

ASSIGNMENT 1

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.

```
SELECT UPPER(CONCAT(first_name, ' ',last_name)) AS `Actor Name` FROM actor  
ORDER BY `Actor Name`;
```

```
| zero cage |  
+-----+  
200 rows in set (0.01 sec)  
  
mysql> SELECT UPPER(CONCAT(first_name,' ',last_name)) AS `Actor Name` FROM actor ORDER BY `Actor Name`;  
+-----+  
| 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 |  
| CAMERON ZELLWEGER |  
| CARMEN HUNT |  
| CARY MCCONAUGHEY |  
| CATE HARRIS |  
| CATE MCQUEEN |  
| CHARLIZE DENCH |  
| CHRIS BRIDGES |  
| CHRIS DEPP |  
| CHRISTIAN AKROYD |
```

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

```
SELECT CONCAT(first_name,' ',last_name) AS `Actor Name` FROM actor WHERE  
last_name LIKE '%GEN%' ORDER BY `Actor Name`;
```

```

Windows PowerShell x Windows PowerShell x + -
+-----+
| VIVIEN BASINGER |
| VIVIEN BERGEN |
| WALTER TORN |
| WARREN JACKMAN |
| WARREN NOLTE |
| WHOOP! HURT |
| WILL WILSON |
| WILLIAM HACKMAN |
| WOODY HOFFMAN |
| WOODY JOLIE |
| ZERO CAGE |
+-----+
200 rows in set (0.04 sec)

mysql> SELECT CONCAT(first_name, ' ', last_name) AS 'Actor Name' FROM actor WHERE last_name LIKE '%GEN%' ORDER BY 'Actor Name';
+-----+
| Actor Name |
+-----+
| GINA DEGENERES |
| JODIE DEGENERES |
| NICK DEGENERES |
| VIVIEN BERGEN |
+-----+
4 rows in set (0.01 sec)

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 (0.02 sec)

mysql> SELECT last_name,count(last_name) AS Count FROM actor GROUP BY last_name;
+-----+-----+
| last_name | Count |
+-----+-----+
| AKROYD | 3 |
+-----+-----+

```

32°C Haze 19:17 ENG IN 15-06-2023

0. 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');`

```

Windows PowerShell x Windows PowerShell x + -
+-----+
| VIVIEN BASINGER |
| VIVIEN BERGEN |
| WALTER TORN |
| WARREN JACKMAN |
| WARREN NOLTE |
| WHOOP! HURT |
| WILL WILSON |
| WILLIAM HACKMAN |
| WOODY HOFFMAN |
| WOODY JOLIE |
| ZERO CAGE |
+-----+
200 rows in set (0.04 sec)

mysql> SELECT CONCAT(first_name, ' ',last_name) AS 'Actor Name' FROM actor WHERE last_name LIKE '%GEN%' ORDER BY 'Actor Name';
+-----+
| Actor Name |
+-----+
| GINA DEGENERES |
| JODIE DEGENERES |
| NICK DEGENERES |
| VIVIEN BERGEN |
+-----+
4 rows in set (0.01 sec)

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 (0.02 sec)

mysql> SELECT last_name,count(last_name) AS Count FROM actor GROUP BY last_name;
+-----+-----+
| last_name | Count |
+-----+-----+
| AKROYD | 3 |
+-----+-----+

```

32°C Haze 19:17 ENG IN 15-06-2023

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

`SELECT last_name,count(last_name) AS Count FROM actor GROUP BY last_name;`

```

Windows PowerShell x Windows PowerShell x + -
3 rows in set (0.02 sec)

mysql> SELECT last_name, count(last_name) AS Count FROM actor GROUP BY last_name;
+-----+-----+
| last_name | Count |
+-----+-----+
| AKROYD   |    3 |
| ALLEN    |    3 |
| ASTAIRE   |    1 |
| BACALL   |    1 |
| BATLEY   |    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 |
| CRUTSE   |    1 |
| CRUZ     |    1 |
| DAMON    |    1 |
| DAVIS    |    3 |
| DAY-LEWIS |    1 |
| DEAN     |    2 |
+-----+-----+

```

0. 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

```
SELECT last_name, count(last_name) AS Count FROM actor GROUP BY last_name HAVING Count >= 2;
```

```

Windows PowerShell x Windows PowerShell x + -
121 rows in set (0.01 sec)

mysql> SELECT last_name, count(last_name) AS Count FROM actor GROUP BY last_name HAVING Count >= 2;
+-----+-----+
| last_name | Count |
+-----+-----+
| ZELLWEGER |    3 |
| AKROYD   |    3 |
| ALLEN    |    3 |
| BATLEY   |    2 |
| BENING   |    2 |
| 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 |
| GUINNESS |    3 |
| HACKMAN  |    2 |
| HARRIS   |    3 |
| HOFFMAN  |    3 |
| HOPKINS  |    3 |
| HOPPER   |    2 |
| JACKMAN  |    2 |
| JOHANSSON |    3 |
| KEITEL   |    3 |
| KILMER   |    5 |
| MCCONAUGHEY |    2 |
+-----+-----+

```

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

```
UPDATE actor SET first_name='HARPO' WHERE first_name='GROUCHO' AND last_name='WILLIAMS';
```

Windows PowerShell Windows PowerShell

```

| OLIVIER | 2 |
| PALTROW | 2 |
| PECK | 3 |
| PENN | 2 |
| SILVERSTONE | 2 |
| STREEP | 2 |
| TANDY | 2 |
| TEMPLE | 4 |
| TORN | 3 |
| TRACY | 2 |
| WAHLBERG | 2 |
| WEST | 2 |
| WILLIAMS | 3 |
| WILLIS | 3 |
| WINSLET | 2 |
| WOOD | 2 |
| ZELLWEGER | 3 |
+-----+
55 rows in set (0.00 sec)

mysql> UPDATE actor SET first_name='GROUCHO' WHERE first_name='HARPO' AND last_name='WILLIAMS';
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE actor SET first_name='HARPO' WHERE first_name='GROUCHO' AND last_name='WILLIAMS';
Query OK, 1 row affected (0.02 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)

mysql> SELECT first_name,last_name,address FROM staff JOIN address USING(address_id);
+-----+-----+-----+
| first_name | last_name | address        |
+-----+-----+-----+
| Mike      | Hillyer   | 23 Workhaven Lane |
| Jon       | Stephens  | 1411 Lillydale Drive |
+-----+-----+-----+
2 rows in set (0.03 sec)

32°C Haze
ENG IN 1918 15-06-2023

```

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

```
SELECT first_name,last_name,address FROM staff JOIN address USING(address_id);
```

Windows PowerShell Windows PowerShell

```

| HARPO | WILLIAMS |
+-----+
3 rows in set (0.00 sec)

mysql> SELECT first_name,last_name,address FROM staff JOIN address USING(address_id);
+-----+-----+-----+
| first_name | last_name | address        |
+-----+-----+-----+
| Mike      | Hillyer   | 23 Workhaven Lane |
| Jon       | Stephens  | 1411 Lillydale Drive |
+-----+-----+-----+
2 rows in set (0.03 sec)

mysql> SELECT title,COUNT(actor_id) count FROM film INNER JOIN film_actor USING(film_id) GROUP BY title;
+-----+-----+
| title          | count |
+-----+
| 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 |
| AMERICAN CIRCUS | 5 |
| AMISTAD MIDSUMMER | 4 |
| ANACONDA CONFESSIONS | 5 |
+-----+
10 rows in set (0.03 sec)

32°C Haze
ENG IN 1919 15-06-2023

```

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

```
SELECT title,COUNT(actor_id) count FROM film INNER JOIN film_actor USING(film_id)
GROUP BY title;
```

```

Windows PowerShell x Windows PowerShell x + v
| HARPO | WILLIAMS |
+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT first_name,last_name,address FROM staff JOIN address USING(address_id);
+-----+-----+-----+
| first_name | last_name | address      |
+-----+-----+-----+
| Mike      | Hillyer   | 23 Workhaven Lane |
| Jon       | Stephens  | 1411 Lillydale Drive |
+-----+-----+-----+
2 rows in set (0.03 sec)

mysql> SELECT title,COUNT(actor_id) count FROM film INNER JOIN film_actor USING(film_id) GROUP BY title;
+-----+-----+
| title          | count |
+-----+-----+
| 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 |
| ALT FOREVER |     5 |
| ALICE FANTASIA |     4 |
| ALIEN CENTER |     6 |
| ALLEY EVOLUTION |    5 |
| ALONE TRIP |     8 |
| ALTER VICTORY |     4 |
| AMADEUS HOLY |     6 |
| AMELIE HELLFIGHTERS |    6 |
| AMERICAN CIRCUS |    5 |
| AMISTAD MIDSUMMER |    4 |
| ANACONDA CONFESSIONS |    5 |
+-----+-----+

```

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

```
SELECT title,COUNT(i.film_id) Copy FROM film JOIN inventory i USING(film_id) WHERE
title='Hunchback Impossible';
```

```

Windows PowerShell x Windows PowerShell x + v
| WORLD LEATHERNECKS |     8 |
| WORST BANGER |     4 |
| WRATH MILE |     4 |
| WRONG BEHAVIOR |    9 |
| WYOMING STORM |     6 |
| YENTL IDAHO |     1 |
| YOUNG LANGUAGE |     5 |
| YOUTH KICK |     5 |
| ZHIVAGO CORE |     6 |
| ZOOLANDER FICTION |    5 |
| ZORRO ARK |     3 |
+-----+-----+
997 rows in set (0.03 sec)

mysql> SELECT title,COUNT(i.film_id) Copy FROM film JOIN inventory i USING(film_id) WHERE title='Hunchback Impossible';
+-----+-----+
| title          | Copy |
+-----+-----+
| HUNCHBACK IMPOSSIBLE |    6 |
+-----+-----+
1 row in set (0.01 sec)

mysql> SELECT first_name,last_name,SUM(amount) 'total paid' FROM payment JOIN customer USING(customer_id) GROUP BY customer_id ORDER BY last_name;
+-----+-----+-----+
| first_name | last_name | total paid |
+-----+-----+-----+
| RAFAEL    | ABNEY    |    97.79 |
| NATHANIEL | ADAM     |   133.72 |
| KATHLEEN  | ADAMS    |    92.73 |
| DIANA     | ALEXANDER |   185.73 |
| GORDON    | ALLARD   |   169.68 |
| 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 |
+-----+-----+-----+

```

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

```
SELECT first_name,last_name,SUM(amount) `total paid` FROM payment JOIN customer
USING(customer_id) GROUP BY customer_id ORDER BY last_name;
```

```
mysql> SELECT first_name,last_name,SUM(amount) `total paid` FROM payment JOIN customer
USING(customer_id) GROUP BY customer_id ORDER BY last_name;
+-----+-----+-----+
| first_name | last_name | total paid |
+-----+-----+-----+
| RAFAEL | ABNEY | 97.79 |
| NATHANIEL | ADAM | 133.72 |
| KATHLEEN | ADAMS | 92.73 |
| DIANA | ALEXANDER | 185.73 |
| 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 | 115.68 |
| VICTOR | BARKLEY | 91.76 |
| RACHEL | BARNES | 84.78 |
| CAROLE | BARNETT | 108.70 |
| TRACEY | BARRETT | 118.73 |
| DATSY | BATES | 162.62 |
| EDWARD | BAUGH | 114.72 |
| ROBERT | BAUGHMAN | 92.79 |
+-----+-----+-----+
```

0. 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 title FROM film WHERE title LIKE 'K%' OR title LIKE 'Q%' AND language_id
IN(SELECT language_id FROM language WHERE name='English');
```

```
mysql> SELECT title FROM film WHERE title LIKE 'K%' OR title LIKE 'Q%' AND language_id IN(SELECT language_id FROM language WHERE name='English');
+-----+
| title |
+-----+
| KANE EXORCIST |
| KARATE MOON |
| KENTUCKIAN GIANT |
| KICK SAVANNAH |
| WILL 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 (0.03 sec)

mysql> |
```

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

```
SELECT first_name,last_name FROM actor WHERE actor_id IN(SELECT actor_id FROM film_actor WHERE film_id IN(SELECT film_id FROM film WHERE title='Alone Trip'));
```

```
+-----+  
| 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 (0.03 sec)  
  
mysql> SELECT first_name,last_name FROM actor WHERE actor_id IN(SELECT actor_id FROM film_actor WHERE film_id IN(SELECT film_id FROM film WHERE title='Alone Trip'));  
+-----+-----+  
| first_name | last_name |  
+-----+-----+  
| ED         | CHASE    |  
| KARL       | BERRY    |  
| UMA        | WOOD     |  
| WOODY      | JOLIE    |  
| SPENCER    | DEPP     |  
| CHRIS      | DEPP     |  
| LAURENCE   | BULLOCK  |  
| RENEE      | BALL     |  
+-----+-----+  
8 rows in set (0.01 sec)  
  
mysql> |
```

0. 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 name,email FROM customer_list JOIN customer ON customer_id=ID WHERE country='Canada';
```

```
+-----+  
| 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 (0.03 sec)  
  
mysql> SELECT first_name,last_name FROM actor WHERE actor_id IN(SELECT actor_id FROM film_actor WHERE film_id IN(SELECT film_id FROM film WHERE title='Alone Trip'));  
+-----+-----+  
| first_name | last_name |  
+-----+-----+  
| ED         | CHASE    |  
| KARL       | BERRY    |  
| UMA        | WOOD     |  
| WOODY      | JOLIE    |  
| SPENCER    | DEPP     |  
| CHRIS      | DEPP     |  
| LAURENCE   | BULLOCK  |  
| RENEE      | BALL     |  
+-----+-----+  
8 rows in set (0.01 sec)  
  
mysql> SELECT name,email FROM customer_list JOIN customer ON customer_id=ID WHERE country='Canada';  
+-----+-----+  
| 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 (0.02 sec)  
  
mysql> |
```

0. 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.

```
SELECT title,name FROM film f JOIN film_category fc ON f.film_id=fc.film_id JOIN category c ON fc.category_id=c.category_id WHERE c.category_id IN(SELECT category_id FROM category WHERE name='Family');
```

```
| 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 (0.02 sec)

mysql> SELECT title,name FROM film f JOIN film_category fc ON f.film_id=fc.film_id JOIN category c ON fc.category_id=c.category_id WHERE c.category_id IN(SELECT category_id FROM category WHERE name='Family');
+-----+-----+
| title | name |
+-----+-----+
| 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 |
| EFFECT GLADIATOR | Family |
| FEUD FROGMEN | Family |
| FINDING ANACONDA | Family |
| GABLES METROPOLIS | Family |
| GANDHI KWAI | Family |
+-----+
```

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

DELIMITER !

```
CREATE PROCEDURE film_count (
    IN category_name VARCHAR(30),
    OUT count INT )
BEGIN
    SELECT COUNT(category_id) INTO count FROM category JOIN film_category
    USING(category_id) WHERE name=category_name;
END !
DELIMITER ;
CALL film_count('Family',@count);
SELECT @count;
```

```

Windows PowerShell x Windows PowerShell x Windows PowerShell x + v
--> ^C
mysql> use sakila;
Database changed
mysql> Delimiter !
mysql> CREATE PROCEDURE film_count (
    -> IN category_name VARCHAR(30),
    -> OUT count INT )
    -> BEGIN
    ->     SELECT COUNT(category_id) INTO count FROM category JOIN film_category USING(category_id) WHERE name=category_name;
    -> END !
Query OK, 0 rows affected (0.06 sec)

mysql> DELIMITER ;
mysql> CALL film_count('Family',@count);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT @count;
+-----+
| @count |
+-----+
|    69 |
+-----+
1 row in set (0.01 sec)

mysql> select * from category;
+-----+-----+-----+
| category_id | name      | last_update |
+-----+-----+-----+
|      1 | Action    | 2006-02-15 04:46:27 |
|      2 | Animation | 2006-02-15 04:46:27 |
|      3 | Children   | 2006-02-15 04:46:27 |
|      4 | Classics   | 2006-02-15 04:46:27 |
|      5 | Comedy    | 2006-02-15 04:46:27 |
|      6 | Documentary| 2006-02-15 04:46:27 |
|      7 | Drama     | 2006-02-15 04:46:27 |
|      8 | Family    | 2006-02-15 04:46:27 |
|      9 | Foreign   | 2006-02-15 04:46:27 |
|     10 | Games    | 2006-02-15 04:46:27 |
|     11 | Horror    | 2006-02-15 04:46:27 |
|     12 | Music    | 2006-02-15 04:46:27 |
+-----+-----+-----+

```

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

```

SELECT title,count(r.inventory_id) rented 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 title ORDER
BY rented DESC;

```

```

Windows PowerShell x Windows PowerShell x + v
| @count |
+-----+
|    69 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT title,count(r.inventory_id) rented 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
title ORDER BY rented DESC;
+-----+-----+
| title          | rented |
+-----+-----+
| 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 |
| MASSACRE USUAL | 30 |
| MUSCLE BRIGHT | 30 |
| PULP BEVERLY | 30 |
| RUGRATS SHAKESPEARE | 30 |
+-----+-----+

```

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

```
SELECT store_id, city, country FROM store s JOIN address a ON
s.address_id=a.address_id JOIN city c ON a.city_id=c.city_id JOIN country co ON
c.country_id=co.country_id;
```

```
| SIMON NORTH      | 6 |
| SLING LUKE      | 6 |
| TEQUILA PAST    | 6 |
| TERMINATOR CLUB | 6 |
| TEXAS WATCH     | 6 |
| WARLOCK WEREWOLF | 6 |
| WATERSHIP FRONTIER | 6 |
| WILD APOLLO     | 6 |
| YOUTH KICK      | 6 |
| BRAVEHEART HUMAN | 5 |
| BUNCH MINDS     | 5 |
| CONSPIRACY SPIRIT | 5 |
| FEVER EMPIRE    | 5 |
| FREEDOM CLEOPATRA | 5 |
| FULL FLATLINERS | 5 |
| GLORY TRACY     | 5 |
| HUNTER ALTER     | 5 |
| INFORMER DOUBLE | 5 |
| MANNEQUIN WORST  | 5 |
| MUSSOLINI SPOILERS | 5 |
| PRIVATE DROP     | 5 |
| SEVEN SWARM      | 5 |
| TRAFFIC HOBBIT   | 5 |
| HARDLY ROBBERS   | 4 |
| MIXED DOORS      | 4 |
| TRAIN BUNCH      | 4 |
+-----+
958 rows in set (0.05 sec)

mysql> SELECT store_id, city, country FROM store s JOIN address a ON s.address_id=a.address_id JOIN city c ON a.city_id=c.city_id JOIN country co ON c.count
ry_id=co.country_id;
+-----+-----+-----+
| store_id | city      | country   |
+-----+-----+-----+
|      1  | Lethbridge | Canada   |
|      2  | Woodridge  | Australia |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> |
```

0. List the genres and its gross revenue.

```
SELECT name genre,sum(amount) `gross revenue` FROM category JOIN film_category
USING(category_id) JOIN inventory USING(film_id) JOIN rental USING(inventory_id)
JOIN payment USING(rental_id) GROUP BY name;
```

```
| MIXED DOORS      | 4 |
| TRAIN BUNCH      | 4 |
+-----+
958 rows in set (0.05 sec)

mysql> SELECT store_id, city, country FROM store s JOIN address a ON s.address_id=a.address_id JOIN city c ON a.city_id=c.city_id JOIN country co ON c.count
ry_id=co.country_id;
+-----+-----+-----+
| store_id | city      | country   |
+-----+-----+-----+
|      1  | Lethbridge | Canada   |
|      2  | Woodridge  | Australia |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> SELECT name genre,sum(amount) `gross revenue` FROM category JOIN film_category USING(category_id) JOIN inventory USING(film_id) JOIN rental USING(inv
entory_id) JOIN payment USING(rental_id) GROUP BY name;
+-----+-----+
| 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      |
| New         | 4351.62      |
| Sci-Fi      | 4756.98      |
| Sports      | 5314.21      |
| Travel      | 3549.64      |
+-----+-----+
16 rows in set (0.10 sec)

mysql> |
```

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

```
CREATE VIEW gross_revenue_by_genre AS SELECT name genre,sum(amount) `gross revenue` FROM category JOIN film_category USING(category_id) JOIN inventory USING(film_id) JOIN rental USING(inventory_id) JOIN payment USING(rental_id) GROUP BY name;
```

```

Windows PowerShell x Windows PowerShell x + -
Documentary | 4217.52 |
Drama | 4587.39 |
Family | 4226.07 |
Foreign | 4278.67 |
Games | 4281.33 |
Horror | 3722.54 |
Music | 3417.72 |
New | 4351.62 |
Sci-Fi | 4756.98 |
Sports | 5314.21 |
Travel | 3549.64 |
+-----+
16 rows in set (0.10 sec)

mysql> CREATE VIEW gross_revenue_by_genre AS SELECT name genre,sum(amount) `gross revenue` FROM category JOIN film_category USING(category_id) JOIN inventory USING(film_id) JOIN rental USING(inventory_id) JOIN payment USING(rental_id) GROUP BY name;
ERROR 1050 (42S01): Table 'gross_revenue_by_genre' already exists
mysql> DROP VIEW gross_revenue_by_genre;
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE VIEW gross_revenue_by_genre AS SELECT name genre,sum(amount) `gross revenue` FROM category JOIN film_category USING(category_id) JOIN inventory USING(film_id) JOIN rental USING(inventory_id) JOIN payment USING(rental_id) GROUP BY name;
Query OK, 0 rows affected (0.02 sec)

mysql> SHOW FULL TABLES WHERE Table_type='VIEW';
+-----+
| Tables_in_sakila | Table_type |
+-----+
| actor_info | VIEW |
| customer_list | VIEW |
| film_list | VIEW |
| gross_revenue_by_genre | VIEW |
| nicer_but_slower_film_list | VIEW |
| sales_by_film_category | VIEW |
| sales_by_store | VIEW |
| staff_list | VIEW |
+-----+
8 rows in set (0.03 sec)

mysql>

```

0. Select top 5 genres in gross revenue view.

```
SELECT * FROM gross_revenue_by_genre ORDER BY `gross revenue` DESC LIMIT 5;
```

```

Windows PowerShell x Windows PowerShell x + -
16 rows in set (0.10 sec)

mysql> CREATE VIEW gross_revenue_by_genre AS SELECT name genre,sum(amount) `gross revenue` FROM category JOIN film_category USING(category_id) JOIN inventory USING(film_id) JOIN rental USING(inventory_id) JOIN payment USING(rental_id) GROUP BY name;
ERROR 1050 (42S01): Table 'gross_revenue_by_genre' already exists
mysql> DROP VIEW gross_revenue_by_genre;
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE VIEW gross_revenue_by_genre AS SELECT name genre,sum(amount) `gross revenue` FROM category JOIN film_category USING(category_id) JOIN inventory USING(film_id) JOIN rental USING(inventory_id) JOIN payment USING(rental_id) GROUP BY name;
Query OK, 0 rows affected (0.02 sec)

mysql> SHOW FULL TABLES WHERE Table_type='VIEW';
+-----+
| Tables_in_sakila | Table_type |
+-----+
| actor_info | VIEW |
| customer_list | VIEW |
| film_list | VIEW |
| gross_revenue_by_genre | VIEW |
| nicer_but_slower_film_list | VIEW |
| sales_by_film_category | VIEW |
| sales_by_store | VIEW |
| staff_list | VIEW |
+-----+
8 rows in set (0.03 sec)

mysql> SELECT * FROM gross_revenue_by_genre ORDER BY `gross revenue` DESC LIMIT 5;
+-----+
| genre | gross revenue |
+-----+
| Sports | 5314.21 |
| Sci-Fi | 4756.98 |
| Animation | 4656.38 |
| Drama | 4587.39 |
| Comedy | 4383.58 |
+-----+
5 rows in set (0.12 sec)

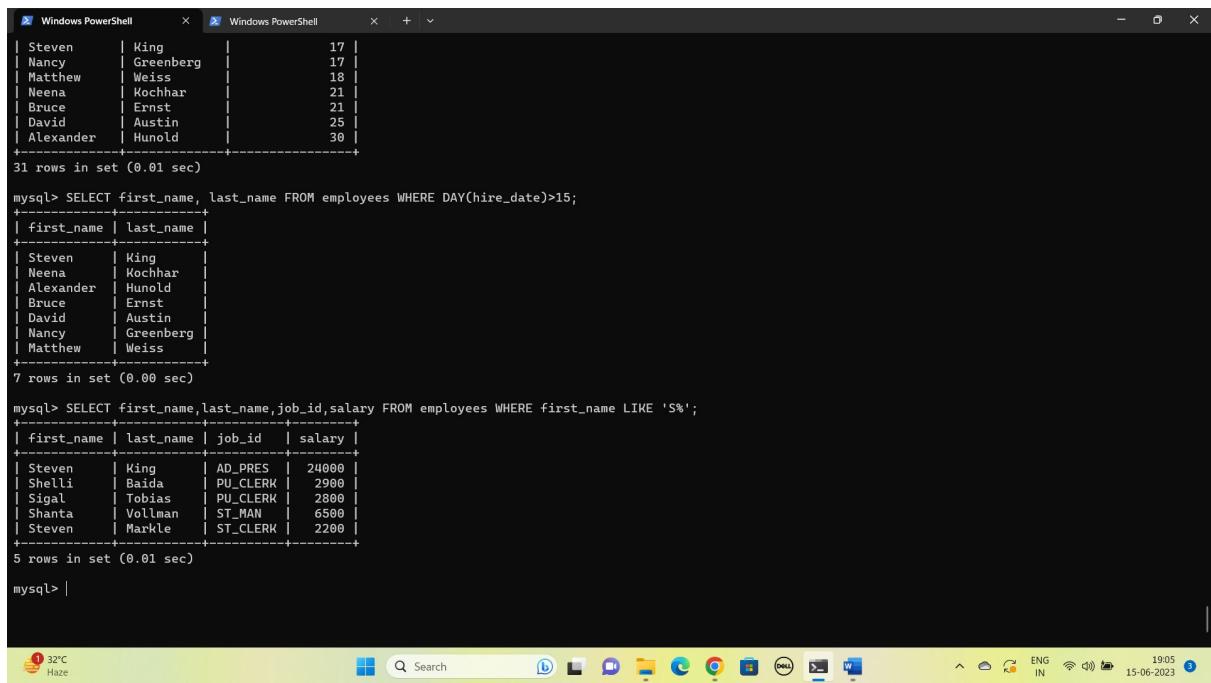
mysql>

```

ASSIGNMENT 2

1. Select employees first name, last name, job_id and salary whose first name starts with alphabet S

```
SELECT first_name,last_name,job_id,salary FROM employees WHERE first_name LIKE 'S%';
```



The screenshot shows a Windows desktop environment. At the top, there are two PowerShell windows side-by-side. The left window displays a table of employee names and hire dates. The right window shows a query result for employees hired after day 15. Below these, a MySQL command-line interface window is open, displaying three separate SELECT statements. The first statement lists employees with first names starting with 'S'. The second statement lists employees hired after day 15. The third statement lists employees with first names starting with 'S' along with their job IDs and salaries. The MySQL prompt 'mysql>' is visible at the bottom of this window. At the very bottom of the screen, a taskbar is visible with various icons and system status indicators.

```
| Steven | King | 17 |
| Nancy | Greenberg | 17 |
| Matthew | Weiss | 18 |
| Neena | Kochhar | 21 |
| Bruce | Ernst | 21 |
| David | Austin | 25 |
| Alexander | Hunold | 30 |
+-----+-----+-----+
31 rows in set (0.01 sec)

mysql> SELECT first_name, last_name FROM employees WHERE DAY(hire_date)>15;
+-----+-----+
| first_name | last_name |
+-----+-----+
| Steven | King |
| Neena | Kochhar |
| Alexander | Hunold |
| Bruce | Ernst |
| David | Austin |
| Nancy | Greenberg |
| Matthew | Weiss |
+-----+-----+
7 rows in set (0.00 sec)

mysql> SELECT first_name, last_name, job_id, salary FROM employees WHERE first_name LIKE 'S%';
+-----+-----+-----+-----+
| first_name | last_name | job_id | salary |
+-----+-----+-----+-----+
| Steven | King | AD_PRES | 24000 |
| Shelli | Baida | PU_CLERK | 2900 |
| Sigal | Tobias | PU_CLERK | 2800 |
| Shanta | Vollman | ST_MAN | 6500 |
| Steven | Markle | ST_CLERK | 2200 |
+-----+-----+-----+-----+
5 rows in set (0.01 sec)

mysql> |
```

2. Write a query to select employee with the highest salary (using inner query)

```
SELECT first_name,last_name,salary FROM employees WHERE salary IN(SELECT MAX(salary) FROM employees);
```

```

Windows PowerShell x Windows PowerShell x + v

mysql> SELECT first_name, last_name FROM employees WHERE DAY(hire_date)>15;
+-----+-----+
| first_name | last_name |
+-----+-----+
| Steven     | King      |
| Neena      | Kochhar   |
| Alexander  | Hunold    |
| Bruce      | Ernst     |
| David      | Austin    |
| Nancy      | Greenberg|
| Matthew    | Weiss    |
+-----+-----+
7 rows in set (0.00 sec)

mysql> SELECT first_name, last_name, job_id, salary FROM employees WHERE first_name LIKE 'S%';
+-----+-----+-----+-----+
| first_name | last_name | job_id | salary |
+-----+-----+-----+-----+
| Steven     | King      | AD_PRES | 24000 |
| Shelli     | Baida    | PU_CLERK | 2900  |
| Sigal      | Tobias   | PU_CLERK | 2800  |
| Shanta    | Vollman  | ST_MAN  | 6500  |
| Steven     | Markle   | ST_CLERK | 2200  |
+-----+-----+-----+-----+
5 rows in set (0.01 sec)

mysql> SELECT first_name, last_name, salary FROM employees WHERE salary IN(SELECT MAX(salary) FROM employees);
+-----+-----+-----+
| first_name | last_name | salary |
+-----+-----+-----+
| Steven     | King      | 24000 |
+-----+-----+-----+
1 row in set (0.01 sec)

mysql>

```

32°C Haze

Search ENG IN 15-06-2023 19:08

3. Select employee with the second highest salary

```
SELECT first_name, last_name, salary FROM employees WHERE salary IN(SELECT MAX(salary) FROM employees WHERE salary < (SELECT MAX(salary) FROM employees));
```

```

Windows PowerShell x Windows PowerShell x + v

| David     | Austin    | 4800 |
| Valli    | Pataballa | 4800 |
| Diana    | Lorentz   | 4200 |
| Nancy    | Greenberg | 12000 |
| Daniel   | Faviet    | 9000 |
| John     | Chen      | 8200 |
| Ismael   | Sciarra   | 7700 |
| Jose Manuel | Uzman    | 7800 |
| Luis     | Popp      | 6900 |
| Den      | Raphaely  | 11000 |
| Alexander | Khoo      | 3100 |
| Shelli   | Baida    | 2900 |
| Sigal    | Tobias   | 2800 |
| Guy      | Himuro   | 2600 |
| Karen    | Colmenares | 2500 |
| Matthew   | Weiss    | 8000 |
| Adam     | Fipp      | 8200 |
| Payam    | Kaufling  | 7900 |
| Shanta   | Vollman  | 6500 |
| Kevin    | Mourgos  | 5800 |
| Julia    | Nayer    | 3200 |
| Irene    | Mikkilineni | 2700 |
| James    | Landry   | 2400 |
| Steven   | Markle   | 2200 |
| Laura    | Bissot   | 3300 |
| Mozhe    | Atkinson  | 2800 |
+-----+-----+-----+
31 rows in set (0.00 sec)

mysql> SELECT first_name, last_name, salary FROM employees WHERE salary IN(SELECT MAX(salary) FROM employees WHERE salary < (SELECT MAX(salary) FROM employees));
+-----+-----+-----+
| first_name | last_name | salary |
+-----+-----+-----+
| Neena     | Kochhar   | 17000 |
| Lex       | De Haan   | 17000 |
+-----+-----+-----+
2 rows in set (0.01 sec)

mysql>

```

Top Events Event Brief

Search ENG IN 22:21 15-06-2023

4. Write a query to select employees and their corresponding managers and their salaries (SELF Join)

```
SELECT CONCAT(a.first_name, ',', a.last_name) employee,
CONCAT(b.first_name, ',', b.last_name) manager, a.salary FROM employees
a LEFT JOIN employees b ON b.Employee_id=a.manager_id;
```

```
mysql> SELECT CONCAT(a.first_name, ',', a.last_name) employee, CONCAT(b.first_name, ',', b.last_name) manager, a.salary FROM employees a LEFT JOIN employees b ON b.Employee_id=a.manager_id;
+-----+-----+-----+
| employee | manager | salary |
+-----+-----+-----+
| Steven King | NULL | 24000 |
| Neena Kochhar | Steven King | 17000 |
| Lex De Haan | Steven King | 17000 |
| Alexander Hunold | Lex De Haan | 9000 |
| Bruce Ernst | Alexander Hunold | 6000 |
| David Austin | Alexander Hunold | 4800 |
| Valli Pataballa | Alexander Hunold | 4800 |
| Diana Lorentz | Alexander Hunold | 4200 |
| Nancy Greenberg | Neena Kochhar | 12000 |
| Daniel Faviet | Nancy Greenberg | 9000 |
| John Chen | Nancy Greenberg | 8200 |
| Ismael Sciarra | Nancy Greenberg | 7700 |
| Jose Manuel Urman | Nancy Greenberg | 7800 |
| Luis Popp | Nancy Greenberg | 6900 |
| Den Raphaely | Steven King | 11000 |
| Alexander Khoo | Den Raphaely | 3100 |
| Shelli Baida | Den Raphaely | 2900 |
| Sigal Tobias | Den Raphaely | 2800 |
| Guy Himuro | Den Raphaely | 2600 |
| Karen Colmenares | Den Raphaely | 2500 |
| Matthew Weiss | Steven King | 8000 |
| Adam Fripp | Steven King | 8200 |
| Payam Kaufling | Steven King | 7900 |
| Shanta Vollman | Steven King | 6500 |
| Kevin Murgos | Steven King | 5800 |
| Julia Nayer | Matthew Weiss | 3200 |
| Irene Mikkilineni | Matthew Weiss | 2700 |
| James Landry | Matthew Weiss | 2400 |
| Steven Markle | Matthew Weiss | 2200 |
| Laura Bissot | Adam Fripp | 3300 |
| Mozhe Atkinson | Adam Fripp | 2800 |
+-----+-----+-----+
31 rows in set (0.00 sec)
```

5. Create a view for the above query

```
CREATE VIEW manager_salary_of_employee AS SELECT
CONCAT(a.first_name, ',', a.last_name) employee, CONCAT(b.first_name,
',', b.last_name) manager, a.salary FROM employees a LEFT JOIN
employees b ON b.Employee_id=a.manager_id;
```

```
mysql> CREATE VIEW manager_salary_of_employee AS SELECT CONCAT(a.first_name, ',', a.last_name) employee, CONCAT(b.first_name, ',', b.last_name) manager, a.salary FROM employees a LEFT JOIN employees b ON b.Employee_id=a.manager_id;C
mysql> DROP VIEW `C
mysql> CREATE VIEW manager_salary_of_employee AS SELECT CONCAT(a.first_name, ',', a.last_name) employee, CONCAT(b.first_name, ',', b.last_name) manager, a.salary FROM employees a LEFT JOIN employees b ON b.Employee_id=a.manager_id;C
mysql> DROP VIEW manager_salary_of_employee;
Query OK, 0 rows affected (0.11 sec)

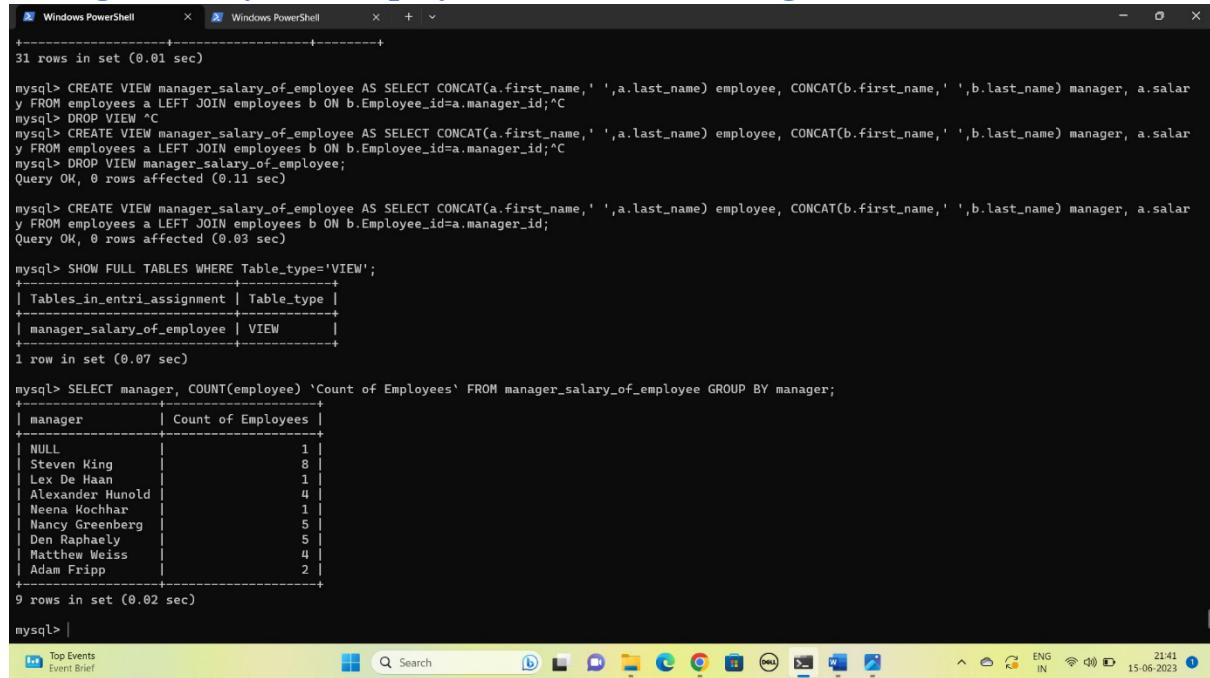
mysql> CREATE VIEW manager_salary_of_employee AS SELECT CONCAT(a.first_name, ',', a.last_name) employee, CONCAT(b.first_name, ',', b.last_name) manager, a.salary FROM employees a LEFT JOIN employees b ON b.Employee_id=a.manager_id;
Query OK, 0 rows affected (0.03 sec)

mysql> SHOW FULL TABLES WHERE Table_type='VIEW';
+-----+-----+
| Tables_in_employees | Table_type |
+-----+-----+
| manager_salary_of_employee | VIEW      |
+-----+-----+
1 row in set (0.07 sec)

mysql> |
```

6. Write a query to show count of employees under each manager in descending order (from view)

SELECT manager, COUNT(employee) `Count of Employees` FROM manager_salary_of_employee GROUP BY manager;



```
mysql> CREATE VIEW manager_salary_of_employee AS SELECT CONCAT(a.first_name, ' ', a.last_name) employee, CONCAT(b.first_name, ' ', b.last_name) manager, a.salary FROM employees a LEFT JOIN employees b ON b.Employee_id=a.manager_id;"C
mysql> DROP VIEW 'C
mysql> CREATE VIEW manager_salary_of_employee AS SELECT CONCAT(a.first_name, ' ', a.last_name) employee, CONCAT(b.first_name, ' ', b.last_name) manager, a.salary FROM employees a LEFT JOIN employees b ON b.Employee_id=a.manager_id;"C
mysql> DROP VIEW manager_salary_of_employee;
Query OK, 0 rows affected (0.11 sec)

mysql> CREATE VIEW manager_salary_of_employee AS SELECT CONCAT(a.first_name, ' ', a.last_name) employee, CONCAT(b.first_name, ' ', b.last_name) manager, a.salary FROM employees a LEFT JOIN employees b ON b.Employee_id=a.manager_id;
Query OK, 0 rows affected (0.03 sec)

mysql> SHOW FULL TABLES WHERE Table_type='VIEW';
+-----+-----+
| Tables_in_entri_assignment | Table_type |
+-----+-----+
| manager_salary_of_employee | VIEW      |
+-----+-----+
1 row in set (0.07 sec)

mysql> SELECT manager, COUNT(employee) `Count of Employees` FROM manager_salary_of_employee GROUP BY manager;
+-----+-----+
| manager | Count of Employees |
+-----+-----+
| NULL    | 1                |
| Steven King | 8               |
| Lex De Haan | 1               |
| Alexander Hunold | 4          |
| Neena Kochhar | 1               |
| Nancy Greenberg | 5           |
| Den Raphaely | 5               |
| Matthew Weiss | 4               |
| Adam Fripp | 2               |
+-----+-----+
9 rows in set (0.02 sec)

mysql> |
```

7. Find the count of employees in each department

SELECT Department_name AS Department, COUNT(Employee_id) AS `count of employees` FROM departments JOIN employees USING(department_id) GROUP BY Department;

```

Windows PowerShell x Windows PowerShell x Windows PowerShell x
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| Employee_id | smallint unsigned | NO   | PRI | NULL    |       |
| first_name  | varchar(15)     | NO   |      | NULL    |       |
| last_name   | varchar(15)     | NO   |      | NULL    |       |
| email       | varchar(40)     | YES  |      | NULL    |       |
| phone_number| varchar(20)     | NO   |      | NULL    |       |
| hire_date   | date           | NO   |      | NULL    |       |
| job_id      | varchar(30)    | NO   |      | NULL    |       |
| salary      | smallint unsigned | NO   |      | NULL    |       |
| commission_pct | varchar(10)   | YES  |      | NULL    |       |
| manager_id  | smallint unsigned | YES  |      | NULL    |       |
| department_id | smallint unsigned | NO   | MUL | NULL    |       |
+-----+
11 rows in set (0.00 sec)

mysql> SELECT Department_name AS Department, COUNT(Employee_id) AS `count of employees` FROM departments JOIN employees USING(department_id) GROUP BY Department;
+-----+
| Department | count of employees |
+-----+
| Marketing  | 2                 |
| Purchasing | 3                 |
| Human Resources | 3             |
| Shipping   | 7                 |
| IT         | 4                 |
| Public Relations | 1             |
| Sales      | 2                 |
| Executive  | 1                 |
| Finance    | 1                 |
| Accounting | 1                 |
| Corporate Tax | 1             |
| Control And Credit | 1             |
| Shareholder Services | 1             |
| Benefits   | 1                 |
| Payroll    | 2                 |
+-----+
15 rows in set (0.01 sec)

mysql> |
28°C Mostly cloudy

```

8. Get the count of employees hired year wise

```

SELECT COUNT(Employee_id) `count of employees`, YEAR(hire_date) year FROM
employees GROUP BY year;

```

```

Windows PowerShell x Windows PowerShell x Windows PowerShell x
+-----+
| 114 | Den        | Raphaelly | 1994-11-08 |
| 115 | Alexander | Khoo      | 1995-05-12 |
| 116 | Shelli    | Baida    | 1997-12-13 |
| 117 | Sigal     | Tobias   | 1997-09-10 |
| 118 | Guy        | Hinuro   | 1998-01-02 |
| 119 | Karen     | Colmenares | 1999-04-08 |
| 120 | Matthew   | Weiss    | 1996-07-18 |
| 121 | Adam      | Fripp    | 1997-08-09 |
| 122 | Payam     | Kaufling | 1995-05-01 |
| 123 | Shanta   | Vollman  | 1997-10-12 |
| 124 | Kevin     | Mourgos  | 1999-11-12 |
| 125 | Julia     | Nayer    | 1997-07-02 |
| 126 | Irene     | Milkilineni | 1998-11-12 |
| 127 | James     | Landry   | 1999-01-02 |
| 128 | Steven   | Markle   | 2000-03-04 |
| 129 | Laura    | Bissot   | 1997-09-10 |
| 130 | Mozhe    | Atkinson | 1997-10-12 |
+-----+
31 rows in set (0.00 sec)

mysql> SELECT COUNT(Employee_id) `count of employees`, YEAR(hire_date) year FROM employees GROUP BY year;
+-----+
| count of employees | year |
+-----+
| 1 | 1987 |
| 1 | 1989 |
| 1 | 1993 |
| 1 | 1998 |
| 1 | 1991 |
| 10 | 1997 |
| 4 | 1998 |
| 5 | 1999 |
| 3 | 1994 |
| 2 | 1995 |
| 1 | 1996 |
| 1 | 2000 |
+-----+
12 rows in set (0.01 sec)

mysql> |
28°C Mostly cloudy

```

9 . create a stored procedure to get the “ Get the count of employees hired in the input year”(IN year , OUT count)

DELIMITER !

```

CREATE PROCEDURE employees_count_by_year (
    IN year INT,
    OUT count INT)
BEGIN
    SELECT COUNT(employee_id) INTO count FROM employees WHERE
YEAR(hire_date)=year;
END !
DELIMITER ;
CALL employees_count_by_year('1997',@count);

```

```

mysql> CREATE PROCEDURE employees_count_by_year (
-> IN year INT,
-> OUT count INT)
-> BEGIN
->     SELECT COUNT(employee_id) INTO count FROM employees WHERE YEAR(hire_date)=year;
-> END !
Query OK, 0 rows affected (0.03 sec)

mysql> DELIMITER ;
mysql> CALL employees_count_by_year('1997',@count);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT @count;
+-----+
| @count |
+-----+
|      10 |
+-----+
1 row in set (0.00 sec)

mysql> CALL employees_count_by_year('2000',@count);
Query OK, 1 row affected (0.01 sec)

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

mysql> CALL employees_count_by_year('1999',@count);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @count;
+-----+
| @count |
+-----+
|      5 |
+-----+
1 row in set (0.00 sec)

```

10. Select the employees whose first_name contains “an”

```
SELECT first_name, last_name FROM employees WHERE first_name LIKE '%an%';
```

```

mysql> CALL employees_count_by_year('2000',@count);
Query OK, 1 row affected (0.01 sec)

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

mysql> CALL employees_count_by_year('1999',@count);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @count;
+-----+
| @count |
+-----+
|      5 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT first_name,last_name FROM employees WHERE first_name LIKE '%an%';
+-----+-----+
| first_name | last_name |
+-----+-----+
| Alexander | Hunold   |
| Diana     | Lorentz  |
| Nancy     | Greenberg|
| Daniel    | Faviet   |
| Jose Manuel | Urman   |
| Alexander | Kho   |
| Shanta    | Vollman |
+-----+-----+
7 rows in set (0.01 sec)

mysql>

```

28°C Mostly cloudy Search 01:08 14-06-2023

11. Select employee first name and the corresponding phone number in the format (___)-(___)-(___)

`SELECT GROUP_CONCAT(first_name,'-',phone_number SEPARATOR '-') FROM employees;`

```

mysql> SELECT GROUP_CONCAT(first_name,'-',phone_number SEPARATOR '-') FROM employees;
+-----+
| GROUP_CONCAT(first_name,'-',phone_number SEPARATOR '-') |
+-----+
| Steven-515.123.4567-Neena-515.123.4568-Lex-515.123.4569-Alexander-590.423.4567-Bruce-590.423.4568-David-590.423.4569-Valli-590.423.4560-Diana-590.423.5567-Nancy-515.124.4569-Daniel-515.124.4169-John-515.124.4269-Ismail-515.124.4369-Jose Manuel-515.124.4469-Luis-515.124.4567-Den-515.127.4561-Alexander-515.127.4562-Shell-515.127.4563-Sigal-515.127.4564-Guy-515.127.4565-Karen-515.127.4566-Matthew-650.123.1234-Adam-650.123.2234-Payam-650.123.3234-Shanta-650.123.4234-Kevin-650.123.5234-Julia-650.124.1214-Irene-650.124.1224-James-650.124.1334-Steven-650.124.1434-Laura-650.124.5234-Mozhe-650.124.6234 |
+-----+
1 row in set

mysql>

```

28°C Mostly cloudy Search 01:52 14-06-2023

12. Find the employees who joined in August, 1994.

`SELECT first_name, last_name FROM employees WHERE hire_date LIKE '1994-08-%';`

```

Windows PowerShell x Windows PowerShell x Windows PowerShell x + -
102 | Lex | De Haan | LDEHAAN | 515.123.4569 | 1993-09-12 | AD_VP | 17000 | NULL | 100 | 30 |
109 | Daniel | Faviet | DFAVIET | 515.124.4169 | 1994-08-12 | FI_ACCOUNT | 9000 | NULL | 108 | 170 |
108 | Nancy | Greenberg | NGREENBE | 515.124.4569 | 1994-08-17 | FI_MGR | 12000 | NULL | 101 | 100 |
114 | Den | Raphaely | DRAPHEAL | 515.127.4561 | 1994-11-08 | PU_MAN | 11000 | NULL | 100 | 30 |
122 | Payam | Kaufling | PKAUFLIN | 650.123.3234 | 1995-05-01 | ST_MAN | 7900 | NULL | 108 | 40 |
115 | Alexander | Kho | AKHOO | 515.127.4562 | 1995-05-12 | PU_CLERK | 3100 | NULL | 114 | 80 |
120 | Matthew | Weiss | MWIESS | 650.123.1234 | 1996-07-18 | ST_MAN | 8000 | NULL | 100 | 50 |
111 | Ismael | Sciarra | ISCIARRA | 515.124.4369 | 1997-02-01 | FI_ACCOUNT | 7700 | NULL | 108 | 160 |
110 | John | Chen | JCHEN | 515.124.4269 | 1997-04-09 | FI_ACCOUNT | 8200 | NULL | 108 | 170 |
105 | David | Austin | DAUSTIN | 590.423.4569 | 1997-06-25 | IT_PROG | 4800 | NULL | 103 | 60 |
125 | Julia | Nayer | JNAYER | 650.124.1214 | 1997-07-02 | ST_CLERK | 3200 | NULL | 120 | 50 |
121 | Adam | Fripp | AFRIPP | 650.123.2234 | 1997-08-09 | ST_MAN | 8200 | NULL | 100 | 50 |
117 | Sigal | Tobias | STOBIAS | 515.127.4564 | 1997-09-10 | PU_CLERK | 2800 | NULL | 114 | 30 |
129 | Laura | Bissot | LBISSTO | 650.124.5234 | 1997-09-10 | ST_CLERK | 3300 | NULL | 121 | 50 |
123 | Shanta | Vollman | SVOLLMAN | 650.123.4234 | 1997-10-12 | ST_MAN | 6500 | NULL | 100 | 50 |
130 | Mozhe | Atkinson | MATKINSO | 650.124.6234 | 1997-10-12 | ST_CLERK | 2800 | NULL | 121 | 110 |
116 | Shelli | Baida | SBaida | 515.127.4563 | 1997-12-13 | PU_CLERK | 2900 | NULL | 114 | 70 |
118 | Guy | Himuro | GHIMURO | 515.127.4565 | 1998-01-02 | PU_CLERK | 2600 | NULL | 114 | 60 |
106 | Valli | Pataballa | VPATABAL | 590.423.4560 | 1998-02-05 | IT_PROG | 4800 | NULL | 103 | 40 |
112 | Jose Manuel | Urman | JMURMAN | 515.124.4469 | 1998-06-03 | FI_ACCOUNT | 7800 | NULL | 108 | 150 |
126 | Irene | Mikkilineni | IMIKKILI | 650.124.1224 | 1998-11-12 | ST_CLERK | 2700 | NULL | 120 | 50 |
127 | James | Landry | JLANDRY | 650.124.1334 | 1999-01-02 | ST_CLERK | 2400 | NULL | 120 | 90 |
107 | Diana | Lorentz | DLORENTZ | 590.423.5567 | 1999-02-09 | IT_PROG | 4200 | NULL | 103 | 40 |
119 | Karen | Colmenares | KCOLMENA | 515.127.4566 | 1999-04-08 | PU_CLERK | 2500 | NULL | 114 | 130 |
124 | Kevin | Mourgos | KMOURGOS | 650.123.5234 | 1999-11-12 | ST_MAN | 5800 | NULL | 100 | 80 |
113 | Luis | Popp | LPOPP | 515.124.4567 | 1999-12-07 | FI_ACCOUNT | 6900 | NULL | 108 | 140 |
128 | Steven | Markle | SMARLKE | 650.124.1434 | 2000-03-04 | ST_CLERK | 2200 | NULL | 120 | 50 |
31 rows in set (0.00 sec)

mysql> SELECT first_name, last_name FROM employees WHERE hire_date LIKE '1994-08-%';
+-----+-----+
| first_name | last_name |
+-----+-----+
| Nancy | Greenberg |
| Daniel | Faviet |
+-----+-----+
2 rows in set (0.00 sec)

mysql>

```

13. Find the maximum salary from each department.

```

SELECT Department_name Department ,MAX(salary) `Maximum Salary` FROM
departments d JOIN employees e ON d.Department_id=e.department_id GROUP BY
Department;

```

```

Windows PowerShell x Windows PowerShell x Windows PowerShell x + -
118 | Guy | Himuro | GHIMURO | 515.127.4565 | 1998-01-02 | PU_CLERK | 2600 | NULL | 114 | 60 |
119 | Karen | Colmenares | KCOLMENA | 515.127.4566 | 1999-04-08 | PU_CLERK | 2500 | NULL | 114 | 130 |
120 | Matthew | Weiss | MWIESS | 650.123.1234 | 1996-07-18 | ST_MAN | 8000 | NULL | 100 | 50 |
121 | Adam | Fripp | AFRIPP | 650.123.2234 | 1997-08-09 | ST_MAN | 8200 | NULL | 100 | 50 |
122 | Payam | Kaufling | PKAUFLIN | 650.123.3234 | 1995-05-01 | ST_MAN | 7900 | NULL | 100 | 40 |
123 | Shanta | Vollman | SVOLLMAN | 650.123.4234 | 1997-10-12 | ST_MAN | 6500 | NULL | 100 | 50 |
124 | Kevin | Mourgos | KMOURGOS | 650.123.5234 | 1999-11-12 | ST_MAN | 5800 | NULL | 100 | 80 |
125 | Julia | Nayer | JNAYER | 650.124.1214 | 1997-07-02 | ST_CLERK | 3200 | NULL | 120 | 50 |
126 | Irene | Mikkilineni | IMIKKILI | 650.124.1224 | 1998-11-12 | ST_CLERK | 2700 | NULL | 120 | 50 |
127 | James | Landry | JLANDRY | 650.124.1334 | 1999-01-02 | ST_CLERK | 2400 | NULL | 120 | 90 |
128 | Steven | Markle | SMARLKE | 650.124.1434 | 2000-03-04 | ST_CLERK | 2200 | NULL | 120 | 50 |
129 | Laura | Bissot | LBISSTO | 650.124.5234 | 1997-09-10 | ST_CLERK | 3300 | NULL | 121 | 50 |
130 | Mozhe | Atkinson | MATKINSO | 650.124.6234 | 1997-10-12 | ST_CLERK | 2800 | NULL | 121 | 110 |
31 rows in set (0.00 sec)

mysql> SELECT Department_name Department ,MAX(salary) `Maximum Salary` FROM departments d JOIN employees e ON d.Department_id=e.department_id GROUP BY Department;
+-----+-----+
| Department | Maximum Salary |
+-----+-----+
| Marketing | 24000 |
| Purchasing | 17000 |
| IT | 9000 |
| Human Resources | 7900 |
| Finance | 12000 |
| Payroll | 9000 |
| Benefits | 7700 |
| Shareholder Services | 7800 |
| Control And Credit | 6900 |
| Sales | 5800 |
| Public Relations | 2900 |
| Corporate Tax | 2500 |
| Shipping | 8200 |
| Executive | 2400 |
| Accounting | 2800 |
+-----+-----+
15 rows in set (0.00 sec)

mysql> |

```

14. Write a SQL query to display the 5 least earning employees

```

SELECT first_name, last_name FROM employees ORDER BY salary LIMIT 5;

```

```

Windows PowerShell x Windows PowerShell x Windows PowerShell x + -
| Julia      | Nayer        | 3200 |
| Laura     | Bissot       | 3300 |
| Diana     | Lorentz      | 4200 |
| David     | Austin       | 4800 |
| Valli     | Pataballa    | 4800 |
| Kevin     | Mourgos     | 5800 |
| Bruce     | Ernst        | 6000 |
| Shanta    | Vollman     | 6500 |
| Luis      | Popp         | 6900 |
| Ismael    | Sciarra      | 7700 |
| Jose Manuel | Urman       | 7800 |
| Payam     | Kaufling     | 7900 |
| Matthew   | Weiss        | 8000 |
| John      | Chen          | 8200 |
| Adam      | Fypp          | 8200 |
| Alexander | Hunold       | 9000 |
| Daniel    | Faviet       | 9000 |
| Den       | Raphaelly    | 11800 |
| Nancy     | Greenberg    | 12000 |
| Neena     | Kochhar      | 17600 |
| Lex       | De Haan       | 17600 |
| Steven    | King          | 24000 |
+-----+
31 rows in set (0.00 sec)

mysql> SELECT first_name, last_name FROM employees ORDER BY salary LIMIT 5;
+-----+-----+
| first_name | last_name |
+-----+-----+
| Steven     | Markle      |
| James      | Landry      |
| Karen      | Colmenares |
| Guy        | Hiru        |
| Irene      | Mikkilineni |
+-----+-----+
5 rows in set (0.01 sec)

mysql>

```

28°C Mostly cloudy Search B C E G DELL W 01:33 14-06-2023

15. Find the employees hired in the 80s

`SELECT first_name, last_name FROM employees WHERE hire_date LIKE '198_-%-%';`

```

Windows PowerShell x Windows PowerShell x Windows PowerShell x + -
| 112 | Jose Manuel | Urman      | JMURMAN | 515.124.4469 | 1998-06-03 | FI_ACCOUNT | 7800 | NULL
| 108 | Irene        | Mikkilineni | IMIKKILI | 650.124.1224 | 1998-11-12 | ST_CLERK  | 2700 | NULL
| 126 | James        | Landry      | JLANDRY  | 650.124.1334 | 1999-01-02 | ST_CLERK  | 2400 | NULL
| 120 | Karen        | Colmenares | KCOLMENA | 515.127.4566 | 1999-04-08 | PU_CLERK  | 2500 | NULL
| 127 | Diana        | Lorentz     | DLORENTZ | 590.423.5567 | 1999-02-09 | IT_PROG   | 4200 | NULL
| 107 | Guy           | Hiru        | KMOURGOS | 650.123.5234 | 1999-11-12 | ST_MAN    | 5800 | NULL
| 103 | Neena         | Kochhar    | LKOCHHAR | 515.124.4567 | 1999-12-07 | FI_ACCOUNT | 6900 | NULL
| 119 | Luis          | Popp        | LPOPP     | 515.124.4567 | 2000-03-04 | ST_CLERK  | 2200 | NULL
| 114 | Kevin          | Mourgos    | KMOURGOS | 650.124.1434 | 2000-03-04 | ST_CLERK  | 2200 | NULL
| 124 | Den            | Raphaelly  | DRAPHAELLY | 650.124.1434 | 2000-03-04 | ST_CLERK  | 2200 | NULL
| 100 | Luis            | Popp        | LPOPP     | 515.124.4567 | 2000-03-04 | ST_CLERK  | 2200 | NULL
| 113 | Daniel          | Faviet      | DFAVIET  | 515.124.4567 | 2000-03-04 | ST_CLERK  | 2200 | NULL
| 108 | Karen            | Colmenares | KCOLMENA | 515.127.4566 | 2000-03-04 | ST_CLERK  | 2200 | NULL
| 128 | Steven          | Markle      | SMARKLE  | 650.124.1434 | 2000-03-04 | ST_CLERK  | 2200 | NULL
| 120 | Irene            | Mikkilineni | IMIKKILI | 650.124.1224 | 2000-03-04 | ST_CLERK  | 2200 | NULL
+-----+
31 rows in set (0.00 sec)

mysql> SELECT first_name, last_name FROM employees WHERE hire_date LIKE '198_--%';
+-----+-----+
| first_name | last_name |
+-----+-----+
| Steven     | King        |
| Neena     | Kochhar    |
+-----+-----+
2 rows in set (0.00 sec)

mysql>

```

28°C Mostly cloudy Search B C E G DELL W 01:37 14-06-2023

16. Find the employees who joined the company after 15th of the month

`SELECT first_name, last_name FROM employees WHERE DAY(hire_date)>15;`

```
Windows PowerShell x Windows PowerShell x Windows PowerShell x + v
+-----+
| 127 | James      | Landry    | JLANDRY | 650.124.1334 | 1999-01-02 | ST_CLERK   | 2400 | NULL
| 128 |          98 |          |
| 107 | Diana      | Lorentz   | DLORENTZ | 590.423.5567 | 1999-02-09 | IT_PROG    | 4200 | NULL
| 119 | Karen      | Colmenares | KCOLMENA | 515.127.4566 | 1999-04-08 | PU_CLERK   | 2500 | NULL
| 114 |          130 |          |
| 124 | Kevin      | Mourgos  | KMOURGOS | 650.123.5234 | 1999-11-12 | ST_MAN    | 5800 | NULL
| 100 |          88 |          |
| 113 | Luis       | Popp      | LPOPP    | 515.124.4567 | 1999-12-07 | FI_ACCOUNT | 6900 | NULL
| 108 |          140 |          |
| 128 | Steven     | Markle   | SMARKLE  | 650.124.1434 | 2000-03-04 | ST_CLERK   | 2200 | NULL
| 120 |          50 |          |
+-----+
31 rows in set (0.00 sec)

mysql> SELECT first_name, last_name FROM employees WHERE DAY(hire_date)>15;
+-----+-----+
| first_name | last_name |
+-----+-----+
| Steven     | King      |
| Neena      | Kochhar   |
| Alexander  | Hunold    |
| Bruce      | Ernst     |
| David      | Austin    |
| Nancy      | Greenberg |
| Matthew    | Weiss    |
+-----+-----+
7 rows in set (0.00 sec)

mysql>
```

