

SQL Project

Library Management System

Database Creation

```
1      -- Library management system
2      -- step 1
3  ●    CREATE DATABASE library;
4  ●    use library;
5      |
6
```

Table Creation

Table 1– books

Table 2– Mmembers

```
1      -- step-2 --creating table
2
3  • Create table books (
4      bookID int auto_increment PRIMARY KEY,
5      title varchar(100) not null ,
6      author varchar(100),
7      genre varchar(50),
8      PublishedYear int ,
9      AopiesAvailable int DEFAULT 1
10  );
11
12
13      -- table creation for members
14  • Create table Members(
15      MemberID int auto_increment primary key,
16      name varchar(100) not null,
17      email varchar(100),
18      phonenumber varchar(15),
19      joindate date
20  );
21
```

Table Creation

Table 3– Borrow History

```
21
22  -- table creation for borrowhistory
23  create table BorrowHistory (
24      BorrowID INT auto_increment PRIMARY KEY,
25      BookID INT,
26      MemberID INT,
27      BorrowDate date,
28      RetunDate date,
29      foreign key (BookID) references BOOKS(BOOKID),
30      foreign key(MEMBERID) references MEMBERS(MEMBERID)
31  );
32
```

Data Insertion into tables –

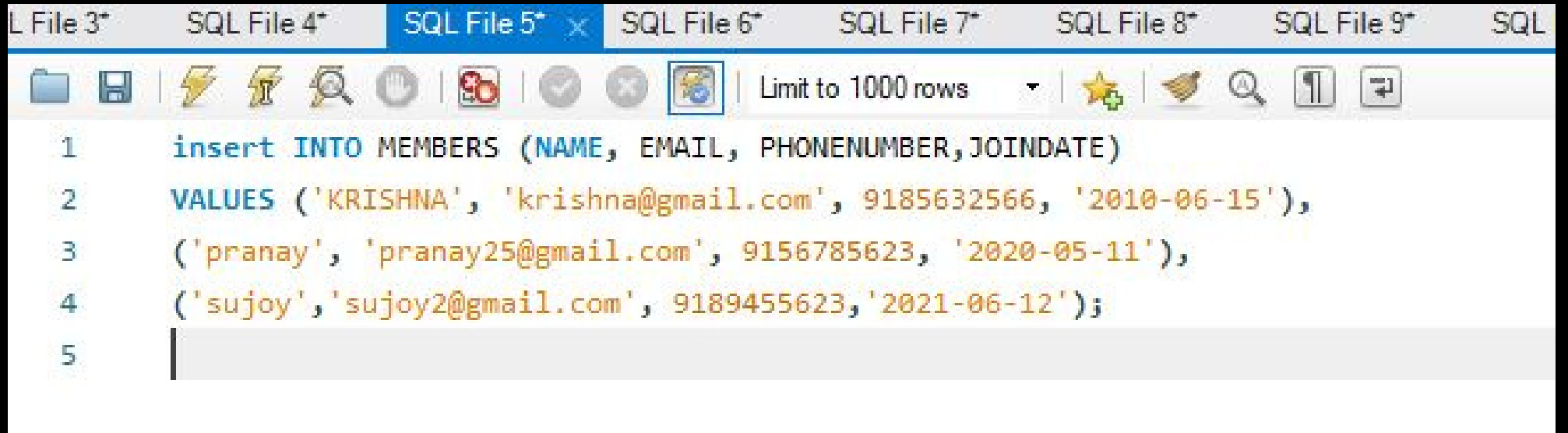
- 1.)book,
- 2.)members,
- 3.)borrow history

```
SQL File 11*  SQL File 3*  SQL File 4* x  SQL File 5*  SQL File 6*  SQL File 7*  SQL File 8*
Limit to 1000 rows
1  -- INSERTION OF DATA
2  •  INSERT INTO books( title , author, genre, publishedyear, copiesavailable)
3  values ('The great mall' , 'scott', 'fiction', 1988, 3),
4  ('horror house' , 'george' , 'fiction', 1949 , 4 ),
5  ('to kill someone' , 'lee', 'classic' , 2001, 6),
6  ('the catcher', 'dharma' , 'fiction', 2010,2);
7  |
```

Result Grid | Filter Rows: | Edit: | Export/Import:

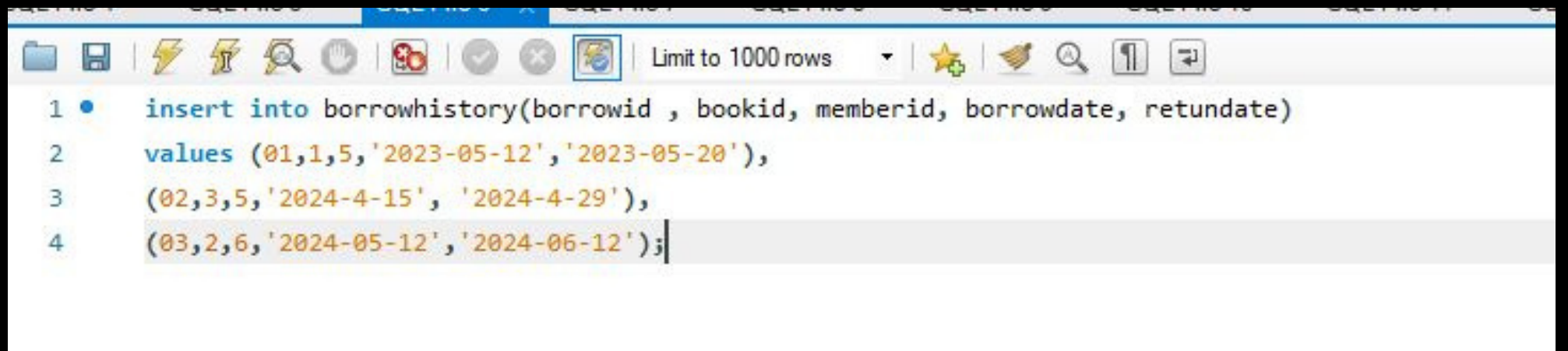
	bookID	title	author	genre	PublishedYear	copiesavailable
▶	1	The great mall	scott	fiction	1988	3
	2	horror house	george	fiction	1949	4
	3	to kill someone	lee	classic	2001	5
	4	the catcher	dharma	fiction	2010	2
•	NULL	NULL	NULL	NULL	NULL	NULL

continue....



The screenshot shows a SQL editor window with multiple tabs. The active tab is 'SQL File 5*'. The editor contains an SQL insert statement for a table named 'MEMBERS'. The statement uses the 'insert INTO' syntax with columns 'NAME', 'EMAIL', 'PHONENUMBER', and 'JOINDATE'. It includes three rows of data: 'KRISHNA' with email 'krishna@gmail.com' and phone number '9185632566' (joined '2010-06-15'), 'pranay' with email 'pranay25@gmail.com' and phone number '9156785623' (joined '2020-05-11'), and 'sujoy' with email 'sujoy2@gmail.com' and phone number '9189455623' (joined '2021-06-12'). The editor has a toolbar with various icons and a 'Limit to 1000 rows' dropdown.

```
1 insert INTO MEMBERS (NAME, EMAIL, PHONENUMBER, JOINDATE)
2 VALUES ('KRISHNA', 'krishna@gmail.com', 9185632566, '2010-06-15'),
3 ('pranay', 'pranay25@gmail.com', 9156785623, '2020-05-11'),
4 ('sujoy', 'sujoy2@gmail.com', 9189455623, '2021-06-12');
5
```



The screenshot shows a SQL editor window with multiple tabs. The active tab is 'SQL File 5*'. The editor contains an SQL insert statement for a table named 'borrowhistory'. The statement uses the 'insert into' syntax with columns 'borrowid', 'bookid', 'memberid', 'borrowdate', and 'retundate'. It includes three rows of data: (01, 1, 5, '2023-05-12', '2023-05-20'), (02, 3, 5, '2024-4-15', '2024-4-29'), and (03, 2, 6, '2024-05-12', '2024-06-12'). The editor has a toolbar with various icons and a 'Limit to 1000 rows' dropdown.

```
1 • insert into borrowhistory(borrowid , bookid, memberid, borrowdate, retundate)
2 values (01,1,5,'2023-05-12','2023-05-20'),
3 (02,3,5,'2024-4-15', '2024-4-29'),
4 (03,2,6,'2024-05-12','2024-06-12');
```

Query-1 view all books

```
-- view all books
```

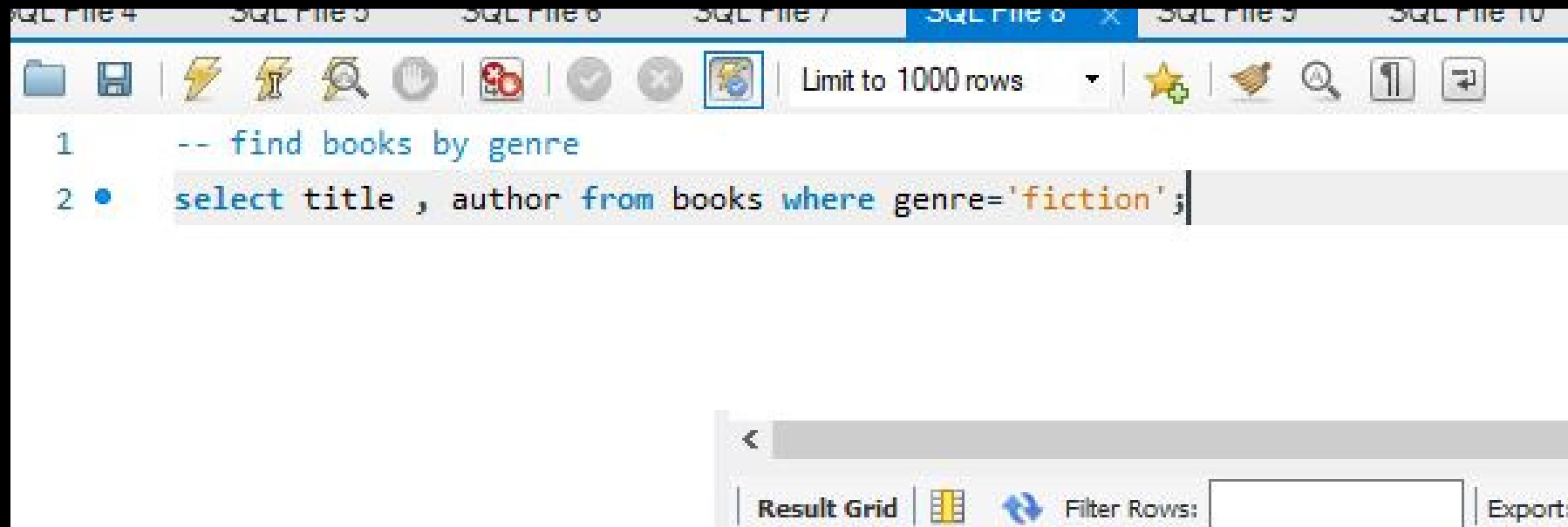
```
select * from books;
```

Filter Rows: Edit:    Export/Import:   Wrap Cell Content

	title	author	genre	PublishedYear	copiesavailable
	The great mall	scott	fiction	1988	3
	horror house	george	fiction	1949	4
	to kill someone	lee	classic	2001	6
	the catcher	dharma	fiction	2010	2
	NULL	NULL	NULL	NULL	NULL

output

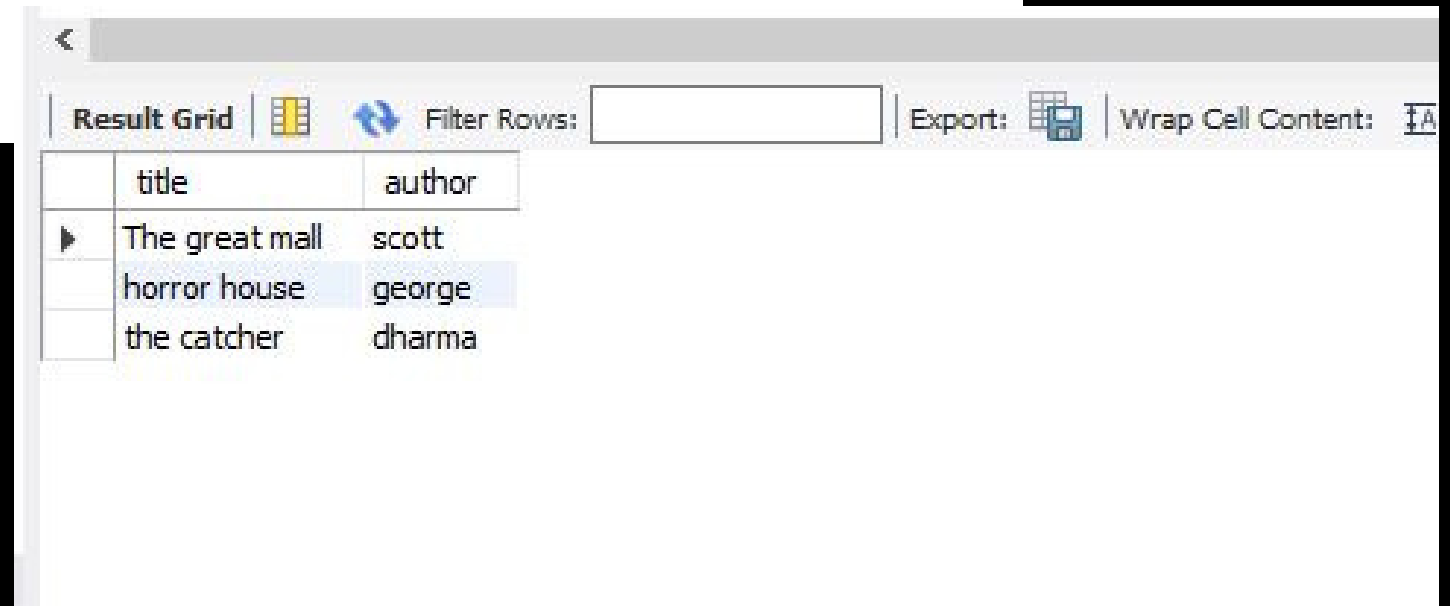
Query-2 Find books by genre



The screenshot shows a SQL editor with multiple tabs labeled 'SQL File 4' through 'SQL File 10'. The active tab is 'SQL File 8'. The editor contains two lines of SQL code: a comment and a select statement. The toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown.

```
1  -- find books by genre
2  select title , author from books where genre='fiction';
```

output



The screenshot shows a 'Result Grid' window with a table containing three rows of data. The table has two columns: 'title' and 'author'. The first row is 'The great mall' by 'scott', the second is 'horror house' by 'george', and the third is 'the catcher' by 'dharma'.

	title	author
▶	The great mall	scott
	horror house	george
	the catcher	dharma


Query-3 Find members who borrowed books

```
1  -- find the members who borrowed books
2  • select members.name as membername , books.title as booktitle, borrowhisto
3
4  from borrowhistory
5
6  join members on borrowhistory.memberid = members.memberid
7  join books on borrowhistory.BookID= books.bookID;
```

output

Result Grid			
Filter Rows:			
	membername	booktitle	BorrowDate
▶	KRISHNA	The great mall	2023-05-12
	KRISHNA	to kill someone	2024-04-15
	pranay	horror house	2024-05-12

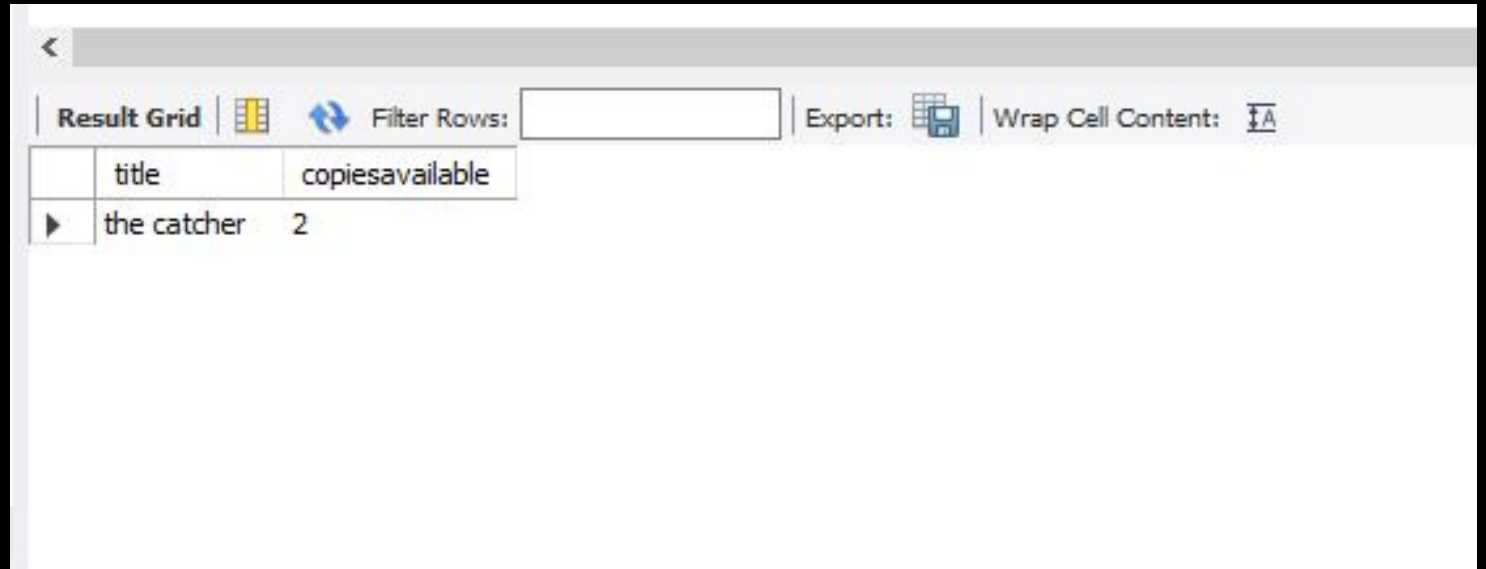
Query-4 Check the books with low availability.



A screenshot of a SQL query editor window. The toolbar at the top includes icons for file operations, execution, and a dropdown menu set to 'Limit to 1000 rows'. The query text is as follows:

```
1  -- check the books with low availability
2
3  •  select title, copiesavailable from books where copiesavailable<3;
```

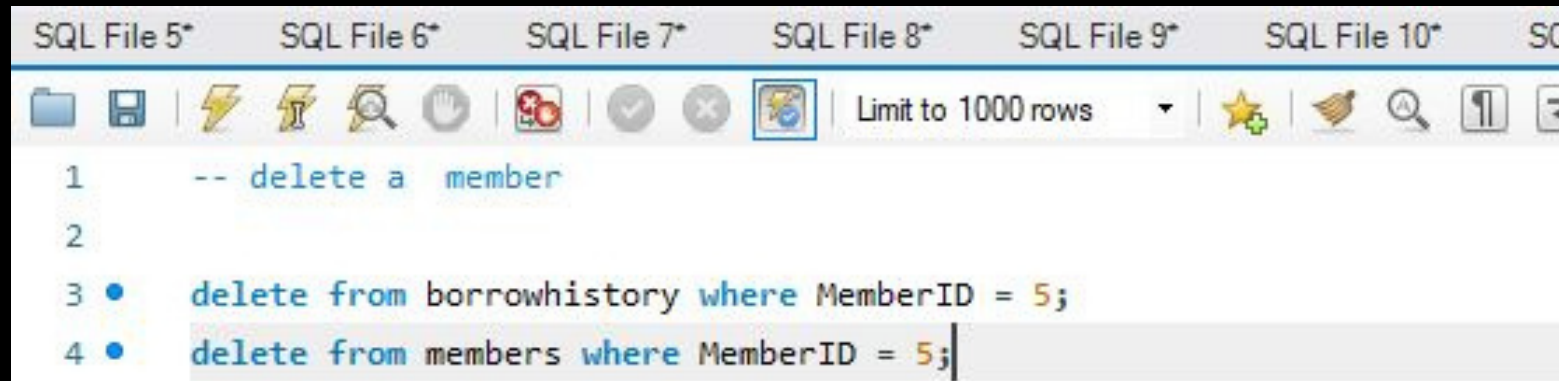
output



A screenshot of a database result grid window. The toolbar at the top includes a 'Result Grid' button, a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' button. The table below shows the results of the query:

	title	copiesavailable
▶	the catcher	2

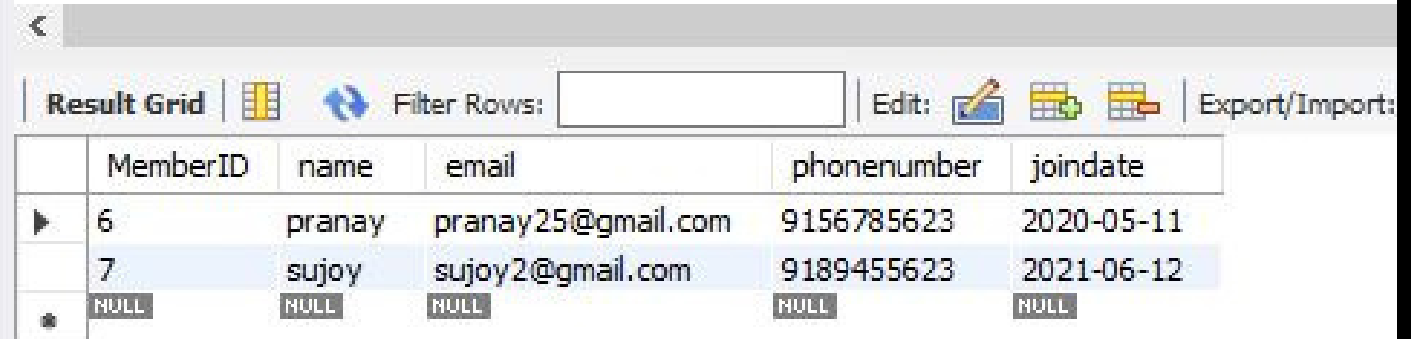
Query-5 Delete a member



The screenshot shows an SQL IDE with multiple tabs labeled 'SQL File 5*' through 'SQL File 10*'. The active tab 'SQL File 5*' contains the following SQL code:

```
1  -- delete a member
2
3  • delete from borrowhistory where MemberID = 5;
4  • delete from members where MemberID = 5;
```

output



The screenshot shows the 'Result Grid' of the SQL IDE. It displays the results of the query, showing a table with 6 columns: MemberID, name, email, phonenumber, and joindate. The table contains 3 rows of data, with the first two rows highlighted in blue. The third row is a summary row with NULL values.

	MemberID	name	email	phonenumber	joindate
▶	6	pranay	pranay25@gmail.com	9156785623	2020-05-11
	7	sujoy	sujoy2@gmail.com	9189455623	2021-06-12
*	NULL	NULL	NULL	NULL	NULL

Query_6 Find overdue books

```
SQL File 6  SQL File 7  SQL File 8  SQL File 9  SQL File 10  SQL File 11  SQL File 12  SQL
Limit to 1000 rows
1  -- find overdue books
2
3  •  SELECT
4      members.name AS membername,
5      books.title AS booktitle,
6      borrowhistory.borrowdate
7  FROM borrowhistory
8  JOIN members ON borrowhistory.memberid = members.MemberID
9  JOIN books ON borrowhistory.bookid = books.bookid
10 WHERE borrowhistory.ReturnDate IS NULL
11 AND borrowhistory.BorrowDate < DATE_SUB(CURDATE(), INTERVAL 7 DAY);
12
```

output

<		
Result Grid	Filter Rows: <input type="text"/>	Export: Wrap Cell Content:
membername	booktitle	borrowdate