MySQL TASK-3

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 upper(concat(first_name,' ',last_name))as Actor_name
    -> from actor
    -> order by concat(first_name,' ',last_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 .

```
mysql> select first_name,last_name
    -> from actor
    -> where last_name like '%GEN%' ;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| VIVIEN     | BERGEN    |
| JODIE      | DEGENERES |
| GINA       | DEGENERES |
| NICK       | DEGENERES |
+------------+-----------+
4 rows in set (0.00 sec)
```

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       |
+------------+-------------+
3 rows in set (0.00 sec)
```

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 |
```

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) as count
    -> from actor
    -> group by last_name
    -> having count(last_name)>=2
    -> limit 10;
+-----------+-------+
| last_name | count |
+-----------+-------+
| AKROYD    |     3 |
| ALLEN     |     3 |
| BAILEY    |     2 |
| BENING    |     2 |
| BERRY     |     3 |
| BOLGER    |     2 |
| BRODY     |     2 |
| CAGE      |     2 |
| CHASE     |     2 |
| CRAWFORD  |     2 |
+-----------+-------+
10 rows in set (0.03 sec)
```

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 first_name='GROUCHO' and last_name='WILLIAMS' ;
Query OK, 1 row affected (0.17 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select first_name,last_name from actor where last_name='WILLIAMS' ;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| SEAN       | WILLIAMS  |
| MORGAN     | WILLIAMS  |
| HARPO      | WILLIAMS  |
+------------+-----------+
3 rows in set (0.00 sec)
```

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 as s
    -> join address as a
    -> ON a.address_id = s.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 title,count(actor_id)as 'Count of Actors'
    -> from film
    -> inner join film_actor USING (film_id)
    -> group by title
    -> Limit 20;
+---------------------+-----------------+
| title               | Count of Actors |
+---------------------+-----------------+
| ACADEMY DINOSAUR    |              10 |
| ACE GOLDFINGER      |               4 |
| ADAPTATION HOLES    |               5 |
| AFFAIR PREJUDICE    |               5 |
| AFRICAN EGG         |               5 |
| AGENT TRUMAN        |               7 |
| AIRPLANE SIERRA     |               5 |
| AIRPORT POLLOCK     |               4 |
| ALABAMA DEVIL       |               9 |
| ALADDIN CALENDAR    |               8 |
| ALAMO VIDEOTAPE     |               4 |
| ALASKA PHANTOM      |               7 |
| ALI FOREVER         |               5 |
| ALICE FANTASIA      |               4 |
| ALIEN CENTER        |               6 |
| ALLEY EVOLUTION     |               5 |
| ALONE TRIP          |               8 |
| ALTER VICTORY       |               4 |
| AMADEUS HOLY        |               6 |
| AMELIE HELLFIGHTERS |               6 |
+---------------------+-----------------+
```

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

```
mysql> select title,count(inventory_id) as 'Count of Copy'
    -> from film f
    -> join inventory i
    -> ON f.film_id = i.film_id
    -> where f.title = 'Hunchback Impossible' ;
+----------------------+---------------+
| title                | Count of Copy |
+----------------------+---------------+
| HUNCHBACK IMPOSSIBLE |             6 |
+----------------------+---------------+
1 row in set (0.00 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.customer_id ,c.last_name ,c.first_name, SUM(p.amount) as Total
    -> from payment p
    -> join customer c
    -> ON p.customer_id = c.customer_id
    -> group by c.customer_id, c.last_name,c.first_name
    -> order by c.last_name limit 10;
+-------------+-----------+------------+--------+
| customer_id | last_name | first_name | Total  |
+-------------+-----------+------------+--------+
|         505 | ABNEY     | RAFAEL     |  97.79 |
|         504 | ADAM      | NATHANIEL  | 133.72 |
|          36 | ADAMS     | KATHLEEN   |  92.73 |
|          96 | ALEXANDER | DIANA      | 105.73 |
|         470 | ALLARD    | GORDON     | 160.68 |
|          27 | ALLEN     | SHIRLEY    | 126.69 |
|         220 | ALVAREZ   | CHARLENE   | 114.73 |
|          11 | ANDERSON  | LISA       | 106.76 |
|         326 | ANDREW    | JOSE       |  96.75 |
|         183 | ANDREWS   | IDA        |  76.77 |
+-------------+-----------+------------+--------+
10 rows in set (0.16 sec)
```

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 film
    -> where (title like 'K%' OR title like 'Q%') AND
    -> language_id=(select language_id from language where name = 'English');
+------------------+
| 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 (0.16 sec)
```

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

```
mysql> select concat(first_name,' ',last_name) as 'Actors in Alone Trip'
    -> from actor
    -> WHERE actor_id IN (select actor_id from film_actor
    -> where film_id = (select film_id from film where title = 'Alone Trip'));
+----------------------+
| Actors in Alone Trip |
+----------------------+
| ED CHASE             |
| KARL BERRY           |
| UMA WOOD             |
| WOODY JOLIE          |
| SPENCER DEPP         |
| CHRIS DEPP           |
| LAURENCE BULLOCK     |
| RENEE BALL           |
+----------------------+
8 rows in set (0.06 sec)
```

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 concat(c.first_name,' ',c.last_name) as 'Name' , c.email as 'E-mail'
    -> from customer as c
    -> join address as a ON c.address_id = a.address_id
    -> join city as ct ON a.city_id = ct.city_id
    -> join country as cn ON cn.country_id = ct.country_id
    -> where cn.country = 'Canada' ;
+-------------------+-------------------------------------+
| Name              | E-mail                              |
+-------------------+-------------------------------------+
| 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 famiy films .

```
mysql> select title , category
    -> from film_list
    -> where category='Family' limit 20 ;
+----------------------+----------+
| title                | category |
+----------------------+----------+
| AFRICAN EGG          | Family   |
| APACHE DIVINE        | Family   |
| ATLANTIS CAUSE       | Family   |
| BAKED CLEOPATRA      | Family   |
| BANG KWAI            | Family   |
| BEDAZZLED MARRIED    | Family   |
| BILKO ANONYMOUS      | Family   |
| BLANKET BEVERLY      | Family   |
| BLOOD ARGONAUTS      | Family   |
| BLUES INSTINCT       | Family   |
| BRAVEHEART HUMAN     | Family   |
| CHASING FIGHT        | Family   |
| CHISUM BEHAVIOR      | Family   |
| CHOCOLAT HARRY       | Family   |
| CONFUSED CANDLES     | Family   |
| CONVERSATION DOWNHILL | Family   |
| DATE SPEED           | Family   |
| DINOSAUR SECRETARY   | Family   |
| DUMBO LUST           | Family   |
| EARRING INSTINCT     | Family   |
+----------------------+----------+
20 rows in set (0.01 sec)
```

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

```
mysql> DELIMITER //
mysql> create procedure FilmCount(IN category_name varchar(40),
    ->                            OUT Count INT )
    -> BEGIN
    ->   select COUNT(category) INTO Count
    ->   from film_list
    ->   where category = category_name;
    -> END //
Query OK, 0 rows affected (0.47 sec)

mysql> DELIMITER ;
mysql>
mysql> CALL FilmCount('Action',@count);
Query OK, 1 row affected (0.09 sec)

mysql> select @count;
+--------+
| @count |
+--------+
|     64 |
+--------+
1 row in set (0.00 sec)
```

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

```
mysql> select f.title as 'Movie' , count(r.rental_date)as 'Count of Rented'
    -> from film as f
    -> join inventory as i ON i.film_id = f.film_id
    -> join rental as r ON r.inventory_id = i.inventory_id
    -> group by f.title
    -> order by count(r.rental_date)desc Limit 10 ;
+--------------------+-----------------+
| Movie              | Count of Rented |
+--------------------+-----------------+
| BUCKET BROTHERHOOD |              34 |
| ROCKETEER MOTHER   |              33 |
| RIDGEMONT SUBMARINE|              32 |
| GRIT CLOCKWORK     |              32 |
| SCALAWAG DUCK      |              32 |
| JUGGLER HARDLY     |              32 |
| FORWARD TEMPLE     |              32 |
| HOBBIT ALIEN       |              31 |
| ROBBERS JOON       |              31 |
| ZORRO ARK          |              31 |
+--------------------+-----------------+
10 rows in set (0.09 sec)
```

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

```
mysql> select store_id, city, country
    -> from store s
    -> join address a ON a.address_id = s.address_id
    -> join city c ON c.city_id = a.city_id
    -> join country cn ON cn.country_id = c.country_id
    -> order by s.store_id ;
+----------+------------+-----------+
| store_id | city       | country   |
+----------+------------+-----------+
|        1 | Lethbridge | Canada    |
|        2 | Woodridge  | Australia |
+----------+------------+-----------+
2 rows in set (0.00 sec)
```

18. List the genres and its gross revenue.

```
mysql> select c.name as 'Film' , SUM(p.amount)as 'Gross Revenue' from category as c
    -> join film_category as fc ON fc.category_id = c.category_id
    -> join inventory as i ON i.film_id =  fc.film_id
    -> join rental as r ON r.inventory_id = i.inventory_id
    -> join payment as p ON p.rental_id = r.rental_id
    -> group by c.name
    -> order by SUM(p.amount) desc ;
+-------------+---------------+
| Film        | Gross Revenue |
+-------------+---------------+
| Sports      |       5314.21 |
| Sci-Fi      |       4756.98 |
| Animation   |       4656.30 |
| Drama       |       4587.39 |
| Comedy      |       4383.58 |
| Action      |       4375.85 |
| New         |       4351.62 |
| Games       |       4281.33 |
| Foreign     |       4270.67 |
| Family      |       4226.07 |
| Documentary |       4217.52 |
| Horror      |       3722.54 |
| Children    |       3655.55 |
| Classics    |       3639.59 |
| Travel      |       3549.64 |
| Music       |       3417.72 |
+-------------+---------------+
16 rows in set (0.24 sec)
```

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

```
mysql> create view film_revenue as
    -> select c.name as 'Film' , SUM(p.amount)as 'Gross Revenue' from category as c
    -> join film_category as fc ON fc.category_id = c.category_id
    -> join inventory as i ON i.film_id =  fc.film_id
    -> join rental as r ON r.inventory_id = i.inventory_id
    -> join payment as p ON p.rental_id = r.rental_id
    -> group by c.name
    -> order by SUM(p.amount) desc ;
Query OK, 0 rows affected (0.15 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_revenue               | VIEW       |
| film_text                  | BASE TABLE |
| 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 |
| staff_list                 | VIEW       |
| store                      | BASE TABLE |
+----------------------------+------------+
24 rows in set (0.00 sec)
```

20. Select top 5 genres in gross revenue view .

```
mysql> select * from film_revenue
    -> Limit 5 ;
+-----------+---------------+
| Film      | Gross Revenue |
+-----------+---------------+
| Sports    |       5314.21 |
| Sci-Fi    |       4756.98 |
| Animation |       4656.30 |
| Drama     |       4587.39 |
| Comedy    |       4383.58 |
+-----------+---------------+
5 rows in set (0.17 sec)
```