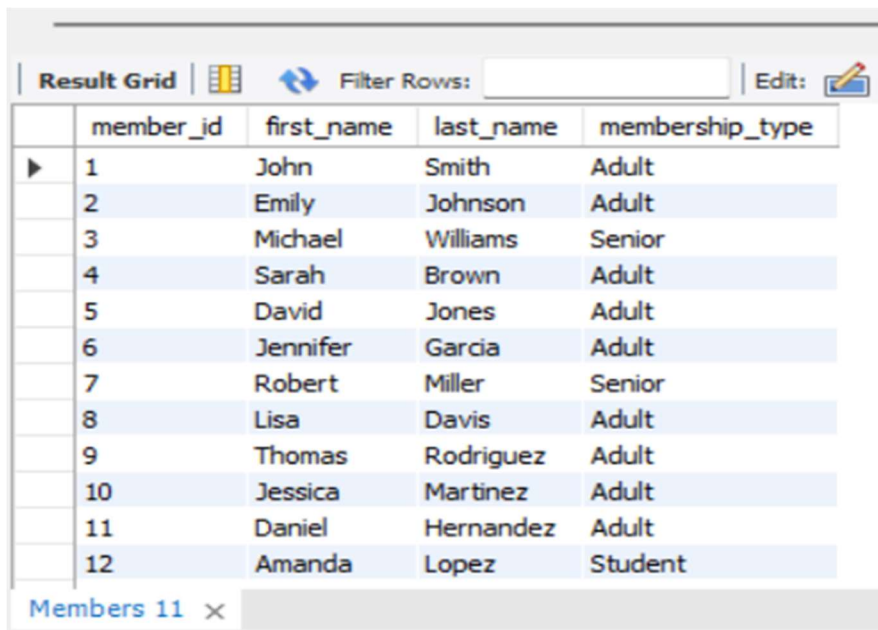


SQL QUERIES WITH OUTPUT

Query 1: List active members and membership type

```
SELECT member_id, first_name, last_name, membership_type FROM Members  
WHERE active_status = TRUE;
```

Output: List of 15 active members.




The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of a SQL query. The grid has a toolbar with icons for 'Result Grid', 'Filter Rows', and 'Edit'. The data is presented in a table with 5 columns: member_id, first_name, last_name, and membership_type. There are 12 rows of data. The first column has a small triangle icon next to it. At the bottom, there is a status bar that says 'Members 11' with a close button (X).

	member_id	first_name	last_name	membership_type
▶	1	John	Smith	Adult
	2	Emily	Johnson	Adult
	3	Michael	Williams	Senior
	4	Sarah	Brown	Adult
	5	David	Jones	Adult
	6	Jennifer	Garcia	Adult
	7	Robert	Miller	Senior
	8	Lisa	Davis	Adult
	9	Thomas	Rodriguez	Adult
	10	Jessica	Martinez	Adult
	11	Daniel	Hernandez	Adult
	12	Amanda	Lopez	Student

Query 2: Books with fewer than 2 available copies

```
SELECT book_id, title, available_copies FROM Books WHERE available_copies  
< 2;
```

Output: Books such as “The Hobbit”, “1984”, etc.

Result Grid			
Filter Rows: <input type="text"/>			
Edit: 			
	book_id	title	available_copies
▶	2	1984	1
	4	The Hobbit	0
	5	Murder on the Orient Express	1
	8	War and Peace	1
	10	The Old Man and the Sea	1
	11	The Great Gatsby	0
	13	A Tale of Two Cities	1
	14	One Hundred Years of Solitude	1
*	NULL	NULL	NULL

Books 12 ×

Query 3: Overdue loans

```
SELECT Loans.loan_id, Members.first_name, Members.last_name, Books.title,
due_date FROM Loans JOIN Members ON Loans.member_id =
Members.member_id JOIN Books ON Loans.book_id = Books.book_id WHERE
return_date IS NULL AND due_date < CURDATE();
```

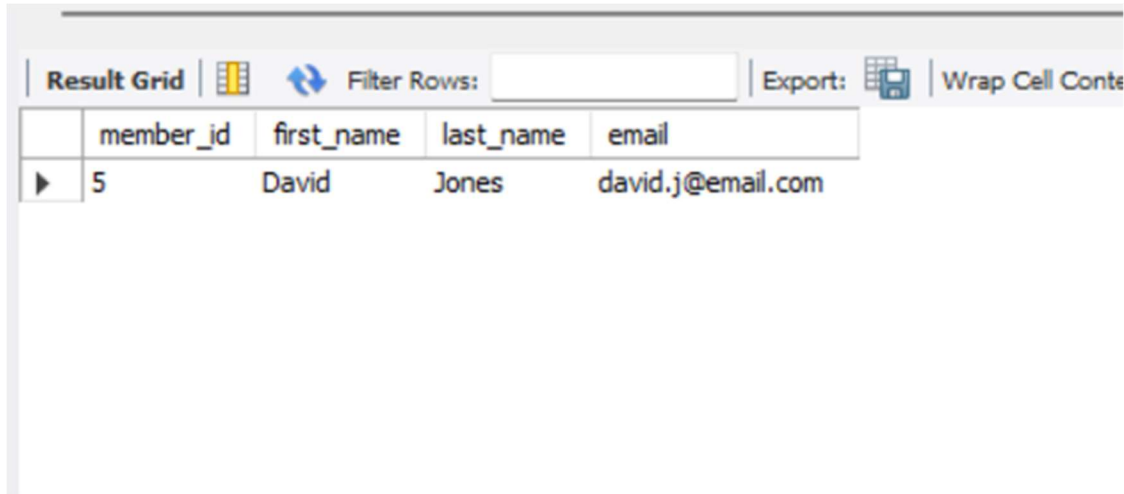
Output: Returns overdue loans for users like Sarah Brown.

Result Grid					
Filter Rows: <input type="text"/>					
Export: <input type="button" value="Export"/>					
Wrap Cell Content: <input type="button" value="Wrap"/>					
	loan_id	first_name	last_name	title	due_date
▶	4	Sarah	Brown	The Hobbit	2023-02-22
	6	Jennifer	Garcia	The Shining	2023-03-03
	8	Lisa	Davis	War and Peace	2023-02-26
	10	Jessica	Martinez	The Old Man and the Sea	2023-03-01
	12	Amanda	Lopez	Mrs Dalloway	2023-03-05
	14	Patricia	Wilson	One Hundred Years of Solitude	2023-03-08

Query 4: Members with unpaid fines

```
SELECT DISTINCT Members.member_id, first_name, last_name, email FROM  
Members JOIN Fines ON Members.member_id = Fines.member_id WHERE  
paid_status = FALSE;
```

Output: David Jones appears with unpaid fine.



The screenshot shows a database interface with a 'Result Grid' tab. The grid contains one row of data. The columns are labeled 'member_id', 'first_name', 'last_name', and 'email'. The data in the row is '5', 'David', 'Jones', and 'david.j@email.com' respectively. There are also buttons for 'Filter Rows', 'Export', and 'Wrap Cell Content'.

	member_id	first_name	last_name	email
▶	5	David	Jones	david.j@email.com

Query 5: Top 5 most borrowed books

```
SELECT Books.title, COUNT(Loans.loan_id) AS times_borrowed FROM Books  
JOIN Loans ON Books.book_id = Loans.book_id GROUP BY Books.book_id  
ORDER BY times_borrowed DESC LIMIT 5;
```

Output: Most borrowed book is “Harry Potter and the Philosopher’s Stone”.

Result Grid			Filter Rows:	Export:
	title	times_borrowed		
▶	Harry Potter and the Philosopher's Stone	2		
	1984	1		
	The Hobbit	1		
	Murder on the Orient Express	1		
	The Shining	1		

Result 15 x

Query 6: Members who borrowed more than 2 books

```
SELECT Members.member_id, first_name, last_name, COUNT(Loans.loan_id)
AS total_loans FROM Members JOIN Loans ON Members.member_id =
Loans.member_id GROUP BY Members.member_id HAVING total_loans > 2;
```

Output: John Smith borrowed more than 2 times.

Result Grid

Filter Rows:

Export:

	member_id	first_name	last_name	total_loans

Result Grid		
	book_id	title
▶	3	To Kill a Mockingbird
*	NULL	NULL

Books 17 x

Query 9: Count of members by type

```
SELECT membership_type, COUNT(*) AS total_members FROM Members
GROUP BY membership_type;
```

Output: Adult – 10, Senior – 3, Student – 1

Result Grid			Filter Rows:	Export
	membership_type	total_members		
▶	Adult	11		
	Senior	3		
	Student	1		

Result 22 x

Query 10: Total fine paid by each member

```
SELECT Members.member_id, first_name, last_name, SUM(fine_amount) AS
total_paid FROM Members JOIN Fines ON Members.member_id =
Fines.member_id WHERE paid_status = TRUE GROUP BY
Members.member_id;
```

Output: Michael Williams paid \$2.50; Daniel Hernandez paid \$2.00