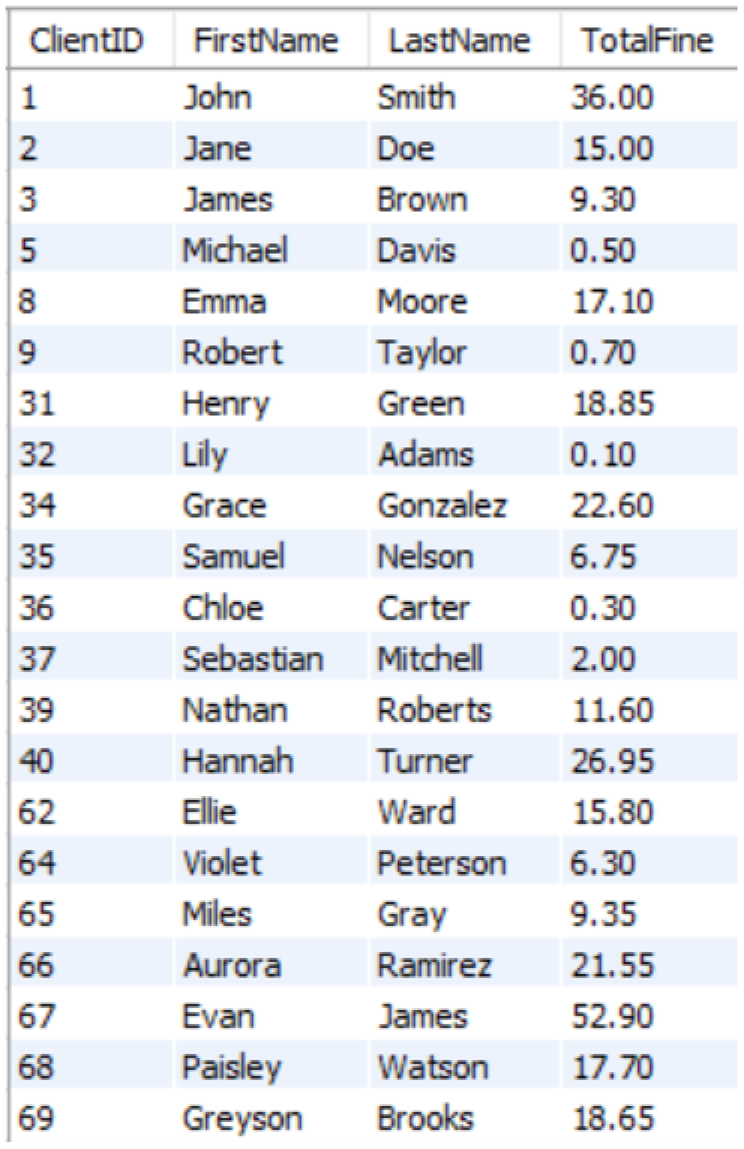
Verification Queries:

1. **Fine Calculation:** Calculate the total fines owed by each member, considering overdue books and a daily fine rate (e.g., $0.25 per day).

```
SELECT c.ClientID, c.FirstName, c.LastName, SUM(l.TotalLateFee) AS TotalFine
FROM Client AS c
JOIN Loans AS l ON c.ClientID = l.ClientID
WHERE l.TotalLateFee > 0
GROUP BY c.ClientID, c.FirstName, c.LastName
ORDER BY ClientID ASC;
```

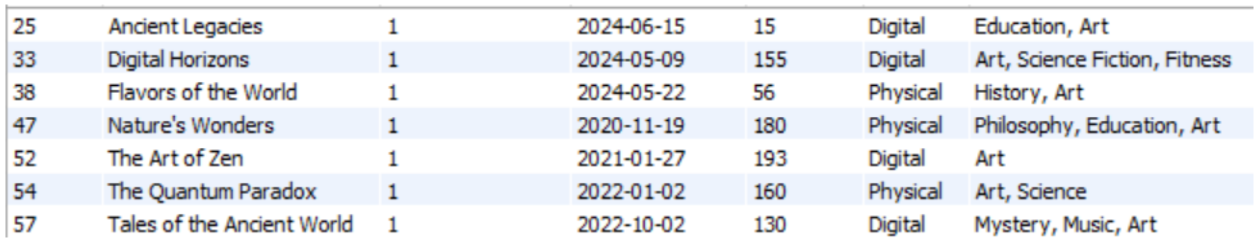


ClientID	FirstName	LastName	TotalFine
1	John	Smith	36.00
2	Jane	Doe	15.00
3	James	Brown	9.30
5	Michael	Davis	0.50
8	Emma	Moore	17.10
9	Robert	Taylor	0.70
31	Henry	Green	18.85
32	Lily	Adams	0.10
34	Grace	Gonzalez	22.60
35	Samuel	Nelson	6.75
36	Chloe	Carter	0.30
37	Sebastian	Mitchell	2.00
39	Nathan	Roberts	11.60
40	Hannah	Turner	26.95
62	Ellie	Ward	15.80
64	Violet	Peterson	6.30
65	Miles	Gray	9.35
66	Aurora	Ramirez	21.55
67	Evan	James	52.90
68	Paisley	Watson	17.70
69	Greyson	Brooks	18.65

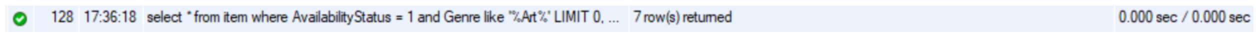
```
127  17:34:48  SELECT c.ClientID, c.FirstName, c.LastName, SUM(l.TotalLateFee) AS TotalF...  22 row(s) returned    0.000 sec / 0.000 sec
```

2. **Book Availability:** Display a list of all available books (not currently borrowed) within a specific genre.

```
select *
from item
where AvailabilityStatus = 1 and Genre like '%Art%'
```



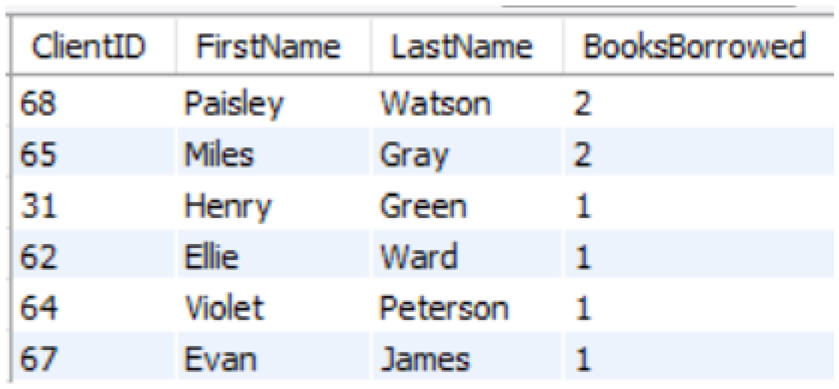
25	Ancient Legacies	1	2024-06-15	15	Digital	Education, Art
33	Digital Horizons	1	2024-05-09	155	Digital	Art, Science Fiction, Fitness
38	Flavors of the World	1	2024-05-22	56	Physical	History, Art
47	Nature's Wonders	1	2020-11-19	180	Physical	Philosophy, Education, Art
52	The Art of Zen	1	2021-01-27	193	Digital	Art
54	The Quantum Paradox	1	2022-01-02	160	Physical	Art, Science
57	Tales of the Ancient World	1	2022-10-02	130	Digital	Mystery, Music, Art



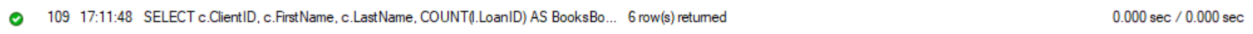
```
128  17:36:18  select * from item where AvailabilityStatus = 1 and Genre like '%Art%' LIMIT 0, ...  7 row(s) returned  0.000 sec / 0.000 sec
```

3. **Frequent Borrowers of a Specific Genre:** Identify the members who have borrowed the most books in a particular genre (e.g., "Mystery") in the last year.

```
SELECT c.ClientID, c.FirstName, c.LastName, COUNT(l.LoanID) AS BooksBorrowed
FROM Client AS c
JOIN Loans AS l ON c.ClientID = l.ClientID
JOIN Item AS i ON l.ItemID = i.ItemID
WHERE i.Genre LIKE '%Romance%'
GROUP BY c.ClientID
ORDER BY BooksBorrowed DESC;
```



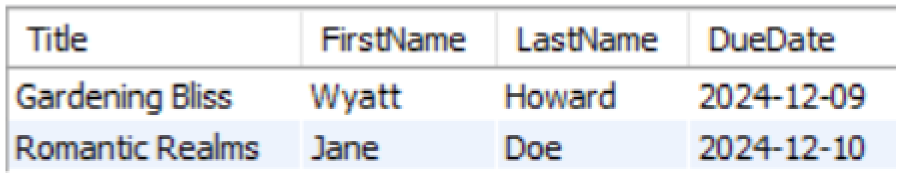
ClientID	FirstName	LastName	BooksBorrowed
68	Paisley	Watson	2
65	Miles	Gray	2
31	Henry	Green	1
62	Ellie	Ward	1
64	Violet	Peterson	1
67	Evan	James	1



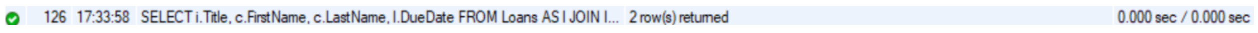
```
109  17:11:48  SELECT c.ClientID, c.FirstName, c.LastName, COUNT(l.LoanID) AS BooksBo...  6 row(s) returned        0.000 sec / 0.000 sec
```

4. **Books Due Soon:** Generate a report of all books due within the next week, sorted by due date.

```
SELECT i.Title, c.FirstName, c.LastName, I.DueDate
FROM Loans AS I
JOIN Item AS i ON I.ItemID = i.ItemID
JOIN Client AS c ON I.ClientID = c.ClientID
WHERE I.DueDate >= CURDATE() AND I.DueDate <= CURDATE() + 7
ORDER BY I.DueDate;
```



Title	FirstName	LastName	DueDate
Gardening Bliss	Wyatt	Howard	2024-12-09
Romantic Realms	Jane	Doe	2024-12-10



```
126  17:33:58  SELECT i.Title, c.FirstName, c.LastName, I.DueDate FROM Loans AS I JOIN I...  2 row(s) returned  0.000 sec / 0.000 sec
```

5. **Members with Overdue Books:** List all members who currently have at least one overdue book, along with the titles of the overdue books.

SELECT c.ClientID, c.FirstName, c.LastName, i.Title

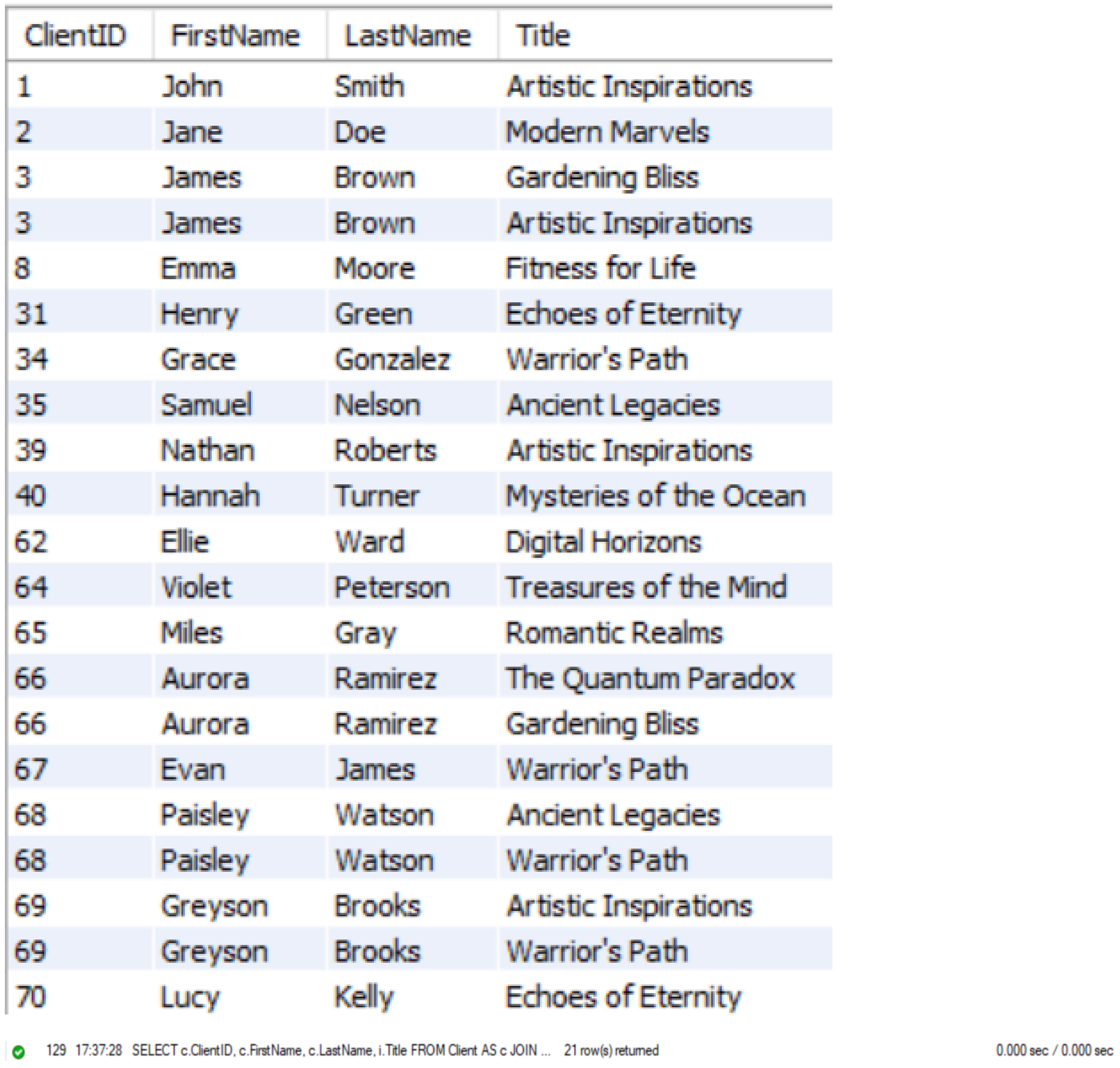
FROM Client AS c

JOIN Loans AS l ON c.ClientID = l.ClientID

JOIN Item AS i ON l.ItemID = i.ItemID

WHERE l.DueDate < CURDATE() AND l.ReturnDate IS NULL

ORDER BY c.ClientID;



ClientID	FirstName	LastName	Title
1	John	Smith	Artistic Inspirations
2	Jane	Doe	Modern Marvels
3	James	Brown	Gardening Bliss
3	James	Brown	Artistic Inspirations
8	Emma	Moore	Fitness for Life
31	Henry	Green	Echoes of Eternity
34	Grace	Gonzalez	Warrior's Path
35	Samuel	Nelson	Ancient Legacies
39	Nathan	Roberts	Artistic Inspirations
40	Hannah	Turner	Mysteries of the Ocean
62	Ellie	Ward	Digital Horizons
64	Violet	Peterson	Treasures of the Mind
65	Miles	Gray	Romantic Realms
66	Aurora	Ramirez	The Quantum Paradox
66	Aurora	Ramirez	Gardening Bliss
67	Evan	James	Warrior's Path
68	Paisley	Watson	Ancient Legacies
68	Paisley	Watson	Warrior's Path
69	Greyson	Brooks	Artistic Inspirations
69	Greyson	Brooks	Warrior's Path
70	Lucy	Kelly	Echoes of Eternity

```
129  17:37:28  SELECT c.ClientID, c.FirstName, c.LastName, i.Title FROM Client AS c JOIN ...  21 row(s) returned  0.000 sec / 0.000 sec
```

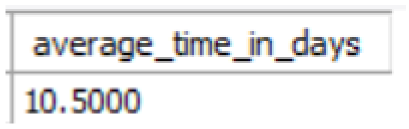
6. **Average Borrowing Time:** Calculate the average number of days members borrow books for a specific genre.

The Example here looks for the Romance genre. This can be replaced with other genres.

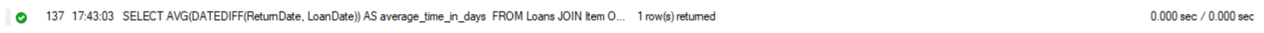
```
SELECT AVG(DATEDIFF(ReturnDate, LoanDate)) AS average_time_in_days

FROM Loans JOIN Item ON Loans.ItemID = Item.ItemID

WHERE Item.Genre like '%Romance%';
```



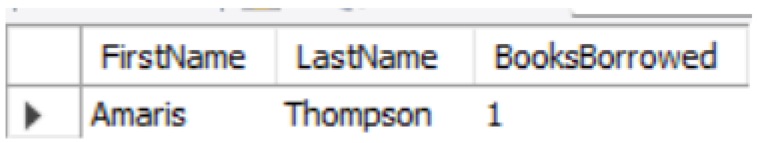
average_time_in_days
10.5000



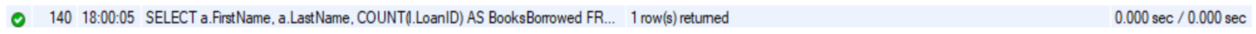
```
137  17:43:03  SELECT AVG(DATEDIFF(ReturnDate, LoanDate)) AS average_time_in_days  FROM Loans JOIN Item O...   1 row(s) returned                                  0.000 sec / 0.000 sec
```

7. **Most Popular Author in the Last Month:** Determine the author whose books have been borrowed the most in the last month.

```
SELECT a.FirstName, a.LastName, COUNT(l.LoanID) AS BooksBorrowed
FROM Loans AS l
JOIN Item AS i ON l.ItemID = i.ItemID
JOIN Author AS a ON i.AuthorID = a.AuthorID
WHERE l.LoanDate >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH)
AND l.ReturnDate IS NOT NULL
GROUP BY a.AuthorID
ORDER BY BooksBorrowed DESC
LIMIT 1;
```



	FirstName	LastName	BooksBorrowed
▶	Amaris	Thompson	1



```
140  18:00:05  SELECT a.FirstName, a.LastName, COUNT(l.LoanID) AS BooksBorrowed FR...  1 row(s) returned  0.000 sec / 0.000 sec
```

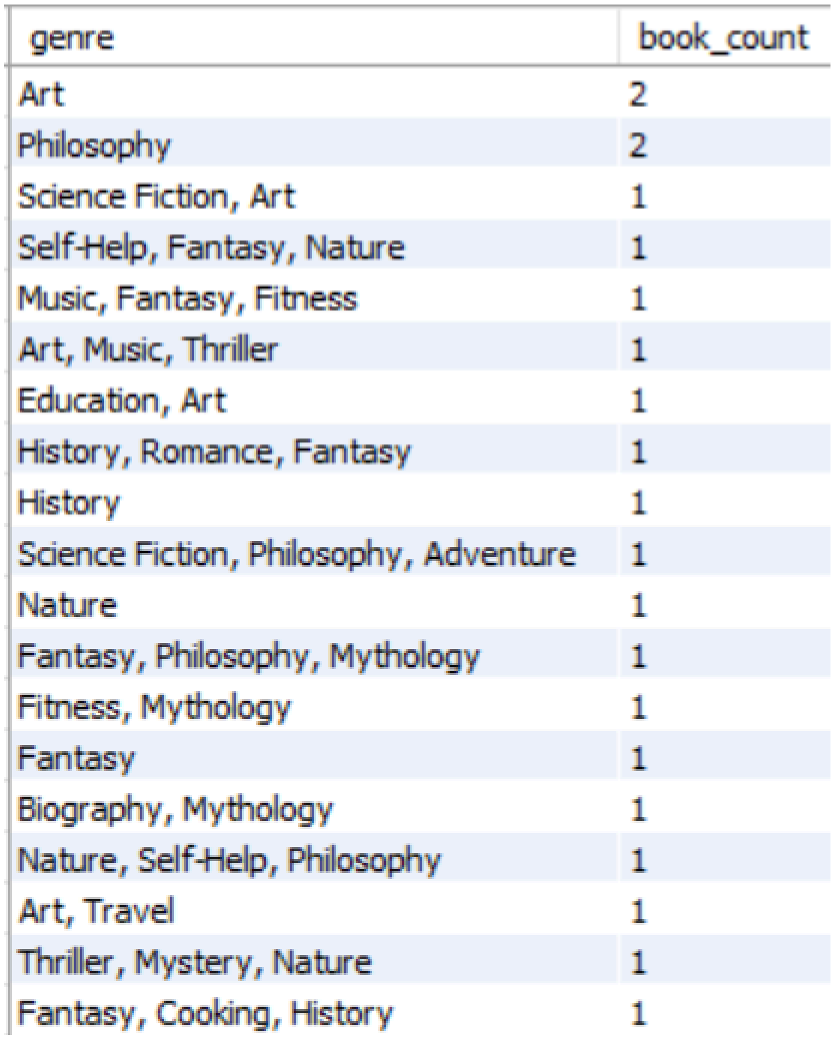
SQL reports (30 points)

For our report we chose the topic below:

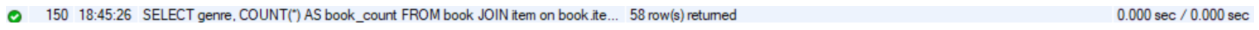
Generate a Collection Analysis Report. This report should provide a comprehensive analysis of the library's book collection, examining the distribution of books by genre, identifying trends in acquisition over the past 5 years, and assessing the age of the collection to identify outdated materials. Highlight books with low circulation and analyze borrowing patterns to identify under-represented genres or authors. Your report should provide insights for collection development and management decisions.

1. The first query examines the distribution of books by genre, showing the most popular ones. For our database, it is realistic, meaning that a book can have multiple genres, making it harder to examine the most popular one. Nonetheless it shows the most popular single genre.

```
SELECT genre, COUNT(*) AS book_count
FROM book
JOIN item on book.itemID = item.itemID
GROUP BY genre
ORDER BY book_count DESC;
```



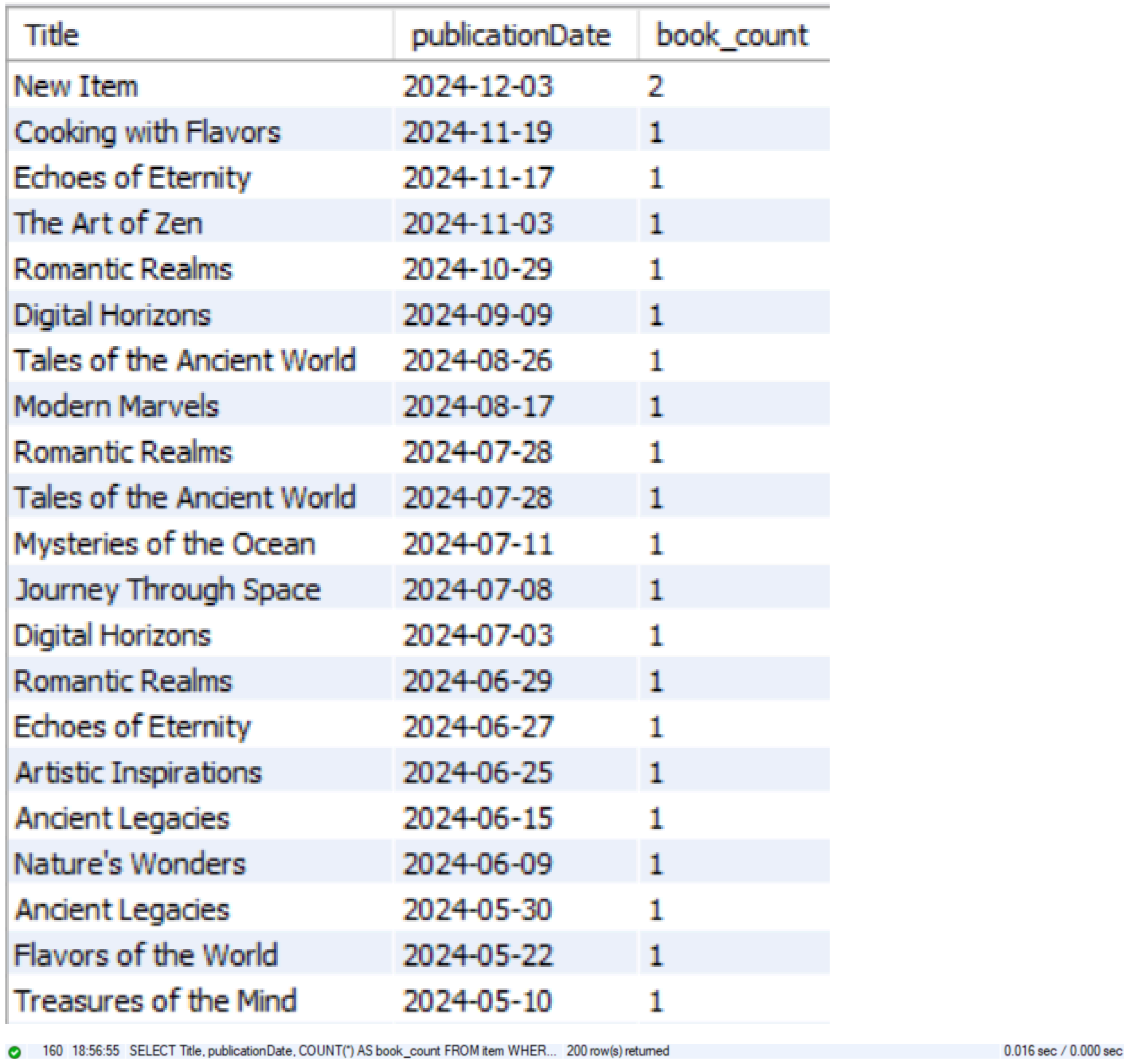
genre	book_count
Art	2
Philosophy	2
Science Fiction, Art	1
Self-Help, Fantasy, Nature	1
Music, Fantasy, Fitness	1
Art, Music, Thriller	1
Education, Art	1
History, Romance, Fantasy	1
History	1
Science Fiction, Philosophy, Adventure	1
Nature	1
Fantasy, Philosophy, Mythology	1
Fitness, Mythology	1
Fantasy	1
Biography, Mythology	1
Nature, Self-Help, Philosophy	1
Art, Travel	1
Thriller, Mystery, Nature	1
Fantasy, Cooking, History	1



```
150 18:45:26 SELECT genre, COUNT(*) AS book_count FROM book JOIN item on book.ite... 58 row(s) returned 0.000 sec / 0.000 sec
```

2. For our second query we show the trends in acquisition over the past 5 years, which shows the count of books the database has. Our results show that only one book in our database with the title ‘new item’ has more than one copy acquired.

```
SELECT publication_year, COUNT(*) AS book_count
FROM item
WHERE publication_year >= YEAR(CURDATE()) - 5
GROUP BY publication_year
ORDER BY publication_year DESC;
```

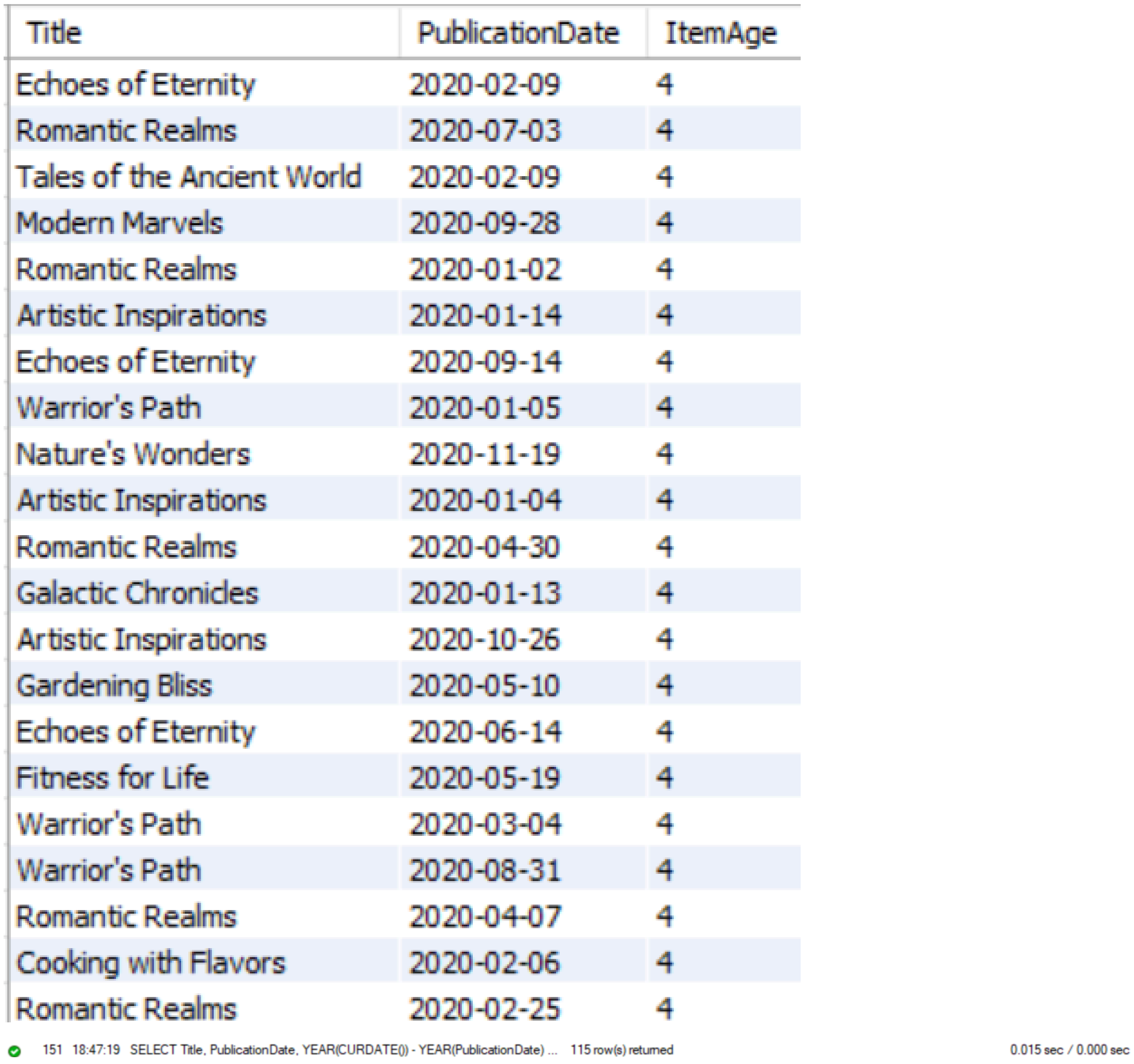


Title	publicationDate	book_count
New Item	2024-12-03	2
Cooking with Flavors	2024-11-19	1
Echoes of Eternity	2024-11-17	1
The Art of Zen	2024-11-03	1
Romantic Realms	2024-10-29	1
Digital Horizons	2024-09-09	1
Tales of the Ancient World	2024-08-26	1
Modern Marvels	2024-08-17	1
Romantic Realms	2024-07-28	1
Tales of the Ancient World	2024-07-28	1
Mysteries of the Ocean	2024-07-11	1
Journey Through Space	2024-07-08	1
Digital Horizons	2024-07-03	1
Romantic Realms	2024-06-29	1
Echoes of Eternity	2024-06-27	1
Artistic Inspirations	2024-06-25	1
Ancient Legacies	2024-06-15	1
Nature's Wonders	2024-06-09	1
Ancient Legacies	2024-05-30	1
Flavors of the World	2024-05-22	1
Treasures of the Mind	2024-05-10	1

```
160  18:56:55  SELECT Title, publicationDate, COUNT(*) AS book_count FROM item WHER...  200 row(s) returned    0.016 sec / 0.000 sec
```

3. This third query shows the age of the collection to identify outdated materials. In simpler terms, any book which is older than the average age is shown in the results table from the query below.

```
SELECT Title, PublicationDate, YEAR(CURDATE()) - YEAR(PublicationDate) AS ItemAge
FROM Item
HAVING ItemAge > (SELECT AVG(YEAR(CURDATE()) - YEAR(PublicationDate)) FROM Item)
ORDER BY ItemAge DESC;
```

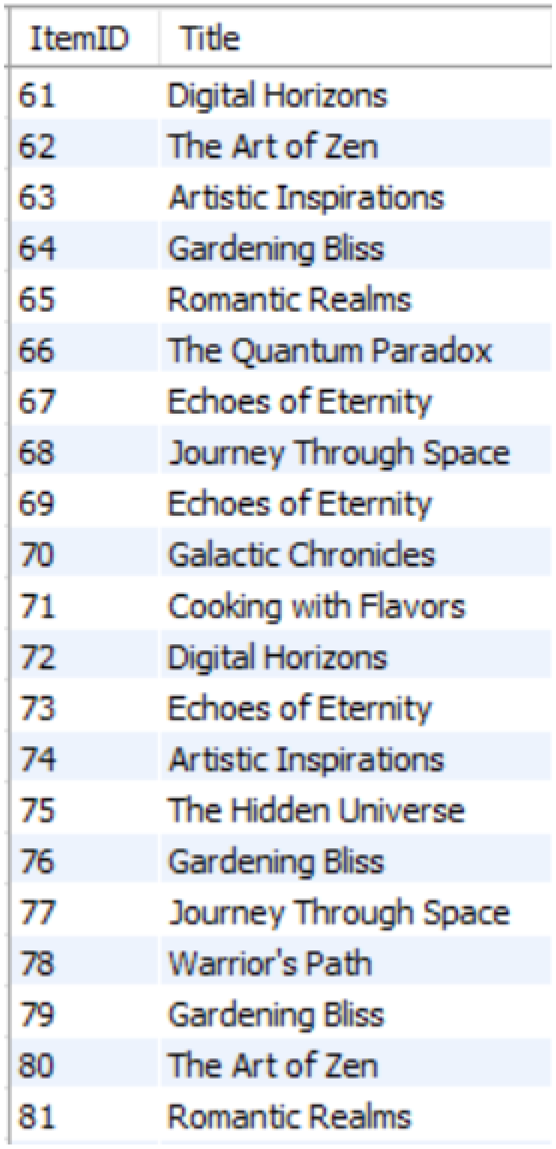


Title	PublicationDate	ItemAge
Echoes of Eternity	2020-02-09	4
Romantic Realms	2020-07-03	4
Tales of the Ancient World	2020-02-09	4
Modern Marvels	2020-09-28	4
Romantic Realms	2020-01-02	4
Artistic Inspirations	2020-01-14	4
Echoes of Eternity	2020-09-14	4
Warrior's Path	2020-01-05	4
Nature's Wonders	2020-11-19	4
Artistic Inspirations	2020-01-04	4
Romantic Realms	2020-04-30	4
Galactic Chronicles	2020-01-13	4
Artistic Inspirations	2020-10-26	4
Gardening Bliss	2020-05-10	4
Echoes of Eternity	2020-06-14	4
Fitness for Life	2020-05-19	4
Warrior's Path	2020-03-04	4
Warrior's Path	2020-08-31	4
Romantic Realms	2020-04-07	4
Cooking with Flavors	2020-02-06	4
Romantic Realms	2020-02-25	4

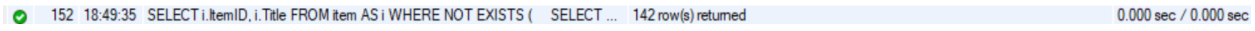
```
151  18:47:19  SELECT Title, PublicationDate, YEAR(CURDATE()) - YEAR(PublicationDate) ...   115 row(s) returned   0.015 sec / 0.000 sec
```

4. This fourth query shows highlighted books with low circulation, meaning that what is returned is books without a loan.

```
SELECT i.ItemID, i.Title
FROM item AS i
WHERE NOT EXISTS (
   SELECT 1
   FROM loans AS l
   WHERE l.ItemID = i.ItemID
);
```



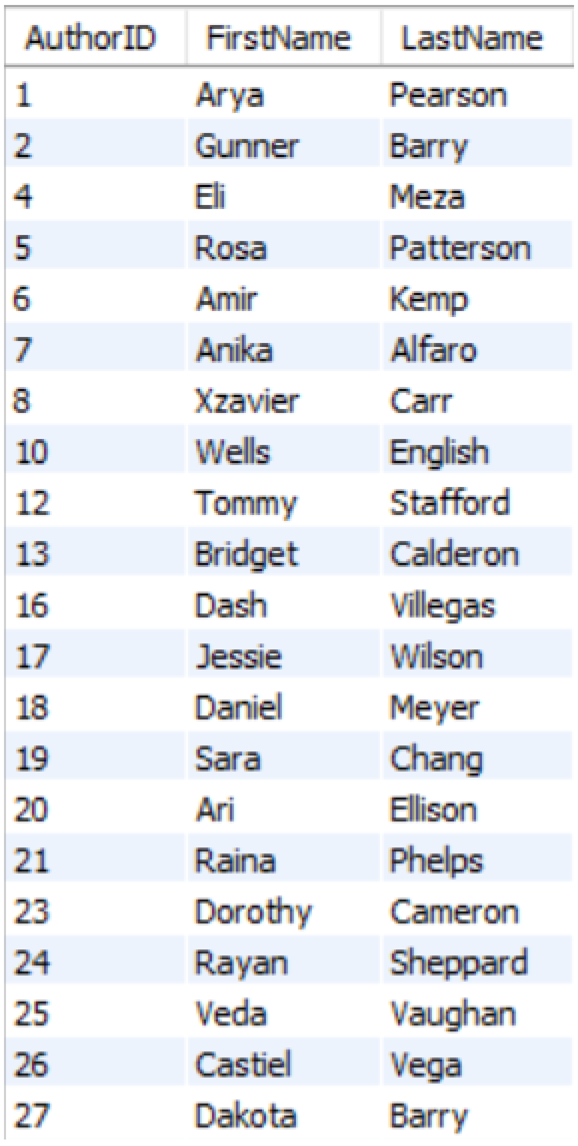
ItemID	Title
61	Digital Horizons
62	The Art of Zen
63	Artistic Inspirations
64	Gardening Bliss
65	Romantic Realms
66	The Quantum Paradox
67	Echoes of Eternity
68	Journey Through Space
69	Echoes of Eternity
70	Galactic Chronicles
71	Cooking with Flavors
72	Digital Horizons
73	Echoes of Eternity
74	Artistic Inspirations
75	The Hidden Universe
76	Gardening Bliss
77	Journey Through Space
78	Warrior's Path
79	Gardening Bliss
80	The Art of Zen
81	Romantic Realms



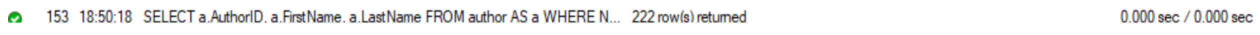
```
152   18:49:35   SELECT i.ItemID, i.Title FROM item AS i WHERE NOT EXISTS (   SELECT ...   142 row(s) returned   0.000 sec / 0.000 sec
```

5. This fifth query analyzes borrowing patterns to identify under-represented authors, which the table shows that because there is an overwhelming amount of authors in our database, a lot go under-represented.

```
SELECT a.AuthorID, a.FirstName, a.LastName
FROM author AS a
WHERE NOT EXISTS (
   SELECT 1
   FROM item AS i
   JOIN loans AS l ON i.ItemID = l.ItemID
   WHERE i.AuthorID = a.AuthorID
);
```



AuthorID	FirstName	LastName
1	Arya	Pearson
2	Gunner	Barry
4	Eli	Meza
5	Rosa	Patterson
6	Amir	Kemp
7	Anika	Alfaro
8	Xzavier	Carr
10	Wells	English
12	Tommy	Stafford
13	Bridget	Calderon
16	Dash	Villegas
17	Jessie	Wilson
18	Daniel	Meyer
19	Sara	Chang
20	Ari	Ellison
21	Raina	Phelps
23	Dorothy	Cameron
24	Rayan	Sheppard
25	Veda	Vaughan
26	Castiel	Vega
27	Dakota	Barry



```
153   18:50:18   SELECT a.AuthorID, a.FirstName, a.LastName FROM author AS a WHERE N...   222 row(s) returned        0.000 sec / 0.000 sec
```

Only return books info

```
SELECT Item.ItemID, Title, AvailabilityStatus PublicationDate, AuthorID, Type, Genre, ISBN
FROM Item
JOIN Book
ON Item.ItemID = Book.ItemID;
```

Only return magazines info

```
SELECT Item.ItemID, Title, AvailabilityStatus PublicationDate, AuthorID, Type, Genre,
IssueNumber
FROM Item
JOIN Magazine
ON Item.ItemID = Magazine.ItemID;
```

Only return digital media info

```
SELECT Item.ItemID, Title, AvailabilityStatus PublicationDate, AuthorID, Type, Genre,
MediaType
FROM Item
JOIN DigitalMedia
ON Item.ItemID = DigitalMedia.ItemID;
```

Return the name of who reserved what item title

```
SELECT r.ReservationID, c.FirstName, c.LastName, i.Title
FROM Reserves r
JOIN Client c ON r.ClientID = c.ClientID
JOIN Item i ON r.ItemID = i.ItemID;
```

Author creates items

```
SELECT a.FirstName, a.LastName, i.Title, i.ItemID
FROM Creates c
JOIN Author a ON c.AuthorID = a.AuthorID
JOIN Item i ON c.ItemID = i.ItemID;
```

Return the late fees and the names of the clients who owe them

```
SELECT C.FirstName, C.LastName, SUM(L.LateFee) AS Charges
FROM Loans L
JOIN Client C ON L.ClientID=C.ClientID
```

```
WHERE L.LateFee IS NOT NULL
GROUP BY C.ClientID
```

Return the accounts of people who are suspended

```
SELECT *
FROM Client
WHERE accountstatus = 'Suspended';
```

Return the client who has loaned the most items and the amount of items they have loaned

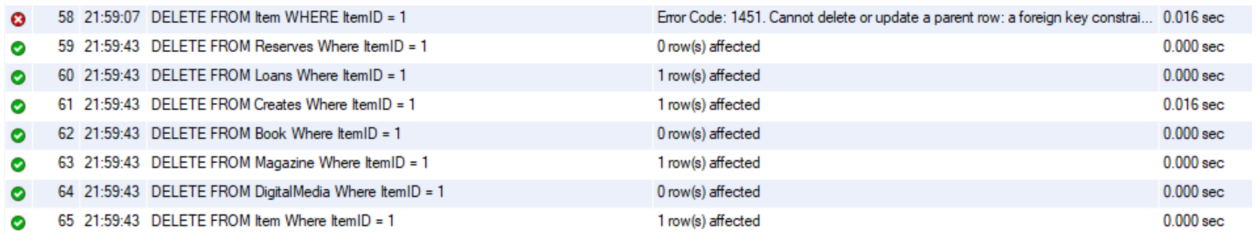
```
SELECT C.FirstName, C.LastName, COUNT(L.ItemID) AS Items
FROM Loans L
JOIN Client C ON C.ClientID = L.ClientID
GROUP BY C.FirstName, C.LastName
ORDER BY Items DESC
LIMIT 1;
```

Attempt to delete ItemID 1 - This should return an error!

```
DELETE FROM Item WHERE ItemID = 1;
```

Attempt to delete ItemID 1 - By cleaning it up in every potential child record first - This should work.

```
DELETE FROM Reserves Where ItemID = 1;
DELETE FROM Loans Where ItemID = 1;
DELETE FROM Creates Where ItemID = 1;
DELETE FROM Book Where ItemID = 1;
DELETE FROM Magazine Where ItemID = 1;
DELETE FROM DigitalMedia Where ItemID = 1;
DELETE FROM Item Where ItemID = 1;
```



```
58  21:59:07  DELETE FROM Item WHERE ItemID = 1              Error Code: 1451. Cannot delete or update a parent row: a foreign key constrai...  0.016 sec
59  21:59:43  DELETE FROM Reserves Where ItemID = 1          0 row(s) affected                                                                   0.000 sec
60  21:59:43  DELETE FROM Loans Where ItemID = 1             1 row(s) affected                                                                   0.000 sec
61  21:59:43  DELETE FROM Creates Where ItemID = 1           1 row(s) affected                                                                   0.016 sec
62  21:59:43  DELETE FROM Book Where ItemID = 1              0 row(s) affected                                                                   0.000 sec
63  21:59:43  DELETE FROM Magazine Where ItemID = 1          1 row(s) affected                                                                   0.000 sec
64  21:59:43  DELETE FROM DigitalMedia Where ItemID = 1      0 row(s) affected                                                                   0.000 sec
65  21:59:43  DELETE FROM Item Where ItemID = 1              1 row(s) affected                                                                   0.000 sec
```

Attempt to insert an Item with a null value in attribute Type - This should return an error!

```
INSERT INTO Item (Title, AvailabilityStatus, PublicationDate, AuthorID, Type, Genre)
Values ( "New Item", 1, '2024-12-03', 1, NULL, "Fantasy");
```

Attempt to insert an Item with a null value in attribute Author - This should work as intended

```
INSERT INTO Item (Title, AvailabilityStatus, PublicationDate, AuthorID, Type, Genre)
Values ("New Item", 1, '2024-12-03', NULL, 'Physical', "Fantasy");
```

Attempt to insert a key and then a duplicate key; the duplicate key should cause an error.

```
INSERT INTO Item (ItemID, Title, AvailabilityStatus, PublicationDate, AuthorID, Type, Genre)
Values (1001, "New Item", 1, '2024-12-03', NULL, 'Physical', "Fantasy"), (1001, "New Item", 1,
'2024-12-03', NULL, 'Physical', "Fantasy");
```