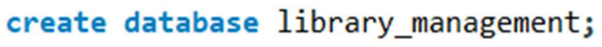
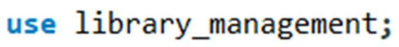
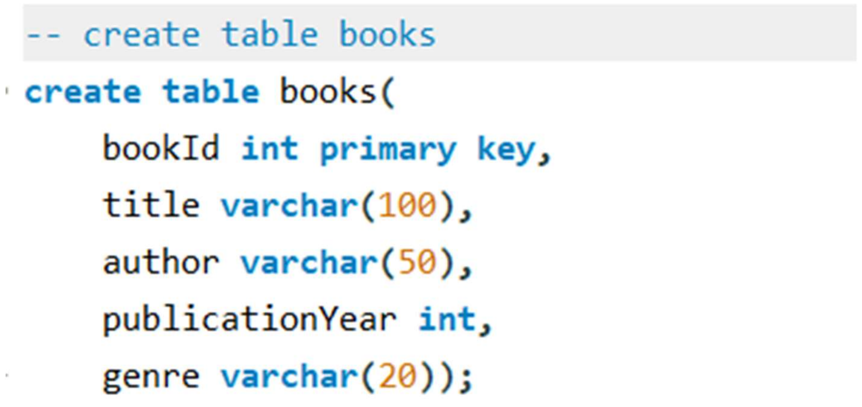
Create Database as Library Management

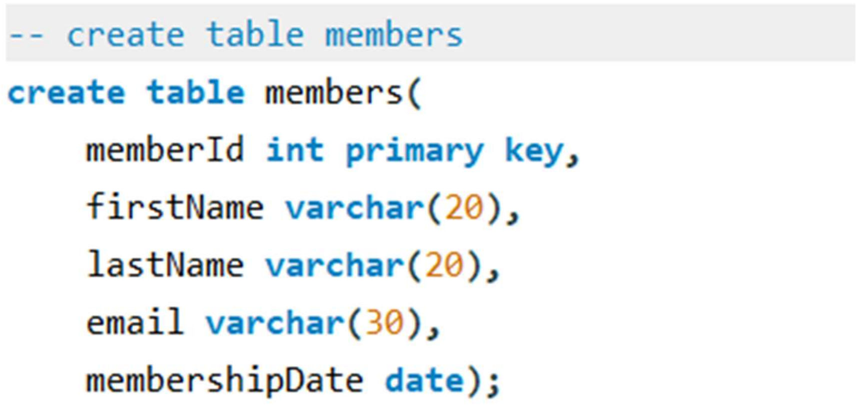

```sql
create database library_management;
```

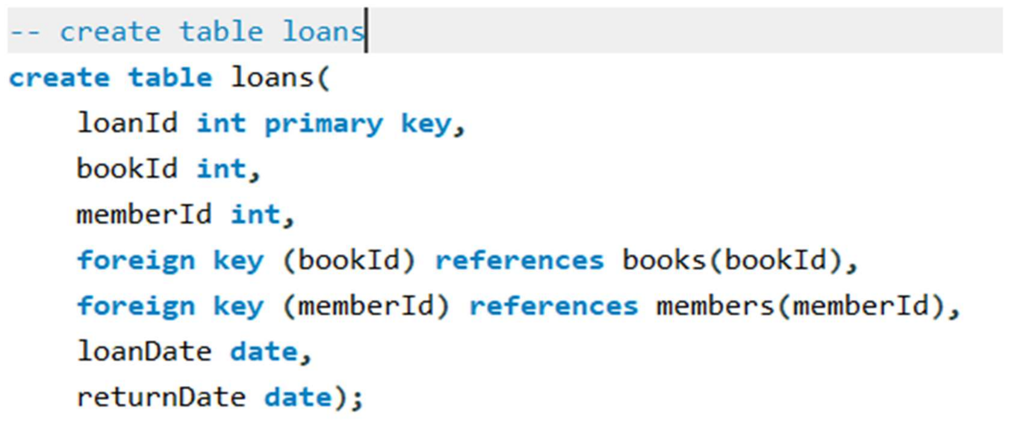
Use Database


```sql
use library_management;
```

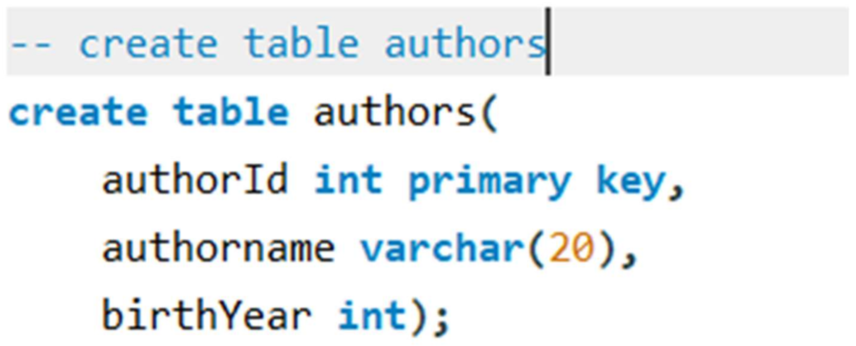
1. **Create Table:**
    - Write an SQL statement to create all tables with the specified columns.


```sql
-- create table books
create table books(
    bookId int primary key,
    title varchar(100),
    author varchar(50),
    publicationYear int,
    genre varchar(20));
```

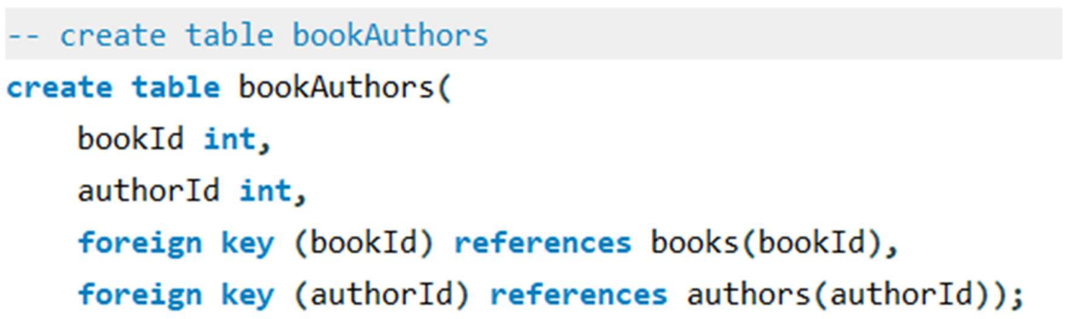

```sql
-- create table members
create table members(
    memberId int primary key,
    firstName varchar(20),
    lastName varchar(20),
    email varchar(30),
    membershipDate date);
```



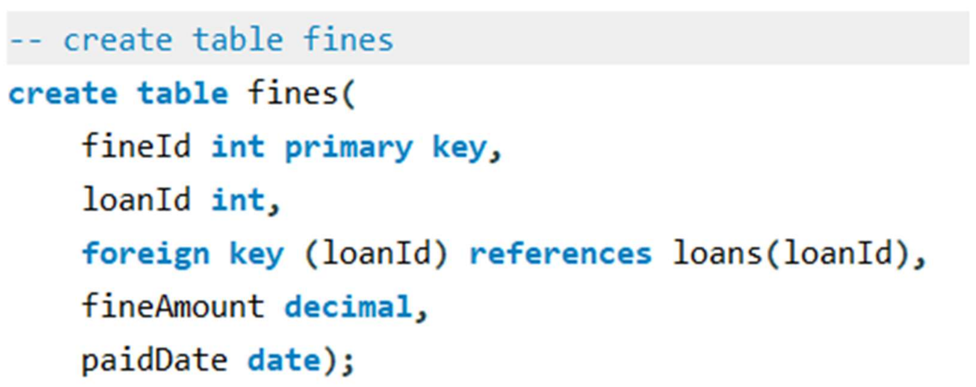
```
-- create table loans
create table loans(
    loanId int primary key,
    bookId int,
    memberId int,
    foreign key (bookId) references books(bookId),
    foreign key (memberId) references members(memberId),
    loanDate date,
    returnDate date);
```



```
-- create table authors
create table authors(
    authorId int primary key,
    authorname varchar(20),
    birthYear int);
```



```
-- create table bookAuthors
create table bookAuthors(
    bookId int,
    authorId int,
    foreign key (bookId) references books(bookId),
    foreign key (authorId) references authors(authorId));
```

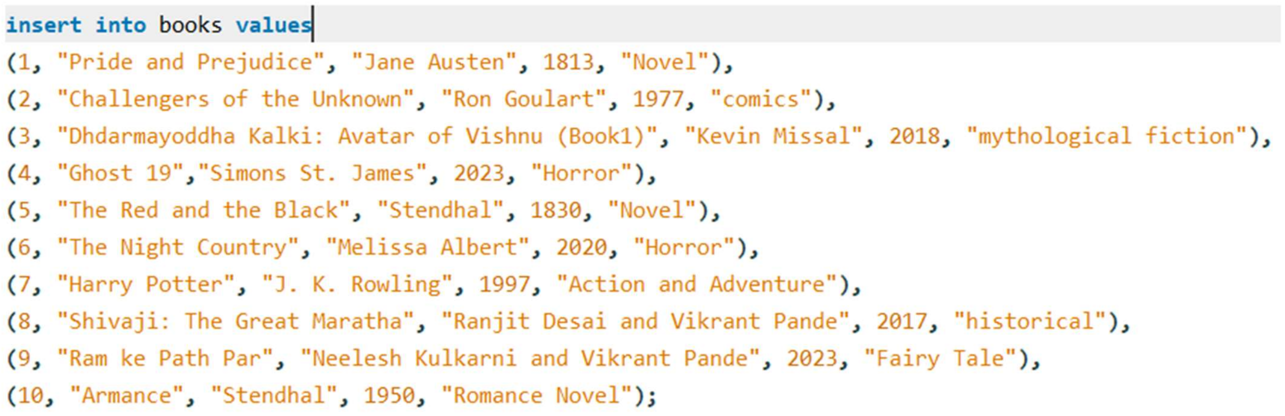


```
-- create table fines
create table fines(
    fineId int primary key,
    loanId int,
    foreign key (loanId) references loans(loanId),
    fineAmount decimal,
    paidDate date);
```

2. **Insert Records:**

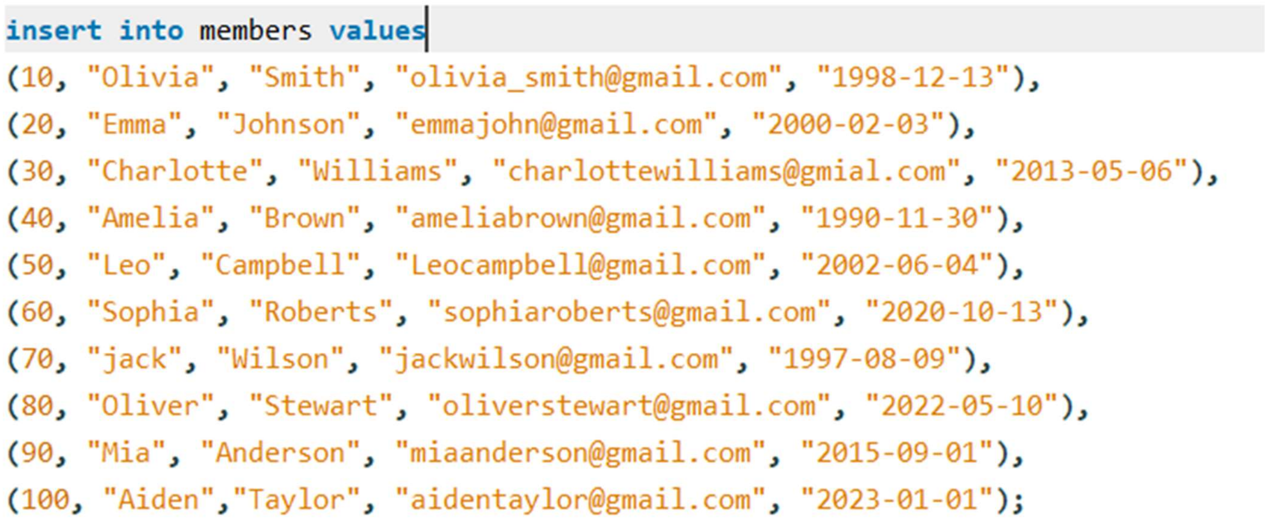
   - Insert at least 10 records in all the tables.

- Inserting records in books table



```
insert into books values
(1, "Pride and Prejudice", "Jane Austen", 1813, "Novel"),
(2, "Challengers of the Unknown", "Ron Goulart", 1977, "comics"),
(3, "Dhdarmayoddha Kalki: Avatar of Vishnu (Book1)", "Kevin Missal", 2018, "mythological fiction"),
(4, "Ghost 19","Simons St. James", 2023, "Horror"),
(5, "The Red and the Black", "Stendhal", 1830, "Novel"),
(6, "The Night Country", "Melissa Albert", 2020, "Horror"),
(7, "Harry Potter", "J. K. Rowling", 1997, "Action and Adventure"),
(8, "Shivaji: The Great Maratha", "Ranjit Desai and Vikrant Pande", 2017, "historical"),
(9, "Ram ke Path Par", "Neelesh Kulkarni and Vikrant Pande", 2023, "Fairy Tale"),
(10, "Armance", "Stendhal", 1950, "Romance Novel");
```

- Inserting records in members table

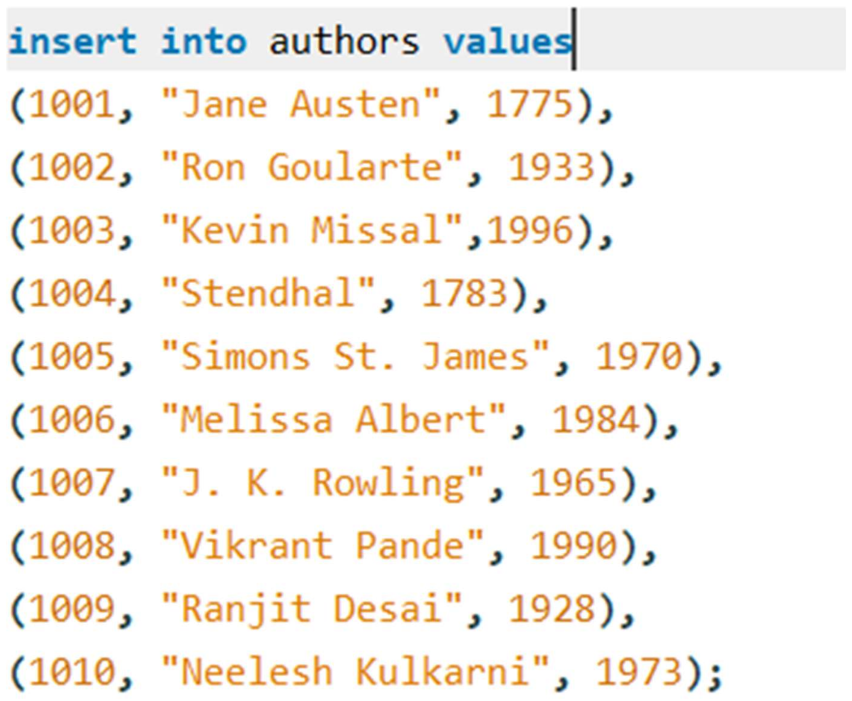


```
insert into members values
(10, "Olivia", "Smith", "olivia_smith@gmail.com", "1998-12-13"),
(20, "Emma", "Johnson", "emmajohn@gmail.com", "2000-02-03"),
(30, "Charlotte", "Williams", "charlottewilliams@gmial.com", "2013-05-06"),
(40, "Amelia", "Brown", "ameliabrown@gmail.com", "1990-11-30"),
(50, "Leo", "Campbell", "Leocampbell@gmail.com", "2002-06-04"),
(60, "Sophia", "Roberts", "sophiaroberts@gmail.com", "2020-10-13"),
(70, "jack", "Wilson", "jackwilson@gmail.com", "1997-08-09"),
(80, "Oliver", "Stewart", "oliverstewart@gmail.com", "2022-05-10"),
(90, "Mia", "Anderson", "miaanderson@gmail.com", "2015-09-01"),
(100, "Aiden","Taylor", "aidentaylor@gmail.com", "2023-01-01");
```

- Inserting records in loans table

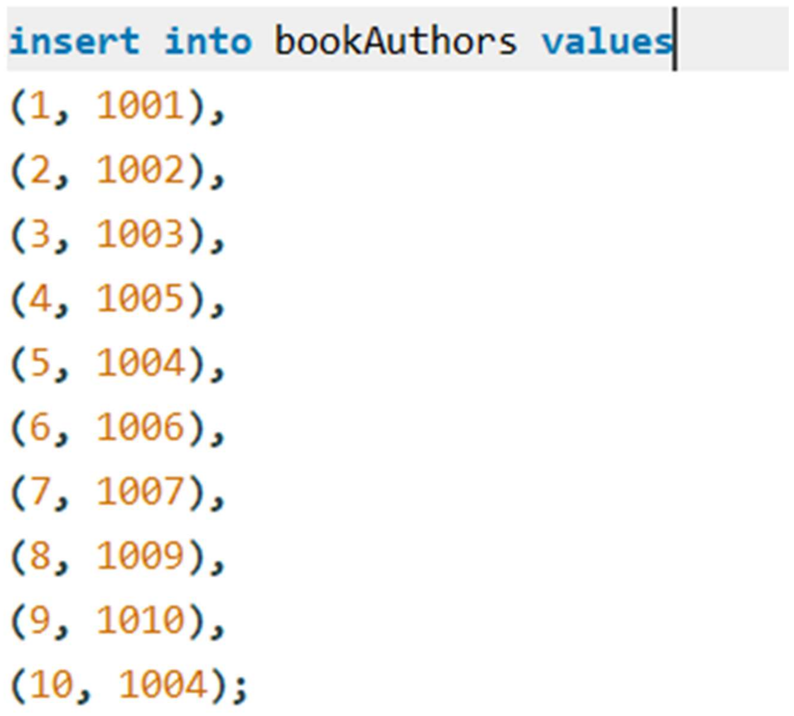
```
insert into loans values
(501, 2, 10, "2000-01-10", "2000-06-01"),
(502, 1, 90, "1995-06-12", "1996-02-21"),
(503, 4, 30, "2014-03-04", "2014-08-10");

insert into loans (loanId, bookId, memberId, loanDate) values
(504, 3, 60, "2022-10-22");

insert into loans values
(505, 5, 50, "2004-07-15", "2005-02-15"),
(506, 10, 90, "2018-05-20", "2018-12-02");

insert into loans (loanId, bookId, memberId, loanDate) values
(507, 8, 60, "2024-03-15"),
(508,5, 30, "2023-05-10");

insert into loans values
(509, 6, 90, "2002-06-14", "2003-01-01"),
(510, 5, 70, "1991-04-25", "1991-12-20"),
(511, 4, 40, "2010-06-14", "2011-01-01"),
(512, 5, 90, "1999-04-25", "1999-12-20");

insert into loans (loanId, bookId, memberId, loanDate) values
(513, 8, 90, "2024-0-19");

insert into loans values
(514, 6, 40, "2015-01-14", "2016-06-05"),
(515, 5, 60, "2001-04-25", "2001-12-20"),
(516, 4, 90, "2005-05-15", "2005-11-25"),
(517, 1, 60, "2012-04-25", "2012-09-30");
```

- Inserting records in authors table



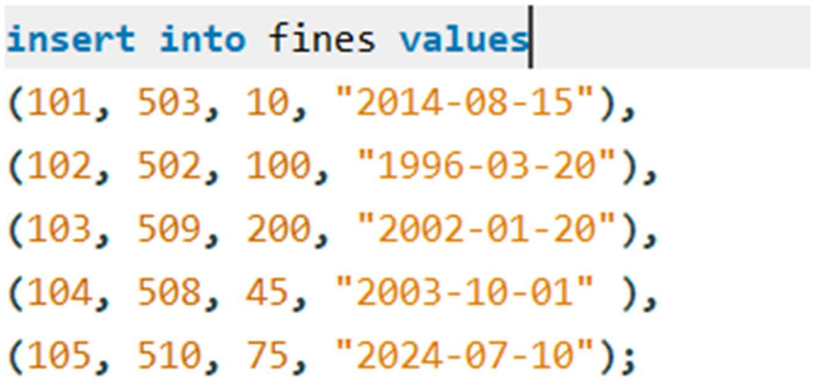
```
insert into authors values
(1001, "Jane Austen", 1775),
(1002, "Ron Goularte", 1933),
(1003, "Kevin Missal",1996),
(1004, "Stendhal", 1783),
(1005, "Simons St. James", 1970),
(1006, "Melissa Albert", 1984),
(1007, "J. K. Rowling", 1965),
(1008, "Vikrant Pande", 1990),
(1009, "Ranjit Desai", 1928),
(1010, "Neelesh Kulkarni", 1973);
```

- Inserting records in bookAuthors table



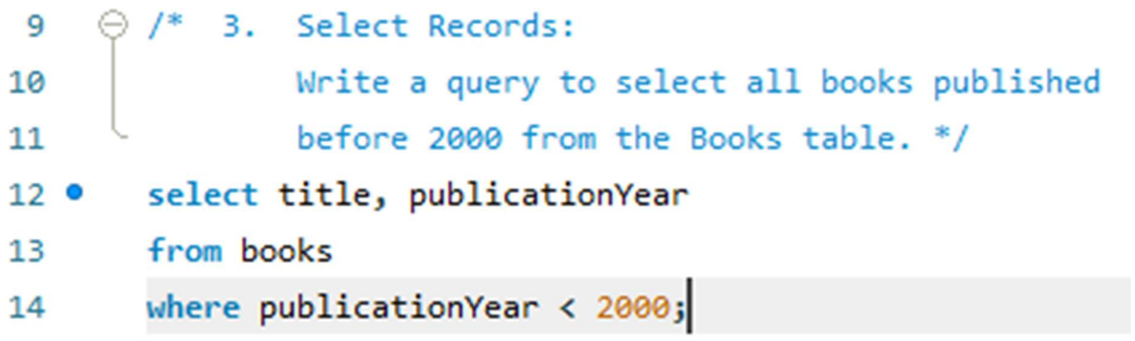
```
insert into bookAuthors values
(1, 1001),
(2, 1002),
(3, 1003),
(4, 1005),
(5, 1004),
(6, 1006),
(7, 1007),
(8, 1009),
(9, 1010),
(10, 1004);
```

- Inserting records in fines table



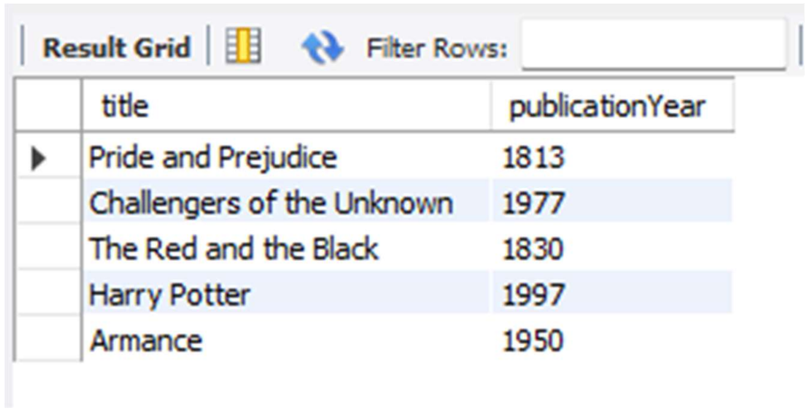
```
insert into fines values
(101, 503, 10, "2014-08-15"),
(102, 502, 100, "1996-03-20"),
(103, 509, 200, "2002-01-20"),
(104, 508, 45, "2003-10-01" ),
(105, 510, 75, "2024-07-10");
```

3. **Select Records:**
    - Write a query to select all books published before 2000 from the Books table.



```
/*  3.  Select Records:
        Write a query to select all books published
        before 2000 from the Books table. */
select title, publicationYear
from books
where publicationYear < 2000;
```

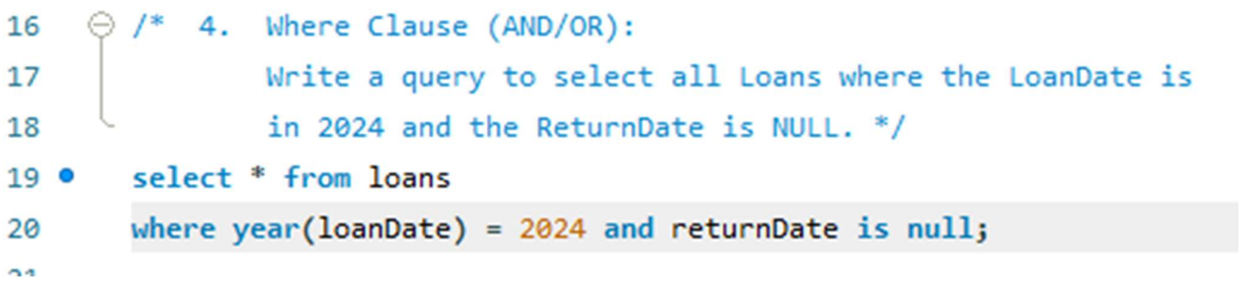
Result:



title	publicationYear
Pride and Prejudice	1813
Challengers of the Unknown	1977
The Red and the Black	1830
Harry Potter	1997
Armance	1950

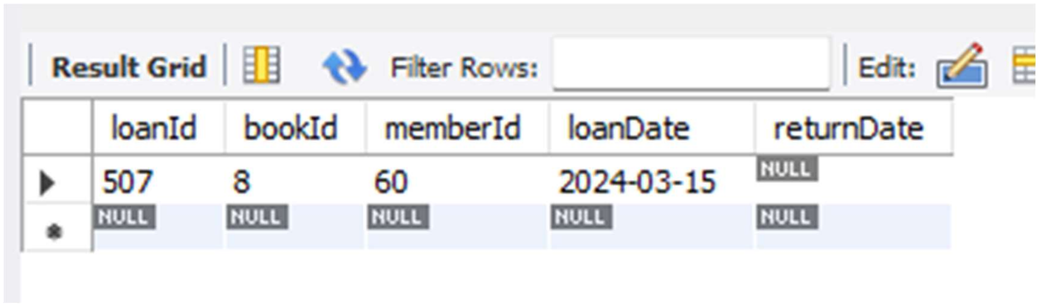
4. **Where Clause (AND/OR):**

   - Write a query to select all Loans where the LoanDate is in 2024 and the ReturnDate is NULL.



```sql
/*  4.  Where Clause (AND/OR):
        Write a query to select all Loans where the LoanDate is
        in 2024 and the ReturnDate is NULL. */
select * from loans
where year(loanDate) = 2024 and returnDate is null;
```

Result:

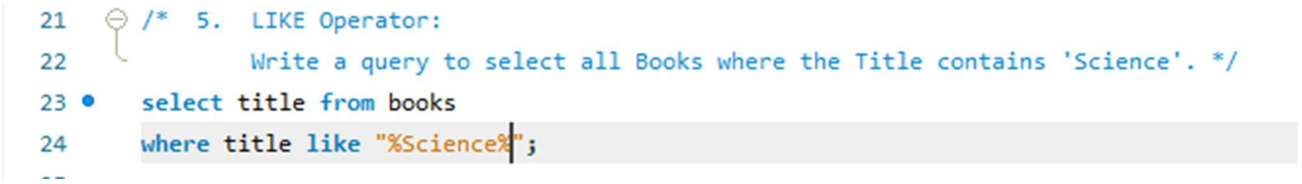


Result Grid | Filter Rows: | Edit:

	loanId	bookId	memberId	loanDate	returnDate
▶	507	8	60	2024-03-15	NULL
*	NULL	NULL	NULL	NULL	NULL

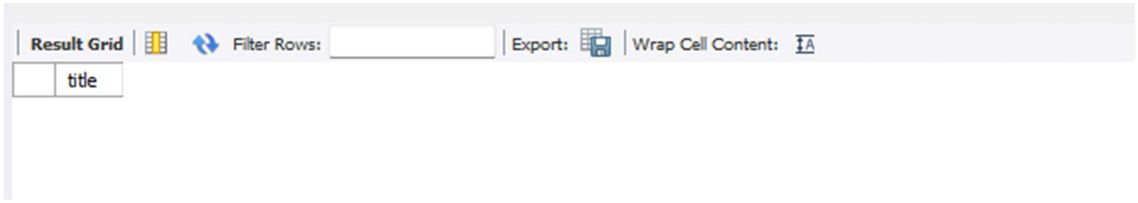
5. **LIKE Operator:**

   - Write a query to select all Books where the Title contains 'Science'.



```sql
/*  5.  LIKE Operator:
        Write a query to select all Books where the Title contains 'Science'. */
select title from books
where title like "%Science%";
```

Result:

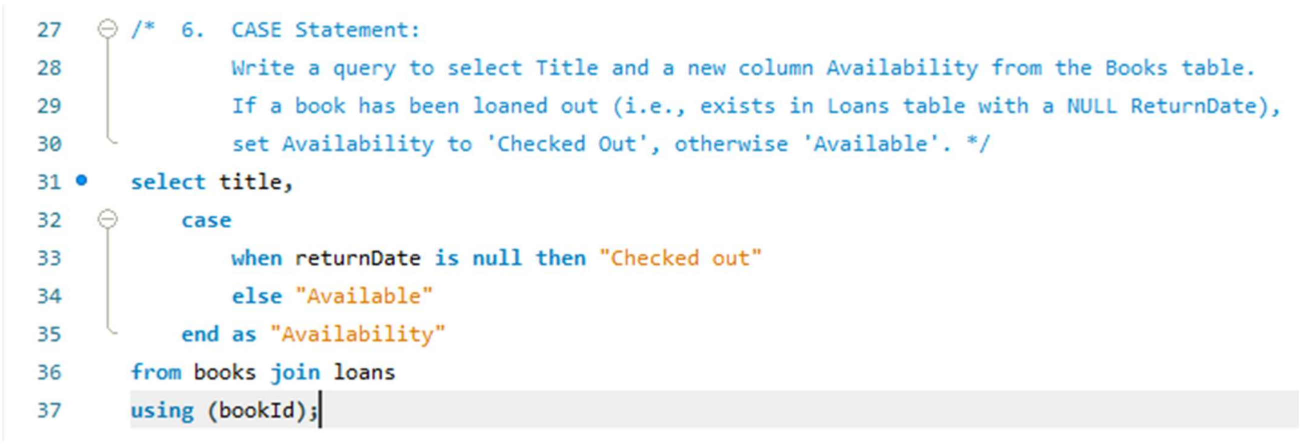


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	title

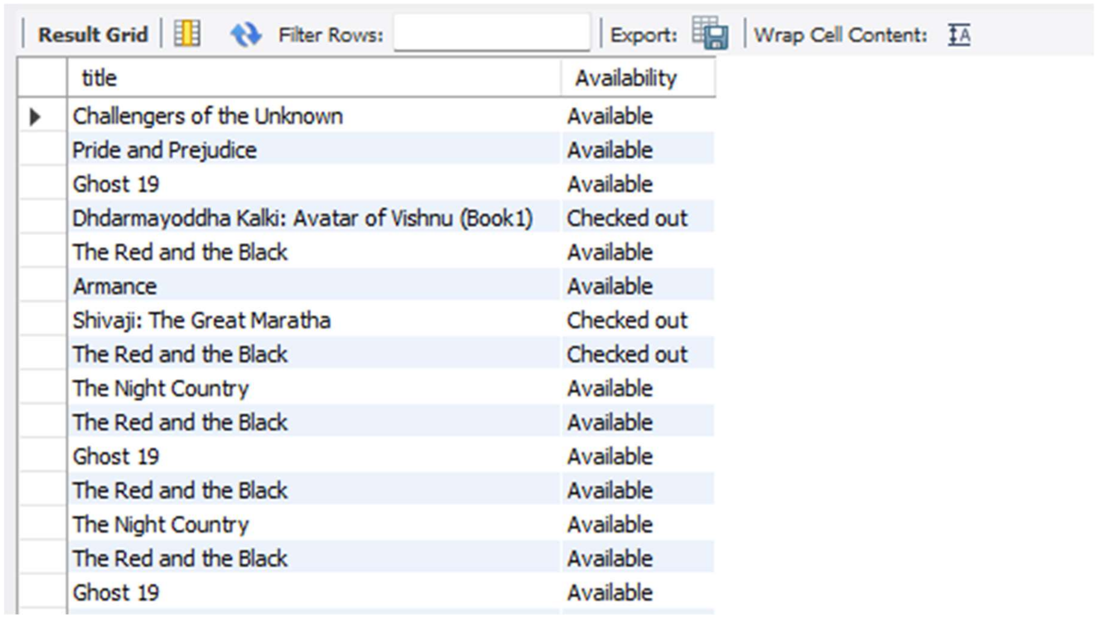
6. **CASE Statement:**

   - Write a query to select Title and a new column Availability from the Books table. If a book has been loaned out (i.e., exists in Loans table with a NULL ReturnDate), set Availability to 'Checked Out', otherwise 'Available'.



```
/*  6.  CASE Statement:
        Write a query to select Title and a new column Availability from the Books table.
        If a book has been loaned out (i.e., exists in Loans table with a NULL ReturnDate),
        set Availability to 'Checked Out', otherwise 'Available'. */
select title,
    case
        when returnDate is null then "Checked out"
        else "Available"
    end as "Availability"
from books join loans
using (bookId);
```

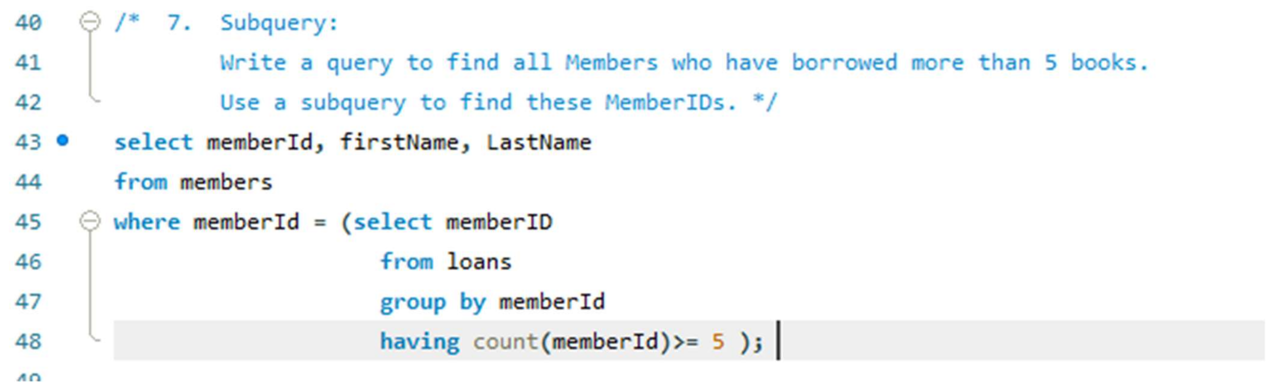
Result:



title	Availability
Challengers of the Unknown	Available
Pride and Prejudice	Available
Ghost 19	Available
Dhdarmayoddha Kalki: Avatar of Vishnu (Book1)	Checked out
The Red and the Black	Available
Armance	Available
Shivaji: The Great Maratha	Checked out
The Red and the Black	Checked out
The Night Country	Available
The Red and the Black	Available
Ghost 19	Available
The Red and the Black	Available
The Night Country	Available
The Red and the Black	Available
Ghost 19	Available

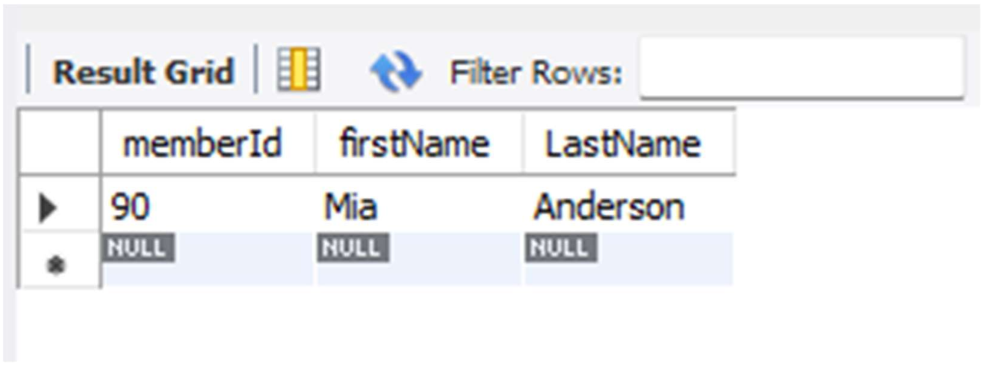
7. **Subquery:**

- Write a query to find all Members who have borrowed more than 5 books. Use a subquery to find these MemberIDs.



```
/*  7.  Subquery:
        Write a query to find all Members who have borrowed more than 5 books.
        Use a subquery to find these MemberIDs. */
select memberId, firstName, LastName
from members
where memberId = (select memberID
                    from loans
                    group by memberId
                    having count(memberId)>= 5 );
```

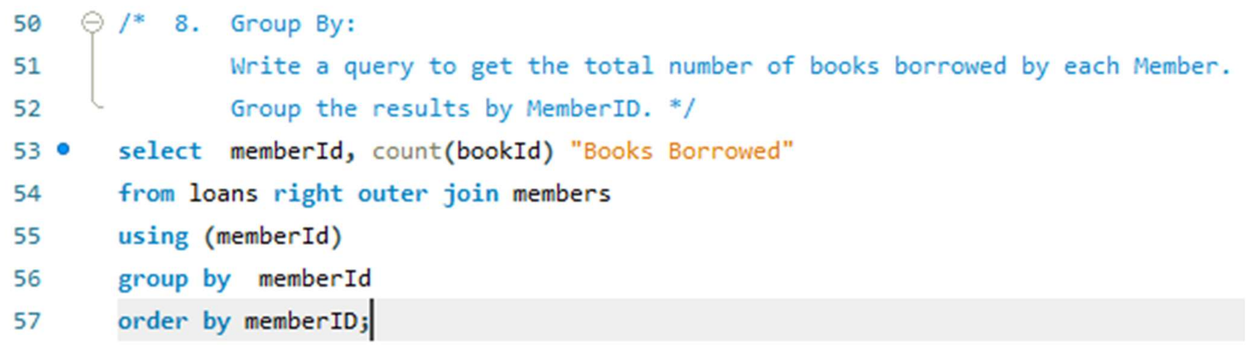
Result:



```
memberId  firstName  LastName
90        Mia        Anderson
NULL      NULL       NULL
```

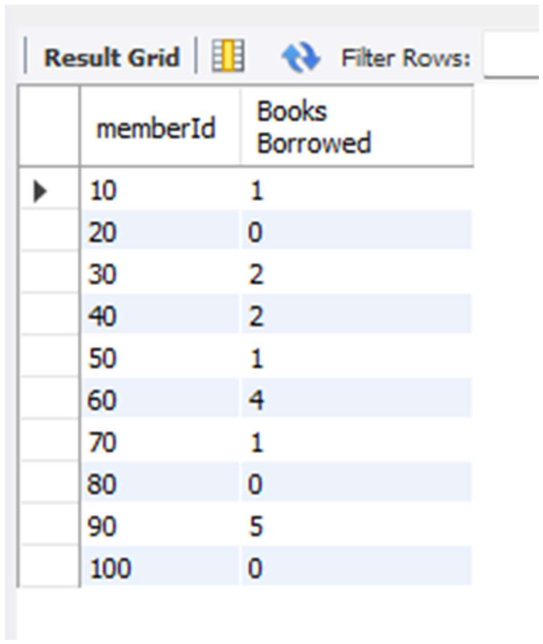
8. **Group By:**

- Write a query to get the total number of books borrowed by each Member. Group the results by MemberID.



```sql
/*  8.  Group By:
        Write a query to get the total number of books borrowed by each Member.
        Group the results by MemberID. */
select  memberId, count(bookId) "Books Borrowed"
from loans right outer join members
using (memberId)
group by  memberId
order by memberID;
```

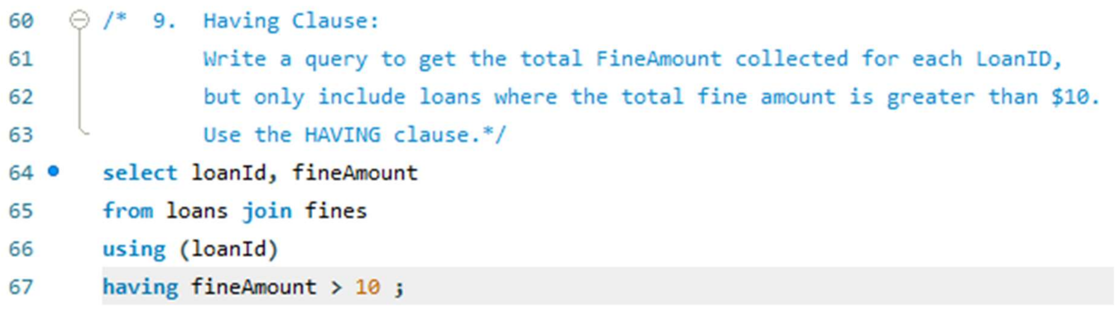
Result:



memberId	Books Borrowed
10	1
20	0
30	2
40	2
50	1
60	4
70	1
80	0
90	5
100	0

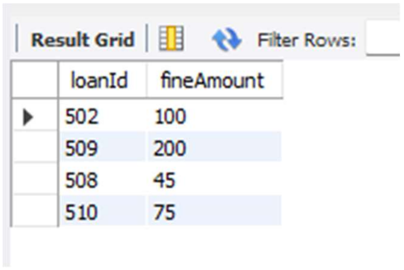
9. **Having Clause:**

   - Write a query to get the total FineAmount collected for each LoanID, but only include loans where the total fine amount is greater than $10. Use the HAVING clause.



```
/*  9.  Having Clause:
        Write a query to get the total FineAmount collected for each LoanID,
        but only include loans where the total fine amount is greater than $10.
        Use the HAVING clause.*/
select loanId, fineAmount
from loans join fines
using (loanId)
having fineAmount > 10 ;
```

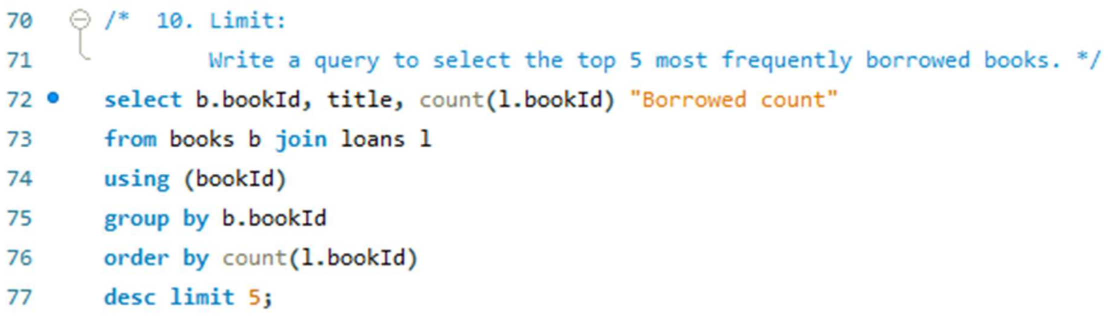
Result:



loanId	fineAmount
502	100
509	200
508	45
510	75

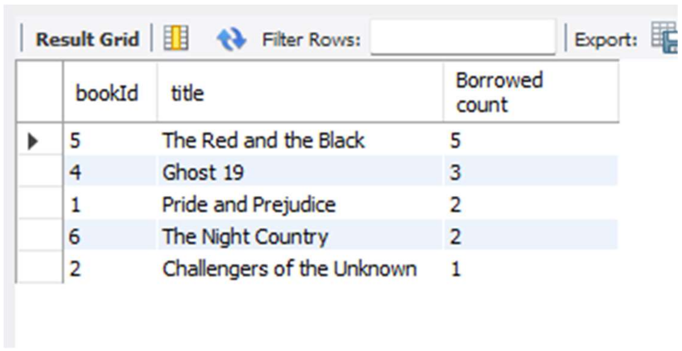
10. **Limit:**

   - Write a query to select the top 5 most frequently borrowed books.



```
/*  10. Limit:
        Write a query to select the top 5 most frequently borrowed books. */
select b.bookId, title, count(l.bookId) "Borrowed count"
from books b join loans l
using (bookId)
group by b.bookId
order by count(l.bookId)
desc limit 5;
```

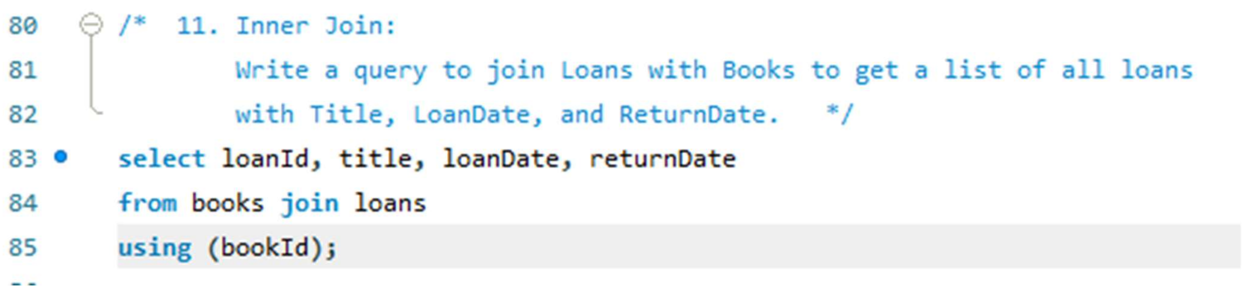
Result:



bookId	title	Borrowed count
5	The Red and the Black	5
4	Ghost 19	3
1	Pride and Prejudice	2
6	The Night Country	2
2	Challengers of the Unknown	1

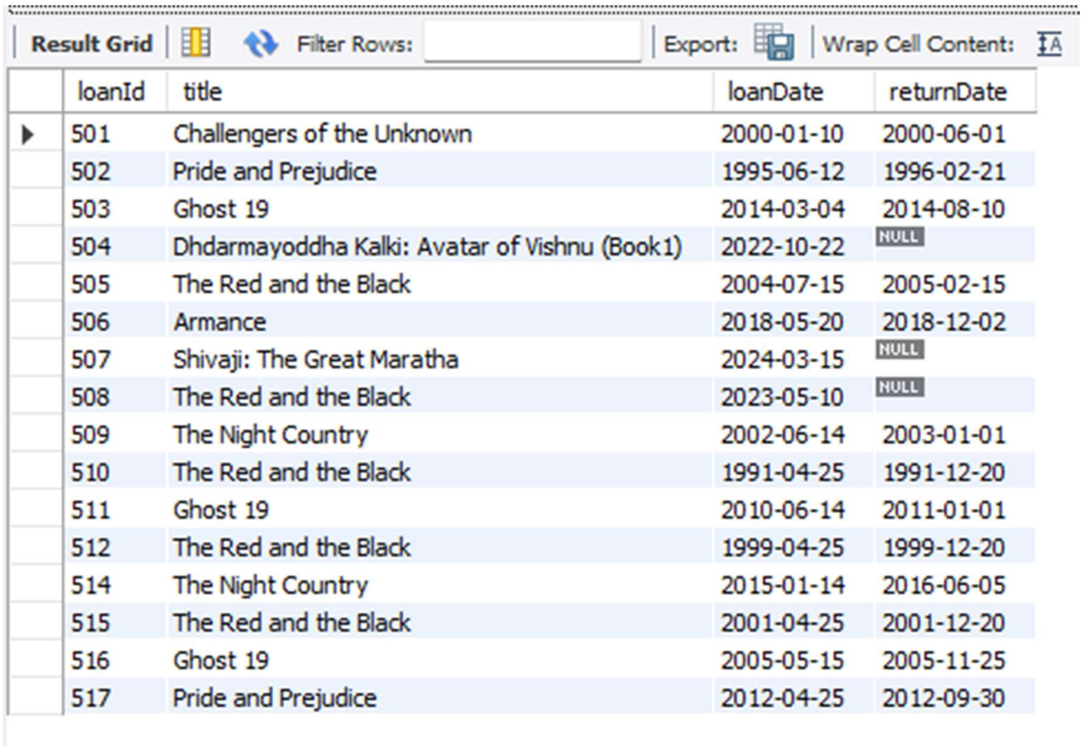
11. **Inner Join:**

- Write a query to join Loans with Books to get a list of all loans with Title, LoanDate, and ReturnDate.



```
/*  11. Inner Join:
        Write a query to join Loans with Books to get a list of all loans
        with Title, LoanDate, and ReturnDate.   */
select loanId, title, loanDate, returnDate
from books join loans
using (bookId);
```

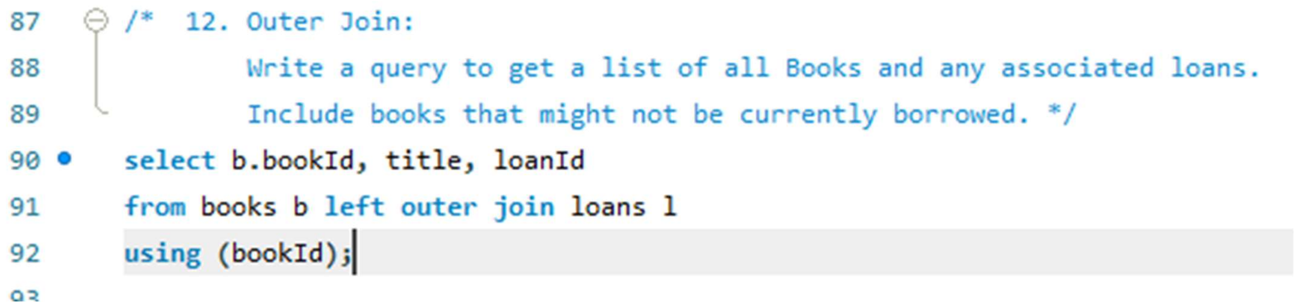
Result:



loanId	title	loanDate	returnDate
501	Challengers of the Unknown	2000-01-10	2000-06-01
502	Pride and Prejudice	1995-06-12	1996-02-21
503	Ghost 19	2014-03-04	2014-08-10
504	Dhdarmayoddha Kalki: Avatar of Vishnu (Book1)	2022-10-22	NULL
505	The Red and the Black	2004-07-15	2005-02-15
506	Armance	2018-05-20	2018-12-02
507	Shivaji: The Great Maratha	2024-03-15	NULL
508	The Red and the Black	2023-05-10	NULL
509	The Night Country	2002-06-14	2003-01-01
510	The Red and the Black	1991-04-25	1991-12-20
511	Ghost 19	2010-06-14	2011-01-01
512	The Red and the Black	1999-04-25	1999-12-20
514	The Night Country	2015-01-14	2016-06-05
515	The Red and the Black	2001-04-25	2001-12-20
516	Ghost 19	2005-05-15	2005-11-25
517	Pride and Prejudice	2012-04-25	2012-09-30

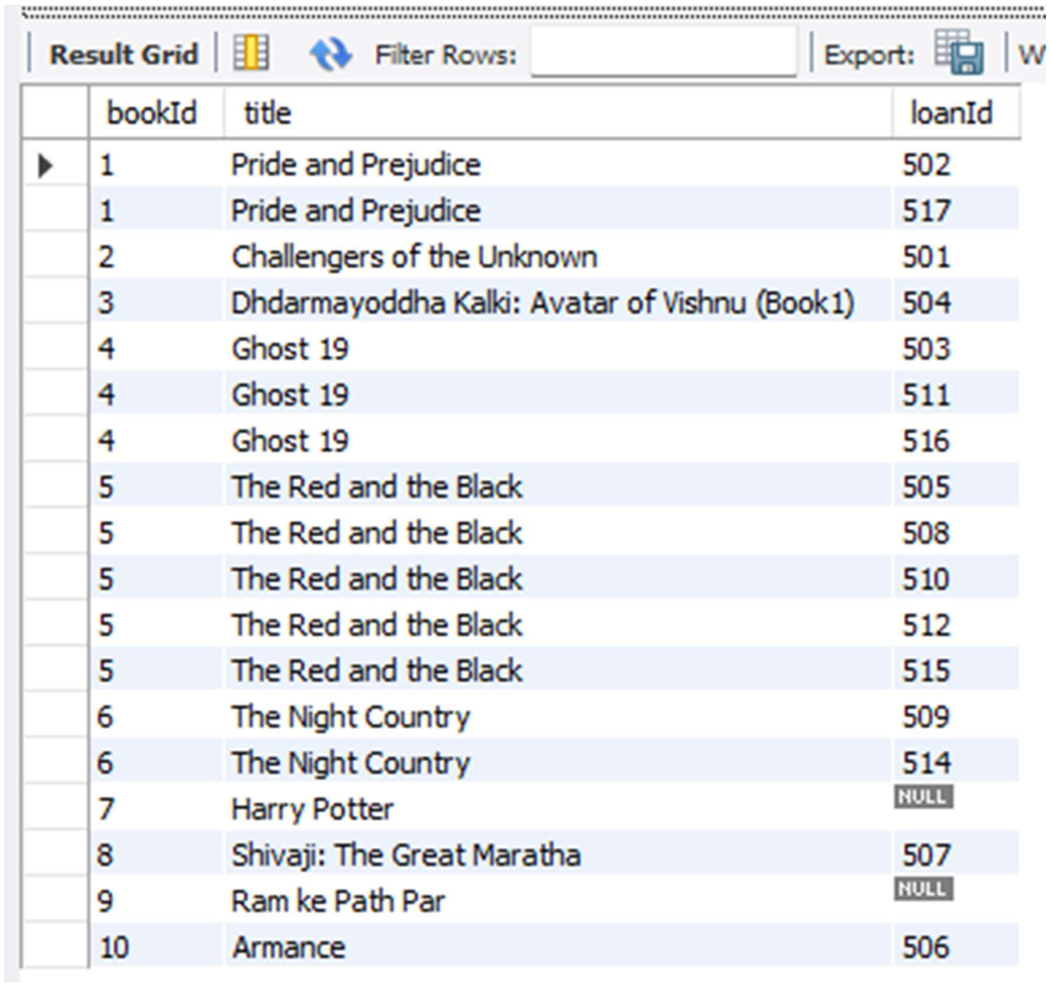
12. **Outer Join:**

- Write a query to get a list of all Books and any associated loans. Include books that might not be currently borrowed.



```
/*  12. Outer Join:
        Write a query to get a list of all Books and any associated loans.
        Include books that might not be currently borrowed. */
select b.bookId, title, loanId
from books b left outer join loans l
using (bookId);
```

Result:

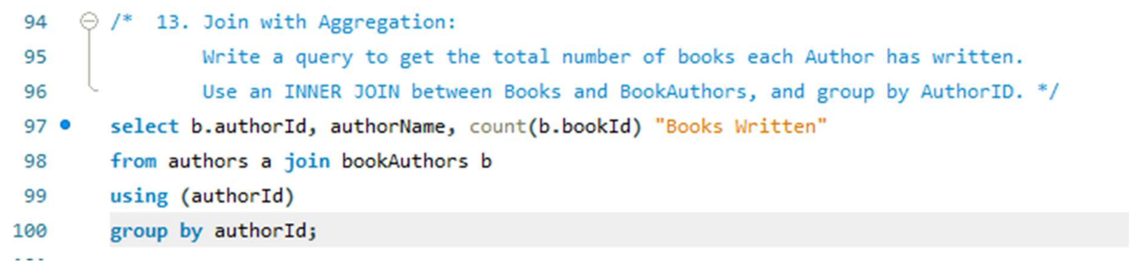


Result Grid | Filter Rows: | Export: | W

bookId	title	loanId
1	Pride and Prejudice	502
1	Pride and Prejudice	517
2	Challengers of the Unknown	501
3	Dhdarmayoddha Kalki: Avatar of Vishnu (Book1)	504
4	Ghost 19	503
4	Ghost 19	511
4	Ghost 19	516
5	The Red and the Black	505
5	The Red and the Black	508
5	The Red and the Black	510
5	The Red and the Black	512
5	The Red and the Black	515
6	The Night Country	509
6	The Night Country	514
7	Harry Potter	NULL
8	Shivaji: The Great Maratha	507
9	Ram ke Path Par	NULL
10	Armance	506

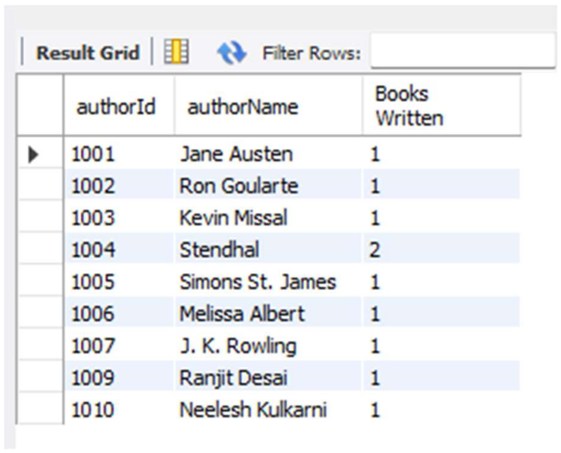
13. **Join with Aggregation:**

- Write a query to get the total number of books each Author has written. Use an INNER JOIN between Books and BookAuthors, and group by AuthorID.



```sql
/*  13. Join with Aggregation:
        Write a query to get the total number of books each Author has written.
        Use an INNER JOIN between Books and BookAuthors, and group by AuthorID. */
select b.authorId, authorName, count(b.bookId) "Books Written"
from authors a join bookAuthors b
using (authorId)
group by authorId;
```

Result:

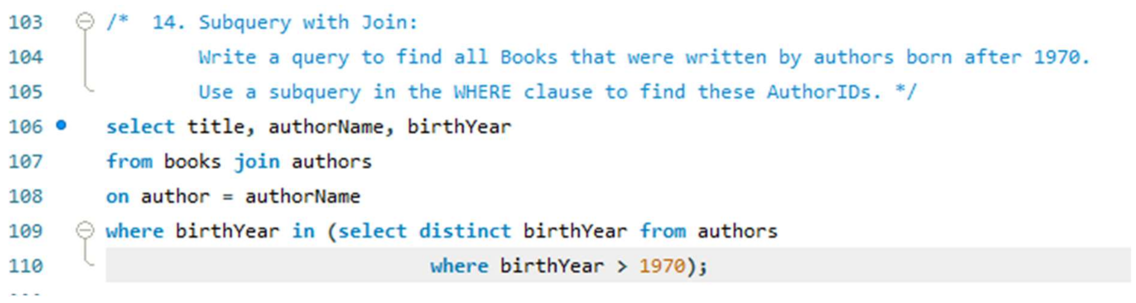


Result Grid | Filter Rows:

authorId	authorName	Books Written
1001	Jane Austen	1
1002	Ron Goularte	1
1003	Kevin Missal	1
1004	Stendhal	2
1005	Simons St. James	1
1006	Melissa Albert	1
1007	J. K. Rowling	1
1009	Ranjit Desai	1
1010	Neelesh Kulkarni	1

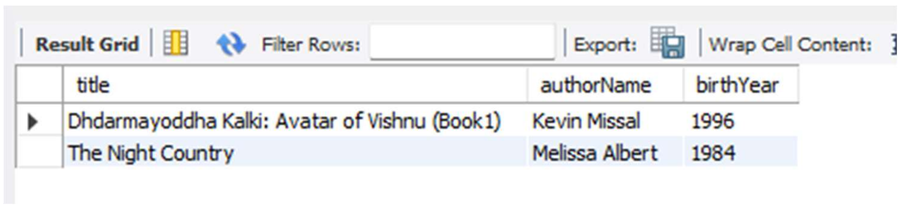
14. **Subquery with Join:**

- Write a query to find all Books that were written by authors born after 1970. Use a subquery in the WHERE clause to find these AuthorIDs.



```sql
/*  14. Subquery with Join:
        Write a query to find all Books that were written by authors born after 1970.
        Use a subquery in the WHERE clause to find these AuthorIDs. */
select title, authorName, birthYear
from books join authors
on author = authorName
where birthYear in (select distinct birthYear from authors
                            where birthYear > 1970);
```

Result:

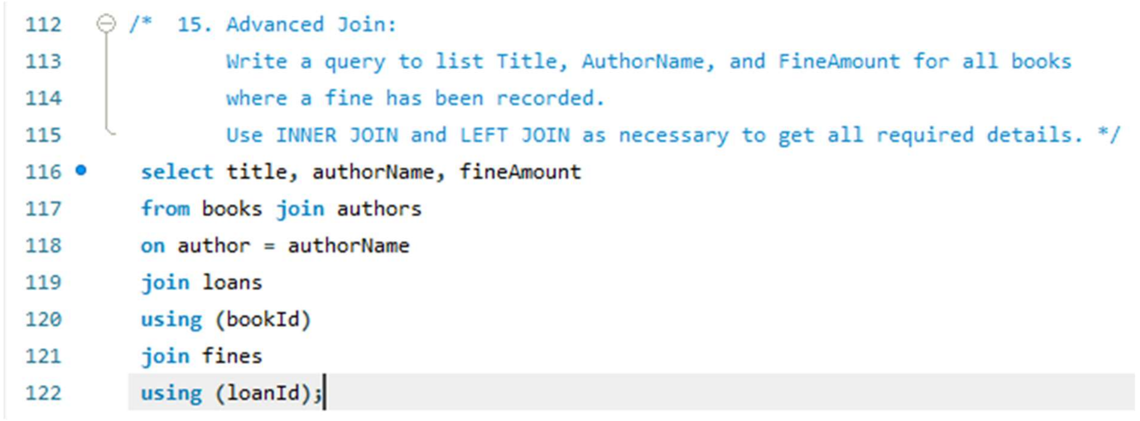


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

title	authorName	birthYear
Dhdarmayoddha Kalki: Avatar of Vishnu (Book1)	Kevin Missal	1996
The Night Country	Melissa Albert	1984

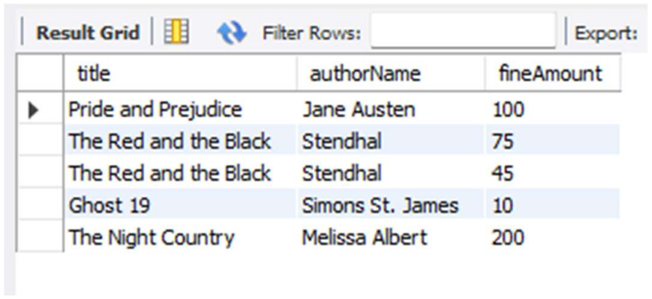
15. **Advanced Join:**

- Write a query to list Title, AuthorName, and FineAmount for all books where a fine has been recorded. Use INNER JOIN and LEFT JOIN as necessary to get all required details.



```sql
/*  15. Advanced Join:
        Write a query to list Title, AuthorName, and FineAmount for all books
        where a fine has been recorded.
        Use INNER JOIN and LEFT JOIN as necessary to get all required details. */
 select title, authorName, fineAmount
 from books join authors
 on author = authorName
 join loans
 using (bookId)
 join fines
 using (loanId);
```

Result:



Result Grid | Filter Rows: | Export:

title	authorName	fineAmount
Pride and Prejudice	Jane Austen	100
The Red and the Black	Stendhal	75
The Red and the Black	Stendhal	45
Ghost 19	Simons St. James	10
The Night Country	Melissa Albert	200