

```

77 -- Delete inactive members (no borrow in last 1 year)
78 WHERE member_id NOT IN (
79     SELECT DISTINCT member_id FROM Transactions
80     WHERE borrow_date >= DATE_SUB(CURDATE(), INTERVAL 1 YEAR)
81 )
82 ;
83
84 -- Retrieve available books
85 • SELECT * FROM Books WHERE available_copies > 0;

```

**Result Grid**

book_id	title	author_id	category	isbn	published_date	price	available_copies
1	Malgudi Days	1	Fiction	1111	2010-05-10	450.00	5
2	Five Point Someone	2	Education	2222	2015-08-12	550.00	3
3	Wings of Fire	3	Science	3333	2020-01-01	300.00	3
4	Ignited Minds	3	Science	4444	2018-03-15	480.00	2
5	New Book	1	Fiction	5555	2022-01-01	600.00	6
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL

**Action Output**

#	Time	Action	Message
7	10:41:53	INSERT INTO Authors (name, email) VALUES ('R.K. Narayan','rk@email.com'), ('Chetan Bhagat','chetan@email.com')	3 row(s) affected Records: 3
8	10:41:59	INSERT INTO Books (title, author_id, category, isbn, published_date, price, available_copies) VALUES ('Malgudi Days',1,'Fiction','1111','2010-05-10',450.00,5)	4 row(s) affected Records: 4
9	10:42:06	INSERT INTO Members (name,email,phone_number,membership_date) VALUES ('Yash Patel','yash@email.co...','9898989898','2024-01-01')	3 row(s) affected Records: 3
10	10:42:14	INSERT INTO Transactions (member_id,book_id,borrow_date,return_date,fine_amount) VALUES (1,3,'2024-01-01','2024-01-15',10.00)	3 row(s) affected Records: 3
11	10:42:23	INSERT INTO Books (title, author_id, category, isbn, published_date, price, available_copies) VALUES ('New Book',1,'Fiction','5555','2022-01-01',600.00,6)	1 row(s) affected
12	10:42:29	UPDATE Books SET available_copies = available_copies - 1 WHERE book_id = 3	1 row(s) affected Rows matched

Books 1 ×

Output:

```

87 -- TASK 2: WHERE, HAVING, LIMIT
88
89 • SELECT * FROM Books WHERE YEAR(published_date) > 2015;
90
91 • SELECT * FROM Books ORDER BY price DESC LIMIT 5;
92
93 • SELECT * FROM Members WHERE YEAR(membership_date) < 2022;
94
95 -- TASK 3: AND, OR, NOT

```

**Result Grid**

book_id	title	author_id	category	isbn	published_date	price	available_copies
3	Wings of Fire	3	Science	3333	2020-01-01	300.00	3
4	Ignited Minds	3	Science	4444	2018-03-15	480.00	2
5	New Book	1	Fiction	5555	2022-01-01	600.00	6
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL

**Action Output**

#	Time	Action	Message
8	10:41:59	INSERT INTO Books (title, author_id, category, isbn, published_date, price, available_copies) VALUES ('Malgudi Days',1,'Fiction','1111','2010-05-10',450.00,5)	4 row(s) affected
9	10:42:06	INSERT INTO Members (name,email,phone_number,membership_date) VALUES ('Yash Patel','yash@email.co...','9898989898','2024-01-01')	3 row(s) affected
10	10:42:14	INSERT INTO Transactions (member_id,book_id,borrow_date,return_date,fine_amount) VALUES (1,3,'2024-01-01','2024-01-15',10.00)	3 row(s) affected

Books 2 ×

Output:

```

87 -- TASK 2: WHERE, HAVING, LIMIT
88
89 • SELECT * FROM Books WHERE YEAR(published_date) > 2015;
90
91 • SELECT * FROM Books ORDER BY price DESC LIMIT 5;
92
93 • SELECT * FROM Members WHERE YEAR(membership_date) < 2022;
94
95 -- TASK 3: AND, OR, NOT

```

**Books 3**

Output

Action Output

book_id	title	author_id	category	isbn	published_date	price	available_copies
5	New Book	1	Fiction	5555	2022-01-01	600.00	6
2	Five Point Someone	2	Education	2222	2015-08-12	550.00	3
4	Ignited Minds	3	Science	4444	2018-03-15	480.00	2
1	Malgudi Days	1	Fiction	1111	2010-05-10	450.00	5
3	Wings of Fire	3	Science	3333	2020-01-01	300.00	3
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Books 3 x

Context Help Snipp

```

87 -- TASK 2: WHERE, HAVING, LIMIT
88
89 • SELECT * FROM Books WHERE YEAR(published_date) > 2015;
90
91 • SELECT * FROM Books ORDER BY price DESC LIMIT 5;
92
93 • SELECT * FROM Members WHERE YEAR(membership_date) < 2022;
94
95 -- TASK 3: AND, OR, NOT

```

**Members 4**

Output

Action Output

member_id	name	email	phone_number	membership_date
2	Rahul Shah	rahul@email.com	8888888888	2021-06-20
*	HULL	HULL	HULL	HULL

Members 4 x

Context Help Snipp

```

87 -- TASK 2: WHERE, HAVING, LIMIT
88
89 • SELECT * FROM Books WHERE YEAR(published_date) > 2015;
90
91 • SELECT * FROM Books ORDER BY price DESC LIMIT 5;
92
93 • SELECT * FROM Members WHERE YEAR(membership_date) < 2022;
94
95 -- TASK 3: AND, OR, NOT

```

```

93 •  SELECT * FROM Members WHERE YEAR(membership_date) < 2022;
94 Save the script to a file.
-- TASK 3: AND, OR, NOT
96
97 •  SELECT * FROM Books
98 WHERE category = 'Science' AND price < 500;
99
100 •  SELECT * FROM Books
101 WHERE available_copies = 0;

```

**Books 5**

Output:

Action Output

book_id	title	author_id	category	isbn	published_date	price	available_copies
3	Wings of Fire	3	Science	3333	2020-01-01	300.00	3
4	Ignited Minds	3	Science	4444	2018-03-15	480.00	2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor |

Books 5 x Apply Revert

Message: 11 10:42:23 INSERT INTO Books (title, author\_id, category, isbn, published\_date, price, available\_copies) VALUES ('New ...') 1 row(s) affected

```

99
100 •  Execute the selected portion of the script or everything, if there is no selection
101 WHERE available_copies = 0;
102
103 •  SELECT * FROM Members
104 WHERE YEAR(membership_date) > 2020
105 OR member_id IN (
106     SELECT member_id FROM Transactions
107     GROUP BY member_id

```

**Books 6**

Output:

Action Output

book_id	title	author_id	category	isbn	published_date	price	available_copies
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor |

Books 6 x Apply Revert Context Help Snippets

```

102
103 •   SELECT * FROM Members
104 WHERE YEAR(membership_date) > 2020
105 OR member_id IN (
106     SELECT member_id FROM Transactions
107     GROUP BY member_id
108     HAVING COUNT(book_id) > 3
109 );
110

```

**Result Grid** | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | **Result Grid**

	member_id	name	email	phone_number	membership_date
▶	1	Yash Patel	yash@email.com	9999999999	2023-01-10
	2	Rahul Shah	rahul@email.com	8888888888	2021-06-20
	3	Priya Mehta	HULL	7777777777	2024-02-01
*	NULL	NULL	NULL	NULL	NULL

Members 7 ×

Apply

Revert

Output:

```

112 -- TASK 4: ORDER BY, GROUP BY
113
114 •   SELECT * FROM Books ORDER BY title ASC;
115
116 •   SELECT member_id, COUNT(book_id) AS total_books
117     FROM Transactions
118     GROUP BY member_id;
119
120 •   SELECT category, COUNT(*) AS total_books

```

**Result Grid** | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | **Result Grid**

	book_id	title	author_id	category	isbn	published_date	price	available_copies
▶	2	Five Point Someone	2	Education	2222	2015-08-12	550.00	3
	4	Ignited Minds	3	Science	4444	2018-03-15	480.00	2
	1	Malgudi Days	1	Fiction	1111	2010-05-10	450.00	5
	5	New Book	1	Fiction	5555	2022-01-01	600.00	6
	3	Wings of Fire	3	Science	3333	2020-01-01	300.00	3
*	NULL	NULL	HULL	NULL	NULL	NULL	NULL	NULL

Books 8 ×

Apply

Revert

Cont

Output:

Action Output

#	Time	Action	Message
✓	14 10:42:50	SELECT * FROM Books WHERE available_copies > 0 LIMIT 0, 1000	5 row(s) returned
✓	15 10:43:13	SELECT * FROM Books WHERE YEAR(published_date) > 2015 LIMIT 0, 1000	3 row(s) returned
✓	16 10:43:25	SELECT * FROM Books ORDER BY price DESC LIMIT 5	5 row(s) returned
✓	17 10:43:34	SELECT * FROM Members WHERE YEAR(membership_date) < 2022 LIMIT 0, 1000	1 row(s) returned
✓	18 10:43:47	SELECT * FROM Books WHERE category = 'Science' AND price < 500 LIMIT 0, 1000	2 row(s) returned
✓	19 10:44:00	SELECT * FROM Books WHERE available_copies = 0 LIMIT 0, 1000	0 row(s) returned
✓	20 10:44:12	SELECT * FROM Books WHERE YEAR(published_date) > 2020 OR published_date < 2010 SELECT * FROM Books WHERE available_copies > 0 LIMIT 0, 1000	2 row(s) returned

```

112      -- TASK 4: ORDER BY, GROUP BY
113
114 •  SELECT * FROM Books ORDER BY title ASC;
115
116 •  SELECT member_id, COUNT(book_id) AS total_books
117   FROM Transactions
118   GROUP BY member_id;
119
120 •  SELECT category, COUNT(*) AS total_books

```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

member_id	total_books
1	2
2	1

Result 9 Read Only

Output ::

Action Output

#	Time	Action	N
15	10:43:13	SELECT * FROM Books WHERE YEAR(published_date) > 2015 LIMIT 0, 1000	3
16	10:43:25	SELECT * FROM Books ORDER BY price DESC LIMIT 5	5
17	10:43:34	SELECT * FROM Members WHERE YEAR(membership_date) < 2022 LIMIT 0, 1000	1
18	10:43:47	SELECT * FROM Books WHERE category = 'Science' AND price < 500 LIMIT 0, 1000	2
19	10:44:00	SELECT * FROM Books WHERE available_copies = 0 LIMIT 0, 1000	0
20	10:44:12	SELECT * FROM Members WHERE YEAR(membership_date) > 2020 OR member_id IN (    SELECT membe... )	3
21	10:44:27	SELECT * FROM Books ORDER BY title ASC LIMIT 0, 1000	5

```

119 Execute the selected portion of the script or everything, if there is no selection
120 • SELECT category, COUNT(*) AS total_books
121   FROM Books
122   GROUP BY category;
123
124 -- TASK 5: AGGREGATE FUNCTIONS
125
126 • SELECT category, COUNT(*) FROM Books GROUP BY category;
127

```

**Result Grid** | Filter Rows: Export: Wrap Cell Content:

category	total_books
Fiction	2
Education	1
Science	2

Result Grid Form Editor

**Result 10** × Read Only Context I

Output:

Action Output

#	Time	Action	Message
16	10:43:25	SELECT * FROM Books ORDER BY price DESC LIMIT 5	5 row(s) returned
17	10:43:34	SELECT * FROM Members WHERE YEAR(membership_date) < 2022 LIMIT 0, 1000	1 row(s) returned
18	10:43:47	SELECT * FROM Books WHERE category = 'Science' AND price < 500 LIMIT 0, 1000	2 row(s) returned
19	10:44:00	SELECT * FROM Books WHERE available_copies = 0 LIMIT 0, 1000	0 row(s) returned
20	10:44:12	SELECT * FROM Members WHERE YEAR(membership_date) > 2020 OR member_id IN ( SELECT membe... )	3 row(s) returned
21	10:44:27	SELECT * FROM Books ORDER BY title ASC LIMIT 0, 1000	5 row(s) returned

```
125
126 •   SELECT category, COUNT(*) FROM Books GROUP BY category;
127
128 •   SELECT AVG(price) FROM Books;
129
130 •   SELECT book_id, COUNT(*) AS borrow_count
131     FROM Transactions
132     GROUP BY book_id
```

Result Grid | Filter Rows: [ ] | Export: [ ] | Wrap Cell Content: [ ]

Result Grid

category	COUNT(*)
Fiction	2
Education	1
Science	2

Form Editor

Result 11 × Read Only

Action Output

#	Time	Action	Message
17	10:43:34	SELECT * FROM Members WHERE YEAR(membership_date) < 2022 LIMIT 0, 1000	1 row(s) returned
18	10:43:47	SELECT * FROM Books WHERE category = 'Science' AND price < 500 LIMIT 0, 1000	2 row(s) returned
19	10:44:00	SELECT * FROM Books WHERE available_copies = 0 LIMIT 0, 1000	0 row(s) returned
20	10:44:12	SELECT * FROM Members WHERE YEAR(membership_date) > 2020 OR member_id IN ( SELECT membe... )	3 row(s) returned
21	10:44:27	SELECT * FROM Books ORDER BY title ASC LIMIT 0, 1000	5 row(s) returned
22	10:44:38	SELECT member_id, COUNT(book_id) AS total_books FROM Transactions GROUP BY member_id LIMIT 0, 1...	2 row(s) returned
23	10:44:51	SELECT category, COUNT(*) AS total_books FROM Books GROUP BY category LIMIT 0, 1000	3 row(s) returned
24	10:45:16	SELECT category, COUNT(*) FROM Books GROUP BY category LIMIT 0, 1000	3 row(s) returned

Query 1 pr1.Data digger pr2.data transformer pr.finel.university management SQL File 6\*

124 -- TASK 5: AGGREGATE FUNCTIONS  
125 Execute the selected portion of the script or everything, if there is no selection  
126 • SELECT category, COUNT(\*) FROM Books GROUP BY category;  
127  
128 • SELECT AVG(price) FROM Books;  
129  
130 • SELECT book\_id, COUNT(\*) AS borrow\_count  
131 FROM Transactions  
132 GROUP BY book\_id

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid Form Editor

	AVG(price)
▶	476.000000

Result 12 × Read Only Co

Output:

Action Output

#	Time	Action	Message
18	10:43:47	SELECT * FROM Books WHERE category = 'Science' AND price < 500 LIMIT 0, 1000	2 row(s) returned
19	10:44:00	SELECT * FROM Books WHERE available_copies = 0 LIMIT 0, 1000	0 row(s) returned
20	10:44:12	SELECT * FROM Members WHERE YEAR(membership_date) > 2020 OR member_id IN ( SELECT membe... )	3 row(s) returned
21	10:44:27	SELECT * FROM Books ORDER BY title ASC LIMIT 0, 1000	5 row(s) returned
22	10:44:38	SELECT member_id, COUNT(book_id) AS total_books FROM Transactions GROUP BY member_id LIMIT 0, 1...	2 row(s) returned
23	10:44:51	SELECT category, COUNT(*) AS total_books FROM Books GROUP BY category LIMIT 0, 1000	3 row(s) returned
24	10:45:16	SELECT category, COUNT(*) FROM Books GROUP BY category LIMIT 0, 1000	3 row(s) returned
25	10:45:26	SELECT AVG(price) FROM Books LIMIT 0, 1000	1 row(s) returned

```
129
130 •   SELECT book_id, COUNT(*) AS borrow_count
131     FROM Transactions
132     GROUP BY book_id
133     ORDER BY borrow_count DESC
134     LIMIT 1;
135
136 •   SELECT SUM(fine_amount) FROM Transactions;
137
```

---

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	book_id	borrow_count			
▶	1	1			

Result 13 x

Output ::::::::::::

Action Output

#	Time	Action
19	10:44:00	SELECT * FROM Books WHERE available_copies = 0 LIMIT 0, 1000
20	10:44:12	SELECT * FROM Members WHERE YEAR(membership_date) > 2020 OR member_id IN (
21	10:44:27	SELECT * FROM Books ORDER BY title ASC LIMIT 0, 1000
22	10:44:38	SELECT member_id, COUNT(book_id) AS total_books FROM Transactions GROUP BY member_id
23	10:44:51	SELECT category, COUNT(*) AS total_books FROM Books GROUP BY category LIMIT 0, 1000

```
135
136 • SE Execute the selected portion of the script or everything, if there is no selection
137
138
139 -- TASK 6: PRIMARY & FOREIGN KEY
140
141
142 -- Already established during table creation
143
```

---

Result Grid | Filter Rows: Export: Wrap Cell Content: □

SUM(fine_amount)
50.00

Result 14 × Read Only

Output:

Action Output

#	Time	Action	Message
✓ 20	10:44:12	SELECT * FROM Members WHERE YEAR(membership_date) > 2020 OR member_id IN ( SELECT membe...	3 row(s)
✓ 21	10:44:27	SELECT * FROM Books ORDER BY title ASC LIMIT 0, 1000	5 row(s)
✓ 22	10:44:38	SELECT member_id, COUNT(book_id) AS total_books FROM Transactions GROUP BY member_id LIMIT 0, 1...	2 row(s)
✓ 23	10:44:51	SELECT category, COUNT(*) AS total_books FROM Books GROUP BY category LIMIT 0, 1000	3 row(s)
✓ 24	10:45:16	SELECT category, COUNT(*) FROM Books GROUP BY category LIMIT 0, 1000	3 row(s)
✓ 25	10:45:26	SELECT AVG(price) FROM Books LIMIT 0, 1000	1 row(s)
✓ 26	10:45:40	SELECT book_id, COUNT(*) AS borrow_count FROM Transactions GROUP BY book_id ORDER BY borrow...	1 row(s)

```

147 • SELECT b.title, a.name
148   Execute the selected portion of the script or everything, if there is no selection
149   INNER JOIN Authors a ON b.author_id = a.author_id;
150
151   -- LEFT JOIN
152 • SELECT m.name, t.book_id
153   FROM Members m
154   LEFT JOIN Transactions t ON m.member_id = t.member_id;
155

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor

	title	name
▶	Malgudi Days	R.K. Narayan
	New Book	R.K. Narayan
	Five Point Someone	Chetan Bhagat
	Wings of Fire	APJ Abdul Kalam
	Ignited Minds	APJ Abdul Kalam

Result 15 × Read Only Con

Output:

Action Output

#	Time	Action	Message
21	10:44:27	SELECT * FROM Books ORDER BY title ASC LIMIT 0, 1000	5 row(s) returned
22	10:44:38	SELECT member_id, COUNT(book_id) AS total_books FROM Transactions GROUP BY member_id LIMIT 0, 1...	2 row(s) returned
23	10:44:51	SELECT category, COUNT(*) AS total_books FROM Books GROUP BY category LIMIT 0, 1000	3 row(s) returned
24	10:45:16	SELECT category, COUNT(*) FROM Books GROUP BY category LIMIT 0, 1000	3 row(s) returned
25	10:45:26	SELECT AVG(price) FROM Books LIMIT 0, 1000	1 row(s) returned

```
147 •  SELECT b.title, a.name
148    FROM Books b
149    INNER JOIN Authors a ON b.author_id = a.author_id;
150
151    -- LEFT JOIN
152 •  SELECT m.name, t.book_id
153    FROM Members m
154    LEFT JOIN Transactions t ON m.member_id = t.member_id;
155
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	name	book_id
▶	Yash Patel	3
	Yash Patel	4
	Rahul Shah	1
	Priya Mehta	NULL

Result 16 × Read Only

Output:

Action Output

#	Time	Action	M
✓ 22	10:44:38	SELECT member_id, COUNT(book_id) AS total_books FROM Transactions GROUP BY member_id LIMIT 0, 1...	21
✓ 23	10:44:51	SELECT category, COUNT(*) AS total_books FROM Books GROUP BY category LIMIT 0, 1000	31
✓ 24	10:45:16	SELECT category, COUNT(*) FROM Books GROUP BY category LIMIT 0, 1000	31
✓ 25	10:45:26	SELECT AVG(price) FROM Books LIMIT 0, 1000	11
✓ 26	10:45:40	SELECT book_id, COUNT(*) AS borrow_count FROM Transactions GROUP BY book_id ORDER BY borrow_...	11

```
156      -- RIGHT JOIN
157 •   SELECT b.title, t.transaction_id
158     FROM Books b
159     RIGHT JOIN Transactions t ON b.book_id = t.book_id
160
161      -- FULL OUTER JOIN (MySQL alternative using UNION)
162 •   SELECT m.name, t.book_id
163     FROM Members m
164     LEFT JOIN Transactions t ON m.member_id = t.member_id
```

Result Grid		Filter Rows:	Export:	Wrap Cell
title	transaction_id			
Malgudi Days	2			
Wings of Fire	1			
Ignited Minds	3			

Action Output		
#	Time	Action
23	10:44:51	SELECT category, COUNT(*) AS total_books FROM Books GROUP BY category
24	10:45:16	SELECT category, COUNT(*) FROM Books GROUP BY category
25	10:45:26	SELECT AVG(price) FROM Books LIMIT 0, 1000
26	10:45:40	SELECT book_id, COUNT(*) AS borrow_count FROM Transactions GROUP BY book_id
27	10:45:53	SELECT SUM(fine_amount) FROM Transactions LIMIT 0, 1000
28	10:46:18	SELECT b.title, a.name FROM Books b INNER JOIN Authors a ON b.author_id = a.id
29	10:46:28	SELECT m.name, t.book_id FROM Members m LEFT JOIN Transactions t ON m.member_id = t.member_id
30	10:46:45	SELECT b.title, t.transaction_id FROM Books b RIGHT JOIN Transactions t ON b.book_id = t.book_id

```

156    -- RIGHT JOIN
157 •  SELECT b.title, t.transaction_id
158   FROM Books b
159   RIGHT JOIN Transactions t ON b.book_id = t.book_id;
160
161    -- FULL OUTER JOIN (MySQL alternative using UNION)
162 •  SELECT m.name, t.book_id
163   FROM Members m
164   LEFT JOIN Transactions t ON m.member_id = t.member_id

```

**Result Grid** | Filter Rows: Export: Wrap Cell Content: Result Grid | Form Editor

	title	transaction_id
▶	Malgudi Days	2
	Wings of Fire	1
	Ignited Minds	3

Result 17 × Read Only

Output

Action Output

#	Time	Action	Message
23	10:44:51	SELECT category, COUNT(*) AS total_books FROM Books GROUP BY category LIMIT 0, 1000	3 row(s) re
24	10:45:16	SELECT category, COUNT(*) FROM Books GROUP BY category LIMIT 0, 1000	3 row(s) re
25	10:45:26	SELECT AVG(price) FROM Books LIMIT 0, 1000	1 row(s) re
26	10:45:40	SELECT book_id, COUNT(*) AS borrow_count FROM Transactions GROUP BY book_id ORDER BY borrow_...	1 row(s) re
27	10:45:53	SELECT SUM(fine_amount) FROM Transactions LIMIT 0, 1000	1 row(s) re
28	10:46:18	SELECT b.title, a.name FROM Books b INNER JOIN Authors a ON b.author_id = a.author_id LIMIT 0, 1000	5 row(s) re
29	10:46:28	SELECT m.name, t.book_id FROM Members m LEFT JOIN Transactions t ON m.member_id = t.member_id LIM... 4 row(s) re	
30	10:46:45	SELECT b.title, t.transaction_id FROM Books b RIGHT JOIN Transactions t ON b.book_id = t.book_id LIMIT 0... 3 row(s) re	

```

161 -- FULL OUTER JOIN (MySQL alternative using UNION)
162 • SELECT m.name, t.book_id
163   FROM Members m
164   LEFT JOIN Transactions t ON m.member_id = t.member_id
165 UNION
166 SELECT m.name, t.book_id
167   FROM Members m
168   RIGHT JOIN Transactions t ON m.member_id = t.member_id;
169

```

Result Grid | Filter Rows: \_\_\_\_\_ | Export: \_\_\_\_\_ | Wrap Cell Content:

	name	book_id
▶	Yash Patel	4
	Yash Patel	3
	Rahul Shah	1
	Priya Mehta	NULL

Result Grid | Form Editor

Result 18 × Read Only Context

Action Output

#	Time	Action	Message
✓ 24	10:45:16	SELECT category, COUNT(*) FROM Books GROUP BY category LIMIT 0, 1000	3 row(s) returned
✓ 25	10:45:26	SELECT AVG(price) FROM Books LIMIT 0, 1000	1 row(s) returned
✓ 26	10:45:40	SELECT book_id, COUNT(*) AS borrow_count FROM Transactions GROUP BY book_id ORDER BY borrow_...	1 row(s) returned
✓ 27	10:45:53	SELECT SUM(fine_amount) FROM Transactions LIMIT 0, 1000	1 row(s) returned
✓ 28	10:46:18	SELECT b.title, a.name FROM Books b INNER JOIN Authors a ON b.author_id = a.author_id LIMIT 0, 1000	5 row(s) returned
✓ 29	10:46:28	SELECT m.name, t.book_id FROM Members m LEFT JOIN Transactions t ON m.member_id = t.member_id LIM...	4 row(s) returned
✓ 30	10:46:45	SELECT b.title, t.transaction_id FROM Books b RIGHT JOIN Transactions t ON b.book_id = t.book_id LIMIT 0...	3 row(s) returned
✓ 31	10:47:18	SELECT m.name, t.book_id FROM Members m LEFT JOIN Transactions t ON m.member_id = t.member_id UN...	4 row(s) returned

```

171
172 •  SELECT * FROM Books
173 Ⓜ WHERE book_id IN (
174     SELECT book_id FROM Transactions
175     WHERE member_id IN (
176         SELECT member_id FROM Members
177         WHERE YEAR(membership_date) > 2022
178     )
179 );

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | R

	book_id	title	author_id	category	isbn	published_date	price	available_copies
▶	3	Wings of Fire	3	Science	3333	2020-01-01	300.00	3
▶	4	Ignited Minds	3	Science	4444	2018-03-15	480.00	2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Books 19 x

Output:

Action Output

#	Time	Action	Me
25	10:45:26	SELECT AVG(price) FROM Books LIMIT 0, 1000	1 ro
26	10:45:40	SELECT book_id, COUNT(*) AS borrow_count FROM Transactions GROUP BY book_id ORDER BY borrow_...	1 ro
27	10:45:53	SELECT SUM(fine_amount) FROM Transactions LIMIT 0, 1000	1 ro
28	10:46:18	SELECT b.title, a.name FROM Books b INNER JOIN Authors a ON b.author_id = a.author_id LIMIT 0, 1000	5 ro
29	10:46:28	SELECT m.name, t.book_id FROM Members m LEFT JOIN Transactions t ON m.member_id = t.member_id LIM...	4 ro
30	10:46:45	SELECT b.title, t.transaction_id FROM Books b RIGHT JOIN Transactions t ON b.book_id = t.book_id LIMIT 0...	3 ro
31	10:47:10	SELECT m.name, t.book_id FROM Members m LEFT JOIN Transactions t ON m.member_id = t.member_id LIM...	4 ro

```

180
181 • Execute the selected portion of the script or everything, if there is no selection
182 WHERE book_id =
183     SELECT book_id FROM Transactions
184     GROUP BY book_id
185     ORDER BY COUNT(*) DESC
186     LIMIT 1
187 );
188

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

book_id	title	author_id	category	isbn	published_date	price	available_copies
1	Malgudi Days	1	Fiction	1111	2010-05-10	450.00	5
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Result Grid | Form Editor

Books 20 x

Output

Action Output

#	Time	Action	Message
26	10:45:40	SELECT book_id, COUNT(*) AS borrow_count FROM Transactions GROUP BY book_id ORDER BY borrow_...	1 row(s) returned
27	10:45:53	SELECT SUM(fine_amount) FROM Transactions LIMIT 0, 1000	1 row(s) returned
28	10:46:18	SELECT b.title, a.name FROM Books b INNER JOIN Authors a ON b.author_id = a.author_id LIMIT 0, 1000	5 row(s) returned
29	10:46:28	SELECT m.name, t.book_id FROM Members m LEFT JOIN Transactions t ON m.member_id = t.member_id LIM...	4 row(s) returned
30	10:46:45	SELECT b.title, t.transaction_id FROM Books b RIGHT JOIN Transactions t ON b.book_id = t.book_id LIMIT 0...	3 row(s) returned
31	10:47:18	SELECT m.name, t.book_id FROM Members m LEFT JOIN Transactions t ON m.member_id = t.member_id UN...	4 row(s) returned
32	10:47:36	SELECT * FROM Books WHERE book_id IN (    SELECT book_id FROM Transactions WHERE member_i...	2 row(s) returned
33	10:47:56	SELECT * FROM Books WHERE book_id = (    SELECT book_id FROM Transactions GROUP BY book_id...	1 row(s) returned

```

188
189 •   SELECT * FROM Members
190     WHERE member_id NOT IN (
191         SELECT DISTINCT member_id FROM Transactions
192     );
193
194     -- TASK 9: DATE FUNCTIONS
195
196 •   SELECT YEAR(published_date), COUNT(*)

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	member_id	name	email	phone_number	membership_date
▶	3	Priya Mehta	NULL	7777777777	2024-02-01
*	NULL	NULL	NULL	NULL	NULL

Result Grid | Form Editor

Members 21 x Apply Revert Context Help

Output:

Action Output

#	Time	Action	Message
27	10:45:53	SELECT SUM(fine_amount) FROM Transactions LIMIT 0, 1000	1 row(s) returned
28	10:46:18	SELECT b.title, a.name FROM Books b INNER JOIN Authors a ON b.author_id = a.author_id LIMIT 0, 1000	5 row(s) returned
29	10:46:28	SELECT m.name, t.book_id FROM Members m LEFT JOIN Transactions t ON m.member_id = t.member_id LIMIT 0, 1000	4 row(s) returned
30	10:46:45	SELECT b.title, t.transaction_id FROM Books b RIGHT JOIN Transactions t ON b.book_id = t.book_id LIMIT 0, 1000	3 row(s) returned
31	10:47:18	SELECT m.name, t.book_id FROM Members m LEFT JOIN Transactions t ON m.member_id = t.member_id UNION ALL	4 row(s) returned
32	10:47:36	SELECT * FROM Books WHERE book_id IN (    SELECT book_id FROM Transactions    WHERE member_id = 3 )	2 row(s) returned
33	10:47:56	SELECT * FROM Books WHERE book_id = (    SELECT book_id FROM Transactions    GROUP BY book_id )	1 row(s) returned

195
196 • SELECT SUM(fine\_amount) FROM Transactions LIMIT 0, 1000
197 FROM Books
198 GROUP BY YEAR(published\_date);
199
200 • SELECT DATEDIFF(return\_date, borrow\_date) AS days\_difference
201 FROM Transactions;
202
203 • SELECT DATE\_FORMAT(borrow\_date, '%d-%m-%Y')

Automatic context help for this page

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

YEAR(published_date)	COUNT(*)
2010	1
2015	1
2020	1
2018	1
2022	1

Result Grid | Form Editor

Result 22 x Read Only Context Help Snippets

Output:

Action Output

#	Time	Action	Message
28	10:46:18	SELECT b.title, a.name FROM Books b INNER JOIN Authors a ON b.author_id = a.author_id LIMIT 0, 1000	5 row(s) returned
29	10:46:28	SELECT m.name, t.book_id FROM Members m LEFT JOIN Transactions t ON m.member_id = t.member_id LIMIT 0, 1000	4 row(s) returned
30	10:46:45	SELECT b.title, t.transaction_id FROM Books b RIGHT JOIN Transactions t ON b.book_id = t.book_id LIMIT 0, 1000	3 row(s) returned

```
195
196 •   SELECT YEAR(published_date), COUNT(*)
197     FROM Books
198     GROUP BY YEAR(published_date);
199
200 •   SELECT DATEDIFF(return_date, borrow_date) AS days_difference
201     FROM Transactions;
202
203 •   SELECT DATE_FORMAT(borrow_date, '%d-%m-%Y')
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	days_difference			
▶	9			
	15			
	NULL			

Result 23 ×			Read
Output			
Action Output			
#	Time	Action	
29	10:46:28	SELECT m.name, t.book_id FROM Members m LEFT JOIN Transactions t ON m.member_id = t.member_id	
30	10:46:45	SELECT b.title, t.transaction_id FROM Books b RIGHT JOIN Transactions t ON b.book_id = t.book_id LIMIT 1	
31	10:47:18	SELECT m.name, t.book_id FROM Members m LEFT JOIN Transactions t ON m.member_id = t.member_id	
32	10:47:36	SELECT * FROM Books WHERE book_id IN (    SELECT book_id FROM Transactions    WHERE member_id = 1 )	
33	10:47:56	SELECT * FROM Books WHERE book_id = (    SELECT book_id FROM Transactions    WHERE member_id = 1 ) GROUP BY book_id	

202     Execute the selected portion of the script or everything, if there is no selection

203 •     SELECT DATE\_FORMAT(borrow\_date, '%d-%m-%Y')  
           FROM Transactions;

204

205     -- TASK 10: STRING FUNCTIONS

206

208 •     SELECT UPPER(title) FROM Books;

209

210 •     SELECT TRIM(name) FROM Authors;

---

**Result Grid** | Filter Rows:  Export: Wrap Cell Content:

	DATE_FORMAT(borrow_date, '%d-%m-%Y')
▶	01-01-2024
	05-05-2023
	01-02-2024

Result Grid  
 Form Editor

Result 24 × Read Only Context Help

Output :::::

Action Output

#	Time	Action	Message
30	10:46:45	SELECT b.title,t.transaction_id FROM Books b RIGHT JOIN Transactions t ON b.book_id = t.book_id LIMIT 0...	3 row(s) returned
31	10:47:18	SELECT m.name,t.book_id FROM Members m LEFT JOIN Transactions t ON m.member_id = t.member_id UN...	4 row(s) returned
32	10:47:36	SELECT * FROM Books WHERE book_id IN ( SELECT book_id FROM Transactions WHERE member_i...	2 row(s) returned
33	10:47:56	SELECT * FROM Books WHERE book_id = ( SELECT book_id FROM Transactions GROUP BY book_id...	1 row(s) returned
34	10:48:11	SELECT * FROM Members WHERE member_id NOT IN ( SELECT DISTINCT member_id FROM Transactio...	1 row(s) returned
35	10:48:25	SELECT YEAR(borrow_date), COUNT(*) FROM Books GROUP BY YEAR(borrow_date) LIMIT 0, 1000	5 rows(s) returned

```
202
203 • SE Execute the selected portion of the script or everything, if there is no selection
204 FROM Transactions;
205
206 -- TASK 10: STRING FUNCTIONS
207
208 • SELECT UPPER(title) FROM Books;
209
210 • SELECT TRIM(name) FROM Authors;
```

Result Grid | Filter Rows: \_\_\_\_\_ | Export: | Wrap Cell Content: |

Result Grid | Form Editor

Result 25 × Read Only

Action Output

#	Time	Action	Message
31	10:47:18	SELECT m.name, t.book_id FROM Members m LEFT JOIN Transactions t ON m.member_id = t.member_id UNION SELECT * FROM Books WHERE book_id IN ( SELECT book_id FROM Transactions WHERE member_id = m.member_id )	4 row(s) returned
32	10:47:36	SELECT * FROM Books WHERE book_id IN ( SELECT book_id FROM Transactions WHERE member_id = m.member_id )	2 row(s) returned
33	10:47:56	SELECT * FROM Books WHERE book_id = ( SELECT book_id FROM Transactions GROUP BY book_id )	1 row(s) returned
34	10:48:11	SELECT * FROM Members WHERE member_id NOT IN ( SELECT DISTINCT member_id FROM Transactions )	1 row(s) returned
35	10:48:25	SELECT YEAR(published_date), COUNT(*) FROM Books GROUP BY YEAR(published_date) LIMIT 0, 1000	5 row(s) returned

```
202
203 • S Execute the selected portion of the script or everything, if there is no selection
204 FROM Transactions;
205
206 -- TASK 10: STRING FUNCTIONS
207
208 • SELECT UPPER(title) FROM Books;
209
210 • SELECT TRIM(name) FROM Authors;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid

TRIM(name)
R.K. Narayan
Chetan Bhagat
APJ Abdul Kalam

Result 26 × Read Only

Output:

Action Output

#	Time	Action	Message
32	10:47:36	SELECT * FROM Books WHERE book_id IN (    SELECT book_id FROM Transactions    WHERE member_i...)	2 row(s)
33	10:47:56	SELECT * FROM Books WHERE book_id = (    SELECT book_id FROM Transactions    GROUP BY book_id...)	1 row(s)
34	10:48:11	SELECT * FROM Members WHERE member_id NOT IN (    SELECT DISTINCT member_id FROM Transactio...)	1 row(s)
35	10:48:25	SELECT YEAR(published_date), COUNT(*) FROM Books GROUP BY YEAR(published_date) LIMIT 0, 1000	5 row(s)
36	10:48:35	SELECT DATEDIFF(return_date, borrow_date) AS days_difference FROM Transactions LIMIT 0, 1000	3 row(s)
37	10:48:50	SELECT DATE_FORMAT(borrow_date,"%d-%m-%Y") FROM Transactions LIMIT 0, 1000	3 row(s)

```

211
212 • SI Execute the selected portion of the script or everything, if there is no selection
213
214 -- TASK 11: WINDOW FUNCTIONS
215
216 • SELECT book_id,
217 COUNT(*) AS borrow_count,
218 RANK() OVER (ORDER BY COUNT(*) DESC) AS rank_position
219 FROM Transactions

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid

IFNULL(email,'Not Provided')
▶ yash@email.com
rahul@email.com
Not Provided

Result 27 × Read Only

Output:

Action Output

#	Time	Action	Message
33	10:47:56	SELECT * FROM Books WHERE book_id = ( SELECT book_id FROM Transactions GROUP BY book_id... )	1 row(s)
34	10:48:11	SELECT * FROM Members WHERE member_id NOT IN ( SELECT DISTINCT member_id FROM Transactions )	1 row(s)
35	10:48:25	SELECT YEAR(published_date), COUNT(*) FROM Books GROUP BY YEAR(published_date) LIMIT 0, 1000	5 row(s)
36	10:48:35	SELECT DATEDIFF(return_date, borrow_date) AS days_difference FROM Transactions LIMIT 0, 1000	3 row(s)
37	10:48:50	SELECT DATE_FORMAT(borrow_date,"%d-%m-%Y") FROM Transactions LIMIT 0, 1000	3 row(s)
38	10:49:01	SELECT UPPER(title) FROM Books LIMIT 0, 1000	5 row(s)
39	10:49:09	SELECT TRIM(name) FROM Authors LIMIT 0, 1000	3 row(s)
40	10:49:24	SELECT IFNULL(email,'Not Provided') FROM Members LIMIT 0, 1000	3 row(s)

```

216 •   SELECT book_id,
217     Execute the selected portion of the script or everything, if there is no selection
218     RANK() OVER (ORDER BY COUNT(*) DESC) AS rank_position
219   FROM Transactions
220   GROUP BY book_id;
221
222 •   SELECT
223       member_id,
224       borrow_date,

```

**Result Grid** | Filter Rows: | Export: | Wrap Cell Content: | **Result Grid** | **Form Editor**

book_id	borrow_count	rank_position
1	1	1
3	1	1
4	1	1

**Result 28** × **Read Only** Context

**Action Output**

#	Time	Action	Message
34	10:48:11	SELECT * FROM Members WHERE member_id NOT IN ( SELECT DISTINCT member_id FROM Transactions )	1 row(s) returned
35	10:48:25	SELECT YEAR(published_date), COUNT(*) FROM Books GROUP BY YEAR(published_date) LIMIT 0, 1000	5 row(s) returned
36	10:48:35	SELECT DATEDIFF(return_date, borrow_date) AS days_difference FROM Transactions LIMIT 0, 1000	3 row(s) returned
37	10:48:50	SELECT DATE_FORMAT(borrow_date, '%d-%m-%Y') FROM Transactions LIMIT 0, 1000	3 row(s) returned
38	10:49:01	SELECT UPPER(title) FROM Books LIMIT 0, 1000	5 row(s) returned
39	10:49:09	SELECT TRIM(name) FROM Authors LIMIT 0, 1000	3 row(s) returned

```

253 •   SELECT
254       title,
255       published_date,
256       CASE
257           WHEN EXTRACT(YEAR FROM published_date) > 2020 THEN 'New Arrival'
258           WHEN EXTRACT(YEAR FROM published_date) < 2000 THEN 'Classic'
259           ELSE 'Regular'
260       END AS book_category
261   FROM Books;

```

**Result Grid** | Filter Rows: | Export: | Wrap Cell Content: | **Result Grid**

title	published_date	book_category
Malgudi Days	2010-05-10	Regular
Five Point Someone	2015-08-12	Regular
Wings of Fire	2020-01-01	Regular
Ignited Minds	2018-03-15	Regular
New Book	2022-01-01	New Arrival

```

213      -- TASK 11: WINDOW FUNCTIONS
214
215 •   SELECT
216     book_id,
217     COUNT(transaction_id) AS borrow_count,
218     RANK() OVER (ORDER BY COUNT(transaction_id) DESC) AS borrow_rank
219   FROM Transactions
220   GROUP BY book_id;
221

```

	book_id	borrow_count	borrow_rank
▶	1	1	1
	3	1	1
	4	1	1

### Result 33 ×

```

222 •   SELECT
223     member_id,
224     borrow_date,
225     COUNT(*) OVER (PARTITION BY member_id ORDER BY borrow_date) AS cumulative_borrows
226   FROM Transactions;
227
228
229 •   SELECT

```

	month	monthly_count	moving_avg
▶	1	1	1.0000
	2	1	1.0000
	5	1	1.0000

### Result 34 ×

Output:

Action Output			Message
#	Time	Action	
✖ 51	11:06:47	WITH MonthlyBorrows AS ( SELECT DATE_TRUNC('month', borrow_date) as borrow_month, COUNT(*) AS monthly_count )	Error Code: 1146. Table 'smartlibrary.borrowed_book' doesn't exist.
✖ 52	11:08:25	WITH MonthlyBorrows AS ( SELECT DATE_FORMAT(borrow_date, '%Y-%m-01') AS borrow_month, COUNT(*) AS monthly_count )	Error Code: 1146. Table 'smartlibrary.borrowed_book' doesn't exist.
✖ 53	11:10:11	SELECT member_id, borrow_date, COUNT(*) OVER (PARTITION BY member_id ORDER BY borrow_date)	Error Code: 1146. Table 'smartlibrary.borrowed_book' doesn't exist.
✓ 54	11:10:29	SELECT book_id, COUNT(*) AS borrow_count, RANK() OVER (ORDER BY COUNT(*) DESC) AS rank_position	3 row(s) returned
✖ 55	11:10:57	SELECT member_id, borrow_date, COUNT(*) OVER (PARTITION BY member_id ORDER BY borrow_date) AS monthly_count	Error Code: 1146. Table 'smartlibrary.borrowed_book' doesn't exist.
✓ 56	11:19:01	SELECT member_id, borrow_date, COUNT(*) OVER (PARTITION BY member_id ORDER BY borrow_date) AS monthly_count	3 row(s) returned

```

233     FROM Transactions;
234     Execute the selected portion of the script or everything, if there is no selection
235 #task 12 - SQLCASE EXPRESSION
236
237 • SELECT m.name,
238     CASE
239         WHEN MAX(t.borrow_date) >= DATE_SUB(CURDATE(), INTERVAL 6 MONTH) THEN 'Active'
240         ELSE 'Inactive'
241     END AS Membership_Status

```

Result Grid | Filter Rows: \_\_\_\_\_ | Export: \_\_\_\_\_ | Wrap Cell Content: \_\_\_\_\_

	name	Membership_Status
▶	Yash Patel	Inactive
	Rahul Shah	Inactive
	Priya Mehta	Inactive

Result 37 ×

Output :

Action Output

#	Time	Action	Message
56	11:19:01	SELECT member_id, borrow_date, COUNT(*) OVER (PARTITION BY member_id ORDER BY borrow...	3 row(s) returned
239		WHEN MAX(t.borrow_date) >= DATE_SUB(CURDATE(), INTERVAL 6 MONTH) THEN 'Active'	Execute the selected portion of the script or everything, if there is no selection
240		ELSE 'Inactive'	
241		END AS Membership_Status	
242		FROM Members m	
243		LEFT JOIN Transactions t ON m.member_id = t.member_id	
244		GROUP BY m.member_id, m.name;	
245		-- 2. Categorize books based on publication year	
246			
247		SELECT	

Result Grid | Filter Rows: \_\_\_\_\_ | Export: \_\_\_\_\_ | Wrap Cell Content: \_\_\_\_\_

	title	published_date	Book_Category
▶	Malgudi Days	2010-05-10	Regular
	Five Point Someone	2015-08-12	Regular
	Wings of Fire	2020-01-01	Regular
	Ignited Minds	2018-03-15	Regular
	New Book	2022-01-01	New Arrival

Result 38 ×

Output :

Action Output

#	Time	Action	Message
57	11:19:28	SELECT book_id, COUNT(transaction_id) AS borrow_count, RANK() OVER (ORDER BY COUNT(tr...	3 row(s) returned
58	11:20:11	MONTH(borrow_date) AS month, COUNT(transaction_id) AS monthly_count, AVG(COUNT(tr...	3 row(s) returned