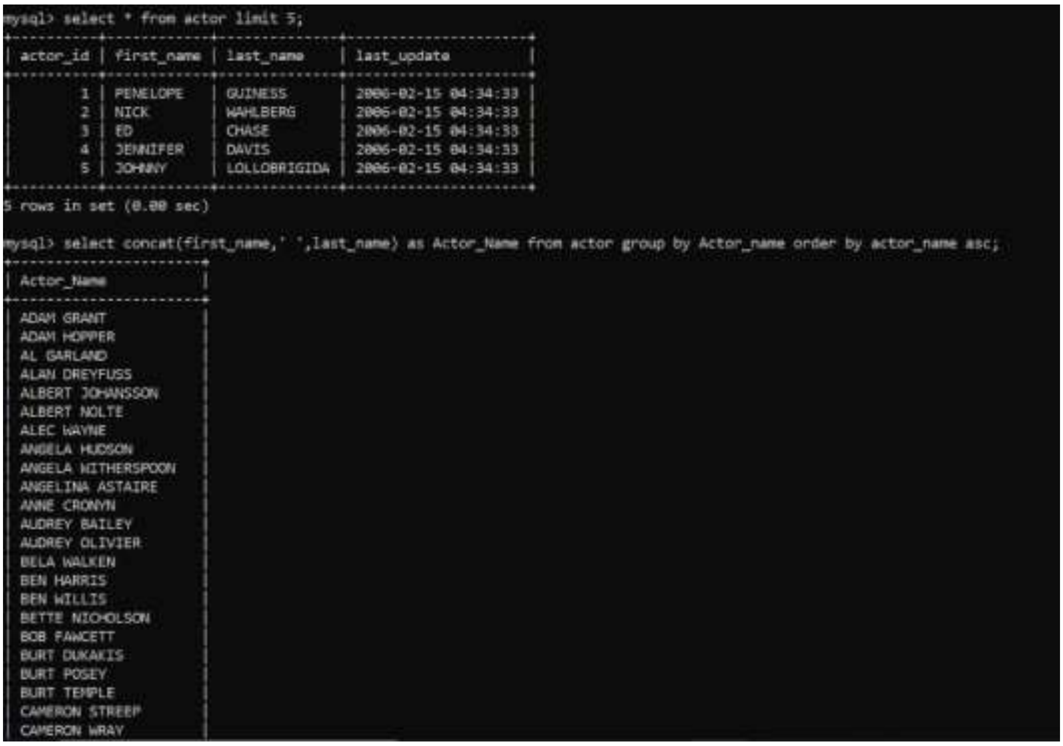
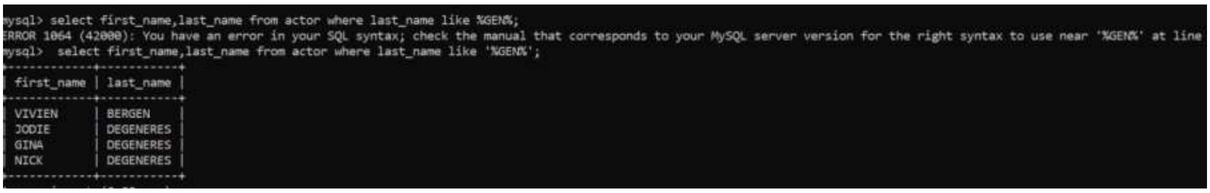
1. Display the first and last name of each actor in a single column in upper case letters in alphabetic order. Name the column Actor Name.



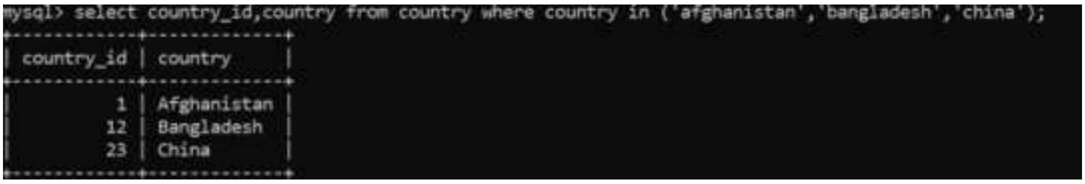
```
mysql> select * from actor limit 5;
+----------+------------+--------------+---------------------+
| actor_id | first_name | last_name    | last_update         |
+----------+------------+--------------+---------------------+
|        1 | PENELOPE   | GUINESS      | 2006-02-15 04:34:33 |
|        2 | NICK       | WAHLBERG     | 2006-02-15 04:34:33 |
|        3 | ED         | CHASE        | 2006-02-15 04:34:33 |
|        4 | JENNIFER   | DAVIS        | 2006-02-15 04:34:33 |
|        5 | JOHNNY     | LOLLOBRIGIDA | 2006-02-15 04:34:33 |
+----------+------------+--------------+---------------------+
5 rows in set (0.00 sec)

mysql> select concat(first_name,' ',last_name) as Actor_Name from actor group by Actor_name order by actor_name asc;
+----------------------+
| Actor_Name           |
+----------------------+
| ADAM GRANT           |
| ADAM HOPPER          |
| AL GARLAND           |
| ALAN DREYFUSS        |
| ALBERT JOHANSSON     |
| ALBERT NOLTE         |
| ALEC WAYNE           |
| ANGELA HUDSON        |
| ANGELA WITHERSPOON   |
| ANGELINA ASTAIRE     |
| ANNE CRONYN          |
| AUDREY BAILEY        |
| AUDREY OLIVIER       |
| BELA WALKEN          |
| BEN HARRIS           |
| BEN WILLIS           |
| BETTE NICHOLSON      |
| BOB FAWCETT          |
| BURT DUKAKIS         |
| BURT POSEY           |
| BURT TEMPLE          |
| CAMERON STREEP       |
| CAMERON WRAY         |
```

2. Find all actors whose last name contain the letters GEN:



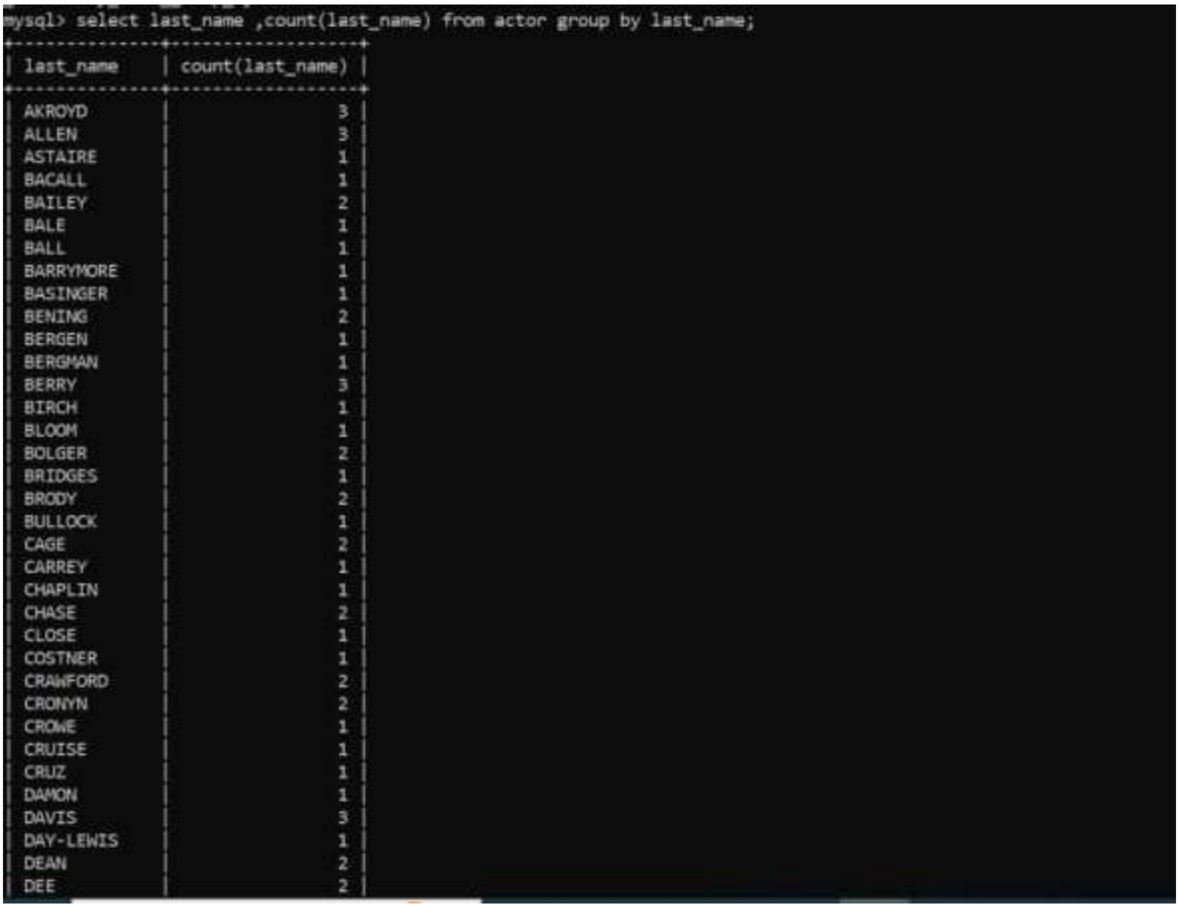
```
ysql> select first_name,last_name from actor where last_name like %GEN%;
RROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '%GEN%' at line
ysql>  select first_name,last_name from actor where last_name like '%GEN%';
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| VIVIEN     | BERGEN    |
| JODIE      | DEGENERES |
| GINA       | DEGENERES |
| NICK       | DEGENERES |
+------------+-----------+
```

3. Using IN, display the country_id and country columns of the following countries: Afghanistan, Bangladesh, and China:



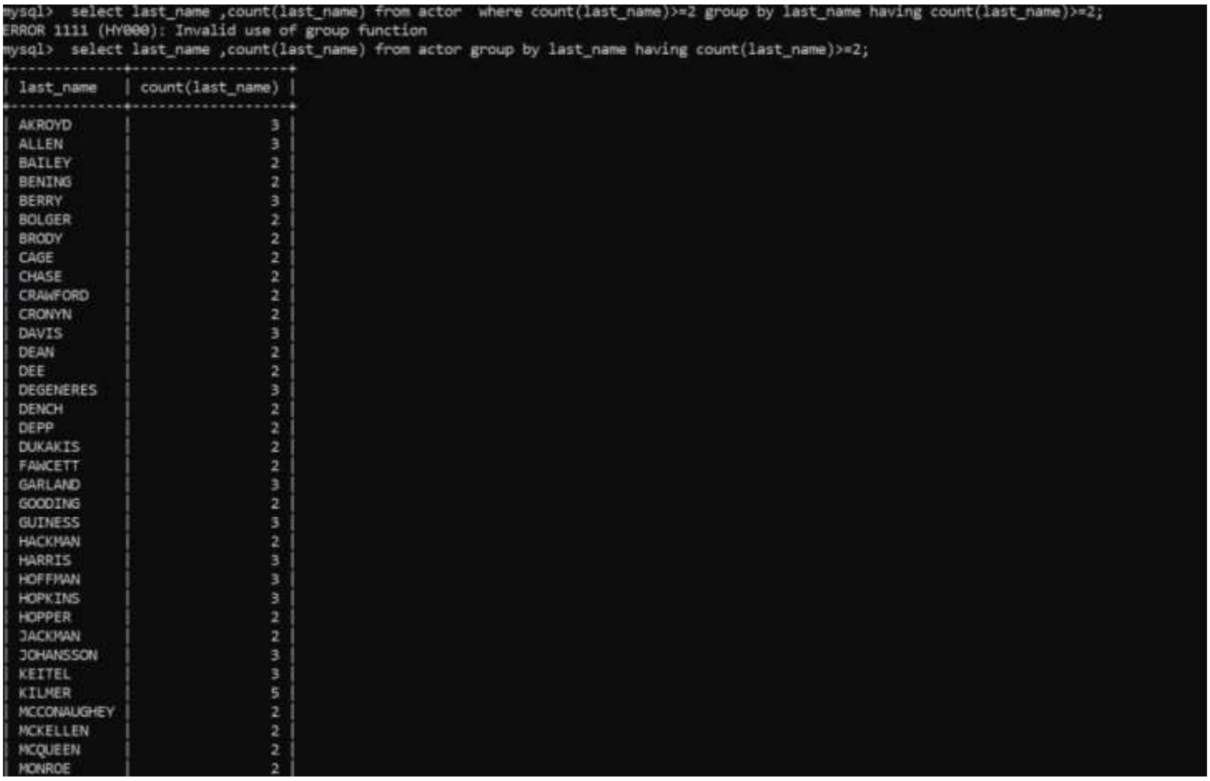
```
mysql> select country_id,country from country where country in ('afghanistan','bangladesh','china');
+------------+-------------+
| country_id | country     |
+------------+-------------+
|          1 | Afghanistan |
|         12 | Bangladesh  |
|         23 | China       |
+------------+-------------+
```

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



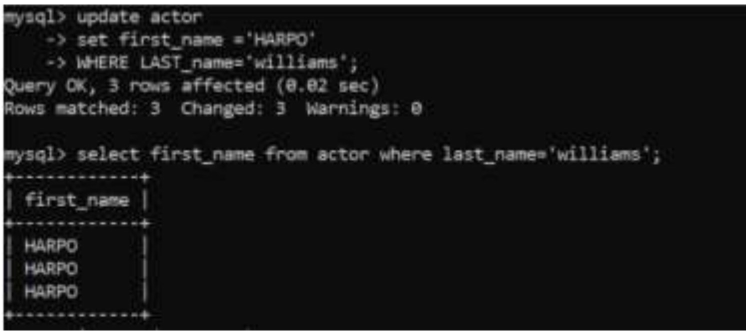
```
mysql> select last_name ,count(last_name) from actor group by last_name;
+--------------+------------------+
| last_name    | count(last_name) |
+--------------+------------------+
| AKROYD       |                3 |
| ALLEN        |                3 |
| ASTAIRE      |                1 |
| BACALL       |                1 |
| BAILEY       |                2 |
| BALE         |                1 |
| BALL         |                1 |
| BARRYMORE    |                1 |
| BASINGER     |                1 |
| BENING       |                2 |
| BERGEN       |                1 |
| BERGMAN      |                1 |
| BERRY        |                3 |
| BIRCH        |                1 |
| BLOOM        |                1 |
| BOLGER       |                2 |
| BRIDGES      |                1 |
| BRODY        |                2 |
| BULLOCK      |                1 |
| CAGE         |                2 |
| CARREY       |                1 |
| CHAPLIN      |                1 |
| CHASE        |                2 |
| CLOSE        |                1 |
| COSTNER      |                1 |
| CRAWFORD     |                2 |
| CRONYN       |                2 |
| CROWE        |                1 |
| CRUISE       |                1 |
| CRUZ         |                1 |
| DAMON        |                1 |
| DAVIS        |                3 |
| DAY-LEWIS    |                1 |
| DEAN         |                2 |
| DEE          |                2 |
```

5. List last names of actors and the number of actors who have that last name, but only
for names that are shared by at least two actors



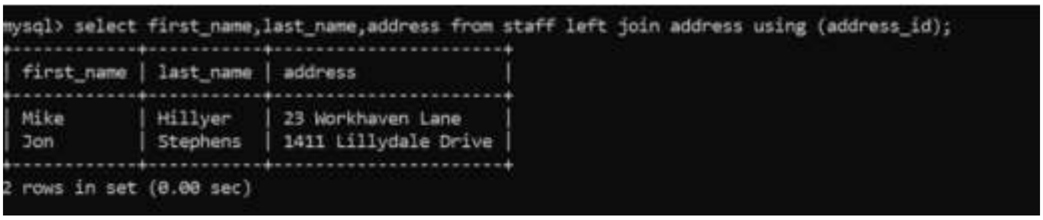
```
mysql>  select last_name ,count(last_name) from actor  where count(last_name)>=2 group by last_name having count(last_name)>=2;
ERROR 1111 (HY000): Invalid use of group function
mysql>  select last_name ,count(last_name) from actor group by last_name having count(last_name)>=2;
+-------------+------------------+
| last_name   | count(last_name) |
+-------------+------------------+
| AKROYD      |                3 |
| ALLEN       |                3 |
| BAILEY      |                2 |
| BENING      |                2 |
| BERRY       |                3 |
| BOLGER      |                2 |
| BRODY       |                2 |
| CAGE        |                2 |
| CHASE       |                2 |
| CRAWFORD    |                2 |
| CRONYN      |                2 |
| DAVIS       |                3 |
| DEAN        |                2 |
| DEE         |                2 |
| DEGENERES   |                3 |
| DENCH       |                2 |
| DEPP        |                2 |
| DUKAKIS     |                2 |
| FAWCETT     |                2 |
| GARLAND     |                3 |
| GOODING     |                2 |
| GUINESS     |                3 |
| HACKMAN     |                2 |
| HARRIS      |                3 |
| HOFFMAN     |                3 |
| HOPKINS     |                3 |
| HOPPER      |                2 |
| JACKMAN     |                2 |
| JOHANSSON   |                3 |
| KEITEL      |                3 |
| KILMER      |                5 |
| MCCONAUGHEY |                2 |
| MCKELLEN    |                2 |
| MCQUEEN     |                2 |
| MONROE      |                2 |
```

6. The actor HARPO WILLIAMS was accidentally entered in the actor table as GROUCHO WILLIAMS. Write a query to fix the record.



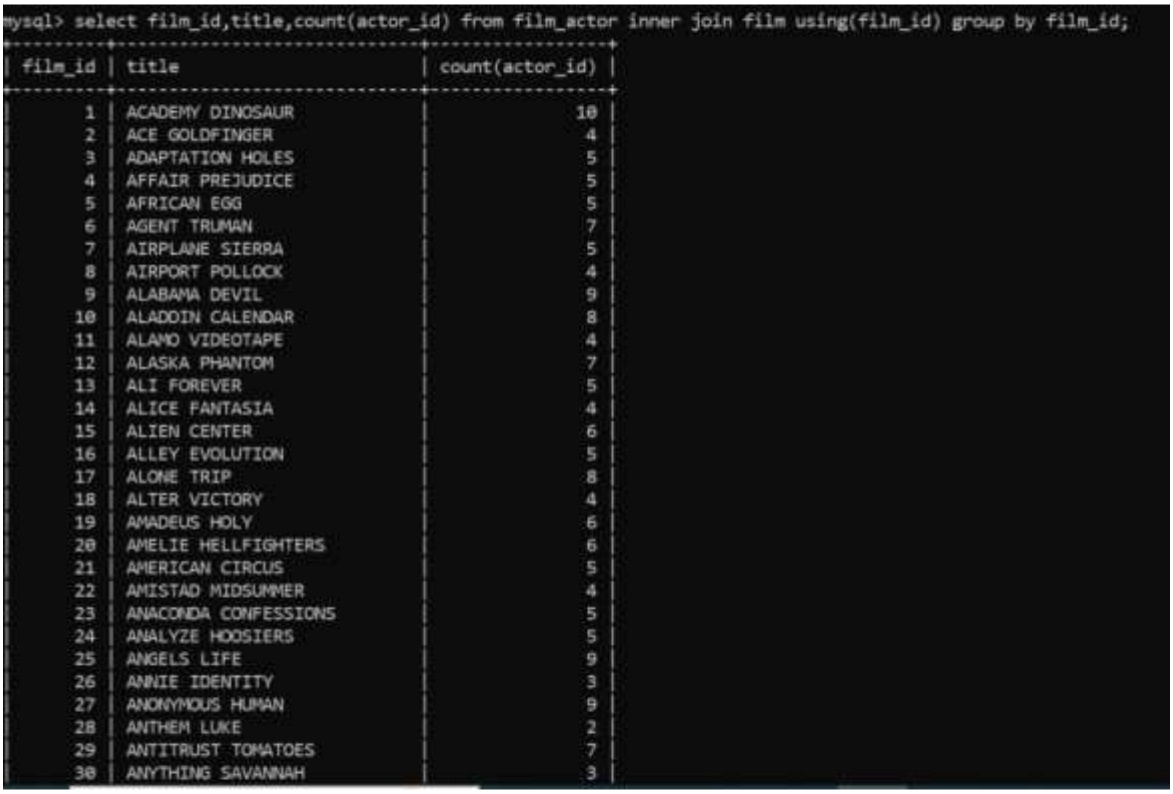
```
mysql> update actor
    -> set first_name ='HARPO'
    -> WHERE LAST_name='williams';
Query OK, 3 rows affected (0.02 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> select first_name from actor where last_name='williams';
+------------+
| first_name |
+------------+
| HARPO      |
| HARPO      |
| HARPO      |
+------------+
```

7. Use JOIN to display the first and last names, as well as the address, of each staff member. Use the tables staff and address:



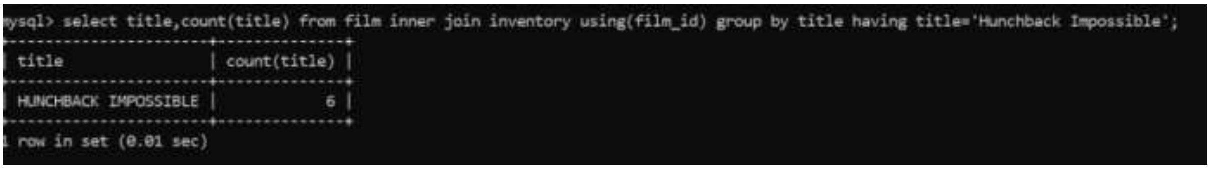
```
mysql> select first_name,last_name,address from staff left join address using (address_id);
+------------+-----------+----------------------+
| first_name | last_name | address              |
+------------+-----------+----------------------+
| Mike       | Hillyer   | 23 Workhaven Lane    |
| Jon        | Stephens  | 1411 Lillydale Drive |
+------------+-----------+----------------------+
2 rows in set (0.00 sec)
```

8. List each film and the number of actors who are listed for that film. Use tables film_actor and film. Use inner join.



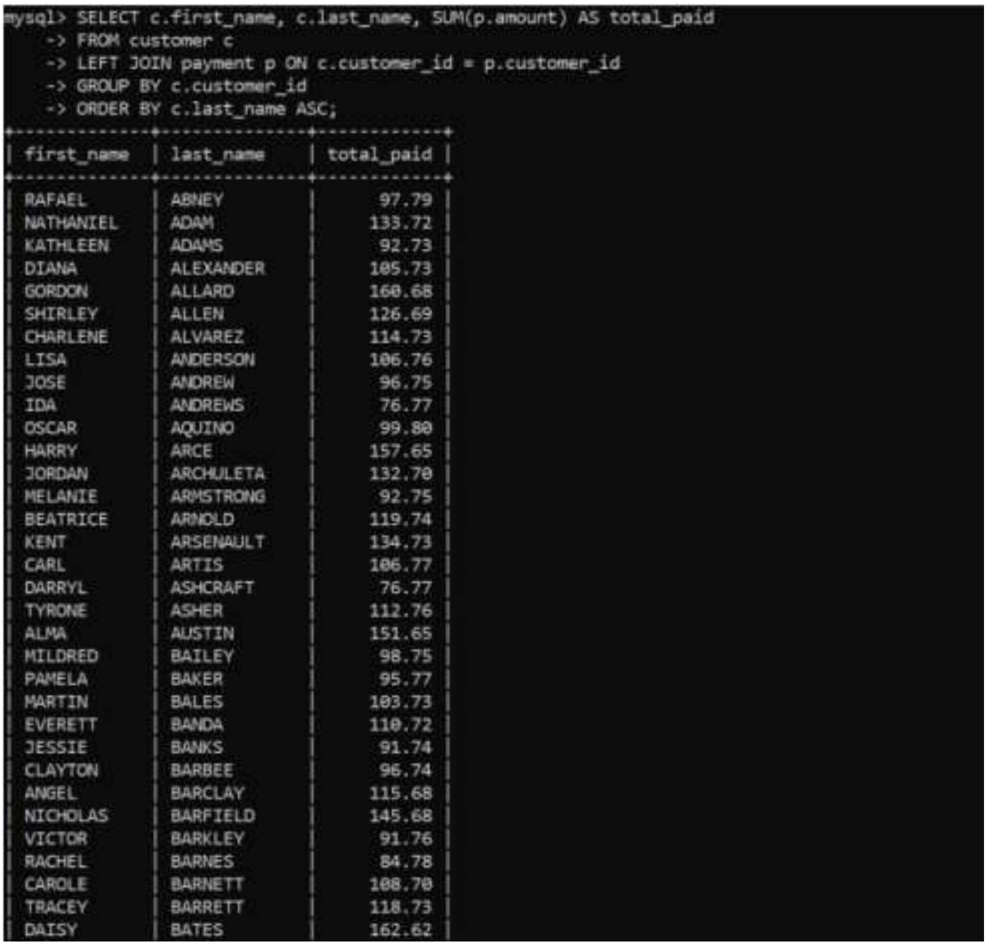
```
mysql> select film_id,title,count(actor_id) from film_actor inner join film using(film_id) group by film_id;
+---------+-----------------------------+-----------------+
| film_id | title                       | count(actor_id) |
+---------+-----------------------------+-----------------+
|       1 | ACADEMY DINOSAUR            |              10 |
|       2 | ACE GOLDFINGER              |               4 |
|       3 | ADAPTATION HOLES            |               5 |
|       4 | AFFAIR PREJUDICE            |               5 |
|       5 | AFRICAN EGG                 |               5 |
|       6 | AGENT TRUMAN                |               7 |
|       7 | AIRPLANE SIERRA             |               5 |
|       8 | AIRPORT POLLOCK             |               4 |
|       9 | ALABAMA DEVIL               |               9 |
|      10 | ALADDIN CALENDAR            |               8 |
|      11 | ALAMO VIDEOTAPE             |               4 |
|      12 | ALASKA PHANTOM              |               7 |
|      13 | ALI FOREVER                 |               5 |
|      14 | ALICE FANTASIA              |               4 |
|      15 | ALIEN CENTER                |               6 |
|      16 | ALLEY EVOLUTION             |               5 |
|      17 | ALONE TRIP                  |               8 |
|      18 | ALTER VICTORY               |               4 |
|      19 | AMADEUS HOLY                |               6 |
|      20 | AMELIE HELLFIGHTERS         |               6 |
|      21 | AMERICAN CIRCUS             |               5 |
|      22 | AMISTAD MIDSUMMER           |               4 |
|      23 | ANACONDA CONFESSIONS        |               5 |
|      24 | ANALYZE HOOSIERS            |               5 |
|      25 | ANGELS LIFE                 |               9 |
|      26 | ANNIE IDENTITY              |               3 |
|      27 | ANONYMOUS HUMAN             |               9 |
|      28 | ANTHEM LUKE                 |               2 |
|      29 | ANTITRUST TOMATOES          |               7 |
|      30 | ANYTHING SAVANNAH           |               3 |
```

9. How many copies of the film Hunchback Impossible exist in the inventory system?



```
mysql> select title,count(title) from film inner join inventory using(film_id) group by title having title='Hunchback Impossible';
+---------------------+--------------+
| title               | count(title) |
+---------------------+--------------+
| HUNCHBACK IMPOSSIBLE |            6 |
+---------------------+--------------+
1 row in set (0.01 sec)
```

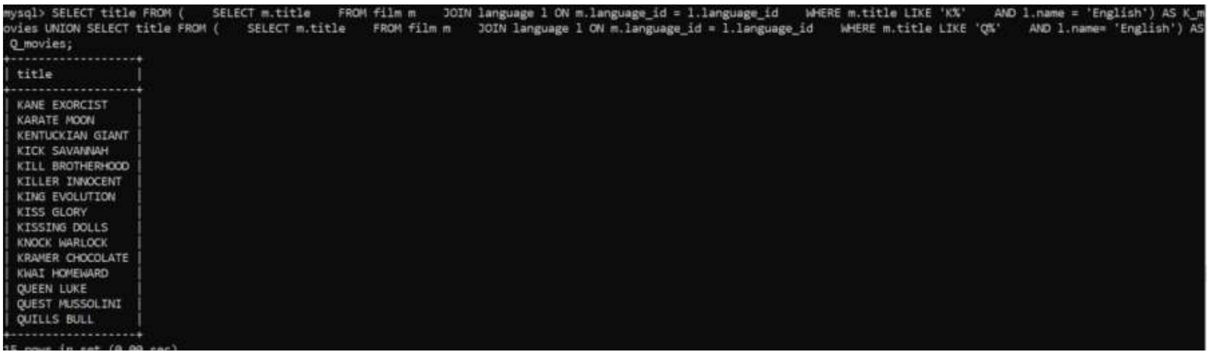
10. Using the tables payment and customer and the JOIN command, list the total paid by
each customer. List the customers alphabetically by last name



```
mysql> SELECT 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 ASC;
+-------------+-------------+------------+
| first_name  | last_name   | total_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 |
| CHARLENE    | ALVAREZ     |     114.73 |
| LISA        | ANDERSON    |     106.76 |
| JOSE        | ANDREW      |      96.75 |
| IDA         | ANDREWS     |      76.77 |
| OSCAR       | AQUINO      |      99.80 |
| HARRY       | ARCE        |     157.65 |
| JORDAN      | ARCHULETA   |     132.70 |
| MELANIE     | ARMSTRONG   |      92.75 |
| BEATRICE    | ARNOLD      |     119.74 |
| KENT        | ARSENAULT   |     134.73 |
| CARL        | ARTIS       |     106.77 |
| DARRYL      | ASHCRAFT    |      76.77 |
| TYRONE      | ASHER       |     112.76 |
| ALMA        | AUSTIN      |     151.65 |
| MILDRED     | BAILEY      |      98.75 |
| PAMELA      | BAKER       |      95.77 |
| MARTIN      | BALES       |     103.73 |
| EVERETT     | BANDA       |     110.72 |
| JESSIE      | BANKS       |      91.74 |
| CLAYTON     | BARBEE      |      96.74 |
| ANGEL       | BARCLAY     |     115.68 |
| NICHOLAS    | BARFIELD    |     145.68 |
| VICTOR      | BARKLEY     |      91.76 |
| RACHEL      | BARNES      |      84.78 |
| CAROLE      | BARNETT     |     108.70 |
| TRACEY      | BARRETT     |     118.73 |
| DAISY       | BATES       |     162.62 |
```

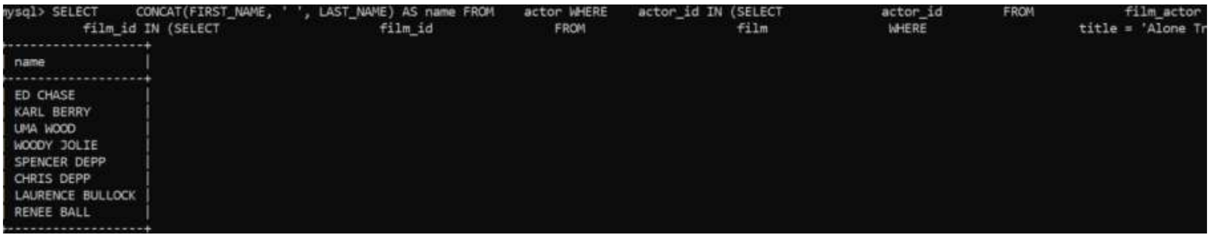
11.The music of Queen and Kris Kristofferson have seen an unlikely resurgence.

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.



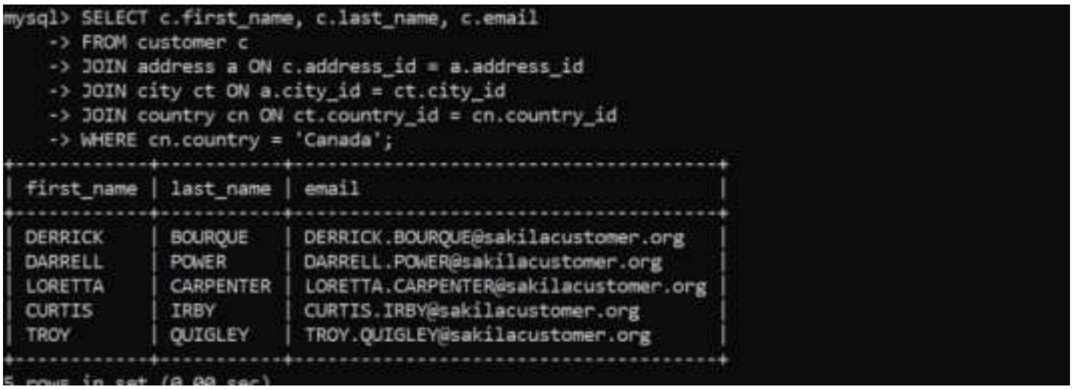
```
mysql> SELECT title FROM (    SELECT m.title    FROM film m    JOIN language l ON m.language_id = l.language_id    WHERE m.title LIKE 'K%'    AND l.name = 'English') AS K_m
ovies UNION SELECT title FROM (    SELECT m.title    FROM film m    JOIN language l ON m.language_id = l.language_id    WHERE m.title LIKE 'Q%'    AND l.name= 'English') AS
 Q_movies;
+------------------+
| title            |
+------------------+
| KANE EXORCIST    |
| KARATE MOON      |
| KENTUCKIAN GIANT |
| KICK SAVANNAH    |
| KILL BROTHERHOOD |
| KILLER INNOCENT  |
| KING EVOLUTION   |
| KISS GLORY       |
| KISSING DOLLS    |
| KNOCK WARLOCK    |
| KRAMER CHOCOLATE |
| KWAI HOMEWARD    |
| QUEEN LUKE       |
| QUEST MUSSOLINI  |
| QUILLS BULL      |
+------------------+
15 rows in set (0.00 sec)
```

12.use subqueries to display all actors who appear in the film `Alone Trip`.



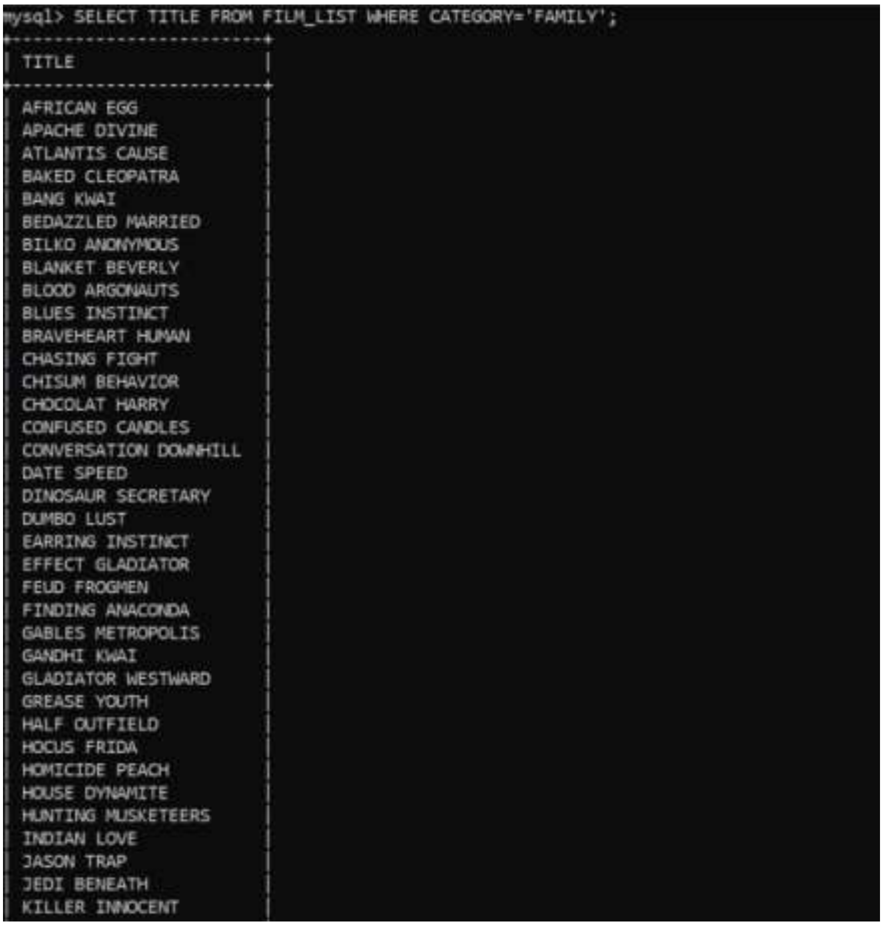
```
mysql> SELECT     CONCAT(FIRST_NAME, ' ', LAST_NAME) AS name FROM     actor WHERE     actor_id IN (SELECT             actor_id        FROM             film_actor
        film_id IN (SELECT                     film_id               FROM                   film             WHERE             title = 'Alone Tr
+------------------+
| name             |
+------------------+
| ED CHASE         |
| KARL BERRY       |
| UMA WOOD         |
| WOODY JOLIE      |
| SPENCER DEPP     |
| CHRIS DEPP       |
| LAURENCE BULLOCK |
| RENEE BALL       |
+------------------+
```

13.You want to run an email marketing campaign in Canada, for which you will
need the names and email addresses of all Canadian customers. Use joins to retrieve this information.



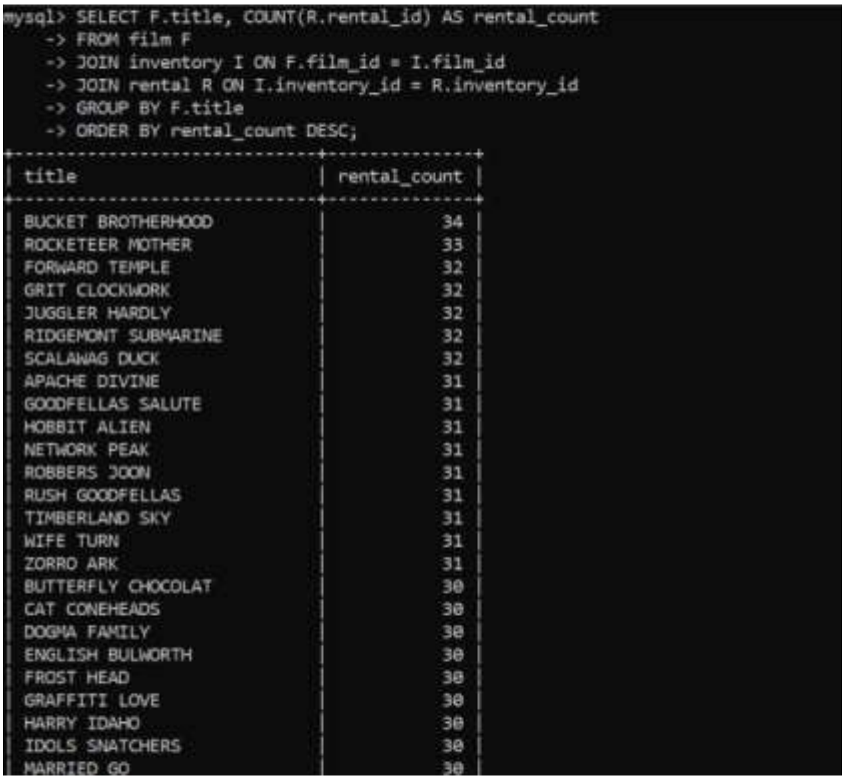
```
mysql> SELECT c.first_name, c.last_name, c.email
    -> FROM customer c
    -> JOIN address a ON c.address_id = a.address_id
    -> JOIN city ct ON a.city_id = ct.city_id
    -> JOIN country cn ON ct.country_id = cn.country_id
    -> WHERE cn.country = 'Canada';
+------------+-----------+---------------------------------------+
| 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       |
+------------+-----------+---------------------------------------+
5 rows in set (0.00 sec)
```

14.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.



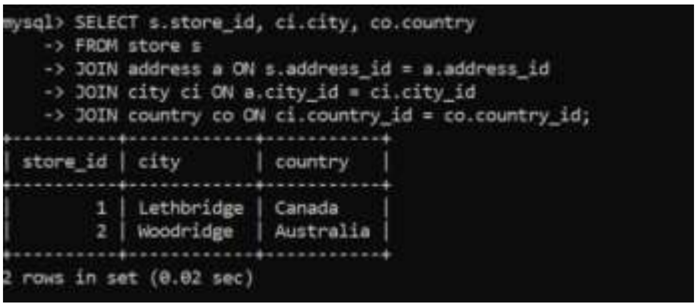
```
mysql> SELECT TITLE FROM FILM_LIST WHERE CATEGORY='FAMILY';
+-------------------------+
| TITLE                   |
+-------------------------+
| AFRICAN EGG             |
| APACHE DIVINE           |
| ATLANTIS CAUSE          |
| BAKED CLEOPATRA         |
| BANG KWAI               |
| BEDAZZLED MARRIED       |
| BILKO ANONYMOUS         |
| BLANKET BEVERLY         |
| BLOOD ARGONAUTS         |
| BLUES INSTINCT          |
| BRAVEHEART HUMAN        |
| CHASING FIGHT           |
| CHISUM BEHAVIOR         |
| CHOCOLAT HARRY          |
| CONFUSED CANDLES        |
| CONVERSATION DOWNHILL   |
| DATE SPEED              |
| DINOSAUR SECRETARY      |
| DUMBO LUST              |
| EARRING INSTINCT        |
| EFFECT GLADIATOR        |
| FEUD FROGMEN            |
| FINDING ANACONDA        |
| GABLES METROPOLIS       |
| GANDHI KWAI             |
| GLADIATOR WESTWARD      |
| GREASE YOUTH            |
| HALF OUTFIELD           |
| HOCUS FRIDA             |
| HOMICIDE PEACH          |
| HOUSE DYNAMITE          |
| HUNTING MUSKETEERS      |
| INDIAN LOVE             |
| JASON TRAP              |
| JEDI BENEATH            |
| KILLER INNOCENT         |
```

16.Display the most frequently rented movies in descending order.



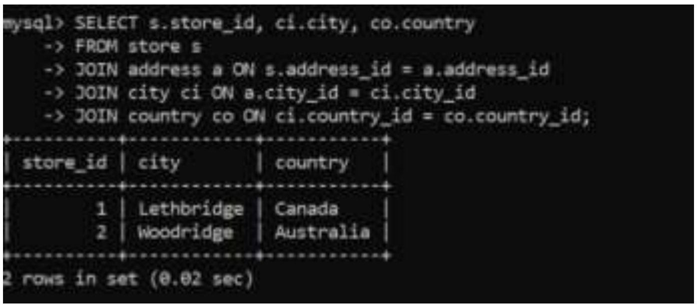
```
mysql> SELECT F.title, COUNT(R.rental_id) AS rental_count
    -> FROM film F
    -> JOIN inventory I ON F.film_id = I.film_id
    -> JOIN rental R ON I.inventory_id = R.inventory_id
    -> GROUP BY F.title
    -> ORDER BY rental_count DESC;
+-----------------------------+--------------+
| title                       | rental_count |
+-----------------------------+--------------+
| BUCKET BROTHERHOOD          |           34 |
| ROCKETEER MOTHER            |           33 |
| FORWARD TEMPLE              |           32 |
| GRIT CLOCKWORK              |           32 |
| JUGGLER HARDLY              |           32 |
| RIDGEMONT SUBMARINE         |           32 |
| SCALAWAG DUCK               |           32 |
| APACHE DIVINE               |           31 |
| GOODFELLAS SALUTE           |           31 |
| HOBBIT ALIEN                |           31 |
| NETWORK PEAK                |           31 |
| ROBBERS JOON                |           31 |
| RUSH GOODFELLAS             |           31 |
| TIMBERLAND SKY              |           31 |
| WIFE TURN                   |           31 |
| ZORRO ARK                   |           31 |
| BUTTERFLY CHOCOLAT          |           30 |
| CAT CONEHEADS               |           30 |
| DOGMA FAMILY                |           30 |
| ENGLISH BULWORTH            |           30 |
| FROST HEAD                  |           30 |
| GRAFFITI LOVE               |           30 |
| HARRY IDAHO                 |           30 |
| IDOLS SNATCHERS             |           30 |
| MARRIED GO                  |           30 |
```

17.Write a query to display for each store its store ID, city, and country.



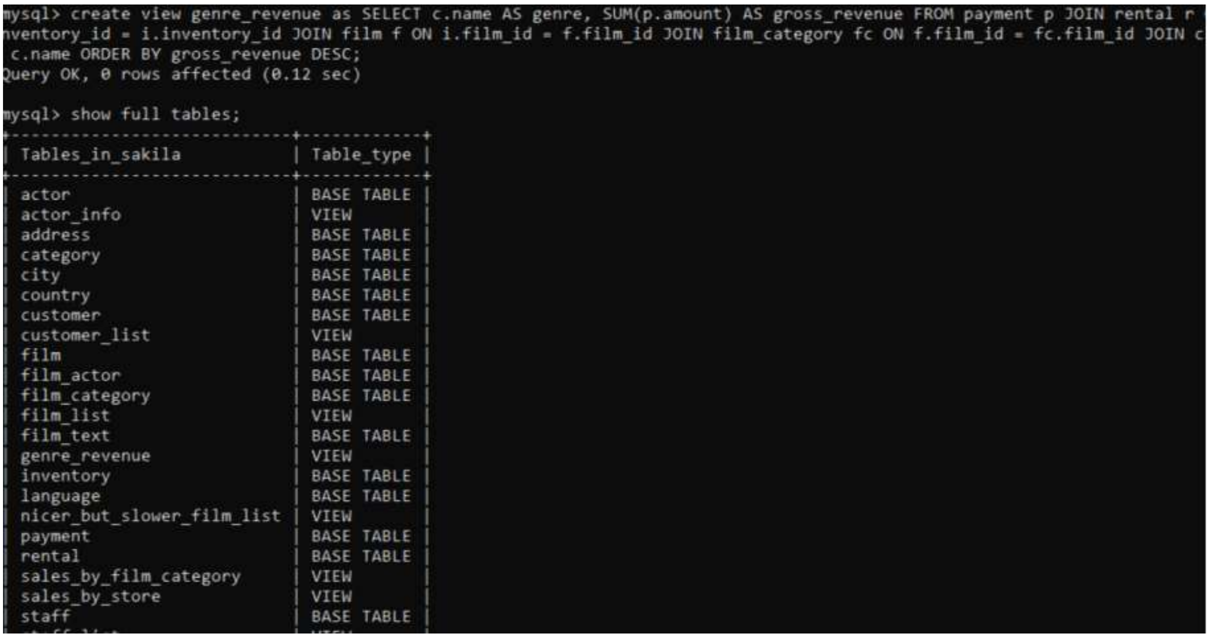
```
mysql> SELECT s.store_id, ci.city, co.country
    -> FROM store s
    -> JOIN address a ON s.address_id = a.address_id
    -> JOIN city ci ON a.city_id = ci.city_id
    -> JOIN country co ON ci.country_id = co.country_id;
+----------+------------+-----------+
| store_id | city       | country   |
+----------+------------+-----------+
|        1 | Lethbridge | Canada    |
|        2 | Woodridge  | Australia |
+----------+------------+-----------+
2 rows in set (0.02 sec)
```

18. List the genres and its gross revenue.



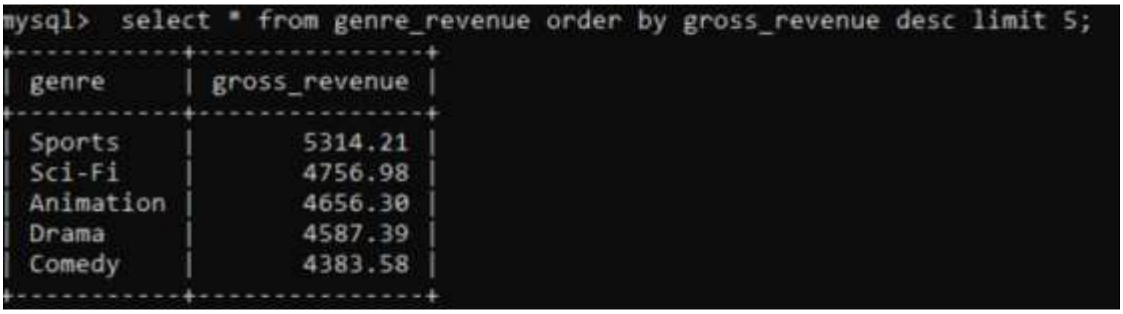
```
mysql> SELECT s.store_id, ci.city, co.country
    -> FROM store s
    -> JOIN address a ON s.address_id = a.address_id
    -> JOIN city ci ON a.city_id = ci.city_id
    -> JOIN country co ON ci.country_id = co.country_id;
+----------+------------+-----------+
| store_id | city       | country   |
+----------+------------+-----------+
|        1 | Lethbridge | Canada    |
|        2 | Woodridge  | Australia |
+----------+------------+-----------+
2 rows in set (0.02 sec)
```

19.Create a View for the above query(18)



```
mysql> create view genre_revenue as SELECT c.name AS genre, SUM(p.amount) AS gross_revenue FROM payment p JOIN rental r
nventory_id = i.inventory_id JOIN film f ON i.film_id = f.film_id JOIN film_category fc ON f.film_id = fc.film_id JOIN c
 c.name ORDER BY gross_revenue DESC;
Query OK, 0 rows affected (0.12 sec)

mysql> show full tables;
+----------------------------+------------+
| Tables_in_sakila           | Table_type |
+----------------------------+------------+
| actor                      | BASE TABLE |
| actor_info                 | VIEW       |
| address                    | BASE TABLE |
| category                   | BASE TABLE |
| city                       | BASE TABLE |
| country                    | BASE TABLE |
| customer                   | BASE TABLE |
| customer_list              | VIEW       |
| film                       | BASE TABLE |
| film_actor                 | BASE TABLE |
| film_category              | BASE TABLE |
| film_list                  | VIEW       |
| film_text                  | BASE TABLE |
| genre_revenue              | VIEW       |
| inventory                  | BASE TABLE |
| language                   | BASE TABLE |
| nicer_but_slower_film_list | VIEW       |
| payment                    | BASE TABLE |
| rental                     | BASE TABLE |
| sales_by_film_category     | VIEW       |
| sales_by_store             | VIEW       |
| staff                      | BASE TABLE |
```

20.Select top 5 genres in gross revenue view.



```
mysql>  select * from genre_revenue order by gross_revenue desc limit 5;
+-----------+---------------+
| genre     | gross_revenue |
+-----------+---------------+
| Sports    |       5314.21 |
| Sci-Fi    |       4756.98 |
| Animation |       4656.30 |
| Drama     |       4587.39 |
| Comedy    |       4383.58 |
+-----------+---------------+
```