

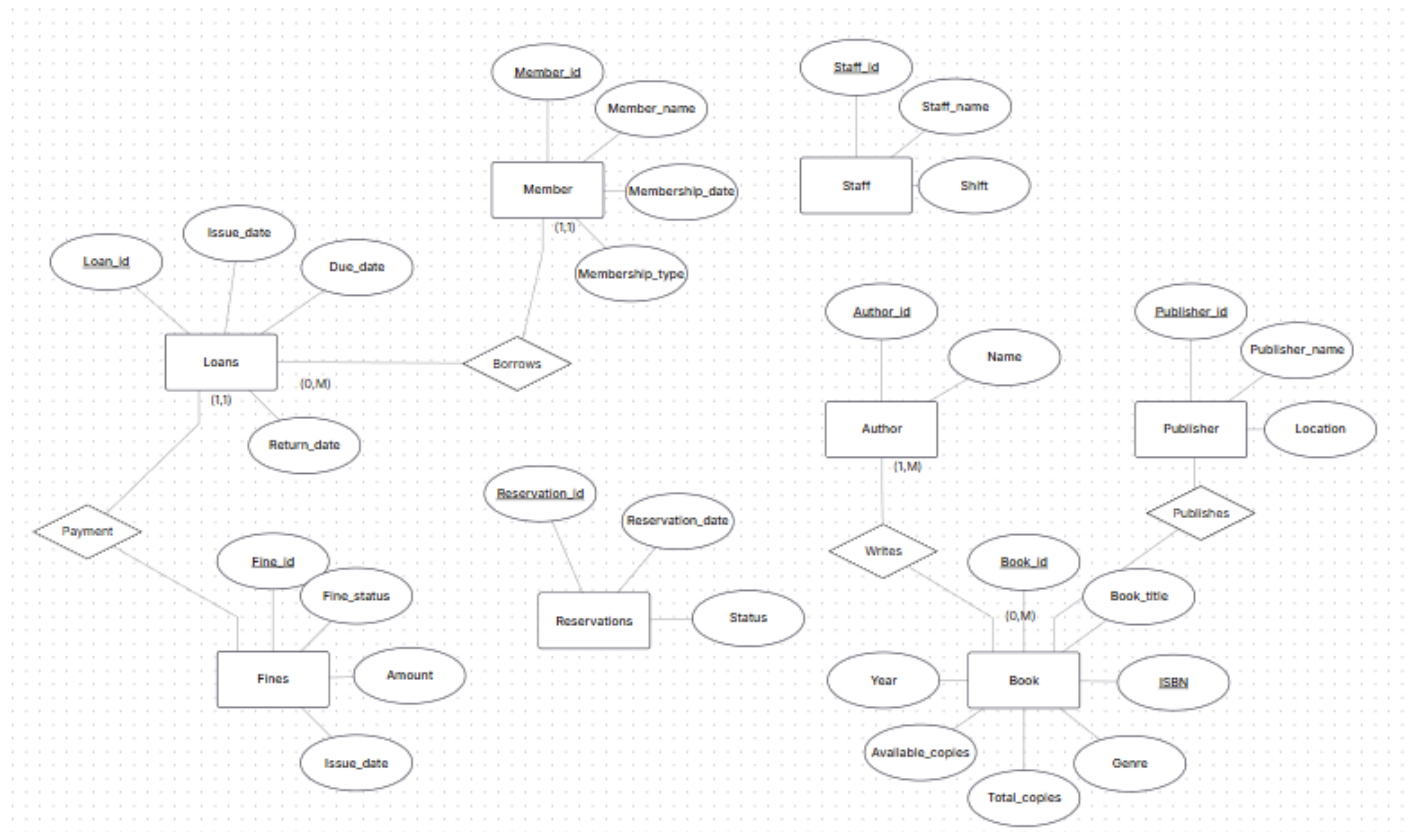
Case Study 3: City Central Library Management System

Main Document

717823P352

717823P247

ER DIAGRAM



SQL Code With Screenshots

Author

```
CREATE TABLE Author (  
  Author_id VARCHAR(20) PRIMARY KEY,  
  Name VARCHAR(100) NOT NULL  
);
```

```
INSERT INTO Author(Author_id,Name) VALUES  
( 'A001', 'J.K. Rowling'),  
( 'A002', 'Niccolò Machiavelli'),  
( 'A003', 'Agatha Christie'),  
( 'A004', 'Yuval Noah Harari'),  
( 'A005', 'Chetan Bhagat'),  
( 'A006', 'George Orwell'),  
( 'A007', 'J.R.R. Tolkien'),  
( 'A008', 'Dan Brown'),  
( 'A009', 'Stephen King'),  
( 'A010', 'Paulo Coelho');
```

```
SELECT * FROM Author;
```

Result Grid			Filter Rows:
	Author_id	Name	
▶	A001	J.K. Rowling	
	A002	Niccolò Machiavelli	
	A003	Agatha Christie	
	A004	Yuval Noah Harari	
	A005	Chetan Bhagat	
	A006	George Orwell	
	A007	J.R.R. Tolkien	
	A008	Dan Brown	
	A009	Stephen King	
	A010	Paulo Coelho	
•	NULL	NULL	

Publisher

```
CREATE TABLE Publisher (
Publisher_id VARCHAR(20) PRIMARY KEY,
Publisher_name VARCHAR(100) NOT NULL,
Location VARCHAR(255)
);
```

```
INSERT INTO Publisher (Publisher_id, Publisher_name, Location) VALUES
('P001', 'Penguin Random House', 'New York, USA'),
('P002', 'HarperCollins', 'London, UK'),
('P003', 'Bloomsbury Publishing', 'London, UK'),
('P004', 'Scholastic', 'Washington, USA'),
('P005', 'Macmillan Publishers', 'Berlin, Germany'),
('P006', 'Oxford University Press', 'Oxford, UK'),
('P007', 'Simon & Schuster', 'New York, USA'),
('P008', 'Hachette Livre', 'Paris, France'),
('P009', 'Random House', 'Berlin, Germany'),
('P010', 'Minotaur Books', 'New York, USA');
```

```
SELECT * FROM Publisher;
```

Result Grid				Filter Rows:	Edit:
	Publisher_id	Publisher_name	Location		
▶	P001	Penguin Random House	New York, USA		
	P002	HarperCollins	London, UK		
	P003	Bloomsbury Publishing	London, UK		
	P004	Scholastic	Washington, USA		
	P005	Macmillan Publishers	Berlin, Germany		
	P006	Oxford University Press	Oxford, UK		
	P007	Simon & Schuster	New York, USA		
	P008	Hachette Livre	Paris, France		
	P009	Random House	Berlin, Germany		
	P010	Minotaur Books	New York, USA		
•	NULL	NULL	NULL		

Book

```
CREATE TABLE Book (
Book_id VARCHAR(20) PRIMARY KEY,
```

```

Book_title VARCHAR(150) NOT NULL,
ISBN VARCHAR(30) UNIQUE,
Genre VARCHAR(50),
Year INT,
Total_copies INT,
Available_copies INT,
Author_id VARCHAR(20),
Publisher_id VARCHAR(20),
FOREIGN KEY (Author_id) REFERENCES Author(Author_id),
FOREIGN KEY (Publisher_id) REFERENCES Publisher(Publisher_id)
);

```

```

INSERT INTO Book (Book_id, Book_title, ISBN, Genre, Year, Total_copies, Available_copies, Author_id, Publisher_id) VALUES
('B001', 'Harry Potter and the Chamber of Secrets', '9780439064873', 'Fantasy', 1998, 10, 8, 'A001', 'P001'),
('B002', 'Pokemon Legends ZA', '9780439136365', 'Fantasy', 1999, 10, 7, 'A001', 'P003'),
('B003', 'The Art of War', '9780140449198', 'Politics', 1521, 5, 3, 'A002', 'P002'),
('B004', 'Discourses on Livy', '9780199536399', 'Politics', 1531, 5, 4, 'A002', 'P002'),
('B005', 'And Then There Were None', '9780062073488', 'Mystery', 1939, 6, 5, 'A003', 'P001'),
('B006', 'The ABC Murders', '9780007136834', 'Mystery', 1936, 5, 4, 'A003', 'P002'),
('B007', 'Homo Deus', '9780062464316', 'History', 2016, 7, 5, 'A004', 'P004'),
('B008', '21 Lessons for the 21st Century', '9780525512172', 'History', 2018, 6, 6, 'A004', 'P004'),
('B009', '2 States', '9788129135568', 'Romance', 2009, 4, 4, 'A005', 'P005'),
('B010', 'One Night Call Center', '9788129135575', 'Science fiction', 2014, 5, 5, 'A005', 'P005');

```

```
SELECT * FROM Book;
```

Book_id	Book_title	ISBN	Genre	Year	Total_copies	Available_copies	Author_id	Publisher_id
B001	Harry Potter and the Chamber of Secrets	9780439064873	Fantasy	1998	10	8	A001	P001
B002	Pokemon Legends ZA	9780439136365	Fantasy	1999	10	7	A001	P003
B003	The Art of War	9780140449198	Politics	1521	5	3	A002	P002
B004	Discourses on Livy	9780199536399	Politics	1531	5	4	A002	P002
B005	And Then There Were None	9780062073488	Mystery	1939	6	5	A003	P001
B006	The ABC Murders	9780007136834	Mystery	1936	5	4	A003	P002
B007	Homo Deus	9780062464316	History	2016	7	5	A004	P004
B008	21 Lessons for the 21st Century	9780525512172	History	2018	6	6	A004	P004
B009	2 States	9788129135568	Romance	2009	4	4	A005	P005
B010	One Night Call Center	9788129135575	Science fiction	2014	5	5	A005	P005
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Member

```

CREATE TABLE Member (
Member_id VARCHAR(20) PRIMARY KEY,
Member_name VARCHAR(100) NOT NULL,
Membership_type VARCHAR(20),
Membership_date DATE,
No_of_books_borrowed INT);

```

```

INSERT INTO Member (Member_id, Member_name, Membership_type, Membership_date, No_of_books_borrowed) VALUES
('M001', 'Revathi Priya', 'Gold', '2025-11-01', 5),
('M002', 'Yogendra', 'Silver', '2025-09-15', 10),
('M003', 'Sasmitha', 'Basic', '2024-12-05', 7),
('M004', 'Arun', 'Silver', '2024-12-18', 8),
('M005', 'Praveen', 'Gold', '2025-01-10', 5),
('M006', 'Barani', 'Gold', '2025-02-01', 9),
('M007', 'Ramlath Nisha', 'Silver', '2025-02-10', 3),
('M008', 'Kavya', 'Basic', '2025-03-05', 7),
('M009', 'Prousika', 'Gold', '2025-03-12', 9),
('M010', 'Sowmi', 'Silver', '2025-04-01', 11);

```

SELECT * FROM Member;

Result Grid					
Filter Rows:		Edit:		Export/Import:	
Member_id	Member_name	Membership_type	Membership_date	No_of_books_borrowed	
M001	Revathi Priya	Gold	2025-11-01	5	
M002	Yogendra	Silver	2025-09-15	10	
M003	Sasmitha	Basic	2024-12-05	7	
M004	Arun	Silver	2024-12-18	8	
M005	Praveen	Gold	2025-01-10	5	
M006	Barani	Gold	2025-02-01	9	
M007	Ramlath Nisha	Silver	2025-02-10	3	
M008	Kavya	Basic	2025-03-05	7	
M009	Prousika	Gold	2025-03-12	9	
M010	Sowmi	Silver	2025-04-01	11	
NULL	NULL	NULL	NULL	NULL	

Loan

```
CREATE TABLE Loan (  
    Loan_id VARCHAR(20) PRIMARY KEY,  
    Member_id VARCHAR(20) NOT NULL,  
    Book_id VARCHAR(20) NOT NULL,  
    Issue_date DATE NULL,  
    Due_date DATE NULL,  
    Return_date DATE NULL,  
    FOREIGN KEY (Member_id) REFERENCES Member(Member_id),  
    FOREIGN KEY (Book_id) REFERENCES Book(Book_id)  
);
```

```
INSERT INTO Loan (Loan_id, Member_id, Book_id, Issue_date, Due_date, Return_date) VALUES  
( 'L001', 'M001', 'B001', '2025-01-10', '2025-11-24', NULL),  
( 'L002', 'M002', 'B004', '2025-11-05', '2025-11-20', '2025-11-18'),  
( 'L003', 'M003', 'B002', NULL, NULL, NULL),  
( 'L004', 'M004', 'B005', '2025-12-02', '2025-12-22', NULL),  
( 'L005', 'M005', 'B005', '2025-01-15', '2025-01-30', '2025-01-29'),  
( 'L006', 'M006', 'B006', '2025-02-10', '2025-02-25', NULL),  
( 'L007', 'M007', 'B002', '2025-03-01', '2025-03-15', '2025-03-14'),  
( 'L008', 'M008', 'B008', '2025-03-05', '2025-03-20', NULL),  
( 'L009', 'M009', 'B009', '2025-04-10', '2025-04-25', NULL),  
( 'L010', 'M010', 'B005', '2025-04-12', '2025-04-28', '2025-04-26'),  
( 'L011', 'M001', 'B004', '2025-05-01', '2025-05-16', NULL),  
( 'L012', 'M002', 'B002', '2025-05-05', '2025-05-20', '2025-05-18'),  
( 'L013', 'M003', 'B003', '2025-05-08', '2025-05-25', NULL),  
( 'L014', 'M004', 'B001', '2025-06-10', '2025-06-30', NULL),  
( 'L015', 'M005', 'B005', '2025-06-12', '2025-07-01', '2025-06-30');
```

SELECT * FROM Loan;

Result Grid						
Filter Rows:				Edit:		
	Loan_id	Member_id	Book_id	Issue_date	Due_date	Return_date
▶	L001	M001	B001	2025-01-10	2025-11-24	NULL
	L002	M002	B004	2025-11-05	2025-11-20	2025-11-18
	L003	M003	B002	NULL	NULL	NULL
	L004	M004	B005	2025-12-02	2025-12-22	NULL
	L005	M005	B005	2025-01-15	2025-01-30	2025-01-29
	L006	M006	B006	2025-02-10	2025-02-25	NULL
	L007	M007	B002	2025-03-01	2025-03-15	2025-03-14
	L008	M008	B008	2025-03-05	2025-03-20	NULL
	L009	M009	B009	2025-04-10	2025-04-25	NULL
	L010	M010	B005	2025-04-12	2025-04-28	2025-04-26
	L011	M001	B004	2025-05-01	2025-05-16	NULL
	L012	M002	B002	2025-05-05	2025-05-20	2025-05-18
	L013	M003	B003	2025-05-08	2025-05-25	NULL
	L014	M004	B001	2025-06-10	2025-06-30	NULL
	L015	M005	B005	2025-06-12	2025-07-01	2025-06-30
✱	NULL	NULL	NULL	NULL	NULL	NULL

Staff

```
CREATE TABLE Staff (
  Staff_id INT PRIMARY KEY,
  Staff_name VARCHAR(100),
  Shift VARCHAR(50)
);
```

```
INSERT INTO Staff (Staff_id,Staff_name,Shift) VALUES
(1, 'Deepa', 'Night'),
(2, 'VijayaBaskar', 'Morning'),
(3, 'Gopishanker', 'Evening'),
(4, 'Vivek', 'Night'),
(5, 'Jeevitha', 'Morning'),
(6, 'Karthik', 'Morning'),
(7, 'Shalini', 'Evening'),
(8, 'Suresh', 'Night'),
(9, 'Meenakshi', 'Morning'),
(10, 'Raghul', 'Evening');
```

```
SELECT * FROM Staff;
```


Result Grid			
Filter Rows:			
	Staff_id	Staff_name	Shift
▶	1	Deepa	Night
	2	VijayaBaskar	Morning
	3	Gopishanker	Evening
	4	Vivek	Night
	5	Jeevitha	Morning
	6	Karthik	Morning
	7	Shalini	Evening
	8	Suresh	Night
	9	Meenakshi	Morning
	10	Raghul	Evening
•	NULL	NULL	NULL

Reservation

```
CREATE TABLE Reservation (
Reservation_id VARCHAR(20) PRIMARY KEY,
Member_id VARCHAR(20),
Book_id VARCHAR(20),
Reservation_date DATE,
Status VARCHAR(30),
FOREIGN KEY (Member_id) REFERENCES Member(Member_id),
FOREIGN KEY (Book_id) REFERENCES Book(Book_id)
);
```

```
INSERT INTO Reservation (Reservation_id, Member_id, Book_id, Reservation_date, Status) VALUES
('R001', 'M001', 'B002', '2025-02-05', 'Pending'),
('R002', 'M003', 'B001', '2025-01-17', 'Approved'),
('R003', 'M002', 'B003', '2025-01-20', 'Cancelled'),
('R004', 'M004', 'B004', '2025-02-02', 'Pending'),
('R005', 'M005', 'B005', '2025-01-25', 'Approved'),
('R006', 'M006', 'B006', '2025-02-12', 'Pending'),
('R007', 'M007', 'B007', '2025-02-15', 'Approved'),
('R008', 'M008', 'B008', '2025-03-01', 'Cancelled'),
('R009', 'M009', 'B009', '2025-03-05', 'Pending'),
('R010', 'M010', 'B010', '2025-03-10', 'Approved');
```

```
SELECT * FROM Reservation;
```

Result Grid					
Filter Rows:					
	Reservation_id	Member_id	Book_id	Reservation_date	Status
▶	R001	M001	B002	2025-02-05	Pending
	R002	M003	B001	2025-01-17	Approved
	R003	M002	B003	2025-01-20	Cancelled
	R004	M004	B004	2025-02-02	Pending
	R005	M005	B005	2025-01-25	Approved
	R006	M006	B006	2025-02-12	Pending
	R007	M007	B007	2025-02-15	Approved
	R008	M008	B008	2025-03-01	Cancelled
	R009	M009	B009	2025-03-05	Pending
	R010	M010	B010	2025-03-10	Approved
•	NULL	NULL	NULL	NULL	NULL

Fine

```
CREATE TABLE Fine (  
Fine_id VARCHAR(20) PRIMARY KEY,  
Member_id VARCHAR(20),  
Amount DECIMAL(10,2),  
Issue_date DATE,  
Fine_status VARCHAR(30),  
FOREIGN KEY (Member_id) REFERENCES Member(Member_id)  
);
```

```
INSERT INTO Fine (Fine_id, Member_id, Amount, Issue_date, Fine_status) VALUES  
(F001, 'M002', 150.00, '2025-10-05', 'Pending'),  
(F002, 'M001', 20.00, '2025-09-10', 'Paid'),  
(F003, 'M003', 75.00, '2025-12-01', 'Pending'),  
(F004, 'M005', 10.00, '2025-07-15', 'Paid'),  
(F005, 'M004', 30.00, '2025-12-29', 'Pending'),  
(F006, 'M006', 50.00, '2025-07-12', 'Pending'),  
(F007, 'M007', 25.00, '2025-03-15', 'Paid'),  
(F008, 'M008', 40.00, '2025-07-05', 'Pending'),  
(F009, 'M009', 60.00, '2025-10-12', 'Paid'),  
(F010, 'M010', 35.00, '2025-03-20', 'Pending');
```

```
SELECT * FROM Fine;
```

	Fine_id	Member_id	Amount	Issue_date	Fine_status
▶	F001	M002	150.00	2025-10-05	Pending
	F002	M001	20.00	2025-09-10	Paid
	F003	M003	75.00	2025-12-01	Pending
	F004	M005	10.00	2025-07-15	Paid
	F005	M004	30.00	2025-12-29	Pending
	F006	M006	50.00	2025-07-12	Pending
	F007	M007	25.00	2025-03-15	Paid
	F008	M008	40.00	2025-07-05	Pending
	F009	M009	60.00	2025-10-12	Paid
	F010	M010	35.00	2025-03-20	Pending
*	NULL	NULL	NULL	NULL	NULL

Query Results With Explanation

Basic Queries

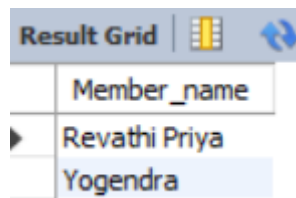
1. List all books published by "Penguin Random House".

```
SELECT Book_title FROM Book  
WHERE Publisher_id=(  
SELECT Publisher_id FROM Publisher  
WHERE Publisher_name='Penguin Random House');
```

	Book_title
▶	Harry Potter and the Chamber of Secrets
	And Then There Were None

2. Find members who joined in the last 3 months.

```
SELECT Member_name FROM Member
WHERE Membership_date >= '2025-09-01';
```

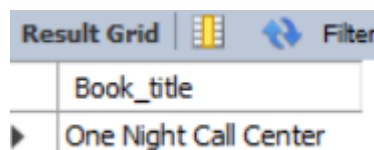


The screenshot shows a 'Result Grid' with a header row containing 'Member_name'. Below the header, there are two rows of data: 'Revathi Priya' and 'Yogendra'. The 'Yogendra' row is highlighted in blue.

Member_name
Revathi Priya
Yogendra

3. Retrieve books classified under the "Science Fiction" genre.

```
SELECT Book_title FROM Book
WHERE Genre='Science Fiction';
```

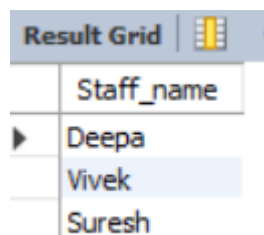


The screenshot shows a 'Result Grid' with a header row containing 'Book_title'. Below the header, there is one row of data: 'One Night Call Center'.

Book_title
One Night Call Center

4. List staff members who work on the "Night Shift".

```
SELECT Staff_name FROM Staff
WHERE Shift = 'Night';
```

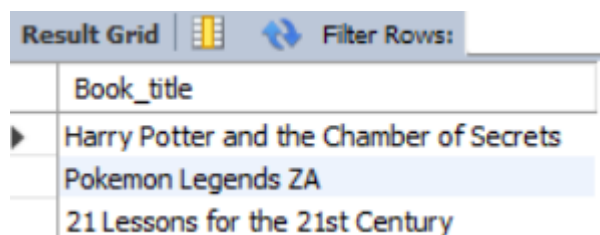


The screenshot shows a 'Result Grid' with a header row containing 'Staff_name'. Below the header, there are three rows of data: 'Deepa', 'Vivek', and 'Suresh'. The 'Vivek' row is highlighted in blue.

Staff_name
Deepa
Vivek
Suresh

5. Find books that have more than 5 copies in stock.

```
SELECT Book_title FROM Book
WHERE Available_copies > '5';
```



The screenshot shows a 'Result Grid' with a header row containing 'Book_title'. Below the header, there are three rows of data: 'Harry Potter and the Chamber of Secrets', 'Pokemon Legends ZA', and '21 Lessons for the 21st Century'. The 'Pokemon Legends ZA' row is highlighted in blue.

Book_title
Harry Potter and the Chamber of Secrets
Pokemon Legends ZA
21 Lessons for the 21st Century

Joins & Subqueries

6. Display member names along with the titles of books they have currently borrowed

```
SELECT Member.Member_name, Book.Book_title
FROM Loan
JOIN Member ON Loan.Member_id = Member.Member_id
JOIN Book ON Loan.Book_id = Book.Book_id
WHERE Loan.Return_date IS NULL;
```

Result Grid	Filter Rows:	Export:
Member_name	Book_title	
▶ Revathi Priya	Harry Potter and the Chamber of Secrets	
Sasmitha	Pokemon Legends ZA	
Arun	And Then There Were None	
Barani	The ABC Murders	
Kavya	21 Lessons for the 21st Century	
Prousika	2 States	
Revathi Priya	Discourses on Livy	
Sasmitha	The Art of War	
Arun	Harry Potter and the Chamber of Secrets	

7. Find members who have accumulated fines greater than \$50.

```
SELECT Member.Member_name
FROM Fine
JOIN Member ON Fine.Member_id = Member.Member_id
WHERE Amount>50;
```

Result Grid	Filter Rows:	Export:
Member_name		
▶ Yogendra		
Sasmitha		
Prousika		

8. List authors who have written books in both "History" and "Politics"

```
SELECT Author.Name
FROM Book
JOIN Author ON Book.Author_id=Author.Author_id
WHERE Genre IN ('History','Politics');
```

Result Grid	Filter Rows:	Export:
Name		
▶ Niccolò Machiavelli		
Niccolò Machiavelli		
Yuval Noah Harari		
Yuval Noah Harari		

9. Identify books that have been reserved but not yet collected.

```
SELECT Book_title
FROM Book
WHERE Book_id IN (
SELECT Book_id FROM Reservation
WHERE Status='Pending' );
```

Result Grid	Filter
Book_title	
Pokemon Legends ZA	
Discourses on Livy	
The ABC Murders	
2 States	

10. List members who have never borrowed a book.

```
SELECT Member_name
FROM Member
WHERE Member_id IN (
SELECT Member_id FROM Loan
WHERE Issue_date IS NULL);
```

Result Grid	Filter
Member_name	
Sasmitha	

Aggregation & Reports

11. Top 5 most borrowed books of the year.

```
SELECT Book.Book_id, Book.Book_title, COUNT(*) AS Borrow_Count
FROM Loan
JOIN Book ON Loan.Book_id = Book.Book_id
GROUP BY Book.Book_id, Book.Book_title
ORDER BY Borrow_Count DESC
LIMIT 5;
```

Result Grid	Filter Rows:	Export:	Wra
Book_id	Book_title	Borrow_Count	
B005	And Then There Were None	4	
B002	Pokemon Legends ZA	3	
B001	Harry Potter and the Chamber of Secrets	2	
B004	Discourses on Livy	2	
B003	The Art of War	1	

12. Total fines collected per month.

```
SELECT MONTH(Issue_date) AS Month,
SUM(Amount) AS Total_Fines FROM Fine
GROUP BY MONTH(Issue_date)
ORDER BY Month;
```

Result Grid	Filter
Month	Total_Fines
3	60.00
7	100.00
9	20.00
10	210.00
12	105.00

13. Identify the author with the highest number of books in the library.

```
SELECT Author.Name,  
COUNT(Book.Book_id) AS Total_Books  
FROM Book  
JOIN Author ON Author.Author_id = Book.Author_id  
GROUP BY Author.Author_id, Author.Name  
ORDER BY Total_Books DESC  
LIMIT 1;
```

Result Grid			Filter Rows
	Name	Total_Books	
▶	J.K. Rowling	2	

14. Average number of days a book is kept by members

```
SELECT b.Book_title,AVG(DATEDIFF(l.Return_date,l.Issue_date)) AS Difference  
FROM Loan l,Book b  
WHERE l.Book_id=b.Book_id  
GROUP BY b.Book_title;
```

Result Grid			Filter Rows:	Exp
	Book_title	Difference		
▶	Harry Potter and the Chamber of Secrets	NULL		
	Pokemon Legends ZA	13.0000		
	The Art of War	NULL		
	Discourses on Livy	13.0000		
	And Then There Were None	15.3333		
	The ABC Murders	NULL		
	21 Lessons for the 21st Century	NULL		
	2 States	NULL		

15. Genres with the highest rate of late returns.

```
UPDATE Loan  
SET Return_date = '2025-12-18'  
WHERE Loan_id = 'L002';  
  
SELECT b.Genre,  
COUNT(b.genre) AS Late_Returns  
FROM Loan l JOIN Book b ON l.Book_id = b.Book_id  
WHERE l.Return_date IS NOT NULL AND l.Return_date > l.Due_date  
GROUP BY b.Genre  
ORDER BY Late_Returns DESC;
```

Result Grid			Filter Rows
	Genre	Late_Returns	
▶	Politics	1	

17. List members who have borrowed more books than the average member.


```
SELECT m.Member_id, m.Member_name, COUNT(l.Loan_id) AS Total_Borrowed  
FROM Member m
```

```

JOIN Loan l ON m.Member_id = l.Member_id
GROUP BY m.Member_id, m.Member_name
HAVING COUNT(l.Loan_id) > (
SELECT AVG(Borrow_Count)
FROM (
SELECT COUNT(*) AS Borrow_Count FROM Loan
GROUP BY Member_id) AS BorrowStats);

```

Result Grid

 Filter Rows:

	Member_id	Member_name	Total_Borrowed
▶	M001	Revathi Priya	2
	M002	Yogendra	2
	M003	Sasmitha	2
	M004	Arun	2
	M005	Praveen	2

18. Identify books that have not been borrowed in the last 2 years.

```

SELECT Book_id, Book_title
FROM Book
WHERE Book_id NOT IN (
SELECT Book_id FROM Loan
WHERE Issue_date >='2023-11-01');

```

Result Grid	Filter Rows:
Book_id	Book_title
B007	Homo Deus
B010	One Night Call Center
NULL	NULL

19. Find members who have paid all their fines (No pending dues)

```

SELECT m.Member_id, m.Member_name
FROM Member m
JOIN Fine f ON f.Member_id=m.Member_id
WHERE f.Fine_status = 'Pending';

```

Result Grid	Filter Rows:
Member_id	Member_name
M002	Yogendra
M003	Sasmitha
M004	Arun
M006	Barani
M008	Kavya
M010	Sowmi

Indexing & Query Optimization

1. Create indexes on isbn, member_id, and due_date.

```
CREATE INDEX idx_book_isbn ON Book(ISBN);
CREATE INDEX idx_member_id ON Member(Member_id);
CREATE INDEX idx_loan_due_date ON Loan(due_date);
```

```
SHOW INDEXES FROM Book;
SHOW INDEXES FROM Member;
SHOW INDEXES FROM Loan;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
book	0	PRIMARY	1	Book_id	A	10	NULL	NULL	NULL	BTREE			YES	NULL
book	0	ISBN	1	ISBN	A	10	NULL	NULL	YES	BTREE			YES	NULL
book	1	Author_id	1	Author_id	A	5	NULL	NULL	YES	BTREE			YES	NULL
book	1	Publisher_id	1	Publisher_id	A	4	NULL	NULL	YES	BTREE			YES	NULL
book	1	idx_book_isbn	1	ISBN	A	10	NULL	NULL	YES	BTREE			YES	NULL

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
member	0	PRIMARY	1	Member_id	A	10	NULL	NULL	NULL	BTREE			YES	NULL
member	1	idx_member_id	1	Member_id	A	10	NULL	NULL	NULL	BTREE			YES	NULL

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
loan	0	PRIMARY	1	Loan_id	A	15	NULL	NULL	NULL	BTREE			YES	NULL
loan	1	Member_id	1	Member_id	A	10	NULL	NULL	NULL	BTREE			YES	NULL
loan	1	Book_id	1	Book_id	A	8	NULL	NULL	NULL	BTREE			YES	NULL
loan	1	idx_loan_due_date	1	Due_date	A	15	NULL	NULL	YES	BTREE			YES	NULL

2. Compare execution plans with vs without indexes.

WITHOUT INDEX

```
SELECT * FROM Book
WHERE ISBN = '9780062073488';
```

WITH INDEX

```
ALTER TABLE Book DROP INDEX idx_book_isbn;

CREATE INDEX idx_book_isbn ON Book(ISBN);
SELECT * FROM Book WHERE ISBN = '9780062073488';
```

Book_id	Book_title	ISBN	Genre	Year	Total_copies	Available_copies	Author_id	Publisher_id
B005	And Then There Were None	9780062073488	Mystery	1939	6	5	A003	P002

Book_id	Book_title	ISBN	Genre	Year	Total_copies	Available_copies	Author_id	Publisher_id
B005	And Then There Were None	9780062073488	Mystery	1939	6	5	A003	P002

3. Rewrite 2 queries (from the list above) into optimized versions.

OPTIMIZED VERSION

```
1) SELECT b.Book_title
FROM Book b
JOIN Publisher p ON b.Publisher_id = p.Publisher_id
WHERE p.Publisher_name = 'Penguin Random House';
```

```
2) SELECT Member_name  
FROM Member  
WHERE Membership_date >= '2025-09-01';
```

```
+-----+  
| Book_title |  
+-----+  
| Discourses on Livy |  
+-----+  
+-----+  
| Member_name |  
+-----+  
| Revathi Priya |  
| Yogendra |  
+-----+
```