Exercises

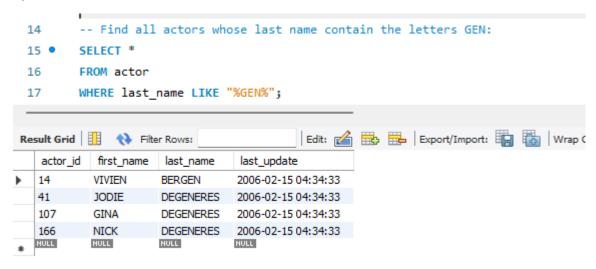
1. Display the first and last name of each actor in a single column in upper case letters in alphabetic order. Name the column Actor Name.

```
→ /* Display the first and last name of each actor in a single column in upper case letters in alphabetic order.

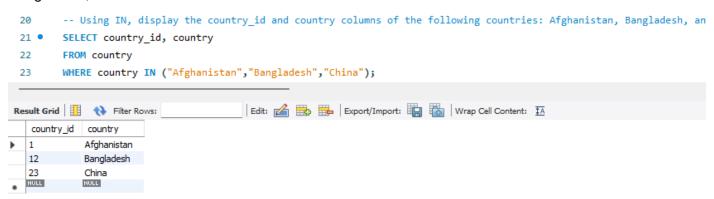
     Name the column Actor Name */
      SELECT
          CONCAT(UPPER(first_name), ' ', UPPER(last_name)) AS "Actor Name"
8
9
      FROM
10
          actor
      ORDER BY
11
          CONCAT(first_name, ' ',last_name);
 Export: Wrap Cell Content: IA
    Actor Name
    ADAM GRANT
    ADAM HOPPER
    AL GARLAND
    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
```

DUDT DUIZAZZO

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



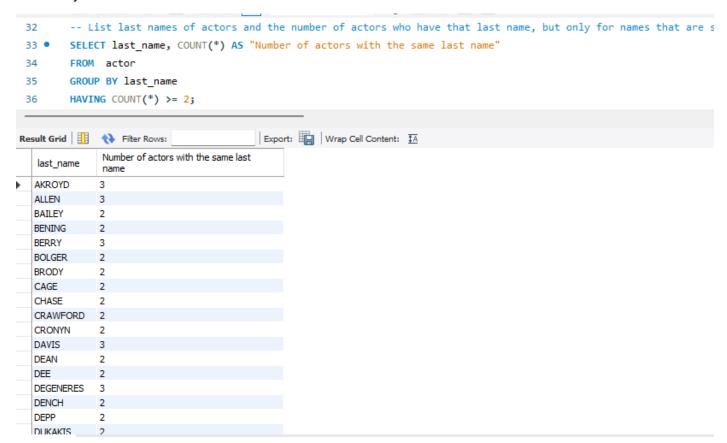
3.Using IN, display the country_id and country columns of the following countries: Afghanistan, Bangladesh, and China:



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

```
-- List the last names of actors, as well as how many actors have that last name
27 •
        SELECT last_name, COUNT(*) AS "Number of actors with the same last name"
28
        FROM actor
        GROUP BY last_name;
29
Export: Wrap Cell Content: IA
              Number of actors with the same last
   last_name
              name
  AKROYD
              3
  ALLEN
              3
  ASTAIRE
              1
  BACALL
              1
  BAILEY
              2
              1
  BALE
  BALL
              1
              1
  BARRYMORE
  BASINGER
              1
  BENING
              2
  BERGEN
              1
              1
  BERGMAN
  BERRY
              3
  BIRCH
              1
  BLOOM
              2
  BOLGER
  BRIDGES
              1
 BRODY
tesult 9 x
```

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



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

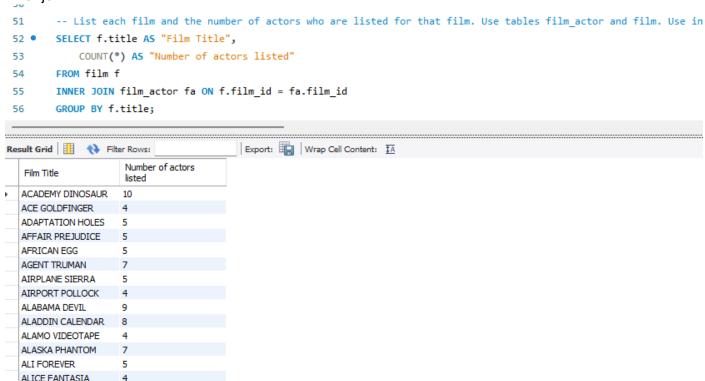
```
39
        -- The actor HARPO WILLIAMS was accidentally entered in the actor table as GROUCHO WILLIAMS. Write a query to fix
 40 •
        UPDATE actor
        SET first_name = "HARPO"
 41
 42
        WHERE first name = "GROUCHO" AND last name = "WILLIAMS";
 43
       SELECT * FROM actor WHERE first_name = 'HARPO';
 44 •
Result Grid Filter Rows:
                                        Edit: 🚄 🖶 Export/Import: 📳 📸 Wrap Cell Content: 🖽
   actor_id first_name last_name last_update
  172
          HARPO
                    WILLIAMS
                              2024-10-30 07:58:37
```

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

```
46 -- Use JOIN to display the first and last names, as well as the address, of each staff member. Use the tables
47 ● SELECT s.first_name, s.last_name, a.address
48 FROM staff s
49 JOIN address a ON s.address_id = a.address_id;

Result Grid 
Filter Rows: | Export: | Wrap Cell Content: 
First_name | last_name | address |
| Mike | Hillyer | 23 Workhaven Lane |
| Jon | Stephens | 1411 Lillydale Drive |
```

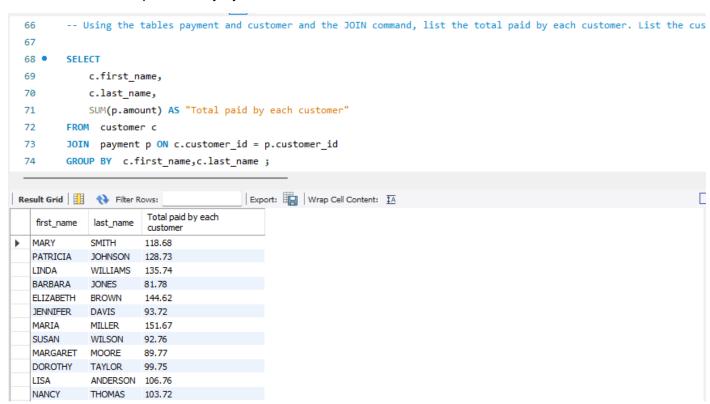
8.List each film and the number of actors who are listed for that film. Use tables film_actor and film. Use inner join.



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

```
57
         -- How many copies of the film Hunchback Impossible exist in the inventory system?
 58
 59
        SELECT COUNT(*) AS "Number of Hunchback Impossible copies in the inventory"
 60 •
        FROM inventory i
 61
        JOIN film f ON i.film_id = f.film_id
 62
        WHERE f.title="HUNCHBACK IMPOSSIBLE";
 63
                                        Export: Wrap Cell Content: IA
Number of Hunchback Impossible copies in
   the inventory
6
```

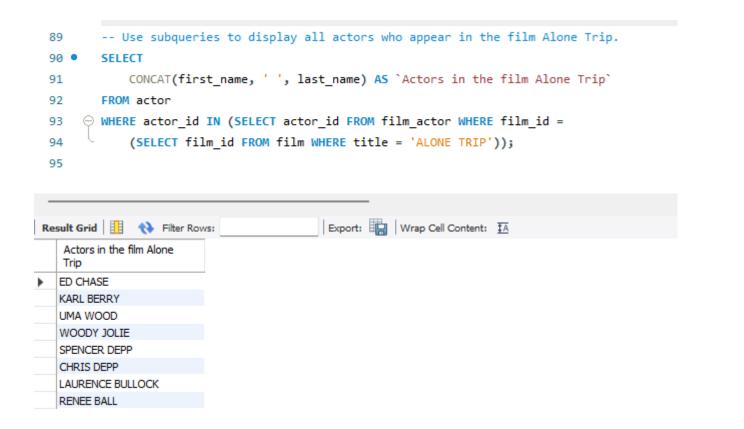
10. Using the tables payment and customer and the JOIN command, list the total paid by each customer. List the customers alphabetically by last name



11.The music of Queen and Kris Kristofferson have seen an unlikely resurgence. As an unintended consequence, films starting with the letters κ and ϱ have also soared in popularity. Use subqueries to display the titles of movies starting with the letters κ and ϱ whose language is English

```
-- The music of Queen and Kris Kristofferson have seen an unlikely resurgence. As an unintended consequence,
77
78
 79 •
        SELECT
            f.title
 81
        FROM
            film f
 82
        WHERE
 83
            (f.title LIKE "K%" OR f.title LIKE "Q%") AND f.language_id =
            ( SELECT l.language_id FROM language l WHERE name = "English");
Export: Wrap Cell Content: IA
  KANE EXORCIST
  KARATE MOON
  KENTUCKIAN GIANT
  KICK SAVANNAH
  KILL BROTHERHOOD
  KILLER INNOCENT
  KING EVOLUTION
  KISS GLORY
  KISSING DOLLS
  KNOCK WARLOCK
  KRAMER CHOCOLATE
  KWAI HOMEWARD
  OUEEN LUKE
```

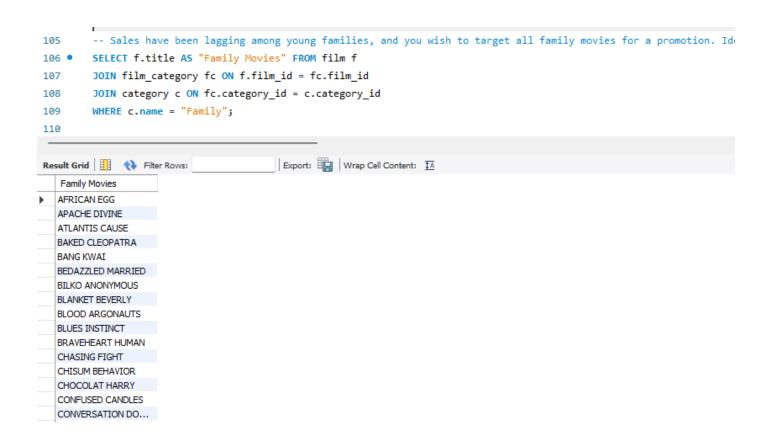
12. Use subqueries to display all actors who appear in the film Alone trip



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

```
96
        -- You want to run an email marketing campaign in Canada, for which you will need the names and email addresses
97
       SELECT c.first_name, c.last_name ,c.email
98 •
99
       FROM customer c
        JOIN address a ON c.address_id = a.address_id
100
        JOIN city ci ON a.city_id = ci.city_id
101
       JOIN country co ON ci.country id = co.country id
102
        WHERE co.country = 'Canada';
103
                                       Export: Wrap Cell Content: 1A
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
```

14. Sales have been lagging among young families, and you wish to target all family movies for a promotion. Identify all movies categorized as family films.



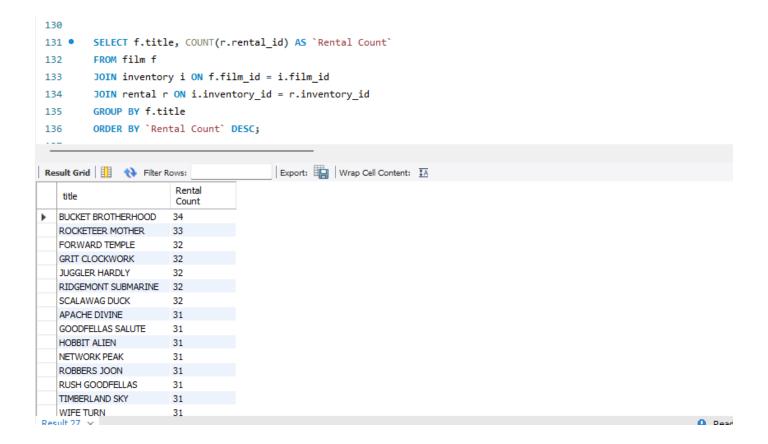
15 .Create a Stored procedure to get the count of films in the input category (IN category_name, OUT count)

```
111
        -- Create a Stored procedure to get the count of films in the input category (IN category_name, OUT count)
        DELIMITER //
112
114 •
        CREATE PROCEDURE FilmCountByCategory(IN category_name VARCHAR(255), OUT film_count INT)

→ BEGIN

115
116
            SELECT COUNT(*) INTO film_count
117
            FROM film f
            JOIN film_category fc ON f.film_id = fc.film_id
118
            JOIN category c ON fc.category_id = c.category_id
119
            WHERE c.name = category_name;
            END //
121
            DELIMITER ;
122
            CALL FilmCountByCategory('Classics', @film_count);
124
125 •
            SELECT @film_count;
126
                                      Export: Wrap Cell Content: IA
@film_count
57
```

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



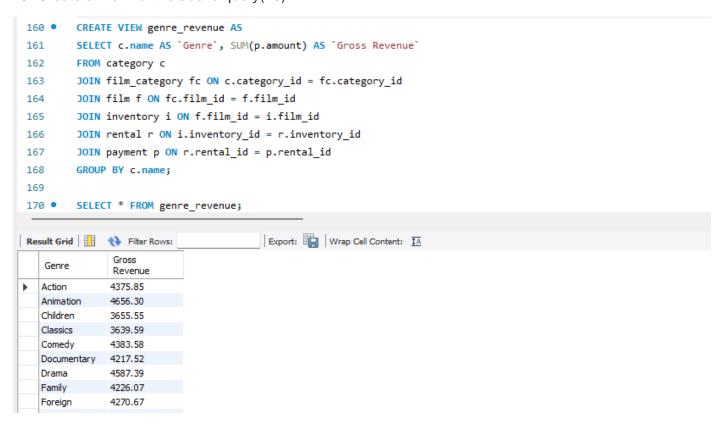
17. Write a query to display for each store its store ID, city, and country.

```
-- Write a query to display for each store its store ID, city, and country.
138
         SELECT s.store_id AS `Store ID`,
 139 •
                ci.city AS `City`,
 140
                co.country A5 `Country`
 141
         FROM store s
142
 143
         JOIN address a ON s.address id = a.address id
         JOIN city ci ON a.city_id = ci.city_id
 144
         JOIN country co ON ci.country_id = co.country_id;
 145
 1/16
Export: Wrap Cell Content: IA
    Store ID
            City
                      Country
   1
           Lethbridge
                     Canada
   2
           Woodridge
                     Australia
```

18. List the genres and its gross revenue.

```
145
         -- List the genres and its gross revenue.
         SELECT c.name AS `Genre`, SUM(p.amount) AS `Gross Revenue`
147
         FROM category c
148
         JOIN film category fc ON c.category id = fc.category id
149
         JOIN film f ON fc.film id = f.film_id
150
         JOIN inventory i ON f.film id = i.film id
151
         JOIN rental r ON i.inventory_id = r.inventory_id
152
         JOIN payment p ON r.rental_id = p.rental_id
153
         GROUP BY c.name;
154
Result Grid
              Filter Rows:
                                            Export: Wrap Cell Content: IA
                Gross
   Genre
                Revenue
               4375.85
  Action
  Animation
               4656.30
  Children
               3655.55
               3639.59
  Classics
               4383.58
  Comedy
  Documentary
               4217.52
  Drama
               4587.39
  Family
               4226.07
  Foreign
               4270.67
               4281.33
  Games
               3722.54
  Horror
```

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



20. Select top 5 genres in gross revenue view

