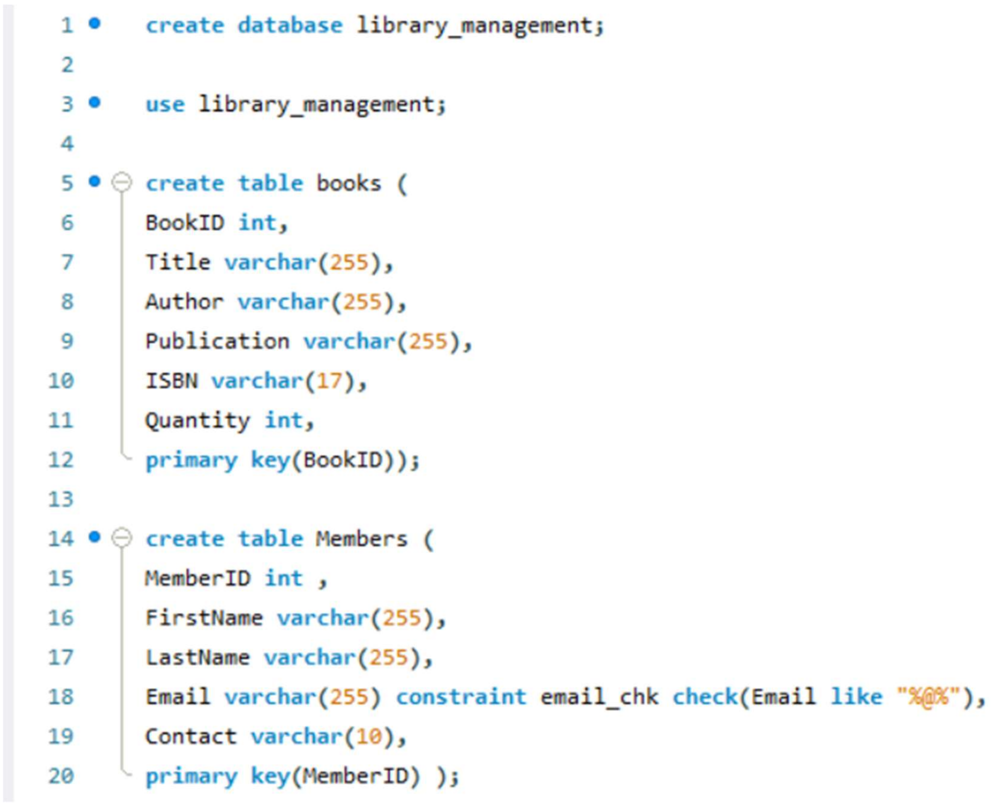
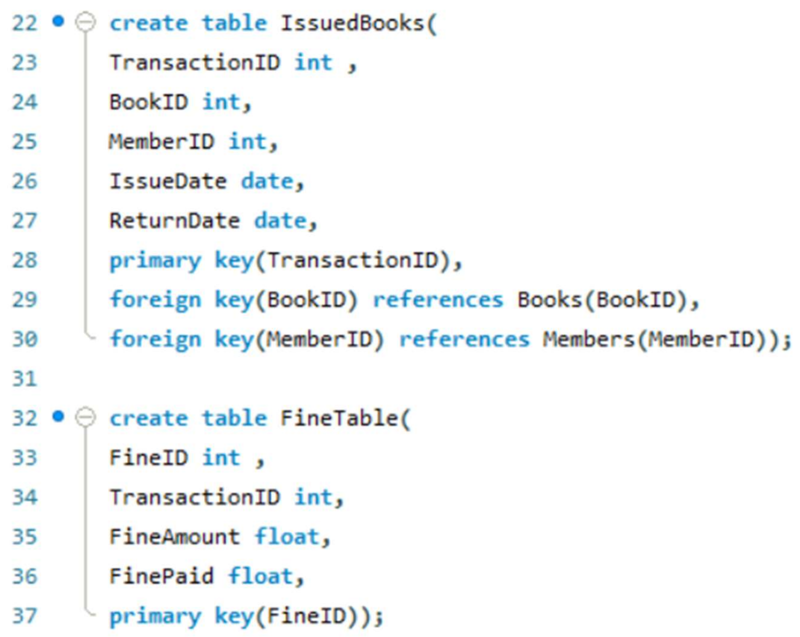
Implementing SQL Queries

1. Write queries to create all the given tables i.e. books, members, issuedbooks, and finetable.

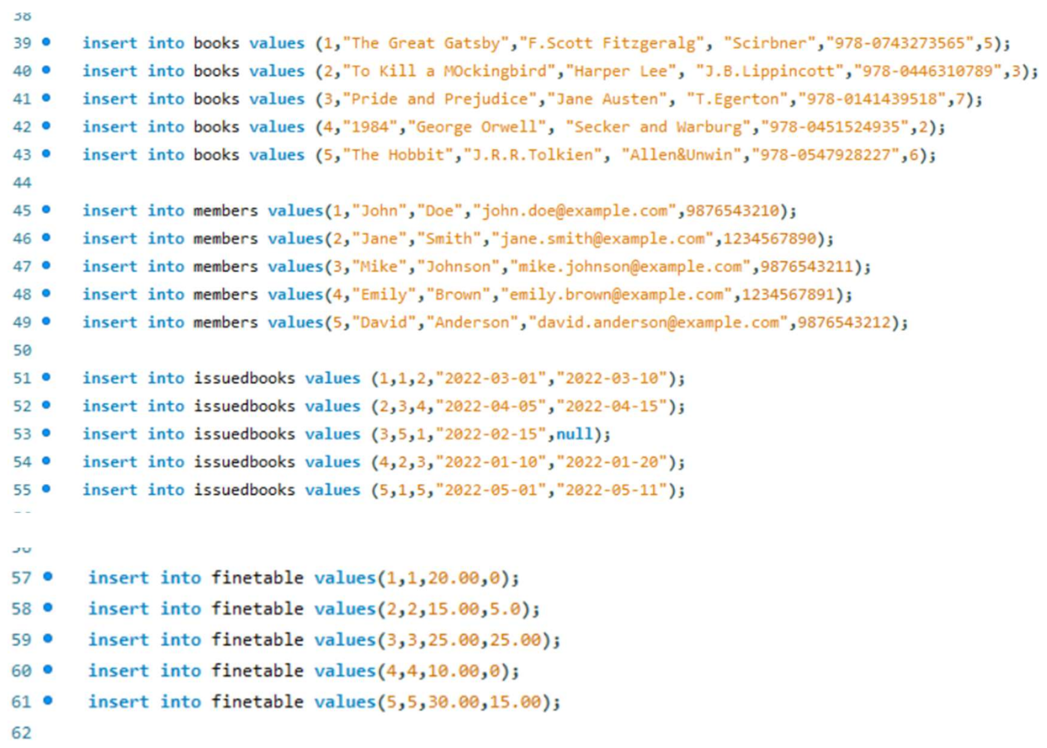


```
create database library_management;

use library_management;

create table books (
BookID int,
Title varchar(255),
Author varchar(255),
Publication varchar(255),
ISBN varchar(17),
Quantity int,
primary key(BookID));

create table Members (
MemberID int ,
FirstName varchar(255),
LastName varchar(255),
Email varchar(255) constraint email_chk check(Email like "%@%"),
Contact varchar(10),
primary key(MemberID) );
```



```sql
create table IssuedBooks(
TransactionID int ,
BookID int,
MemberID int,
IssueDate date,
ReturnDate date,
primary key(TransactionID),
foreign key(BookID) references Books(BookID),
foreign key(MemberID) references Members(MemberID));

create table FineTable(
FineID int ,
TransactionID int,
FineAmount float,
FinePaid float,
primary key(FineID));
```

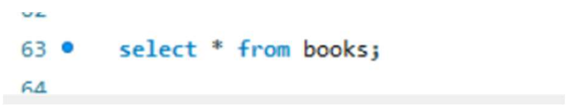
2. Write queries to insert the given dummy values in their respective tables.



```sql
insert into books values (1,"The Great Gatsby","F.Scott Fitzgeralg", "Scirbner","978-0743273565",5);
insert into books values (2,"To Kill a MOckingbird","Harper Lee", "J.B.Lippincott","978-0446310789",3);
insert into books values (3,"Pride and Prejudice","Jane Austen", "T.Egerton","978-0141439518",7);
insert into books values (4,"1984","George Orwell", "Secker and Warburg","978-0451524935",2);
insert into books values (5,"The Hobbit","J.R.R.Tolkien", "Allen&Unwin","978-0547928227",6);

insert into members values(1,"John","Doe","john.doe@example.com",9876543210);
insert into members values(2,"Jane","Smith","jane.smith@example.com",1234567890);
insert into members values(3,"Mike","Johnson","mike.johnson@example.com",9876543211);
insert into members values(4,"Emily","Brown","emily.brown@example.com",1234567891);
insert into members values(5,"David","Anderson","david.anderson@example.com",9876543212);

insert into issuedbooks values (1,1,2,"2022-03-01","2022-03-10");
insert into issuedbooks values (2,3,4,"2022-04-05","2022-04-15");
insert into issuedbooks values (3,5,1,"2022-02-15",null);
insert into issuedbooks values (4,2,3,"2022-01-10","2022-01-20");
insert into issuedbooks values (5,1,5,"2022-05-01","2022-05-11");

insert into finetable values(1,1,20.00,0);
insert into finetable values(2,2,15.00,5.0);
insert into finetable values(3,3,25.00,25.00);
insert into finetable values(4,4,10.00,0);
insert into finetable values(5,5,30.00,15.00);
```

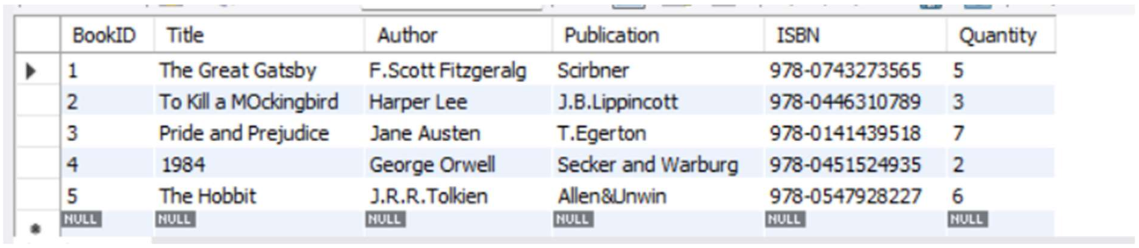
3. Write a query to retrieve all the information from the books table.

Query:



```
select * from books;
```

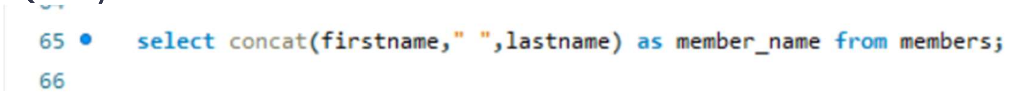
Output:



BookID	Title	Author	Publication	ISBN	Quantity
1	The Great Gatsby	F.Scott Fitzgeralg	Scirbner	978-0743273565	5
2	To Kill a MOckingbird	Harper Lee	J.B.Lippincott	978-0446310789	3
3	Pride and Prejudice	Jane Austen	T.Egerton	978-0141439518	7
4	1984	George Orwell	Secker and Warburg	978-0451524935	2
5	The Hobbit	J.R.R.Tolkien	Allen&Unwin	978-0547928227	6
NULL	NULL	NULL	NULL	NULL	NULL

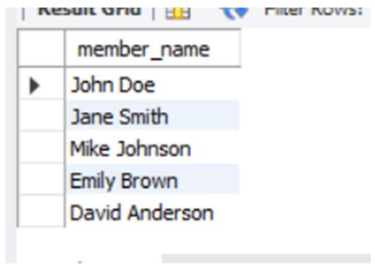
4. Write a query to retrieve the names of all the members from the members' table.

Query:



```
select concat(firstname," ",lastname) as member_name from members;
```

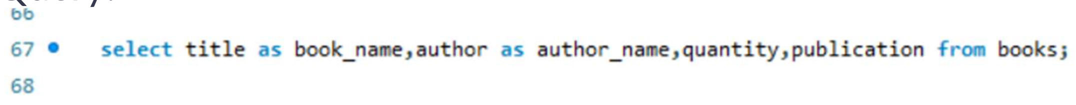
Output:



member_name
John Doe
Jane Smith
Mike Johnson
Emily Brown
David Anderson

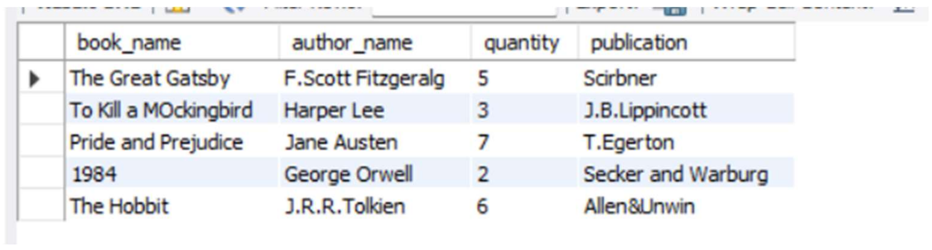
5. Write a query to retrieve the book name, author name, quantity, and publication of all the books.

Query:



```
select title as book_name,author as author_name,quantity,publication from books;
```

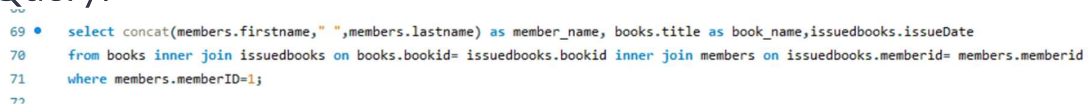
Output:



	book_name	author_name	quantity	publication
▶	The Great Gatsby	F.Scott Fitzgeralg	5	Scirbner
	To Kill a MOckingbird	Harper Lee	3	J.B.Lippincott
	Pride and Prejudice	Jane Austen	7	T.Egerton
	1984	George Orwell	2	Secker and Warburg
	The Hobbit	J.R.R.Tolkien	6	Allen&Unwin

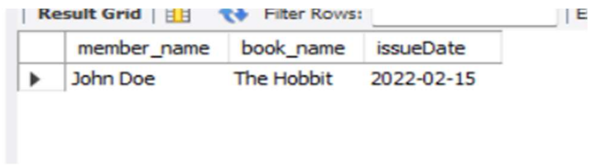
6. Write a query to retrieve the member's name, book name, and issue date for all the books borrowed by the member with ID 1.

Query:



```
select concat(members.firstname," ",members.lastname) as member_name, books.title as book_name,issuedbooks.issueDate
from books inner join issuedbooks on books.bookid= issuedbooks.bookid inner join members on issuedbooks.memberid= members.memberid
where members.memberID=1;
```

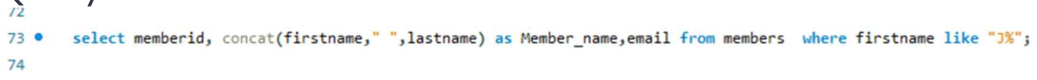
Output:



	member_name	book_name	issueDate
▶	John Doe	The Hobbit	2022-02-15

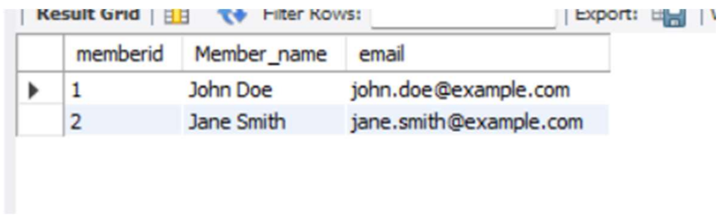
7. Write a query to retrieve the member ID, member name, and email of all the members whose name starts with 'J'.

Query:



```
select memberid, concat(firstname," ",lastname) as Member_name,email from members  where firstname like "J%";
```

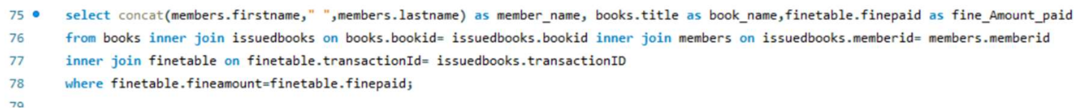
Output:



	memberid	Member_name	email
▶	1	John Doe	john.doe@example.com
	2	Jane Smith	jane.smith@example.com

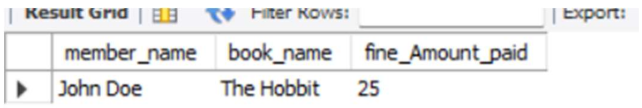
8. Write a query to retrieve the member's name, book name, and fine amount paid for all the members who have paid their complete fine.

Query:



```
select concat(members.firstname," ",members.lastname) as member_name, books.title as book_name,finetable.finepaid as fine_Amount_paid
from books inner join issuedbooks on books.bookid= issuedbooks.bookid inner join members on issuedbooks.memberid= members.memberid
inner join finetable on finetable.transactionId= issuedbooks.transactionID
where finetable.fineamount=finetable.finepaid;
```

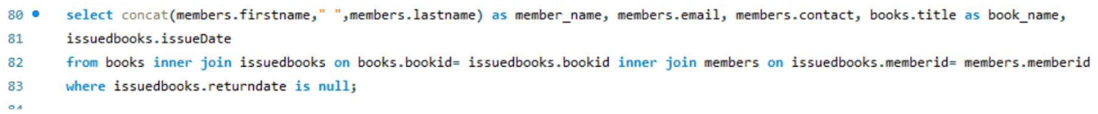
Output:



member_name	book_name	fine_Amount_paid
John Doe	The Hobbit	25

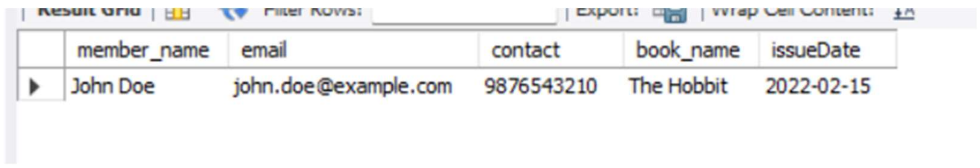
9. Write a query to find the name, email, contact, book name, and issue date of all the members who have not returned any book yet.

Query:



```
select concat(members.firstname," ",members.lastname) as member_name, members.email, members.contact, books.title as book_name,
issuedbooks.issueDate
from books inner join issuedbooks on books.bookid= issuedbooks.bookid inner join members on issuedbooks.memberid= members.memberid
where issuedbooks.returndate is null;
```

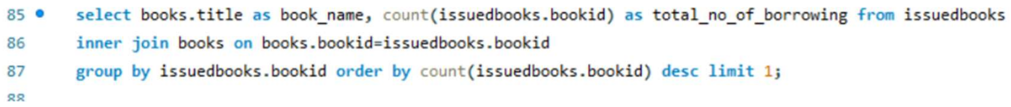
Output:



member_name	email	contact	book_name	issueDate
John Doe	john.doe@example.com	9876543210	The Hobbit	2022-02-15

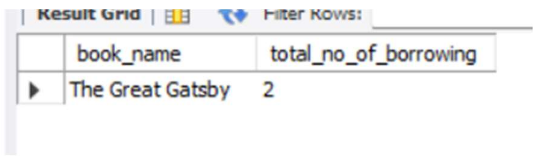
10. Write a query to find the top most borrowed book along with its total number of borrowings.

Query:



```
select books.title as book_name, count(issuedbooks.bookid) as total_no_of_borrowing from issuedbooks
inner join books on books.bookid=issuedbooks.bookid
group by issuedbooks.bookid order by count(issuedbooks.bookid) desc limit 1;
```

Output:

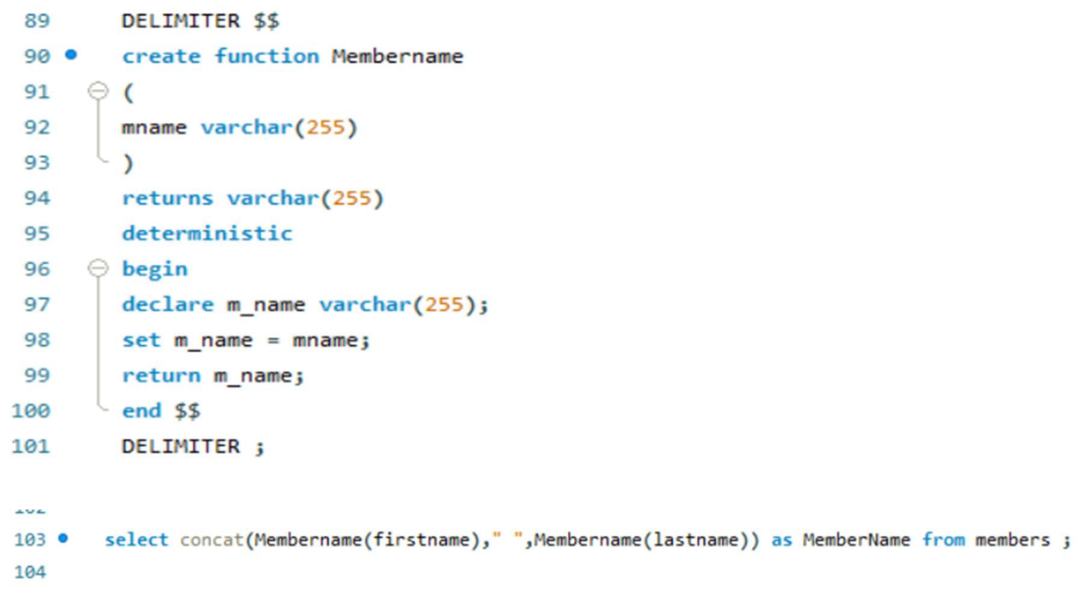


book_name	total_no_of_borrowing
The Great Gatsby	2

User Defined Functions:

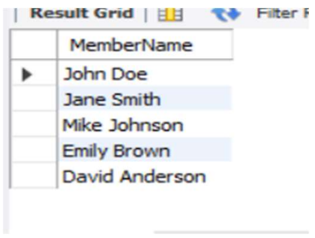
A. To display member’s name.

Query:



```
DELIMITER $$
create function Membername
(
mname varchar(255)
)
returns varchar(255)
deterministic
begin
declare m_name varchar(255);
set m_name = mname;
return m_name;
end $$
DELIMITER ;

select concat(Membername(firstname)," ",Membername(lastname)) as MemberName from members ;
```

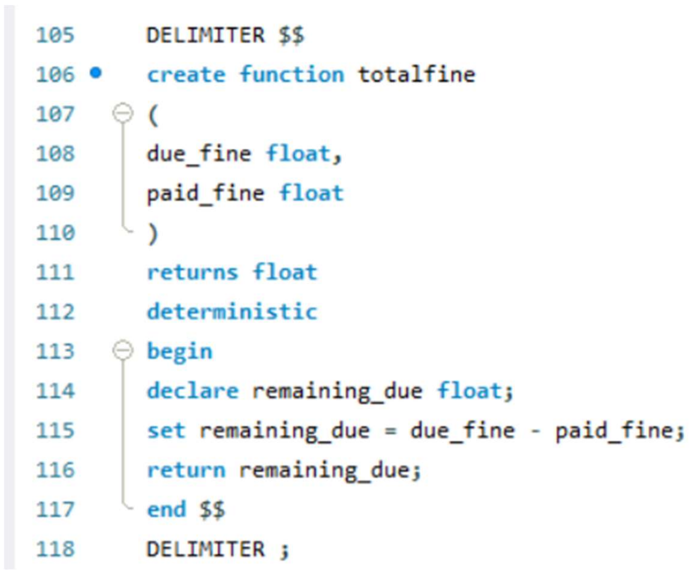
Output:



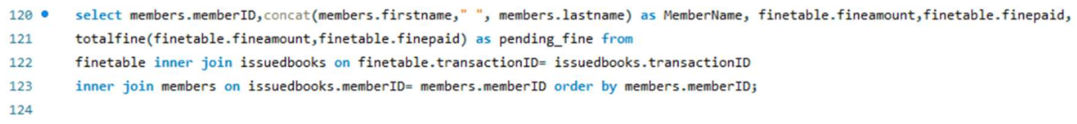
MemberName
John Doe
Jane Smith
Mike Johnson
Emily Brown
David Anderson

B. To calculate remaining fine to be paid by the members.

Query:

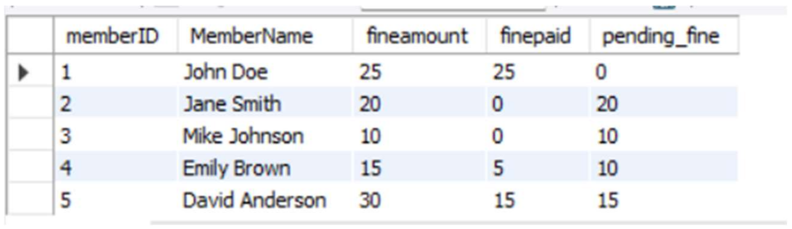


```sql
DELIMITER $$
create function totalfine
(
due_fine float,
paid_fine float
)
returns float
deterministic
begin
declare remaining_due float;
set remaining_due = due_fine - paid_fine;
return remaining_due;
end $$
DELIMITER ;
```



```sql
select members.memberID,concat(members.firstname," ", members.lastname) as MemberName, finetable.fineamount,finetable.finepaid,
totalfine(finetable.fineamount,finetable.finepaid) as pending_fine from
finetable inner join issuedbooks on finetable.transactionID= issuedbooks.transactionID
inner join members on issuedbooks.memberID= members.memberID order by members.memberID;
```

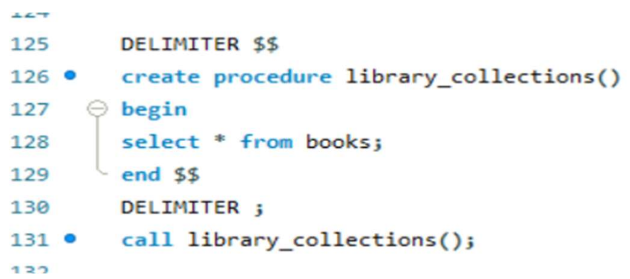
Output:



memberID	MemberName	fineamount	finepaid	pending_fine
1	John Doe	25	25	0
2	Jane Smith	20	0	20
3	Mike Johnson	10	0	10
4	Emily Brown	15	5	10
5	David Anderson	30	15	15

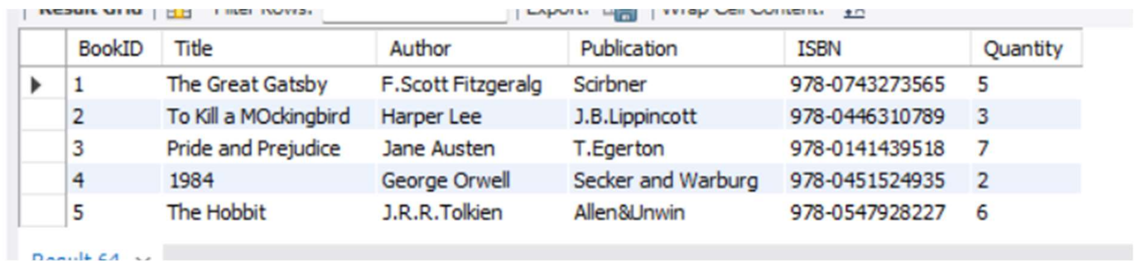
C. To display library collections:

Query:



```
DELIMITER $$
create procedure library_collections()
begin
select * from books;
end $$
DELIMITER ;
call library_collections();
```

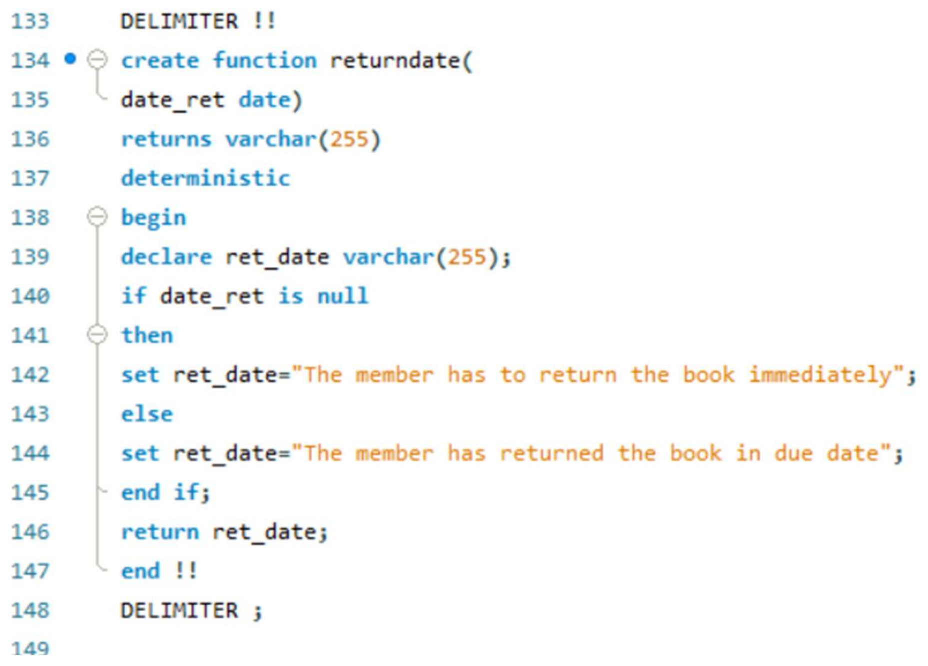
Output:



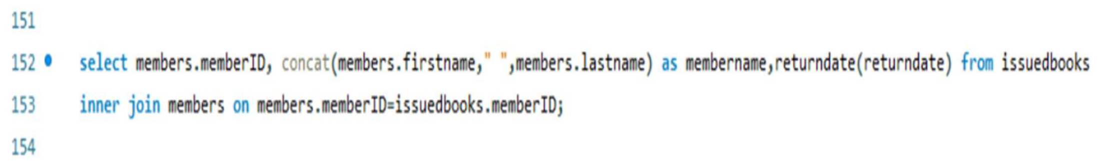
BookID	Title	Author	Publication	ISBN	Quantity
1	The Great Gatsby	F.Scott Fitzgeralg	Scirbner	978-0743273565	5
2	To Kill a MOckingbird	Harper Lee	J.B.Lippincott	978-0446310789	3
3	Pride and Prejudice	Jane Austen	T.Egerton	978-0141439518	7
4	1984	George Orwell	Secker and Warburg	978-0451524935	2
5	The Hobbit	J.R.R.Tolkien	Allen&Unwin	978-0547928227	6

D. To display whether the member returned the book or not.

Query:

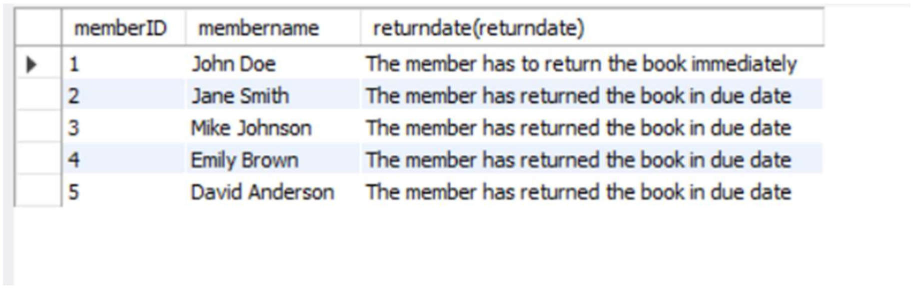


```
DELIMITER !!
create function returndate(
date_ret date)
returns varchar(255)
deterministic
begin
declare ret_date varchar(255);
if date_ret is null
then
set ret_date="The member has to return the book immediately";
else
set ret_date="The member has returned the book in due date";
end if;
return ret_date;
end !!
DELIMITER ;
```



```sql
select members.memberID, concat(members.firstname," ",members.lastname) as membername,returndate(returndate) from issuedbooks
inner join members on members.memberID=issuedbooks.memberID;
```

Output:



memberID	membername	returndate(returndate)
1	John Doe	The member has to return the book immediately
2	Jane Smith	The member has returned the book in due date
3	Mike Johnson	The member has returned the book in due date
4	Emily Brown	The member has returned the book in due date
5	David Anderson	The member has returned the book in due date