

Library Management System Queries using MYSQL

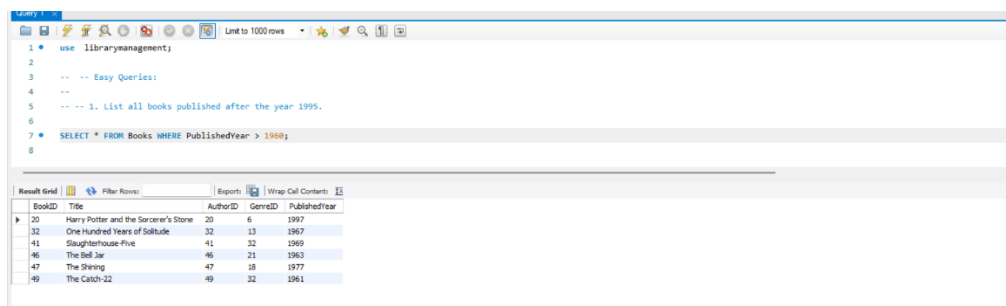
```
use librarymanagement; -- using the required dataset
```

-- -- Easy Queries:

--

-- -- 1. List all books published after the year 1995.

```
SELECT * FROM Books WHERE PublishedYear > 1960;
```



The screenshot shows the MySQL Workbench interface. The query editor at the top contains the following SQL code:

```
1 use librarymanagement;
2
3 -- -- Easy Queries:
4 --
5 -- -- 1. List all books published after the year 1995.
6
7 SELECT * FROM Books WHERE PublishedYear > 1960;
8
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query. The results are as follows:

BookID	Title	AuthorID	GenreID	PublishedYear
20	Harry Potter and the Sorcerer's Stone	20	6	1997
32	One Hundred Years of Solitude	32	13	1967
41	Slaughterhouse-Five	41	32	1969
46	The Bell Jar	46	21	1963
47	The Shining	47	18	1977
49	The Catch-22	49	32	1961

-- 2. Find the name and address of members who joined the library in 2023.

```
SELECT Name, Address FROM Members WHERE MembershipDate LIKE '%2023';
```



The screenshot shows the MySQL Workbench interface. The query editor at the top contains the following SQL code:

```
8
9
10 -- 2. Find the name and address of members who joined the library in 2023.
11
12 SELECT Name, Address FROM Members WHERE MembershipDate LIKE '%2023';
13
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query. The results are as follows:

Name	Address
Bob Smith	456 Oak St
David Brown	101 Maple St
Grace Lee	404 Walnut St
Kathy Turner	808 Redwood St
Paula Clark	1313 Cedar St
Steve Carter	1616 Maple St
Wendy Johnson	2020 Elm St
Aiden Russell	2424 Cedar St
Ethan Rivera	2828 Elm St
Gavin Perez	3030 Walnut St
Kayla Brooks	3434 Ash St
Olivia Barnes	3838 Walnut St
Sean Bell	4242 Ash St
Victor Young	4646 Spruce St

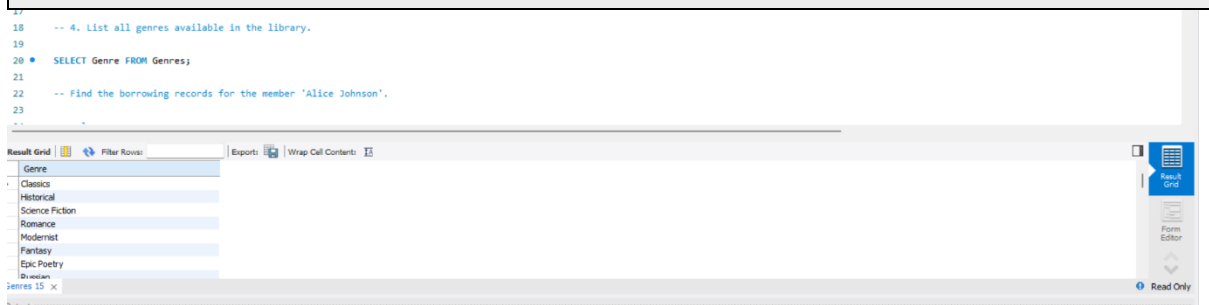
-- 3. Get the titles of all books authored by 'George Orwell'.

```
SELECT Title FROM Books WHERE AuthorID = (SELECT AuthorID FROM Authors WHERE Name = 'George Orwell');
```



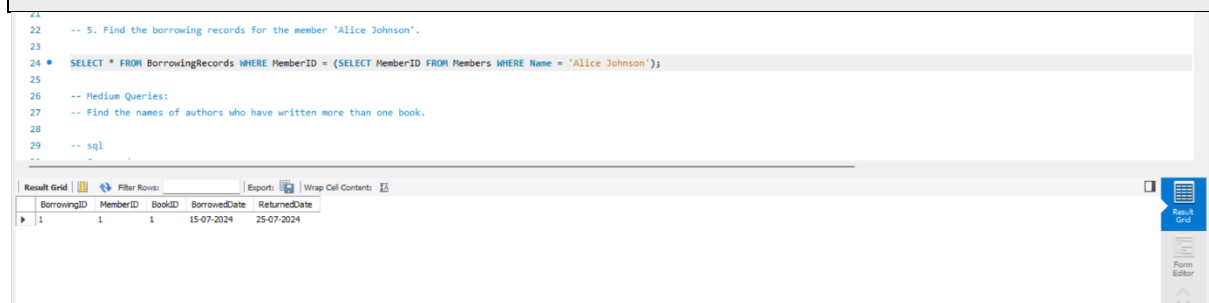
-- 4. List all genres available in the library.

SELECT Genre FROM Genres;



-- 5. Find the borrowing records for the member 'Alice Johnson'.

SELECT * FROM BorrowingRecords WHERE MemberID = (SELECT MemberID FROM Members WHERE Name = 'Alice Johnson');



-- Medium Queries:

-- 6. Find the names of authors who have written more than one book.

SELECT a.Name FROM Authors a
JOIN Books b ON a.AuthorID = b.AuthorID
GROUP BY a.Name
HAVING COUNT(b.BookID) > 1;

```

26
27 -- Medium Queries:
28 -- 6. Find the names of authors who have written more than one book.
29
30 • SELECT a.Name FROM Authors a
31 JOIN Books b ON a.AuthorID = b.AuthorID
32 GROUP BY a.Name
33 HAVING COUNT(b.BookID) > 1;

```

-- 7. Get the details of members who have borrowed books but have not yet returned them.

```

SELECT m.Name, b.Title, br.BorrowedDate
FROM BorrowingRecords br
JOIN Members m ON br.MemberID = m.MemberID
JOIN Books b ON br.BookID = b.BookID
WHERE br.ReturnedDate IS NULL;

```

```

35
36 -- 7. Get the details of members who have borrowed books but have not yet returned them.
37
38 • SELECT m.Name, b.Title, br.BorrowedDate
39 FROM BorrowingRecords br
40 JOIN Members m ON br.MemberID = m.MemberID
41 JOIN Books b ON br.BookID = b.BookID
42 WHERE br.ReturnedDate IS NULL;
43
44 -- List all books and their corresponding genres.
45
46 -- sql
47 -- Copy code
48 -- SELECT b.Title, g.Genre

```

-- 8. List all books and their corresponding genres.

```

SELECT b.Title, g.Genre
FROM Books b
JOIN Genres g ON b.GenreID = g.GenreID;

```

```

43
44 -- 8. List all books and their corresponding genres.
45
46 • SELECT b.Title, g.Genre
47 FROM Books b
48 JOIN Genres g ON b.GenreID = g.GenreID;
49
50 -- Find the most recent book borrowed by each member.
51

```

-- 9. Find the most recent book borrowed by each member.

```

SELECT m.Name, b.Title, MAX(br.BorrowedDate) AS LastBorrowedDate
FROM BorrowingRecords br
JOIN Members m ON br.MemberID = m.MemberID
JOIN Books b ON br.BookID = b.BookID
GROUP BY m.Name, b.Title;

```

```

50 -- 9. Find the most recent book borrowed by each member.
51
52 • SELECT m.Name, b.Title, MAX(br.BorrowedDate) AS LastBorrowedDate
53 FROM BorrowingRecords br
54 JOIN Members m ON br.MemberID = m.MemberID
55 JOIN Books b ON br.BookID = b.BookID
56 GROUP BY m.Name, b.Title;
57
58 -- Get the list of books that have never been borrowed.
59
60 -- sql

```

Name	Title	LastBorrowedDate
Alice Johnson	Moby Dick	15-07-2024
Bob Smith	War and Peace	01-08-2024
Carol White	Pride and Prejudice	20-07-2024
David Brown	The Great Gatsby	25-07-2024
Eve Davis	1984	02-08-2024
Frank Miller	Hamlet	30-07-2024
Grace Lee	The Odyssey	06-08-2024
Hank Williams	Crime and Punishment	15-07-2024
Ivy Green	Don Quixote	07-08-2024
Jack Hill	In Search of Lost Time	10-07-2024
Kathy Turner	The Divine Comedy	09-08-2024
Louis Wright	The Catcher in the Rye	12-07-2024
Mona Kim	Reverie New World	08-08-2024

-- 10. Get the list of books that have never been borrowed.

```

SELECT Title FROM Books
WHERE BookID NOT IN (SELECT BookID FROM BorrowingRecords);

```

```

57
58 -- 10. Get the list of books that have never been borrowed.
59
60 • SELECT Title FROM Books
61 WHERE BookID NOT IN (SELECT BookID FROM BorrowingRecords);
62
63 -- Find the number of books borrowed in each genre.

```

Title

-- 11. Find the number of books borrowed in each genre.

```

SELECT g.Genre, COUNT(br.BorrowingID) AS BooksBorrowed
FROM BorrowingRecords br
JOIN Books b ON br.BookID = b.BookID
JOIN Genres g ON b.GenreID = g.GenreID
GROUP BY g.Genre;

```

```

63 -- 11. Find the number of books borrowed in each genre.
64
65 • SELECT g.Genre, COUNT(br.BorrowingID) AS BooksBorrowed
66 FROM BorrowingRecords br
67 JOIN Books b ON br.BookID = b.BookID
68 JOIN Genres g ON b.GenreID = g.GenreID
69 GROUP BY g.Genre
70
71 -- List the names of members who have borrowed more than 5 books.
72

```

Genre	BooksBorrowed
Classics	1
Historical	1
Science Fiction	1
Romance	3
Modernist	2
Fantasy	3
Epic Poetry	2
Medieval	1
Tragedy	1
Spanish	1
French	4
Magical Realism	1
Fynstverhalet	1

-- 12. List the names of members who have borrowed more than 5 books.

```

SELECT m.Name, COUNT(br.BorrowingID) AS BooksBorrowed
FROM BorrowingRecords br
JOIN Members m ON br.MemberID = m.MemberID
GROUP BY m.Name
HAVING COUNT(br.BorrowingID) > 5;

```

```

78
79 -- 13. Find the title and author of the most borrowed book.
80
81 • SELECT b.Title, a.Name, COUNT(br.BorrowingID) AS TimesBorrowed
82 FROM BorrowingRecords br
83 JOIN Books b ON br.BookID = b.BookID
84 JOIN Authors a ON b.AuthorID = a.AuthorID
85 GROUP BY b.Title, a.Name
86 ORDER BY TimesBorrowed DESC
87 LIMIT 1;
88
89 -- Get the list of all members who borrowed books authored by 'Leo Tolstoy'.
90

```

Title	Name	TimesBorrowed
Moby Dick	Herman Melville	1

-- 13. Find the title and author of the most borrowed book.

```

SELECT b.Title, a.Name, COUNT(br.BorrowingID) AS TimesBorrowed
FROM BorrowingRecords br
JOIN Books b ON br.BookID = b.BookID
JOIN Authors a ON b.AuthorID = a.AuthorID
GROUP BY b.Title, a.Name
ORDER BY TimesBorrowed DESC
LIMIT 1;

```

```

78
79 -- 12. List the names of members who have borrowed more than 5 books.
80
81 • SELECT m.Name, COUNT(br.BorrowingID) AS BooksBorrowed
82 FROM BorrowingRecords br
83 JOIN Members m ON br.MemberID = m.MemberID
84 GROUP BY m.Name
85 HAVING COUNT(br.BorrowingID) > 5;
86
87 -- Find the title and author of the most borrowed book.
88

```

Name	BooksBorrowed
Uma Collins	2
Victor Young	2
Wendy Johnson	2
Xander Reed	2

-- 14. Get the list of all members who borrowed books authored by 'Leo Tolstoy'.

```

SELECT m.Name
FROM BorrowingRecords br
JOIN Members m ON br.MemberID = m.MemberID
JOIN Books b ON br.BookID = b.BookID
WHERE b.AuthorID = (SELECT AuthorID FROM Authors WHERE Name = 'Leo Tolstoy');

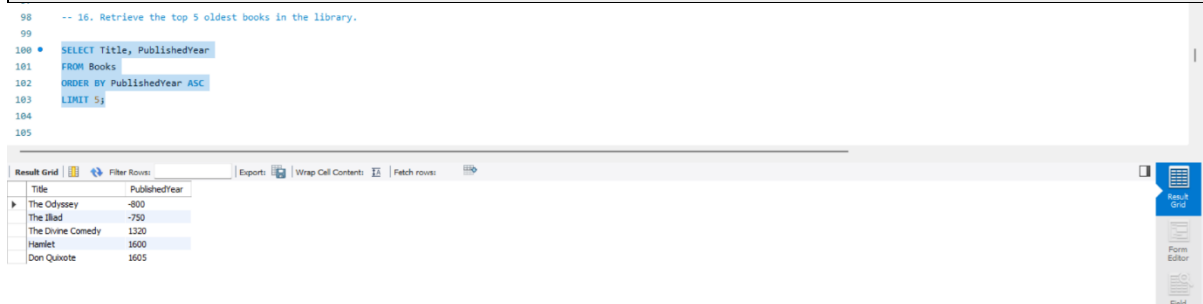
```

-- 15. Retrieve the top 5 oldest books in the library.

```

SELECT Title, PublishedYear
FROM Books
ORDER BY PublishedYear ASC
LIMIT 5;

```



-- Somewhat Hard Queries:

-- 16. Find the number of books borrowed each month in 2024.

```

SELECT DATE_FORMAT(BorrowedDate, '%Y-%m') AS Month, COUNT(BorrowingID) AS BooksBorrowed
FROM BorrowingRecords
WHERE BorrowedDate LIKE '%2024'
GROUP BY Month;

```

-- 16. List the members who have borrowed all books of a particular genre, e.g., 'Fantasy'.

```

SELECT m.Name
FROM Members m
WHERE NOT EXISTS (
    SELECT 1 FROM Books b
    JOIN Genres g ON b.GenreID = g.GenreID
    WHERE g.Genre = 'Fantasy' AND
    NOT EXISTS (
        SELECT 1 FROM BorrowingRecords br
        WHERE br.MemberID = m.MemberID AND br.BookID = b.BookID
    )
);

```

-- 17. Find the average number of books borrowed per member.

```
SELECT AVG(BorrowCount) AS AvgBooksBorrowed
FROM (
    SELECT COUNT(br.BorrowingID) AS BorrowCount
    FROM BorrowingRecords br
    GROUP BY br.MemberID
) AS BorrowCounts;
```

127
128 -- 17. Find the average number of books borrowed per member.
129
130 • SELECT AVG(BorrowCount) AS AvgBooksBorrowed
131 FROM (
132 SELECT COUNT(br.BorrowingID) AS BorrowCount
133 FROM BorrowingRecords br
134 GROUP BY br.MemberID
135) AS BorrowCounts;
136
137 -- Retrieve the books that have been borrowed by at least three different members.
138
139 -- sql
140 -- From code

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

AvgBooksBorrowed
1.0000

Result Grid
Form Editor

-- 18. Retrieve the books that have been borrowed by at least three different members.

```
SELECT b.Title
FROM BorrowingRecords br
JOIN Books b ON br.BookID = b.BookID
GROUP BY b.Title
HAVING COUNT(DISTINCT br.MemberID) >= 3;
```

137 -- 18. Retrieve the books that have been borrowed by at least three different members.
138
139 • SELECT b.Title
140 FROM BorrowingRecords br
141 JOIN Books b ON br.BookID = b.BookID
142 GROUP BY b.Title
143 HAVING COUNT(DISTINCT br.MemberID) >= 3;
144
145 -- List the genres that have never been borrowed.
146
147 -- sql
148 -- From code

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

Title
1984
A Tale of Two Cities
Anna Karenina
Brave New World
Crime and Punishment
Don Quixote
Dracula
Fahrenheit 451
Frankenstein

Result 42 x | Read Only

-- 19. List the genres that have never been borrowed

```
SELECT g.Genre
FROM Genres g
WHERE g.GenreID NOT IN (
    SELECT b.GenreID FROM BorrowingRecords br
    JOIN Books b ON br.BookID = b.BookID
);
```

```

144
145 -- 19. List the genres that have never been borrowed.
146
147 • SELECT g.Genre
148 FROM Genres g
149 WHERE g.GenreID NOT IN (
150     SELECT b.GenreID FROM BorrowingRecords br
151     JOIN Books b ON br.BookID = b.BookID
152 )
153
154 -- Find the authors who have books in more than two different genres.

```

Result Grid

Genre
Russian
Mystery
Thriller
Western
Utopian
Biography
Autobiography
Memoir
Short Stories

Genres 43 x

-- 20. Find the authors who have books in more than two different genres.

```

SELECT a.Name
FROM Authors a
JOIN Books b ON a.AuthorID = b.AuthorID
JOIN Genres g ON b.GenreID = g.GenreID
GROUP BY a.Name
HAVING COUNT(DISTINCT g.GenreID) > 1;

```

```

154 -- 20. Find the authors who have books in more than two different genres.
155
156 • SELECT a.Name
157 FROM Authors a
158 JOIN Books b ON a.AuthorID = b.AuthorID
159 JOIN Genres g ON b.GenreID = g.GenreID
160 GROUP BY a.Name
161 HAVING COUNT(DISTINCT g.GenreID) > 1;
162
163 -- Get the details of members who have borrowed more than one book on the same day.
164
165 -- sql
166 -- from code

```

Result Grid

Name
Charles Dickens
Emily Brontë
Gustave Flaubert
Herman Melville
James Joyce
Joseph Conrad
Leo Tolstoy
William Faulkner

Result 45 x

-- 21. Get the details of members who have borrowed more than one book on the same day.

```

SELECT m.Name, br.BorrowedDate, COUNT(br.BookID) AS BooksBorrowed
FROM BorrowingRecords br
JOIN Members m ON br.MemberID = m.MemberID
GROUP BY m.Name, br.BorrowedDate
HAVING COUNT(br.BookID) >= 1;

```

```

163 -- 21. Get the details of members who have borrowed more than one book on the same day.
164
165 • SELECT m.Name, br.BorrowedDate, COUNT(br.BookID) AS BooksBorrowed
166 FROM BorrowingRecords br
167 JOIN Members m ON br.MemberID = m.MemberID
168 GROUP BY m.Name, br.BorrowedDate
169 HAVING COUNT(br.BookID) >= 1;
170
171 -- Retrieve the list of books borrowed by members who joined in the same year as the book's published year.
172
173

```

Result Grid

Name	BorrowedDate	BooksBorrowed
Alice Johnson	15-07-2024	1
Bob Smith	01-08-2024	1
Carol White	20-07-2024	1
David Brown	25-07-2024	1
Eve Davis	02-08-2024	1
Frank Miller	30-07-2024	1
Grace Lee	06-08-2024	1
Heck Williams	15-07-2024	1
Ivy Green	07-08-2024	1

Result 47 x

-- 22. Retrieve the list of books borrowed by members who joined in the same year as the book's published year.

```
SELECT b.Title, m.Name, m.MembershipDate, b.PublishedYear
FROM BorrowingRecords br
JOIN Members m ON br.MemberID = m.MemberID
JOIN Books b ON br.BookID = b.BookID
WHERE YEAR(m.MembershipDate) = b.PublishedYear;
```

```
170
171 -- 22. Retrieve the list of books borrowed by members who joined in the same year as the book's published year.
172
173 * SELECT b.Title, m.Name, m.MembershipDate, b.PublishedYear
174 FROM BorrowingRecords br
175 JOIN Members m ON br.MemberID = m.MemberID
176 JOIN Books b ON br.BookID = b.BookID
177 WHERE YEAR(m.MembershipDate) = b.PublishedYear;
178
179 -- Find the member who has borrowed the most different genres.
180
181 -- end
```

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

Title	Name	MembershipDate	PublishedYear
-------	------	----------------	---------------

-- 23. Find the member who has borrowed the most different genres.

```
SELECT m.Name, COUNT(DISTINCT g.GenreID) AS DifferentGenres
FROM BorrowingRecords br
JOIN Members m ON br.MemberID = m.MemberID
JOIN Books b ON br.BookID = b.BookID
JOIN Genres g ON b.GenreID = g.GenreID
GROUP BY m.Name
ORDER BY DifferentGenres DESC
LIMIT 1;
```

```
178
179 -- 23. Find the member who has borrowed the most different genres.
180
181 * SELECT m.Name, COUNT(DISTINCT g.GenreID) AS DifferentGenres
182 FROM BorrowingRecords br
183 JOIN Members m ON br.MemberID = m.MemberID
184 JOIN Books b ON br.BookID = b.BookID
185 JOIN Genres g ON b.GenreID = g.GenreID
186 GROUP BY m.Name
187 ORDER BY DifferentGenres DESC
188 LIMIT 1;
189
190 -- list the names of members who have borrowed both 'Classics' and 'Science Fiction' genres
```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: | Patch rows: |

Name	DifferentGenres
Uma Collins	2

-- 24. List the names of members who have borrowed both 'Classics' and 'Science Fiction' genres.

```
SELECT m.Name
FROM Members m
WHERE EXISTS (
    SELECT 1 FROM BorrowingRecords br
    JOIN Books b ON br.BookID = b.BookID
    WHERE br.MemberID = m.MemberID AND b.GenreID = (SELECT GenreID FROM Genres WHERE Genre =
'Classics')
```

```

) AND EXISTS (
  SELECT 1 FROM BorrowingRecords br
  JOIN Books b ON br.BookID = b.BookID
  WHERE br.MemberID = m.MemberID AND b.GenreID = (SELECT GenreID FROM Genres WHERE Genre =
'Science Fiction')
);

```

```

193 FROM Members m
194 WHERE EXISTS (
195   SELECT 1 FROM BorrowingRecords br
196   JOIN Books b ON br.BookID = b.BookID
197   WHERE br.MemberID = m.MemberID AND b.GenreID = (SELECT GenreID FROM Genres WHERE Genre = 'Classics')
198 ) AND EXISTS (
199   SELECT 1 FROM BorrowingRecords br
200   JOIN Books b ON br.BookID = b.BookID
201   WHERE br.MemberID = m.MemberID AND b.GenreID = (SELECT GenreID FROM Genres WHERE Genre = 'Science Fiction')
202 )
203
204

```

25. List the genres with the highest average number of books borrowed per title.

```

SELECT g.Genre, AVG(BorrowCount) AS AvgBooksBorrowed
FROM (
  SELECT g.GenreID, COUNT(br.BorrowingID) AS BorrowCount
  FROM BorrowingRecords br
  JOIN Books b ON br.BookID = b.BookID
  JOIN Genres g ON b.GenreID = g.GenreID
  GROUP BY g.GenreID, b.Title
) AS GenreBorrowCounts
JOIN Genres g ON GenreBorrowCounts.GenreID = g.GenreID
GROUP BY g.Genre
ORDER BY AvgBooksBorrowed DESC;

```

```

193 FROM Members m
194 WHERE EXISTS (
195   SELECT 1 FROM BorrowingRecords br
196   JOIN Books b ON br.BookID = b.BookID
197   WHERE br.MemberID = m.MemberID AND b.GenreID = (SELECT GenreID FROM Genres WHERE Genre = 'Classics')
198 ) AND EXISTS (
199   SELECT 1 FROM BorrowingRecords br
200   JOIN Books b ON br.BookID = b.BookID
201   WHERE br.MemberID = m.MemberID AND b.GenreID = (SELECT GenreID FROM Genres WHERE Genre = 'Science Fiction')
202 )
203
204 SELECT g.Genre, AVG(BorrowCount) AS AvgBooksBorrowed
205

```

Genre	AvgBooksBorrowed
Classics	1.0000
Historical	1.0000
Science Fiction	1.0000
Romance	1.0000
Modernist	1.0000
Fantasy	1.0000
Epic Poetry	1.0000
Medieval	1.0000
Tragedy	1.0000