MY UNIVERSITY

STEP INTO TOMORROW

DepartmentofComputerScience

Spring-2024 DataBase Systems (CS-103)

Lab Task # 06

SQL-Single Rows Function

Date: 21 May 2024

Instructors: Mr. Hamza Javed

Important Instruction

1. No plagiarism allowed.
2. You will required to submit only soft copy via google classroom
3. Late submission not accepted
4. Rename submission file DB24-Name-RegNo. **like: DB24-hamza-1234**
5. You will required to submit single **pdf**

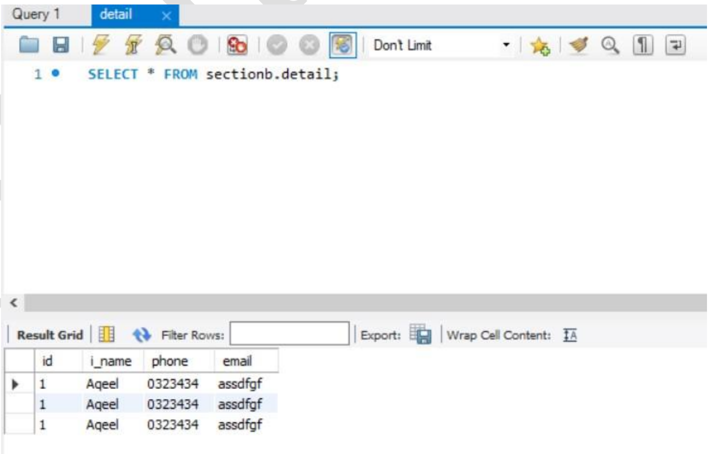
Submission Example

Question .

Write query to select all data from table:

Solution

SELECT * FROM db.detail;

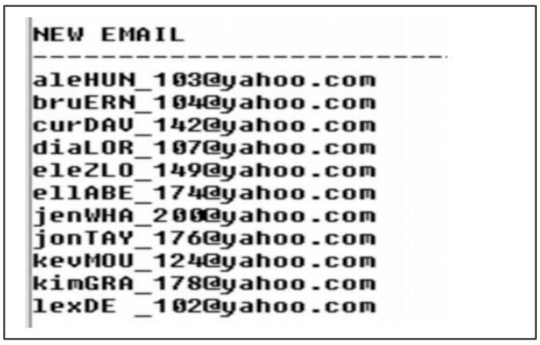


```
1 ●  SELECT * FROM sectionb.detail;
```

id	i_name	phone	email
1	Aqeel	0323434	assdfgf
1	Aqeel	0323434	assdfgf
1	Aqeel	0323434	assdfgf

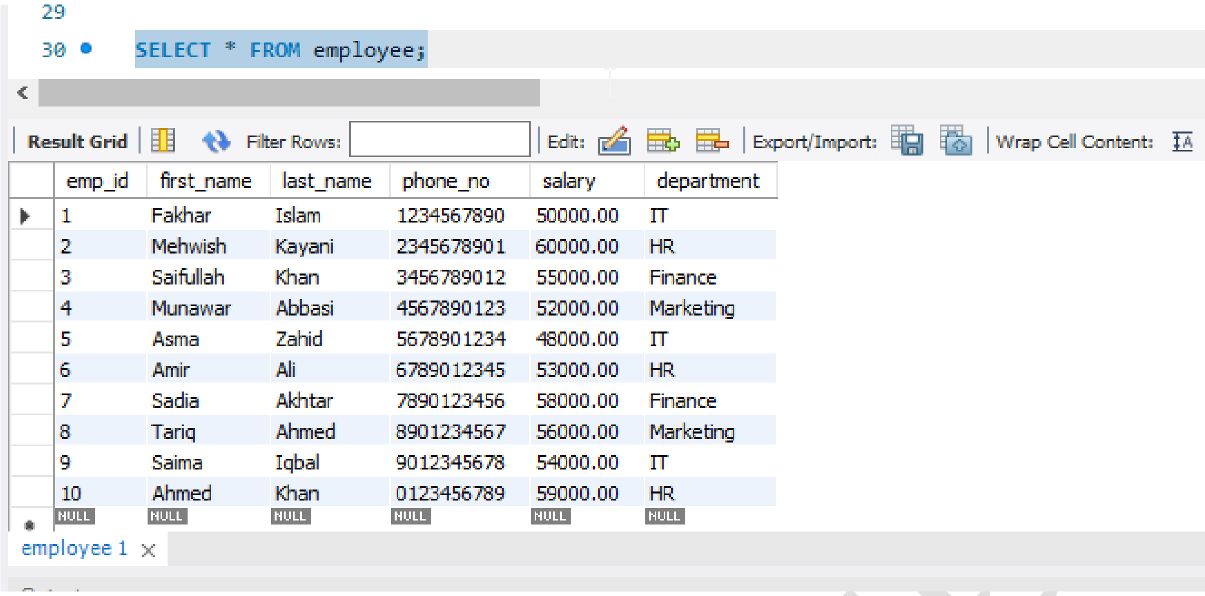
Question 1

1. **Create a new email, which will contain the first 3 letters of the first name and last name and employee id. Sort the data in ascending order. Do not use Concat Operator ||. See sample output below for additional character strings and formatting.**



```
NEW EMAIL
------------------------
aleHUN_103@yahoo.com
bruERN_104@yahoo.com
curDAV_142@yahoo.com
diaLOR_107@yahoo.com
eleZLO_149@yahoo.com
ellABE_174@yahoo.com
jenWHA_200@yahoo.com
jonTAY_176@yahoo.com
kevMOU_124@yahoo.com
kimGRA_178@yahoo.com
lexDE _102@yahoo.com
```

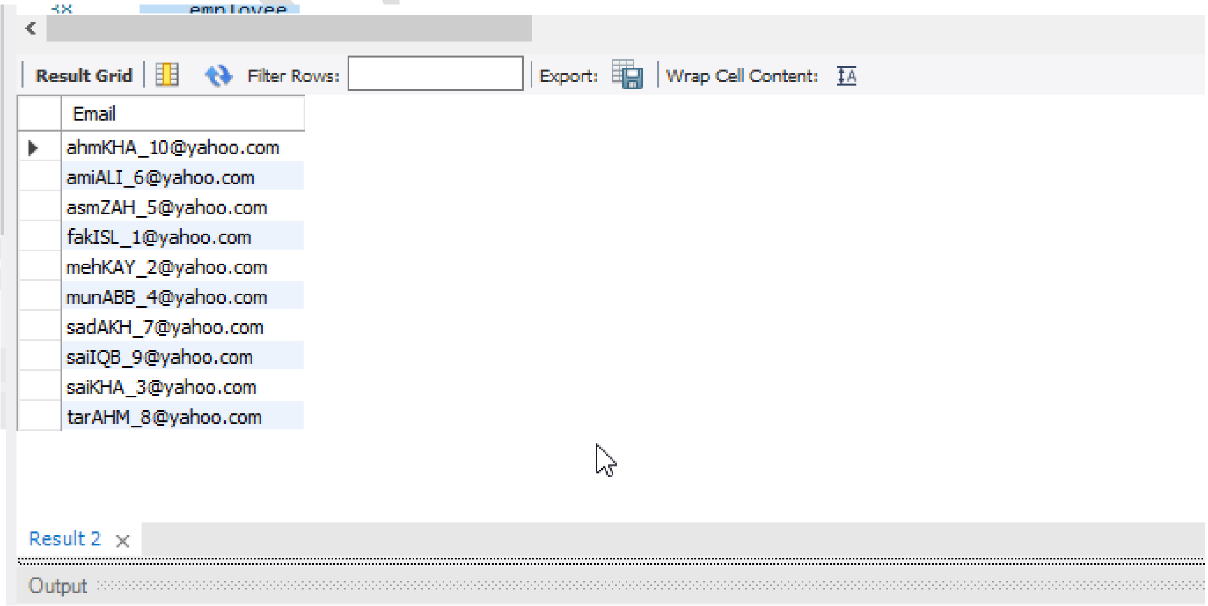
```
CREATE DATABASE IF NOT EXISTS dbp6;

USE dbp6;

CREATE TABLE IF NOT EXISTS employee (
    emp_id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    phone_no VARCHAR(15),
    salary DECIMAL(10, 2),
    department VARCHAR(50)


INSERT INTO employee (first_name, last_name, phone_no, salary, department)
VALUES
('Fakhar', 'Islam', '1234567890', 50000.00, 'IT'),
('Mehwish', 'Kayani', '2345678901', 60000.00, 'HR'),
('Saifullah', 'Khan', '3456789012', 55000.00, 'Finance'),
('Munawar', 'Abbasi', '4567890123', 52000.00, 'Marketing'),
('Asma', 'Zahid', '5678901234', 48000.00, 'IT'),
('Amir', 'Ali', '6789012345', 53000.00, 'HR'),
('Sadia', 'Akhtar', '7890123456', 58000.00, 'Finance'),
('Tariq', 'Ahmed', '8901234567', 56000.00, 'Marketing'),
('Saima', 'Iqbal', '9012345678', 54000.00, 'IT'),
('Ahmed', 'Khan', '0123456789', 59000.00, 'HR');
```



```sql
SELECT * FROM employee;
```

emp_id	first_name	last_name	phone_no	salary	department
1	Fakhar	Islam	1234567890	50000.00	IT
2	Mehwish	Kayani	2345678901	60000.00	HR
3	Saifullah	Khan	3456789012	55000.00	Finance
4	Munawar	Abbasi	4567890123	52000.00	Marketing
5	Asma	Zahid	5678901234	48000.00	IT
6	Amir	Ali	6789012345	53000.00	HR
7	Sadia	Akhtar	7890123456	58000.00	Finance
8	Tariq	Ahmed	8901234567	56000.00	Marketing
9	Saima	Iqbal	9012345678	54000.00	IT
10	Ahmed	Khan	0123456789	59000.00	HR
NULL	NULL	NULL	NULL	NULL	NULL

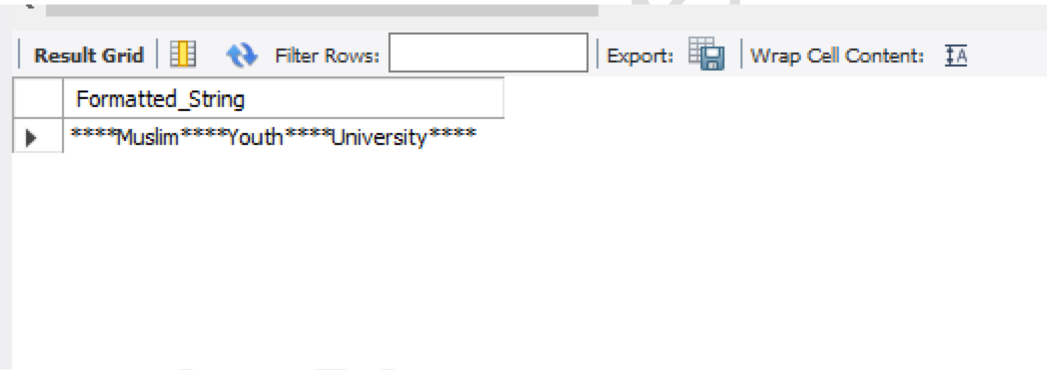
```sql
SELECT
   CONCAT(
      LOWER(LEFT(first_name, 3)),
      UPPER(LEFT(last_name, 3)),
      '_',
      emp_id,
      '@yahoo.com'
   ) AS Email
FROM
   employee
ORDER BY
   Email ASC;
```



Email
ahmKHA_10@yahoo.com
amiALI_6@yahoo.com
asmZAH_5@yahoo.com
fakISL_1@yahoo.com
mehKAY_2@yahoo.com
munABB_4@yahoo.com
sadAKH_7@yahoo.com
saiIQB_9@yahoo.com
saiKHA_3@yahoo.com
tarAHM_8@yahoo.com

2. **Starting with the string “Muslim Youth University”, pad the string to create: ****Muslim****Youth****University******

```
SELECT
    CONCAT(
        RPAD('', 4, '*'),
        'Muslim',
        RPAD('', 4, '*'),
        'Youth',
        RPAD('', 4, '*'),
        'University',
        RPAD('', 4, '*')
    ) AS Formatted_String;
```

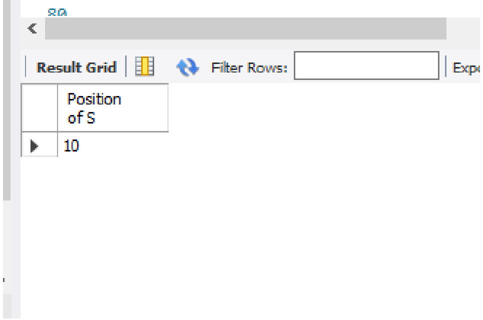


Formatted_String
****Muslim****Youth****University****

3. **How many months between your birthday this year and January 1 next year?**

4. **What’s the position of “S” in “Internet Service”? Label as Position. Use the string “Internet Service” to produce the following output below. Label this column as The Net. Write your SQL Statement**

```
SELECT
INSTR('Internet Service', 'S') AS "Position of S"
```



Position of S
10

Question 2

Library Management System

Given Table are following:

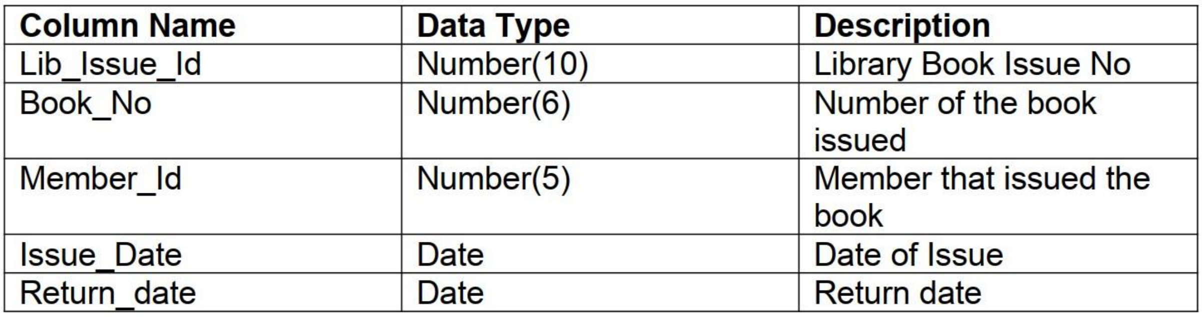
Members: It contains information about the members

Column Name	Data Type	Description
Member_Id	Number(5)	Unique Member ID
Member_Name	Varchar2(30)	Name of the Library member
Member_address	Varchar2(50)	Address of the member
Acc_Open_Date	Date	Date of membership
Membership_type	Varchar2(20)	Type of the membership such as 'Lifetime',' Annual', 'Half Yearly',' Quarterly'
Fees_paid	Number(4)	Membership fees paid
Max_Books_Allowed	Number(2)	Total Number of books that can be issued to the member.
Penalty_Amount	Number(7,2)	Penalty amount due

Books: It contains information about the books belongs to the library

Column Name	Data Type	Description
Book_No	Number(6)	Book identification number
Book_Name	VarChar2(30)	Name of the book
Author_name	Varchar2(30)	Author of the book
Cost	Number(7,2)	Cost of the book
Category	Char(10)	Category like Science , Fiction etc.

Issue: It contains the information about issue of the books



Column Name	Data Type	Description
Lib_Issue_Id	Number(10)	Library Book Issue No
Book_No	Number(6)	Number of the book issued
Member_Id	Number(5)	Member that issued the book
Issue_Date	Date	Date of Issue
Return_date	Date	Return date

a) **Create Database library management system.**

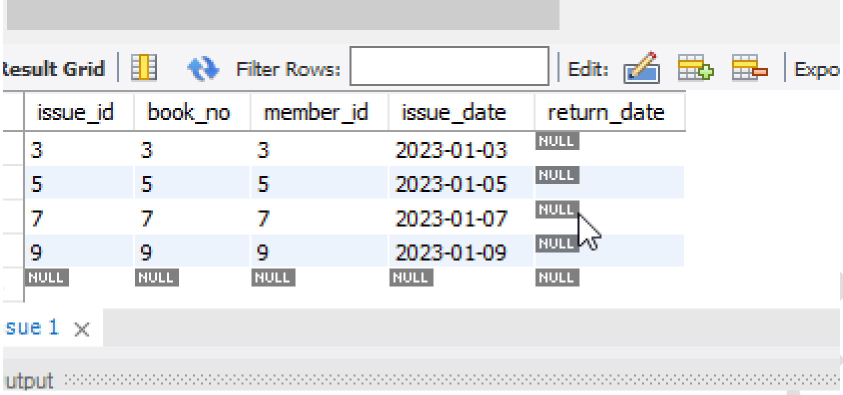
b) **create above mention table and insert upto 15 records. Note: Must use year 2023 as a date in each record like: 01-01-2023**

```
CREATE TABLE members (
  member_id INT PRIMARY KEY,
  member_name VARCHAR(30),
  member_address VARCHAR(30),
  acnt_open_date DATE,
  membership_type VARCHAR(20),
  fees_paid INT,
  max_book_allowed INT,
  penalty_amnt DECIMAL(7,2)
);
INSERT INTO members (
  member_id, member_name, member_address, acnt_open_date, membership_type,
fees_paid, max_book_allowed, penalty_amnt
) VALUES
  (1, 'Alice Johnson', '123 Maple St', '2023-01-01', 'Annual', 50, 5, 0.00),
  (2, 'Bob Smith', '456 Oak St', '2023-01-02', 'Lifetime', 500, 10, 0.00),
  (3, 'Carol White', '789 Pine St', '2023-01-03', 'HalfYearly', 30, 3, 0.00),
  (4, 'David Brown', '101 Birch St', '2023-01-04', 'Annual', 50, 5, 0.00);


INSERT INTO members (
  member_id, member_name, member_address, acnt_open_date, membership_type,
fees_paid, max_book_allowed, penalty_amnt
) VALUES
(5, 'Eva Green', '202 Cedar St', '2023-01-05', 'Annual', 50, 5, 0.00),
  (6, 'Frank Black', '303 Elm St', '2023-01-06', 'Lifetime', 500, 10, 0.00),
  (7, 'Grace Blue', '404 Willow St', '2023-01-07', 'HalfYearly', 30, 3, 0.00),
  (8, 'Hank Gray', '505 Ash St', '2023-01-08', 'Annual', 50, 5, 0.00),
  (9, 'Ivy Gold', '606 Cherry St', '2023-01-09', 'Annual', 50, 5, 0.00),
  (10, 'Jack White', '707 Peach St', '2023-01-10', 'Lifetime', 500, 10, 0.00),
  (11, 'Kathy Silver', '808 Plum St', '2023-01-11', 'HalfYearly', 30, 3, 0.00),
  (12, 'Liam Red', '909 Palm St', '2023-01-12', 'Annual', 50, 5, 0.00),
  (13, 'Mia Brown', '1010 Maple St', '2023-01-13', 'Annual', 50, 5, 0.00),
  (14, 'Noah Green', '1111 Oak St', '2023-01-14', 'Lifetime', 500, 10, 0.00),
  (15, 'Olivia Black', '1212 Pine St', '2023-01-15', 'HalfYearly', 30, 3, 0.00);
```

```
CREATE TABLE book (
   book_id INT PRIMARY KEY,
   book_name VARCHAR(50),
   author_name VARCHAR(50),
   cost DECIMAL(10,2),
   category VARCHAR(30)
);
INSERT INTO book (
   book_id, book_name, author_name, cost, category
) VALUES
   (1, 'To Kill a Mockingbird', 'Harper Lee', 18.99, 'Fiction'),
   (2, '1984', 'George Orwell', 14.99, 'Dystopian'),
   (3, 'Pride and Prejudice', 'Jane Austen', 12.99, 'Classic'),
   (4, 'The Great Gatsby', 'F. Scott Fitzgerald', 15.99, 'Classic'),
   (5, 'Moby Dick', 'Herman Melville', 19.99, 'Classic'),
   (6, 'The Catcher in the Rye', 'J.D. Salinger', 13.99, 'Fiction'),
   (7, 'The Hobbit', 'J.R.R. Tolkien', 25.99, 'Fantasy'),
   (8, 'Harry Potter and the Philosopher\'s Stone', 'J.K. Rowling', 29.99, 'Fantasy'),
   (9, 'The Lord of the Rings', 'J.R.R. Tolkien', 35.99, 'Fantasy'),
   (10, 'Animal Farm', 'George Orwell', 10.99, 'Dystopian'),
   (11, 'War and Peace', 'Leo Tolstoy', 24.99, 'Historical'),
   (12, 'The Odyssey', 'Homer', 16.99, 'Epic'),
   (13, 'Crime and Punishment', 'Fyodor Dostoevsky', 18.99, 'Classic'),
   (14, 'Brave New World', 'Aldous Huxley', 14.99, 'Dystopian'),
   (15, 'Wuthering Heights', 'Emily Brontë', 11.99, 'Classic');


   USE library_management;

CREATE TABLE issue (
   issue_id INT PRIMARY KEY,
   book_no INT,
   member_id INT,
   issue_date DATE,
   return_date DATE,
   FOREIGN KEY (book_no) REFERENCES book(book_id),
   FOREIGN KEY (member_id) REFERENCES members(member_id)
);

INSERT INTO issue (
   issue_id, book_no, member_id, issue_date, return_date
) VALUES
   (1, 1, 1, '2023-01-01', '2023-01-15'),
   (2, 2, 2, '2023-01-02', '2023-01-16'),
   (3, 3, 3, '2023-01-03', NULL),
   (4, 4, 4, '2023-01-04', '2023-01-18'),
   (5, 5, 5, '2023-01-05', NULL),
   (6, 6, 6, '2023-01-06', '2023-01-20'),
   (7, 7, 7, '2023-01-07', NULL),
   (8, 8, 8, '2023-01-08', '2023-01-22'),
   (9, 9, 9, '2023-01-09', NULL),
(10, 10, 10, '2023-01-10', '2023-01-24');
```

c) List the Issue details for the books that are not returned yet.

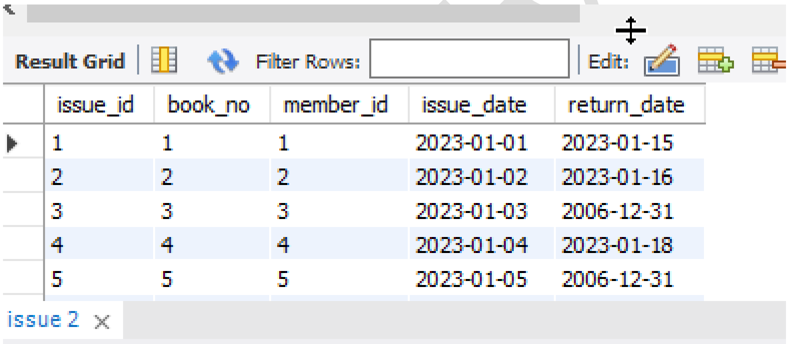
```
SELECT *
FROM issue
WHERE return_date IS NULL;
```



issue_id	book_no	member_id	issue_date	return_date
3	3	3	2023-01-03	NULL
5	5	5	2023-01-05	NULL
7	7	7	2023-01-07	NULL
9	9	9	2023-01-09	NULL
NULL	NULL	NULL	NULL	NULL

d) Update all the blank return date with 31-Dec-06 excluding last two record of tables.

```
UPDATE issue
SET return_date = '2006-12-31'
WHERE return_date IS NULL;
```



issue_id	book_no	member_id	issue_date	return_date
1	1	1	2023-01-01	2023-01-15
2	2	2	2023-01-02	2023-01-16
3	3	3	2023-01-03	2006-12-31
4	4	4	2023-01-04	2023-01-18
5	5	5	2023-01-05	2006-12-31

e) List all the Issue details of books issued for more then 30 days.

```
select * from issue
where  (return_date - issue_date) > 30 ;
```

f) List all the members in the descending order of Penalty due on them

```
SELECT *
FROM members
ORDER BY penalty_amnt DESC;
```

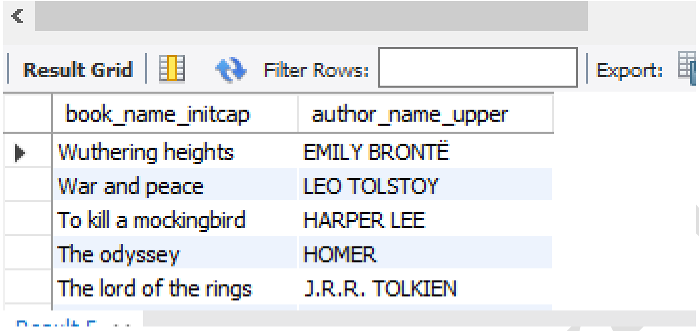
g) List all the members whose name starts with 'R' or 'G' and contains letter 'I' in it.

```
SELECT *
FROM members
```

```
WHERE (member_name LIKE 'R%' OR member_name LIKE 'G%')
  AND member_name LIKE '%I%';
```

h) List the entire book name in Init cap and author in upper case in the descending order of the book name.

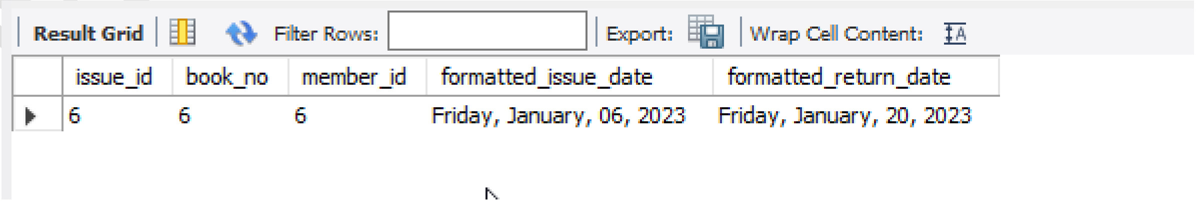
```
 SELECT
   CONCAT(UCASE(LEFT(book_name, 1)), LCASE(SUBSTRING(book_name, 2))) AS book_name_initcap,
   UPPER(author_name) AS author_name_upper
FROM book
ORDER BY book_name DESC;
```



book_name_initcap	author_name_upper
Wuthering heights	EMILY BRONTË
War and peace	LEO TOLSTOY
To kill a mockingbird	HARPER LEE
The odyssey	HOMER
The lord of the rings	J.R.R. TOLKIEN

i) List the Issue Details for all the books issue to member 101 with Issue date and Return Date in following format. 'Tuesday, July, 10, 2023'

```
SELECT
   issue_id,
   book_no,
   member_id,
   DATE_FORMAT(issue_date, '%W, %M, %d, %Y') AS formatted_issue_date,
   DATE_FORMAT(return_date, '%W, %M, %d, %Y') AS formatted_return_date
FROM
   issue
WHERE
   member_id = 6;
```



issue_id	book_no	member_id	formatted_issue_date	formatted_return_date
6	6	6	Friday, January, 06, 2023	Friday, January, 20, 2023

j) List all the members that became the member in the year 2023.

```
SELECT *
FROM members
WHERE YEAR(acnt_open_date) = 2023
```

k) List the Lib Issue Id, Issue Date, Return Date and No of days Book was issued.

```
SELECT
   issue_id AS Lib_Issue_Id,
   issue_date AS Issue_Date,
   return_date AS Return_Date,
   (return_date - issue_date ) as no_of_days
FROM
   issue;
```

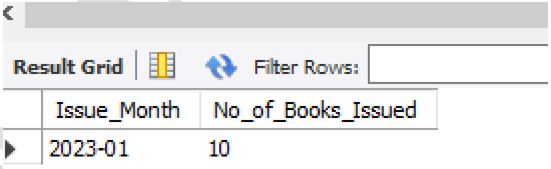
l) Display no of months between issue date and return date for all issues

m) Display the last day of the month of the issue date.

```
SELECT
   issue_id,
   issue_date,
   LAST_DAY(issue_date) AS Last_Day_Of_Issue_Month
FROM
   issue;
```

n) Display the month and no of books issued each month in the descending order of count.

```
SELECT

   DATE_FORMAT(issue_date, '%Y-%m') AS Issue_Month,
   COUNT(*) AS No_of_Books_Issued
FROM
   issue
GROUP BY
   DATE_FORMAT(issue_date, '%Y-%m')
ORDER BY
   No_of_Books_Issued DESC;
```



Issue_Month	No_of_Books_Issued
2023-01	10

o) Find the maximum, minimum, total and average penalty amount in the member table

```
SELECT
   MAX(penalty_amnt) AS Maximum_Penalty,
   MIN(penalty_amnt) AS Minimum_Penalty,
   SUM(penalty_amnt) AS Total_Penalty,
   AVG(penalty_amnt) AS Average_Penalty
FROM
```

members;