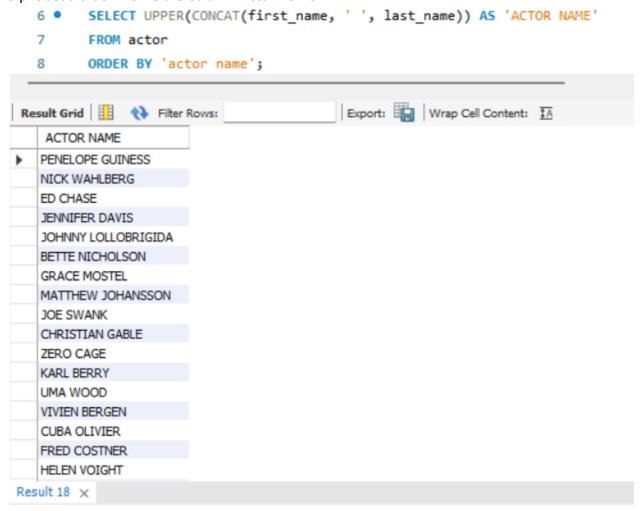
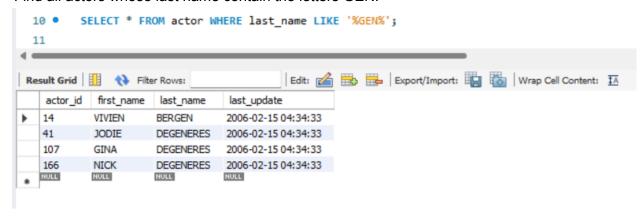
Exercises

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.



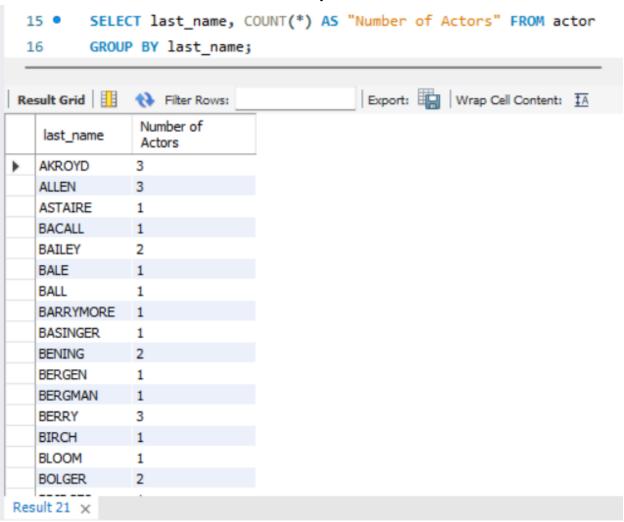
2. Find all actors whose last name contain the letters GEN:



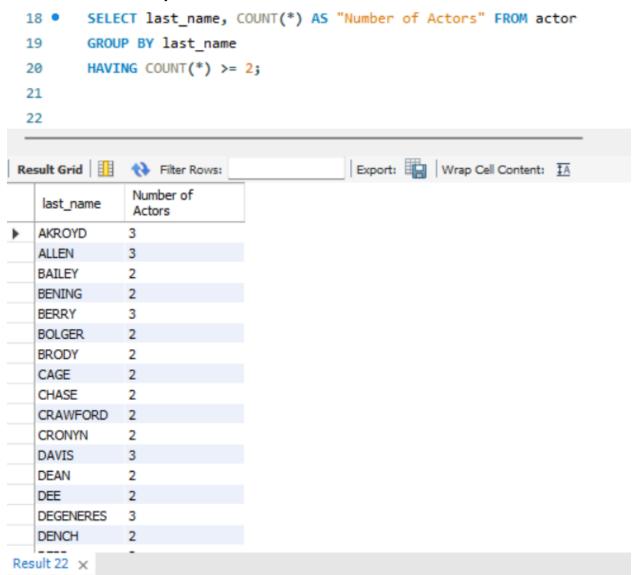
3. Using IN, display the country_id and country columns of the following countries: Afghanistan, Bangladesh, and China:



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



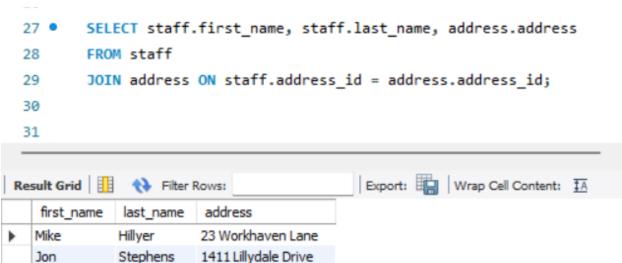
5. List last names of actors and the number of actors who have that last name, but only for names that are shared by at least two actors.



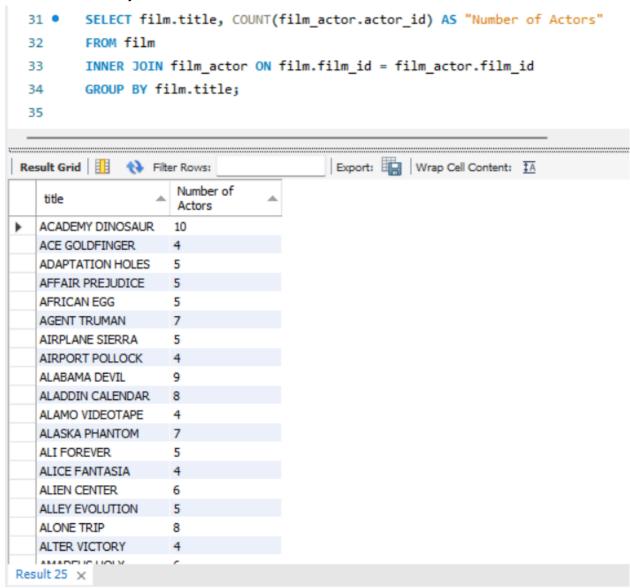
6. The actor HARPO WILLIAMS was accidentally entered in the actor table as GROUCHO WILLIAMS. Write a query to fix the record.



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:



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

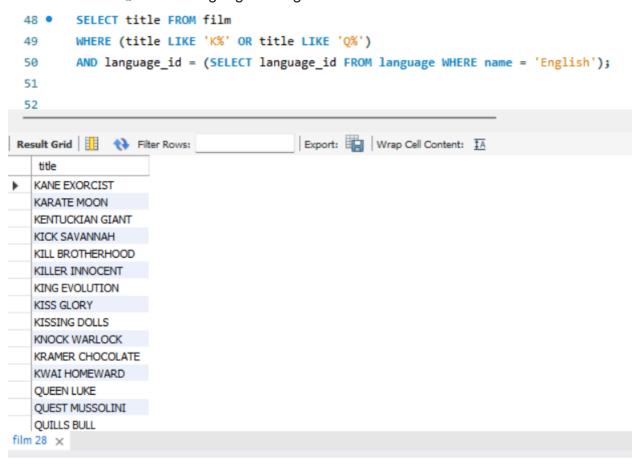


9. How many copies of the film Hunchback Impossible exist in the inventory system?

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

```
41 •
          SELECT c.first_name, c.last_name, SUM(p.amount) AS "Total Paid"
 42
          FROM payment p
          JOIN customer c ON p.customer id = c.customer id
 43
          GROUP BY c.first name, c.last name
 44
 45
         ORDER BY c.last_name, c.first_name;
 46
 47
Result Grid
                                              Export: Wrap Cell Content: $\overline{A}$
               Filter Rows:
                           Total
   first_name
               last_name
                           Paid
                           97.79
   RAFAEL
               ABNEY
   NATHANIEL
               ADAM
                           133.72
   KATHLEEN
               ADAMS
                           92.73
   DIANA
               ALEXANDER
                           105.73
   GORDON
               ALLARD
                           160.68
   SHIRLEY
               ALLEN
                           126.69
   CHARLENE
               ALVAREZ
                           114.73
               ANDERSON
                           106.76
   LISA
               ANDREW
                           96.75
   JOSE
   IDA
               ANDREWS
                           76.77
   OSCAR
               AQUINO
                           99.80
   HARRY
               ARCE
                           157.65
   JORDAN
               ARCHULETA
                           132.70
   MELANIE
               ARMSTRONG 92.75
Result 27 🗶
```

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



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

SELECT first_name, last_name FROM actor

RENEE

BALL

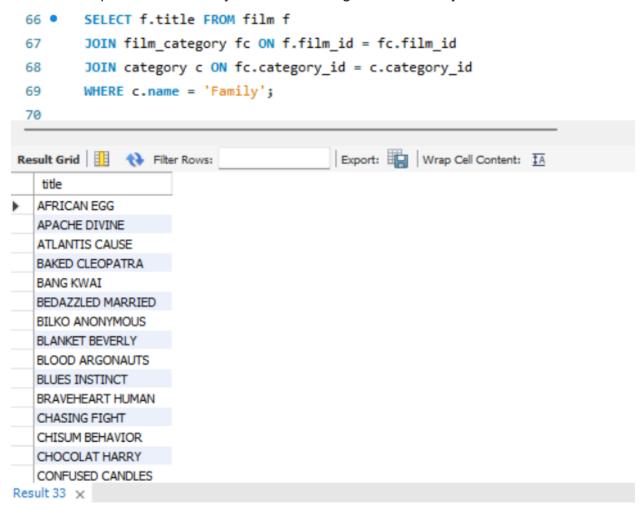
```
55
       WHERE actor_id IN (SELECT actor_id FROM film_actor WHERE film_id = (SELECT film_id FROM film WHERE title = 'Alone Trip'));
 56
 57
                                    Export: Wrap Cell Content: IA
first_name last_name
ED
           CHASE
  KARL
           BERRY
           WOOD
  WOODY
          JOLIE
  SPENCER
           DEPP
  CHRIS
           DEPP
  LAURENCE BULLOCK
```

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.

```
SELECT c.first_name, c.last_name, c.email
         FROM customer c
 59
         JOIN address a ON c.address_id = a.address_id
 60
         JOIN city ci ON a.city_id = ci.city_id
 61
         JOIN country co ON ci.country_id = co.country_id
 62
         WHERE co.country = 'Canada';
 63
 64
                                            Export: Wrap Cell Content: IA
Result Grid
             Filter Rows:
   first_name
             last_name
                         email
  DERRICK
             BOURQUE
                         DERRICK.BOURQUE@sakilacustomer.org
  DARRELL
             POWER
                         DARRELL.POWER@sakilacustomer.org
  LORETTA
                         LORETTA.CARPENTER@sakilacustomer.org
             CARPENTER
                         CURTIS.IRBY@sakilacustomer.org
  CURTIS
             IRBY
                         TROY.QUIGLEY@sakilacustomer.org
  TROY
             QUIGLEY
```

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)

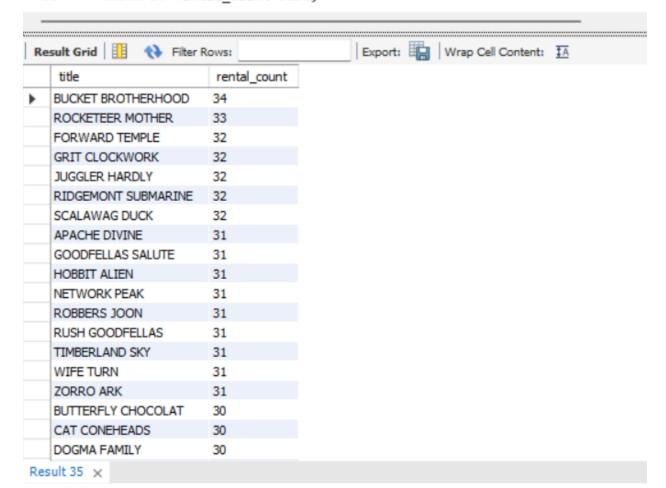
```
72
        DELIMITER //
 73
 74 • ○ CREATE PROCEDURE GetFilmCountByCategory(
             IN category name VARCHAR(255),
 75
            OUT film count INT
 76
 77
        )
 78

→ BEGIN

            SELECT COUNT(*) INTO film_count
 79
            FROM film f
 80
             JOIN film_category fc ON f.film_id = fc.film_id
 81
 82
             JOIN category c ON fc.category_id = c.category_id
            WHERE c.name = category name;
 83
       - END //
 84
 85
 86
        DELIMITER;
 87
 88
        CALL GetFilmCountByCategory('Family', @film_count);
 89 •
        SELECT @film count;
 90 •
 91
 92
                                         Export: Wrap Cell Content: IA
Result Grid
             Filter Rows:
   @film_count
 69
```

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

```
93 • SELECT f.title, COUNT(r.rental_id) AS rental_count
94    FROM film f
95    JOIN inventory i ON f.film_id = i.film_id
96    JOIN rental r ON i.inventory_id = r.inventory_id
97    GROUP BY f.title
98    ORDER BY rental count DESC;
```



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

```
101 • SELECT s.store_id, c.city, co.country
102 FROM store s
103     JOIN address a ON s.address_id = a.address_id
104     JOIN city c ON a.city_id = c.city_id
105     JOIN country co ON c.country_id = co.country_id;
106
```



18. List the genres and its gross revenue.

Result 37 ×

```
SELECT c.name AS genre, SUM(p.amount) AS gross_revenue
         FROM film f
110
         JOIN film_category fc ON f.film_id = fc.film_id
111
         JOIN category c ON fc.category_id = c.category_id
112
         JOIN inventory i ON f.film id = i.film id
113
         JOIN rental r ON i.inventory_id = r.inventory_id
114
         JOIN payment p ON r.rental_id = p.rental_id
115
         GROUP BY c.name;
116
                                            Export: Wrap Cell Content: IA
Result Grid
               Filter Rows:
   genre
                gross_revenue
  Action
               4375.85
  Animation
               4656.30
  Children
               3655.55
  Classics
               3639.59
  Comedy
               4383.58
  Documentary
               4217.52
  Drama
               4587.39
  Family
               4226.07
  Foreign
               4270.67
  Games
               4281.33
  Horror
               3722.54
  Music
               3417.72
```

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

```
CREATE VIEW genre_gross_revenue AS
120
        SELECT c.name AS genre, SUM(p.amount) AS gross revenue
        FROM film f
121
        JOIN film_category fc ON f .film_id = fc.film_id
122
        JOIN category c ON fc.category id = c.category id
123
        JOIN inventory i ON f.film id = i.film id
124
        JOIN rental r ON i.inventory_id = r.inventory_id
125
        JOIN payment p ON r.rental_id = p.rental_id
126
127
        GROUP BY c.name;
128
        SELECT * FROM genre gross revenue;
129 •
```

Export: Wrap Cell Content: IA Result Grid gross revenue genre Action 4375.85 Animation 4656.30 Children 3655.55 Classics 3639.59 Comedy 4383.58 Documentary 4217.52 4587.39 Drama Family 4226.07 4270.67 Foreign Games 4281.33 Horror 3722.54

Travel 3549.64
genre_gross_revenue 38 ×

3417.72

4351.62

4756.98

5314.21

Music

New

Sci-Fi

Sports

20. Select top 5 genres in gross revenue view.

