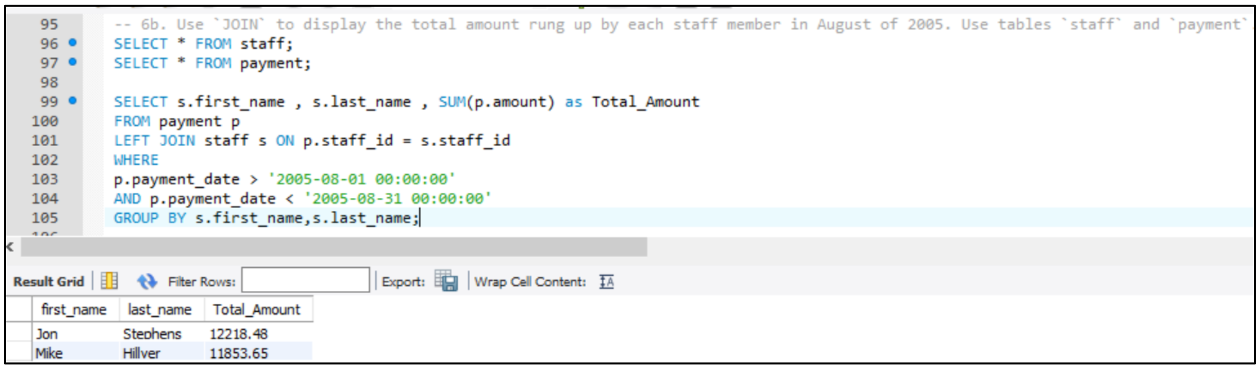
```sql
-- *5a. You cannot locate the schema of the address table. Which query would you use to re-create it? */
DESCRIBE address;
-- Command to show the schema SQL query for the table
SHOW CREATE TABLE address;
```

Field	Type	Null	Key	Default	Extra
address_id	smallint(5) unsigned	NO	PRI	NULL	auto_increment
address	varchar(50)	NO		NULL	
address2	varchar(50)	YES		NULL	
district	varchar(20)	NO		NULL	
city_id	smallint(5) unsigned	NO	MUL	NULL	
postal_code	varchar(10)	YES		NULL	



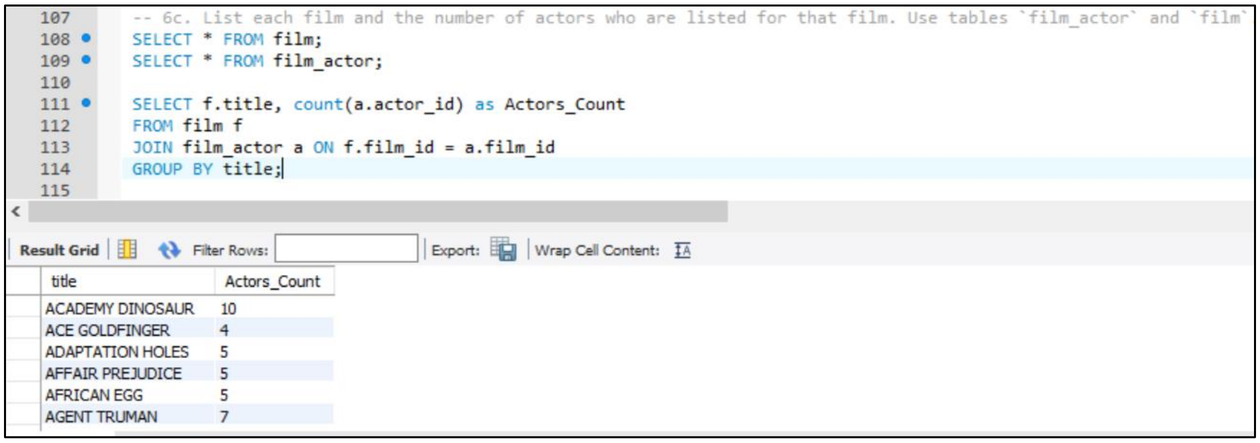
```sql
-- 6a. Use `JOIN` to display the first and last names, as well as the address, of each staff member.
SELECT * FROM staff;

SELECT s.first_name , s.last_name , a.address
FROM staff s
JOIN address a ON s.address_id = a.address_id;
```

first_name	last_name	address
Mike	Hillyer	23 Workhaven Lane
Jon	Stephens	1411 Lillydale Drive



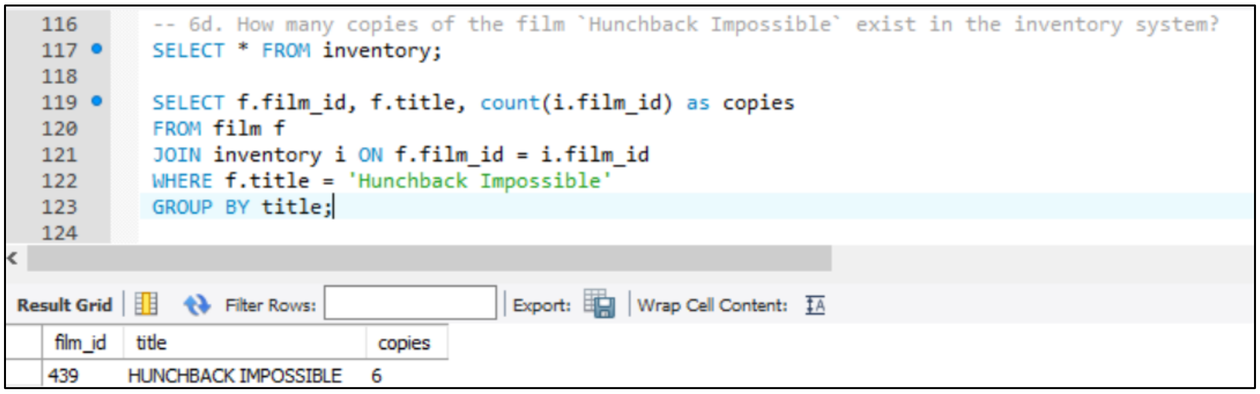
```sql
-- 6b. Use `JOIN` to display the total amount rung up by each staff member in August of 2005. Use tables `staff` and `payment`
SELECT * FROM staff;
SELECT * FROM payment;

SELECT s.first_name , s.last_name , SUM(p.amount) as Total_Amount
FROM payment p
LEFT JOIN staff s ON p.staff_id = s.staff_id
WHERE
p.payment_date > '2005-08-01 00:00:00'
AND p.payment_date < '2005-08-31 00:00:00'
GROUP BY s.first_name,s.last_name;
```

first_name	last_name	Total_Amount
Jon	Stephens	12218.48
Mike	Hillyer	11853.65



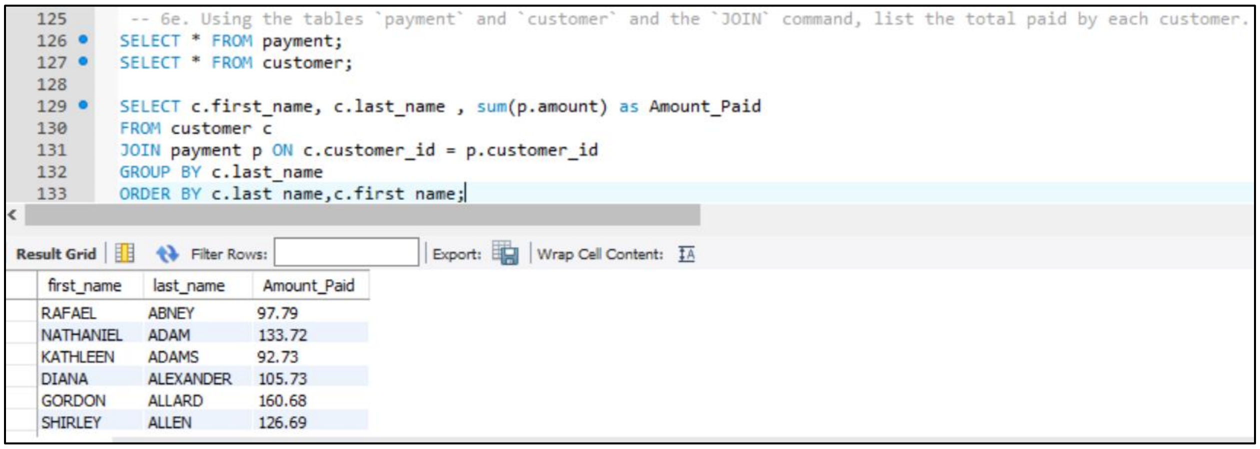
```sql
-- 6c. List each film and the number of actors who are listed for that film. Use tables `film_actor` and `film`
SELECT * FROM film;
SELECT * FROM film_actor;

SELECT f.title, count(a.actor_id) as Actors_Count
FROM film f
JOIN film_actor a ON f.film_id = a.film_id
GROUP BY title;
```

title	Actors_Count
ACADEMY DINOSAUR	10
ACE GOLDFINGER	4
ADAPTATION HOLES	5
AFFAIR PREJUDICE	5
AFRICAN EGG	5
AGENT TRUMAN	7



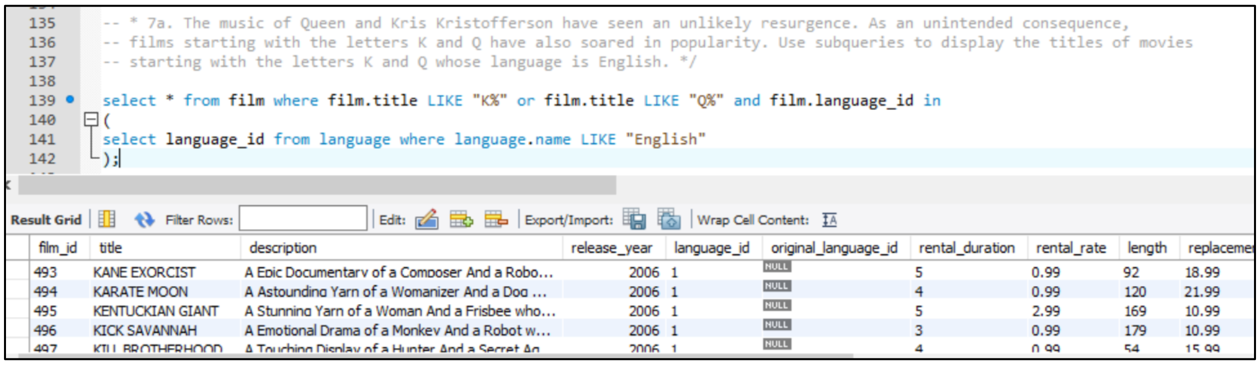
```sql
-- 6d. How many copies of the film `Hunchback Impossible` exist in the inventory system?
SELECT * FROM inventory;

SELECT f.film_id, f.title, count(i.film_id) as copies
FROM film f
JOIN inventory i ON f.film_id = i.film_id
WHERE f.title = 'Hunchback Impossible'
GROUP BY title;
```

film_id	title	copies
439	HUNCHBACK IMPOSSIBLE	6



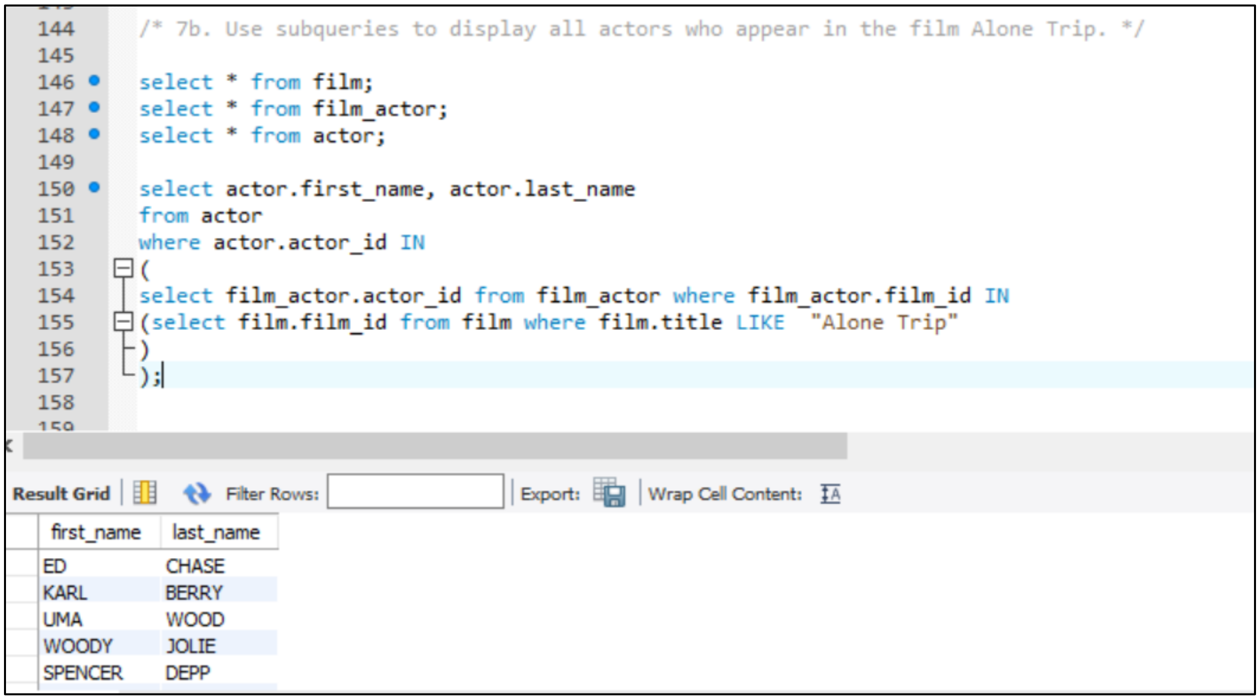
```sql
-- 6e. Using the tables `payment` and `customer` and the `JOIN` command, list the total paid by each customer.
SELECT * FROM payment;
SELECT * FROM customer;

SELECT c.first_name, c.last_name , sum(p.amount) as Amount_Paid
FROM customer c
JOIN payment p ON c.customer_id = p.customer_id
GROUP BY c.last_name
ORDER BY c.last_name,c.first_name;
```

first_name	last_name	Amount_Paid
RAFAEL	ABNEY	97.79
NATHANIEL	ADAM	133.72
KATHLEEN	ADAMS	92.73
DIANA	ALEXANDER	105.73
GORDON	ALLARD	160.68
SHIRLEY	ALLEN	126.69



```sql
-- * 7a. The music of Queen and Kris Kristofferson have seen an unlikely resurgence. As an unintended consequence,
-- films starting with the letters K and Q have also soared in popularity. Use subqueries to display the titles of movies
-- starting with the letters K and Q whose language is English. */

select * from film where film.title LIKE "K%" or film.title LIKE "Q%" and film.language_id in
(
select language_id from language where language.name LIKE "English"
);
```

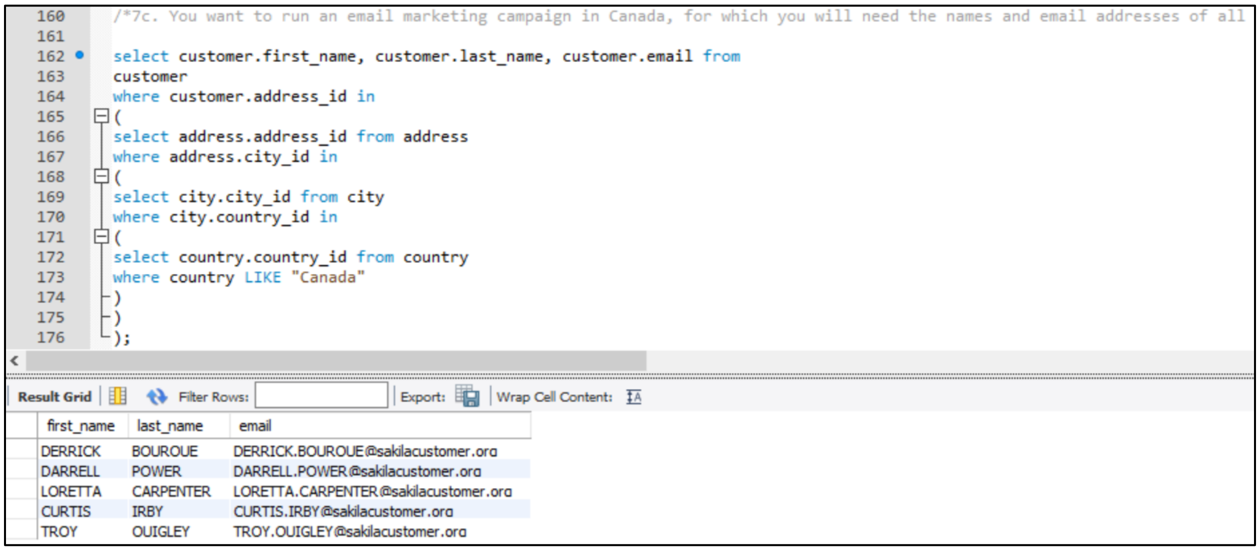
film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replaceme
493	KANE EXORCIST	A Epic Documentary of a Composer And a Robo...	2006	1	NULL	5	0.99	92	18.99
494	KARATE MOON	A Astounding Yarn of a Womanizer And a Dog ...	2006	1	NULL	4	0.99	120	21.99
495	KENTUCKIAN GIANT	A Stunning Yarn of a Woman And a Frisbee who...	2006	1	NULL	5	2.99	169	10.99
496	KICK SAVANNAH	A Emotional Drama of a Monkey And a Robot w...	2006	1	NULL	3	0.99	179	10.99
497	KILL BROTHERHOOD	A Touching Display of a Hunter And a Secret Ag	2006	1	NULL	4	0.99	54	15.99



```sql
/* 7b. Use subqueries to display all actors who appear in the film Alone Trip. */

select * from film;
select * from film_actor;
select * from actor;

select actor.first_name, actor.last_name
from actor
where actor.actor_id IN
(
select film_actor.actor_id from film_actor where film_actor.film_id IN
(select film.film_id from film where film.title LIKE  "Alone Trip"
)
);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

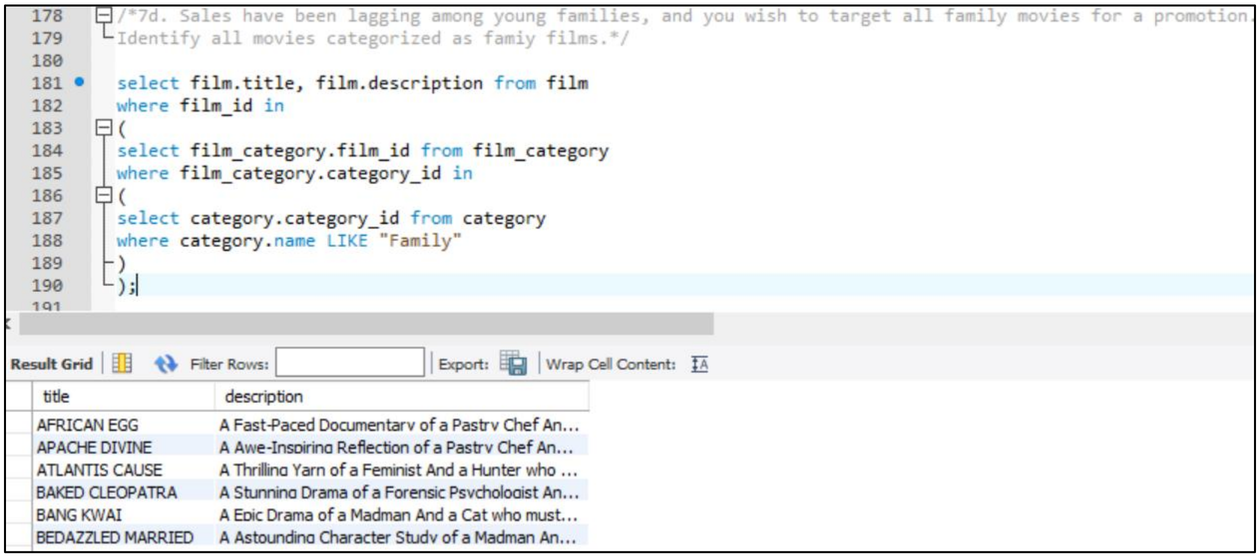
first_name	last_name
ED	CHASE
KARL	BERRY
UMA	WOOD
WOODY	JOLIE
SPENCER	DEPP



```sql
/*7c. You want to run an email marketing campaign in Canada, for which you will need the names and email addresses of all

select customer.first_name, customer.last_name, customer.email from
customer
where customer.address_id in
(
select address.address_id from address
where address.city_id in
(
select city.city_id from city
where city.country_id in
(
select country.country_id from country
where country LIKE "Canada"
)
)
);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

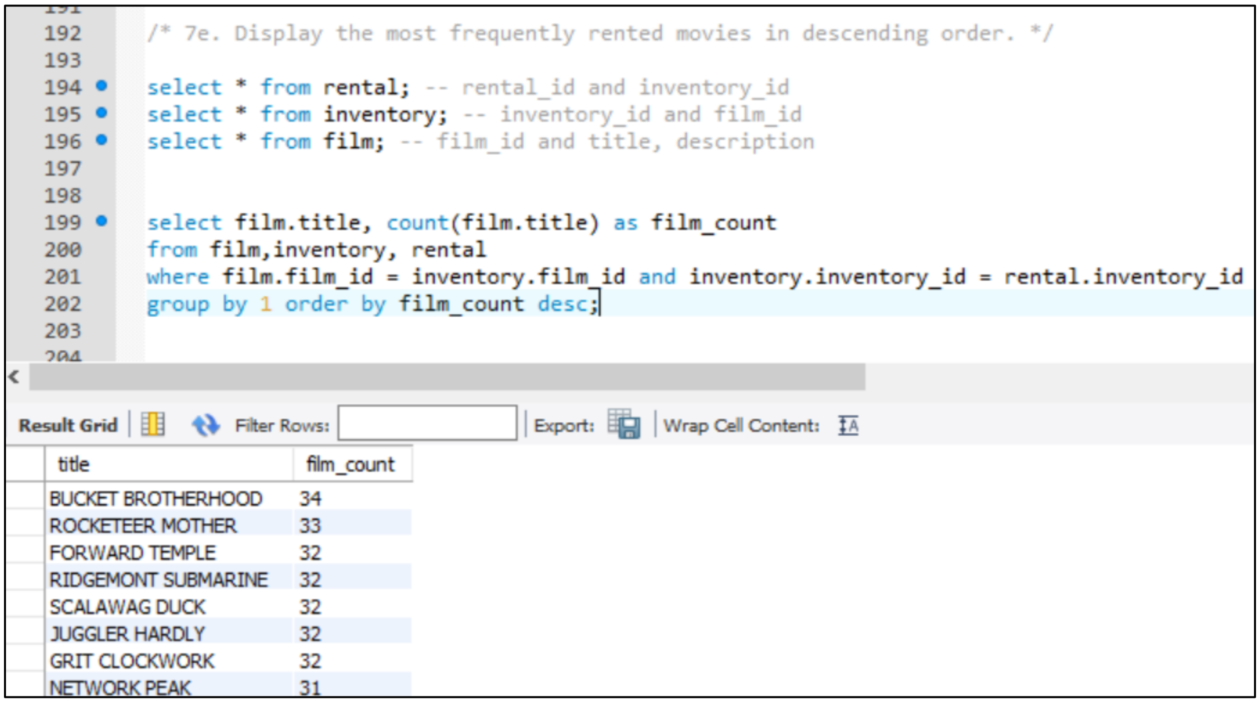
first_name	last_name	email
DERRICK	BOURQUE	DERRICK.BOURQUE@sakilacustomer.org
DARRELL	POWER	DARRELL.POWER@sakilacustomer.org
LORETTA	CARPENTER	LORETTA.CARPENTER@sakilacustomer.org
CURTIS	IRBY	CURTIS.IRBY@sakilacustomer.org
TROY	QUIGLEY	TROY.QUIGLEY@sakilacustomer.org



```
/*7d. Sales have been lagging among young families, and you wish to target all family movies for a promotion.
Identify all movies categorized as famiy films.*/

select film.title, film.description from film
where film_id in
(
select film_category.film_id from film_category
where film_category.category_id in
(
select category.category_id from category
where category.name LIKE "Family"
)
);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

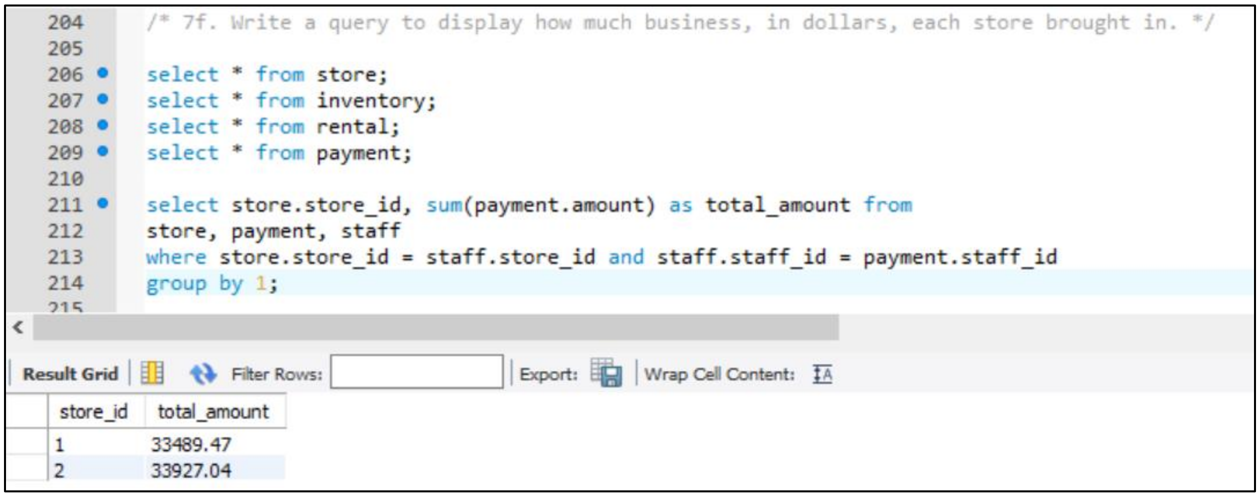
title	description
AFRICAN EGG	A Fast-Paced Documentary of a Pastry Chef An...
APACHE DIVINE	A Awe-Inspiring Reflection of a Pastry Chef An...
ATLANTIS CAUSE	A Thrilling Yarn of a Feminist And a Hunter who ...
BAKED CLEOPATRA	A Stunning Drama of a Forensic Psychologist An...
BANG KWAI	A Epic Drama of a Madman And a Cat who must...
BEDAZZLED MARRIED	A Astounding Character Study of a Madman An...



```
/* 7e. Display the most frequently rented movies in descending order. */

select * from rental; -- rental_id and inventory_id
select * from inventory; -- inventory_id and film_id
select * from film; -- film_id and title, description


select film.title, count(film.title) as film_count
from film,inventory, rental
where film.film_id = inventory.film_id and inventory.inventory_id = rental.inventory_id
group by 1 order by film_count desc;
```

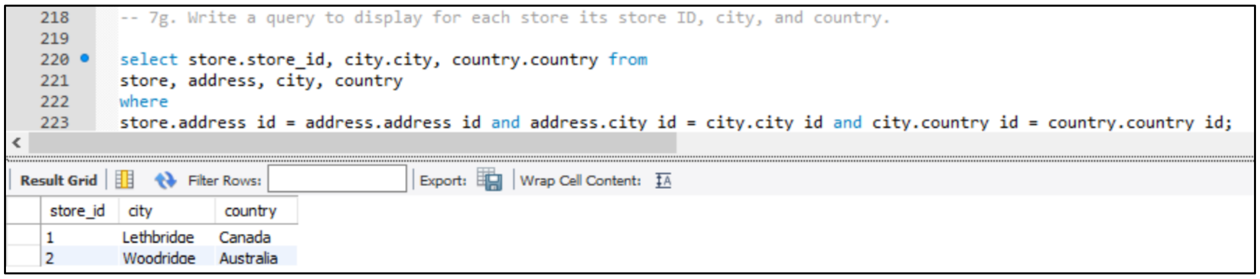
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

title	film_count
BUCKET BROTHERHOOD	34
ROCKETEER MOTHER	33
FORWARD TEMPLE	32
RIDGEMONT SUBMARINE	32
SCALAWAG DUCK	32
JUGGLER HARDLY	32
GRIT CLOCKWORK	32
NETWORK PEAK	31



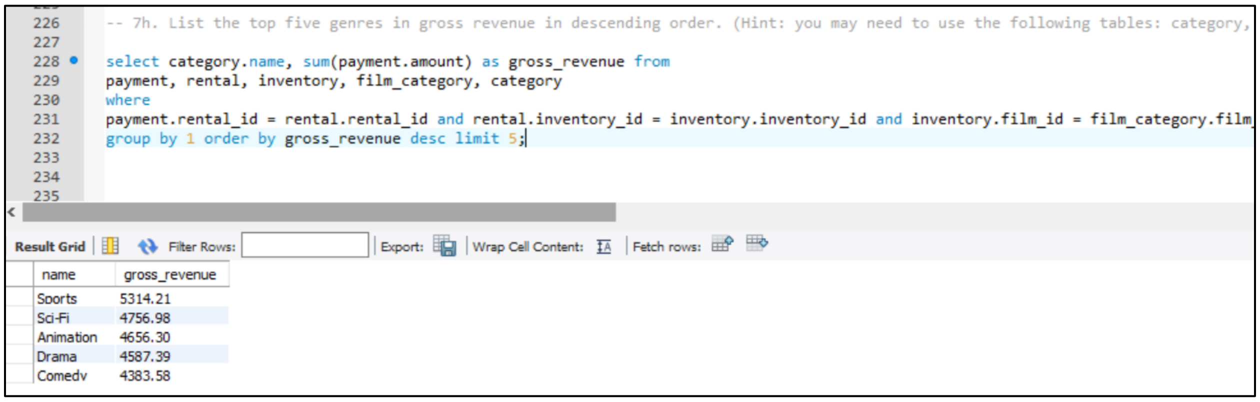
```sql
/* 7f. Write a query to display how much business, in dollars, each store brought in. */

select * from store;
select * from inventory;
select * from rental;
select * from payment;

select store.store_id, sum(payment.amount) as total_amount from
store, payment, staff
where store.store_id = staff.store_id and staff.staff_id = payment.staff_id
group by 1;
```

store_id	total_amount
1	33489.47
2	33927.04



```sql
-- 7g. Write a query to display for each store its store ID, city, and country.

select store.store_id, city.city, country.country from
store, address, city, country
where
store.address_id = address.address_id and address.city_id = city.city_id and city.country_id = country.country_id;
```

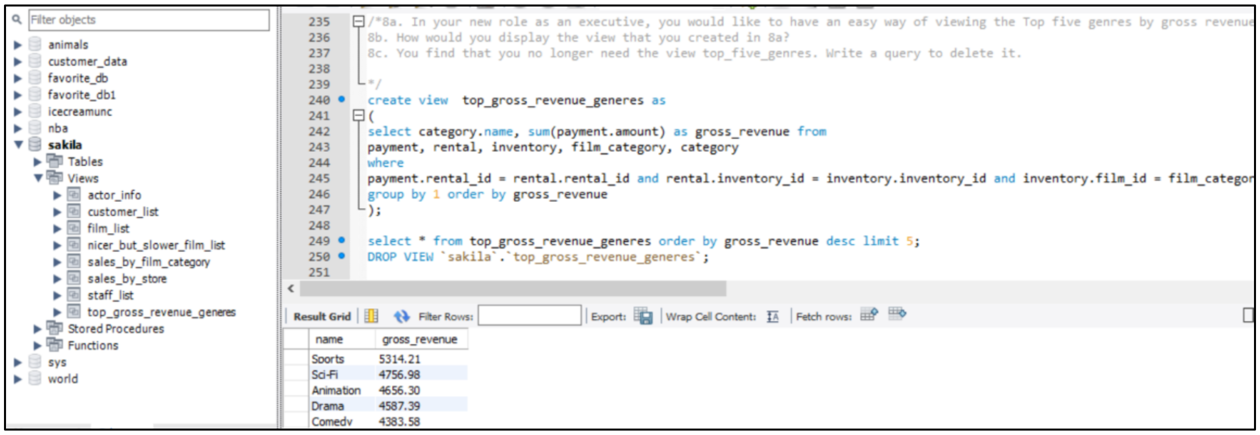
store_id	city	country
1	Lethbridge	Canada
2	Woodridge	Australia



```sql
-- 7h. List the top five genres in gross revenue in descending order. (Hint: you may need to use the following tables: category,

select category.name, sum(payment.amount) as gross_revenue from
payment, rental, inventory, film_category, category
where
payment.rental_id = rental.rental_id and rental.inventory_id = inventory.inventory_id and inventory.film_id = film_category.film
group by 1 order by gross_revenue desc limit 5;
```

name	gross_revenue
Sports	5314.21
Sci-Fi	4756.98
Animation	4656.30
Drama	4587.39
Comedy	4383.58

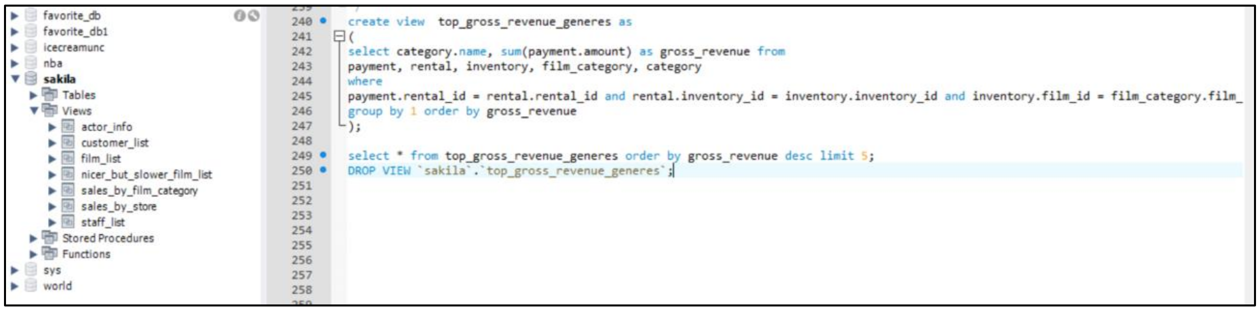
View Created



```
/*8a. In your new role as an executive, you would like to have an easy way of viewing the Top five genres by gross revenue
8b. How would you display the view that you created in 8a?
8c. You find that you no longer need the view top_five_genres. Write a query to delete it.

*/
create view  top_gross_revenue_generes as
(
select category.name, sum(payment.amount) as gross_revenue from
payment, rental, inventory, film_category, category
where
payment.rental_id = rental.rental_id and rental.inventory_id = inventory.inventory_id and inventory.film_id = film_categor
group by 1 order by gross_revenue
);

select * from top_gross_revenue_generes order by gross_revenue desc limit 5;
DROP VIEW `sakila`.`top_gross_revenue_generes`;
```

```
name       gross_revenue
Sports     5314.21
Sci-Fi     4756.98
Animation  4656.30
Drama      4587.39
Comedy     4383.58
```

View Deleted



```
create view  top_gross_revenue_generes as
(
select category.name, sum(payment.amount) as gross_revenue from
payment, rental, inventory, film_category, category
where
payment.rental_id = rental.rental_id and rental.inventory_id = inventory.inventory_id and inventory.film_id = film_category.film_
group by 1 order by gross_revenue
);

select * from top_gross_revenue_generes order by gross_revenue desc limit 5;
DROP VIEW `sakila`.`top_gross_revenue_generes`;
```