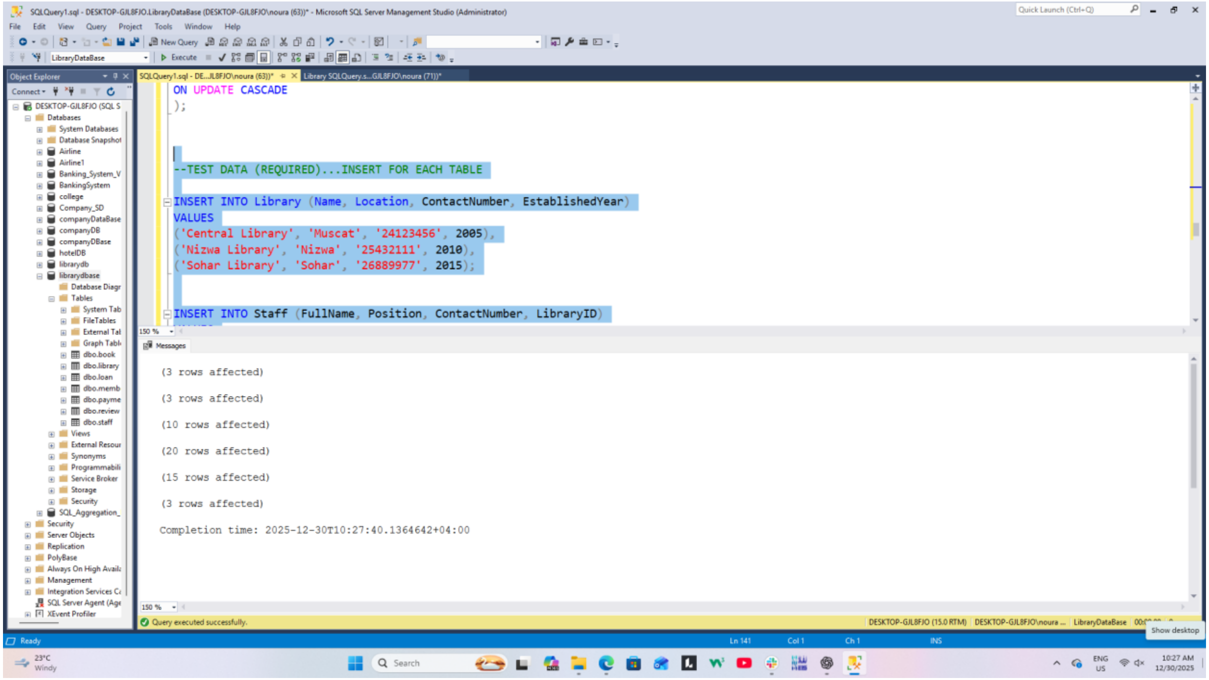
Database Project Part 2

Advanced Queries, Views, and Stored Procedures

1. SQL Script File

Test data inserts demonstrating that your procedures work correctly



```
ON UPDATE CASCADE
);


--TEST DATA (REQUIRED)...INSERT FOR EACH TABLE

INSERT INTO Library (Name, Location, ContactNumber, EstablishedYear)
VALUES
('Central Library', 'Muscat', '24123456', 2005),
('Nizwa Library', 'Nizwa', '25432111', 2010),
('Sohar Library', 'Sohar', '26889977', 2015);


INSERT INTO Staff (FullName, Position, ContactNumber, LibraryID)
```

```
(3 rows affected)

(3 rows affected)

(10 rows affected)

(20 rows affected)

(15 rows affected)

(3 rows affected)

Completion time: 2025-12-30T10:27:40.1364642+04:00
```

Query executed successfully.

2. Documentation Document

Including:

• Brief explanation of your approach for complex queries (queries 5, 6, 7)

Q5: Book Popularity Report

Identify books that are frequently borrowed (at least 3 times) and evaluate reader feedback.

How I did it:

- Joined **Book**, **Loan**, and **Review** tables.
- Used COUNT(LoanID) to calculate how many times each book was loaned.
- Used AVG(Rating) to compute the average review rating per book.

- Applied HAVING COUNT(LoanID) >= 3 to filter popular books only.
- Used LEFT JOIN for reviews to ensure books without reviews are still included.

This approach efficiently combines transactional data (loans) with feedback data (reviews) to provide a meaningful popularity metric without excluding incomplete data.

Q6: Member Reading History
Show a complete borrowing history for each member, including reviews.

How I did it:

- Joined **Member → Loan → Book** to retrieve borrowing records.
- Used a LEFT JOIN with **Review** to include review information only when it exists.
- Displayed both returned and currently borrowed books.
- Sorted by member name and loan date for readability.

Using LEFT JOIN ensures no borrowing record is lost, even if the member did not leave a review, giving a full historical view.

Q7: Revenue Analysis by Genre
Analyze fine revenue collected per book genre.

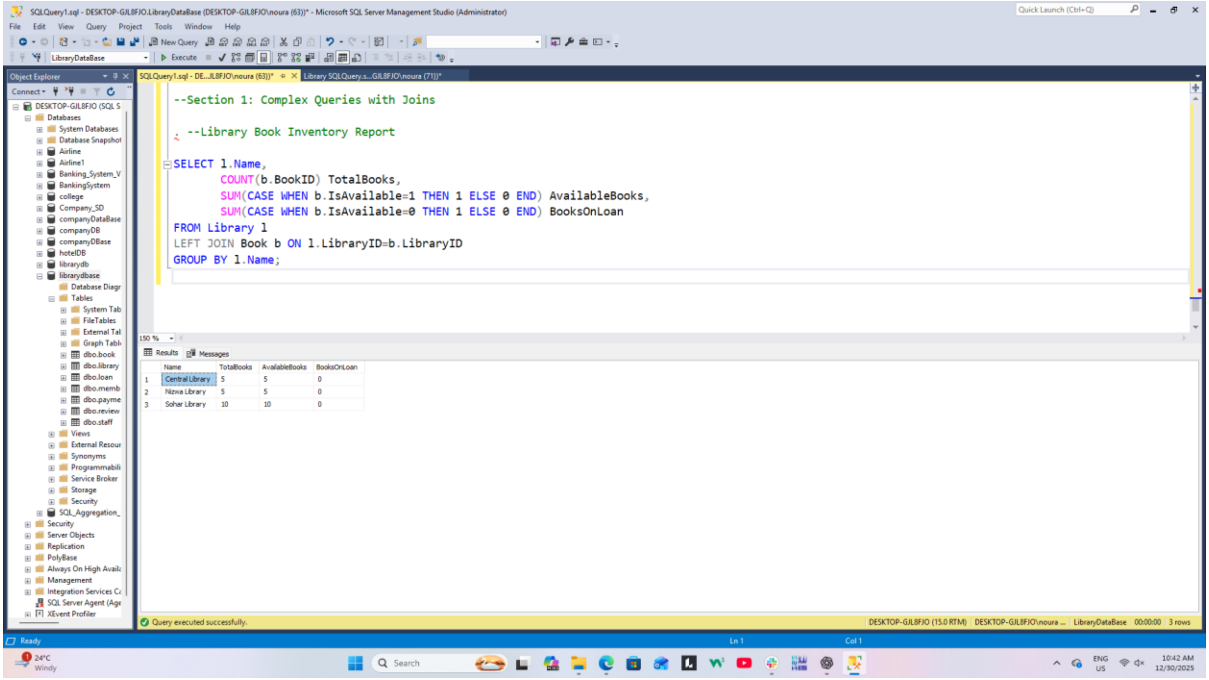
How i did it:

- Joined **Book**, **Loan**, and **Payment** tables.
- Grouped data by book genre.
- Used aggregate functions:
  - COUNT(DISTINCT LoanID) for total loans
  - SUM(Amount) for total revenue
  - AVG(Amount) for average fine per loan

This approach links financial data to book categories, allowing management to identify which genres generate higher fines and borrowing activity.

- **Screenshots of query results for at least 5 different queries**

Section 1: Complex Queries with Joins



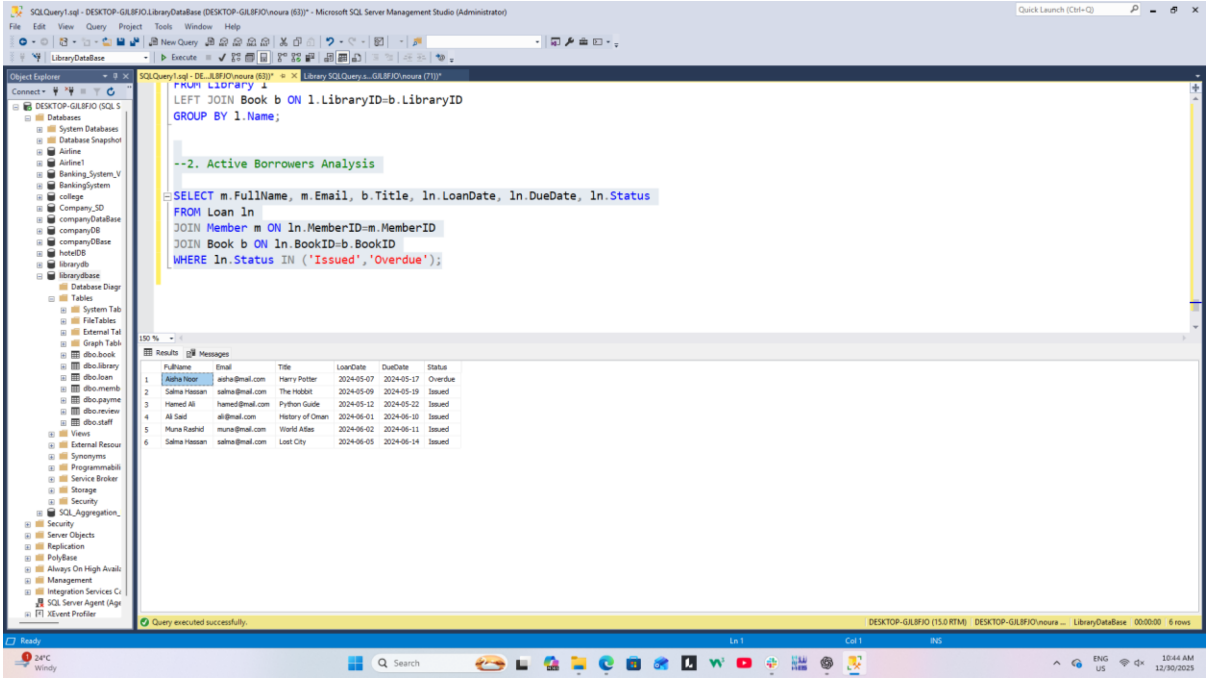
```
--Section 1: Complex Queries with Joins

 --Library Book Inventory Report

SELECT l.Name,
       COUNT(b.BookID) TotalBooks,
       SUM(CASE WHEN b.IsAvailable=1 THEN 1 ELSE 0 END) AvailableBooks,
       SUM(CASE WHEN b.IsAvailable=0 THEN 1 ELSE 0 END) BooksOnLoan
FROM Library l
LEFT JOIN Book b ON l.LibraryID=b.LibraryID
GROUP BY l.Name;
```

	Name	TotalBooks	AvailableBooks	BooksOnLoan
1	Central Library	5	5	0
2	Nizwa Library	5	5	0
3	Sohar Library	10	10	0

Query executed successfully. DESKTOP-GJL8FJO (15.0 RTM) DESKTOP-GJL8FJO\noura ... LibraryDataBase 00:00:00 3 rows



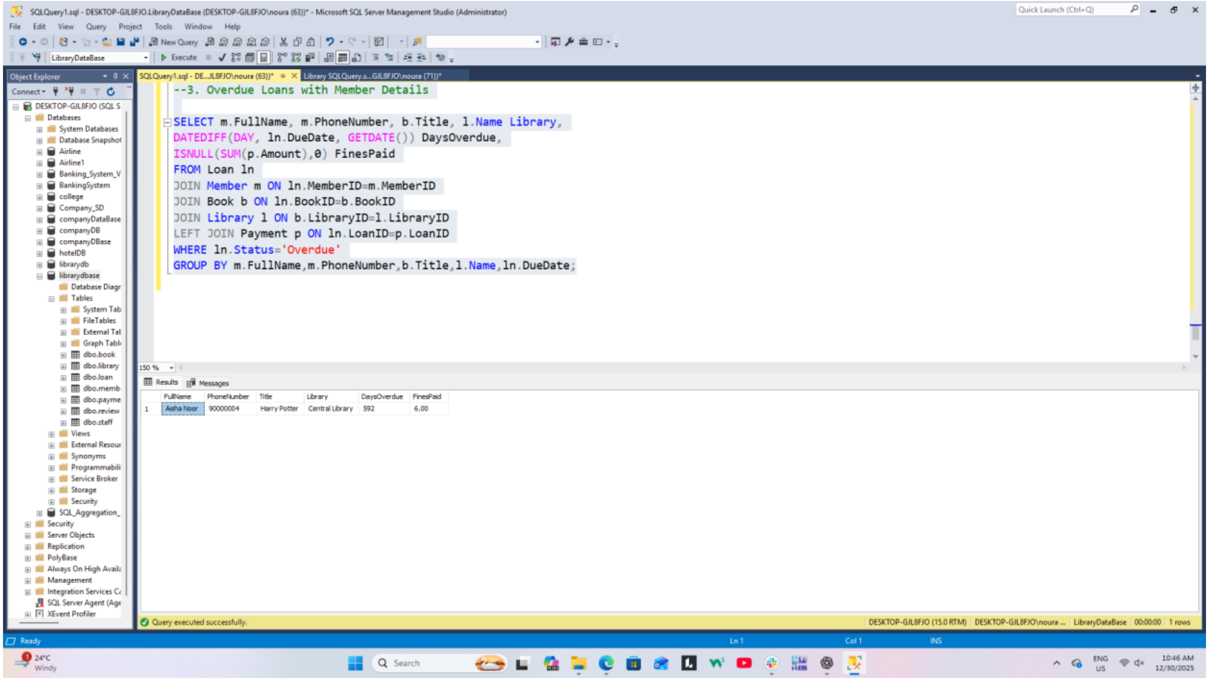
```
FROM Library l
LEFT JOIN Book b ON l.LibraryID=b.LibraryID
GROUP BY l.Name;


--2. Active Borrowers Analysis

SELECT m.FullName, m.Email, b.Title, ln.LoanDate, ln.DueDate, ln.Status
FROM Loan ln
JOIN Member m ON ln.MemberID=m.MemberID
JOIN Book b ON ln.BookID=b.BookID
WHERE ln.Status IN ('Issued','Overdue');
```

	FullName	Email	Title	LoanDate	DueDate	Status
1	Aisha Noor	aisha@mail.com	Harry Potter	2024-05-07	2024-05-17	Overdue
2	Salma Hassan	salma@mail.com	The Hobbit	2024-05-09	2024-05-19	Issued
3	Hamed Ali	hamed@mail.com	Python Guide	2024-05-12	2024-05-22	Issued
4	Ali Said	ali@mail.com	History of Oman	2024-06-01	2024-06-10	Issued
5	Muna Rashid	muna@mail.com	World Atlas	2024-06-02	2024-06-11	Issued
6	Salma Hassan	salma@mail.com	Lost City	2024-06-05	2024-06-14	Issued

Query executed successfully. DESKTOP-GJL8FJO (15.0 RTM) DESKTOP-GJL8FJO\noura ... LibraryDataBase 00:00:00 6 rows


SQLQuery1.sql - DESKTOP-GJL8FJO.LibraryDataBase (DESKTOP-GJL8FJO\noura (63))* - Microsoft SQL Server Management Studio (Administrator)
File Edit View Query Project Tools Window Help
New Query
Execute
LibraryDataBase
Object Explorer

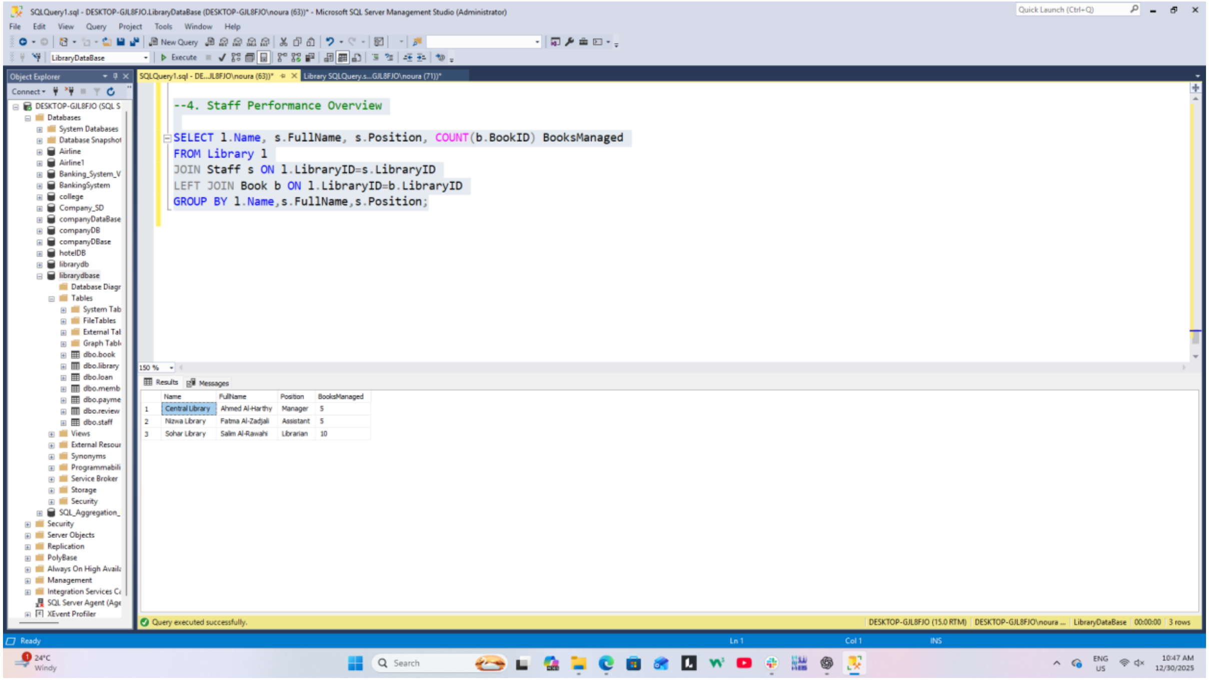
```sql
--3. Overdue Loans with Member Details

SELECT m.FullName, m.PhoneNumber, b.Title, l.Name Library,
DATEDIFF(DAY, ln.DueDate, GETDATE()) DaysOverdue,
ISNULL(SUM(p.Amount),0) FinesPaid
FROM Loan ln
JOIN Member m ON ln.MemberID=m.MemberID
JOIN Book b ON ln.BookID=b.BookID
JOIN Library l ON b.LibraryID=l.LibraryID
LEFT JOIN Payment p ON ln.LoanID=p.LoanID
WHERE ln.Status='Overdue'
GROUP BY m.FullName,m.PhoneNumber,b.Title,l.Name,ln.DueDate;
```

	FullName	PhoneNumber	Title	Library	DaysOverdue	FinesPaid
1	Aisha Noor	90000004	Harry Potter	Central Library	592	6.00

Query executed successfully.

DESKTOP-GJL8FJO (15.0 RTM) DESKTOP-GJL8FJO\noura ... LibraryDataBase 00:00:00 1 rows
10:46 AM
12/30/2025


SQLQuery1.sql - DESKTOP-GJL8FJO.LibraryDataBase (DESKTOP-GJL8FJO\noura (63))* - Microsoft SQL Server Management Studio (Administrator)
File Edit View Query Project Tools Window Help
New Query
Execute
LibraryDataBase
Object Explorer

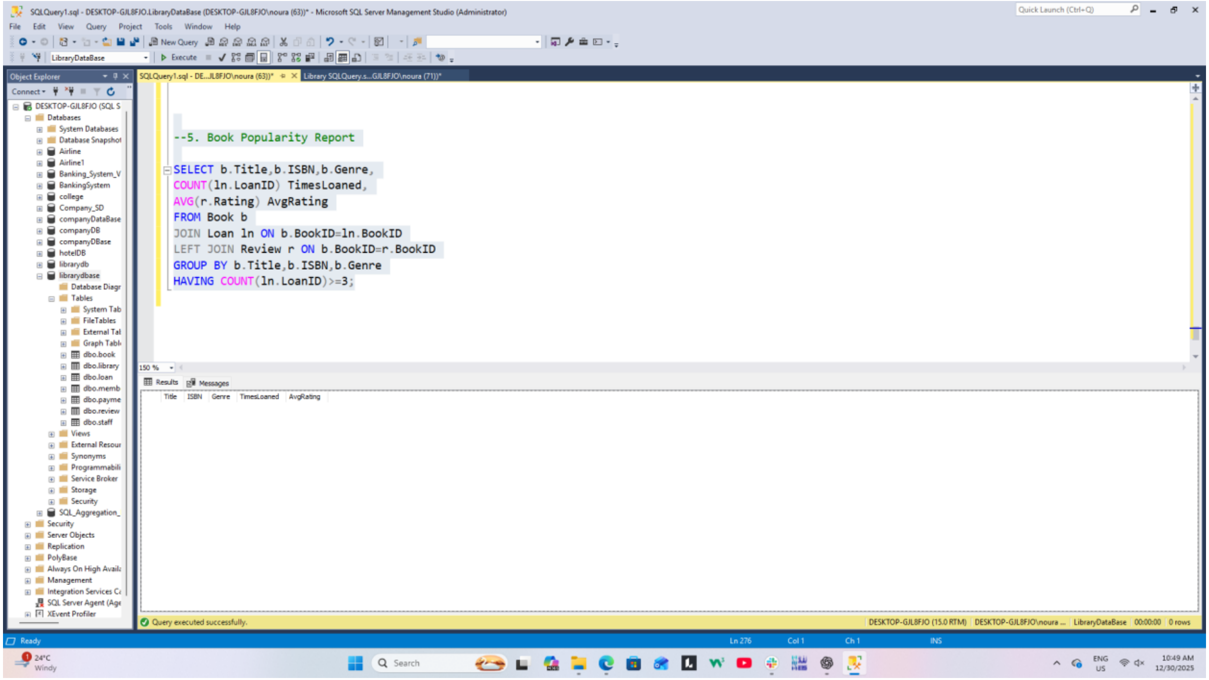
```sql
--4. Staff Performance Overview

SELECT l.Name, s.FullName, s.Position, COUNT(b.BookID) BooksManaged
FROM Library l
JOIN Staff s ON l.LibraryID=s.LibraryID
LEFT JOIN Book b ON l.LibraryID=b.LibraryID
GROUP BY l.Name,s.FullName,s.Position;
```

	Name	FullName	Position	BooksManaged
1	Central Library	Ahmed Al-Harthy	Manager	5
2	Nizwa Library	Fatma Al-Zadjali	Assistant	5
3	Sohar Library	Salim Al-Rawahi	Librarian	10

Query executed successfully.

DESKTOP-GJL8FJO (15.0 RTM) DESKTOP-GJL8FJO\noura ... LibraryDataBase 00:00:00 3 rows
10:47 AM
12/30/2025



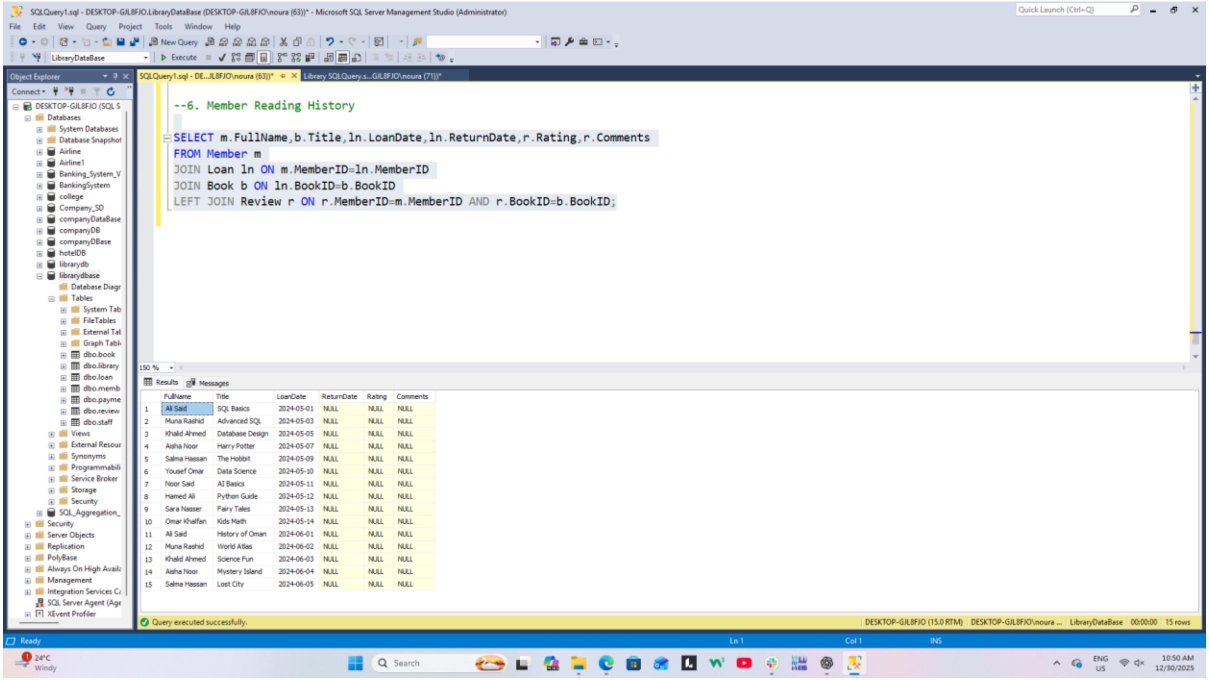
```
--5. Book Popularity Report

SELECT b.Title,b.ISBN,b.Genre,
COUNT(ln.LoanID) TimesLoaned,
AVG(r.Rating) AvgRating
FROM Book b
JOIN Loan ln ON b.BookID=ln.BookID
LEFT JOIN Review r ON b.BookID=r.BookID
GROUP BY b.Title,b.ISBN,b.Genre
HAVING COUNT(ln.LoanID)>=3;
```

Title	ISBN	Genre	TimesLoaned	AvgRating

Query executed successfully.

DESKTOP-GJL8FJO (15.0 RTM) | DESKTOP-GJL8FJO\noura ... | LibraryDataBase | 00:00:00 | 0 rows



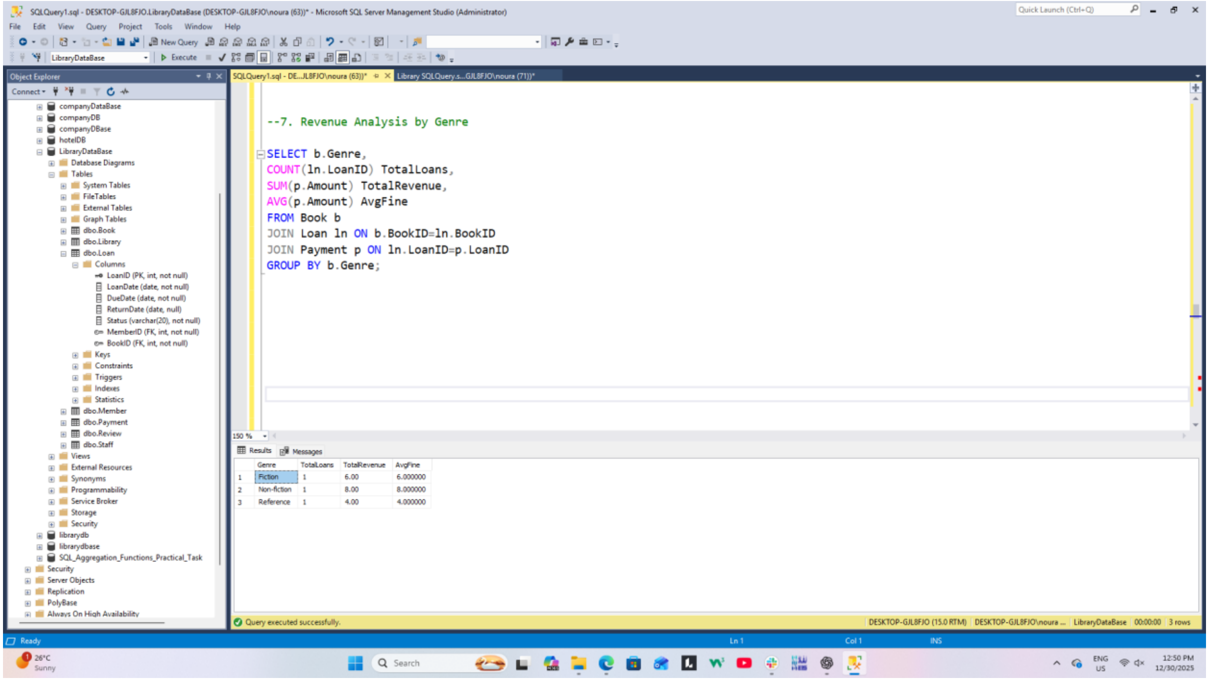
```
--6. Member Reading History

SELECT m.FullName,b.Title,ln.LoanDate,ln.ReturnDate,r.Rating,r.Comments
FROM Member m
JOIN Loan ln ON m.MemberID=ln.MemberID
JOIN Book b ON ln.BookID=b.BookID
LEFT JOIN Review r ON r.MemberID=m.MemberID AND r.BookID=b.BookID;
```

	FullName	Title	LoanDate	ReturnDate	Rating	Comments
1	Ali Said	SQL Basics	2024-05-01	NULL	NULL	NULL
2	Muna Rashid	Advanced SQL	2024-05-03	NULL	NULL	NULL
3	Khalid Ahmed	Database Design	2024-05-05	NULL	NULL	NULL
4	Aisha Noor	Harry Potter	2024-05-07	NULL	NULL	NULL
5	Salma Hassan	The Hobbit	2024-05-09	NULL	NULL	NULL
6	Yousef Omar	Data Science	2024-05-10	NULL	NULL	NULL
7	Noor Said	AI Basics	2024-05-11	NULL	NULL	NULL
8	Hamed Ali	Python Guide	2024-05-12	NULL	NULL	NULL
9	Sara Nasser	Fairy Tales	2024-05-13	NULL	NULL	NULL
10	Omar Khalfan	Kids Math	2024-05-14	NULL	NULL	NULL
11	Ali Said	History of Oman	2024-06-01	NULL	NULL	NULL
12	Muna Rashid	World Atlas	2024-06-02	NULL	NULL	NULL
13	Khalid Ahmed	Science Fun	2024-06-03	NULL	NULL	NULL
14	Aisha Noor	Mystery Island	2024-06-04	NULL	NULL	NULL
15	Salma Hassan	Lost City	2024-06-05	NULL	NULL	NULL

Query executed successfully.

DESKTOP-GJL8FJO (15.0 RTM) | DESKTOP-GJL8FJO\noura ... | LibraryDataBase | 00:00:00 | 15 rows



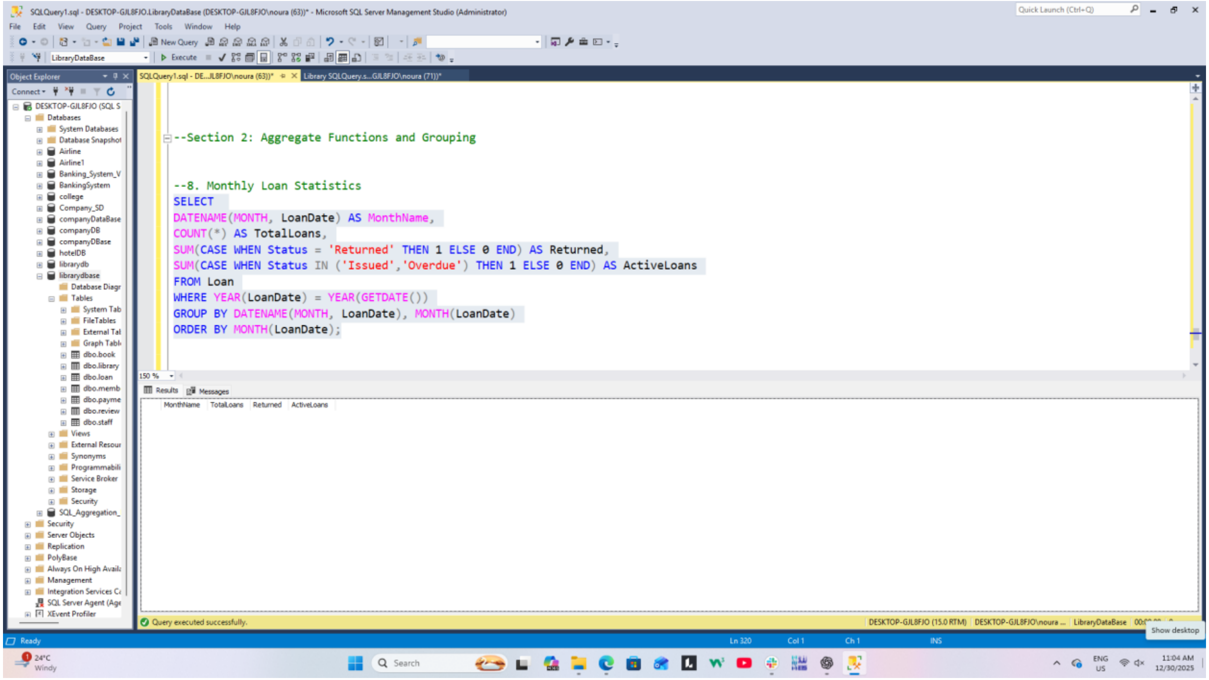
```sql
--7. Revenue Analysis by Genre

SELECT b.Genre,
COUNT(ln.LoanID) TotalLoans,
SUM(p.Amount) TotalRevenue,
AVG(p.Amount) AvgFine
FROM Book b
JOIN Loan ln ON b.BookID=ln.BookID
JOIN Payment p ON ln.LoanID=p.LoanID
GROUP BY b.Genre;
```

	Genre	TotalLoans	TotalRevenue	AvgFine
1	Fiction	1	6.00	6.000000
2	Non-fiction	1	8.00	8.000000
3	Reference	1	4.00	4.000000

Query executed successfully.

Section 2: Aggregate Functions and Grouping



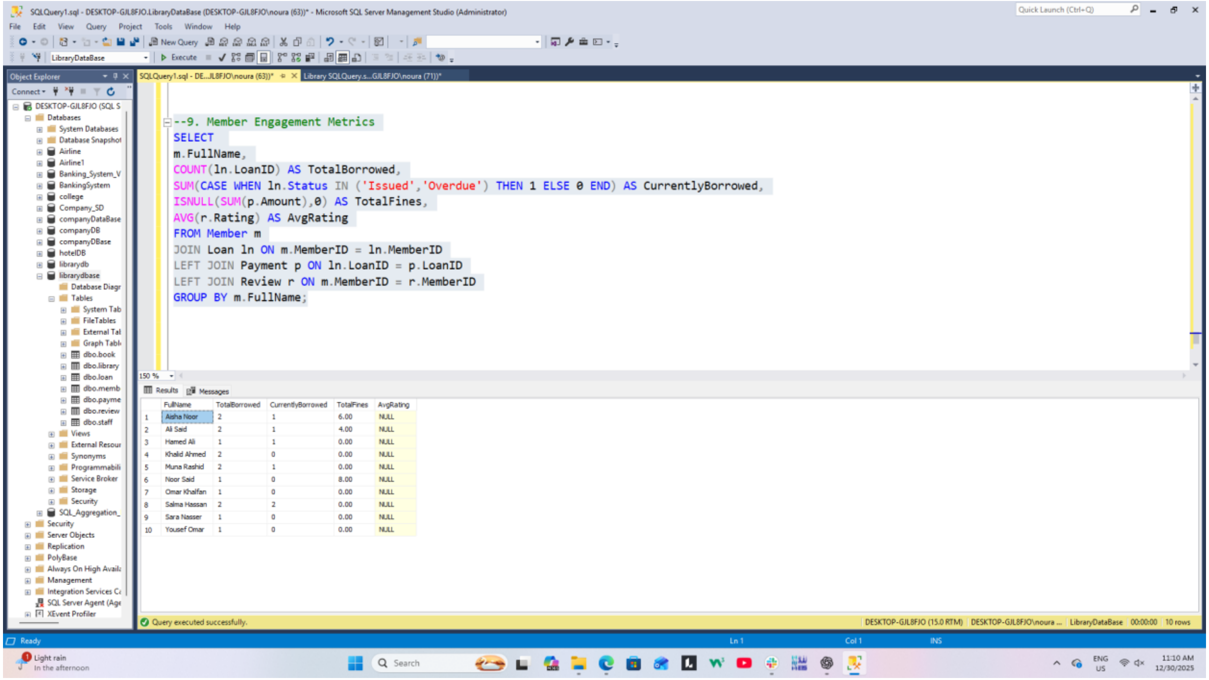
SQLQuery1.sql - DESKTOP-GIL8FJO.LibraryDataBase (DESKTOP-GIL8FJO\noura (63))* - Microsoft SQL Server Management Studio (Administrator)
Object Explorer
Results
Messages

```sql
--Section 2: Aggregate Functions and Grouping


--8. Monthly Loan Statistics
SELECT
DATENAME(MONTH, LoanDate) AS MonthName,
COUNT(*) AS TotalLoans,
SUM(CASE WHEN Status = 'Returned' THEN 1 ELSE 0 END) AS Returned,
SUM(CASE WHEN Status IN ('Issued','Overdue') THEN 1 ELSE 0 END) AS ActiveLoans
FROM Loan
WHERE YEAR(LoanDate) = YEAR(GETDATE())
GROUP BY DATENAME(MONTH, LoanDate), MONTH(LoanDate)
ORDER BY MONTH(LoanDate);
```

MonthName	TotalLoans	Returned	ActiveLoans

Query executed successfully.

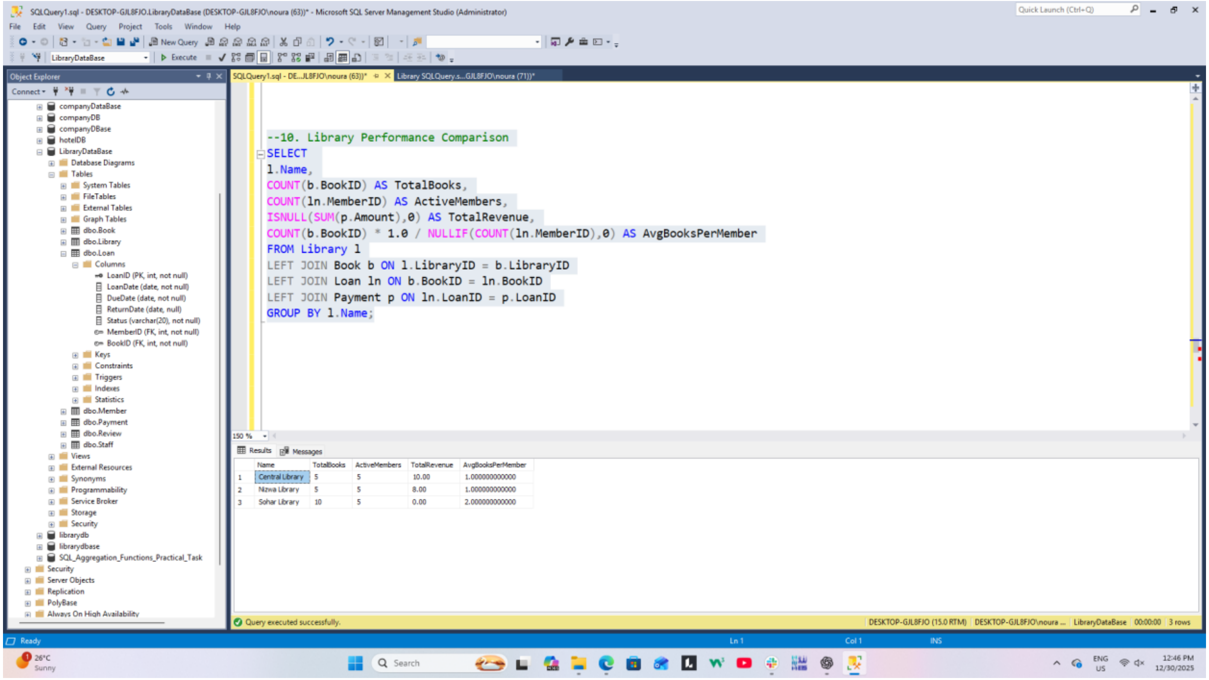


SQLQuery1.sql - DESKTOP-GIL8FJO.LibraryDataBase (DESKTOP-GIL8FJO\noura (63))* - Microsoft SQL Server Management Studio (Administrator)
Object Explorer
Results
Messages

```sql
--9. Member Engagement Metrics
SELECT
m.FullName,
COUNT(ln.LoanID) AS TotalBorrowed,
SUM(CASE WHEN ln.Status IN ('Issued','Overdue') THEN 1 ELSE 0 END) AS CurrentlyBorrowed,
ISNULL(SUM(p.Amount),0) AS TotalFines,
AVG(r.Rating) AS AvgRating
FROM Member m
JOIN Loan ln ON m.MemberID = ln.MemberID
LEFT JOIN Payment p ON ln.LoanID = p.LoanID
LEFT JOIN Review r ON m.MemberID = r.MemberID
GROUP BY m.FullName;
```

	FullName	TotalBorrowed	CurrentlyBorrowed	TotalFines	AvgRating
1	Aisha Noor	2	1	6.00	NULL
2	Ali Said	2	1	4.00	NULL
3	Hamed Ali	1	1	0.00	NULL
4	Khalid Ahmed	2	0	0.00	NULL
5	Muna Rashid	2	1	0.00	NULL
6	Noor Said	1	0	8.00	NULL
7	Omar Khalfan	1	0	0.00	NULL
8	Salma Hassan	2	2	0.00	NULL
9	Sara Nasser	1	0	0.00	NULL
10	Yousef Omar	1	0	0.00	NULL

Query executed successfully.



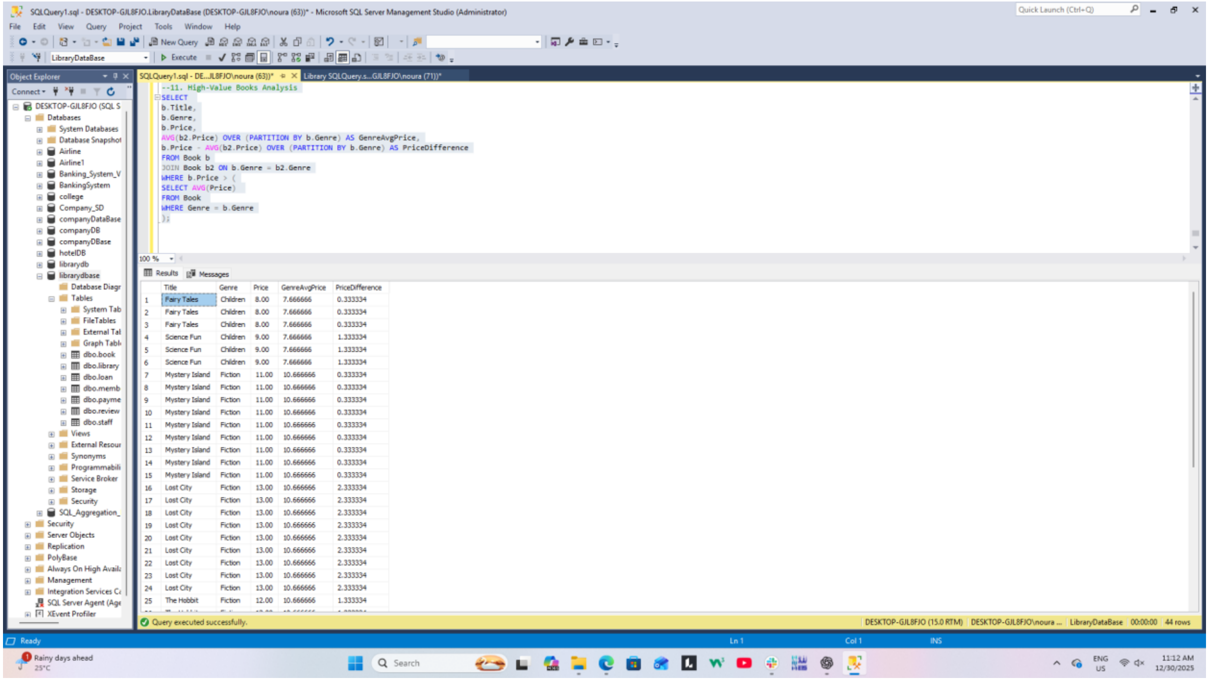
SQLQuery1.sql - DESKTOP-GJL8FJO.LibraryDataBase (DESKTOP-GJL8FJO\noura (63))* - Microsoft SQL Server Management Studio (Administrator)

```sql
--10. Library Performance Comparison
SELECT
l.Name,
COUNT(b.BookID) AS TotalBooks,
COUNT(ln.MemberID) AS ActiveMembers,
ISNULL(SUM(p.Amount),0) AS TotalRevenue,
COUNT(b.BookID) * 1.0 / NULLIF(COUNT(ln.MemberID),0) AS AvgBooksPerMember
FROM Library l
LEFT JOIN Book b ON l.LibraryID = b.LibraryID
LEFT JOIN Loan ln ON b.BookID = ln.BookID
LEFT JOIN Payment p ON ln.LoanID = p.LoanID
GROUP BY l.Name;
```

	Name	TotalBooks	ActiveMembers	TotalRevenue	AvgBooksPerMember
1	Central Library	5	5	10.00	1.000000000000
2	Nizwa Library	5	5	8.00	1.000000000000
3	Sohar Library	10	5	0.00	2.000000000000

Query executed successfully.

DESKTOP-GJL8FJO (15.0 RTM) | DESKTOP-GJL8FJO\noura ... | LibraryDataBase | 00:00:00 | 3 rows



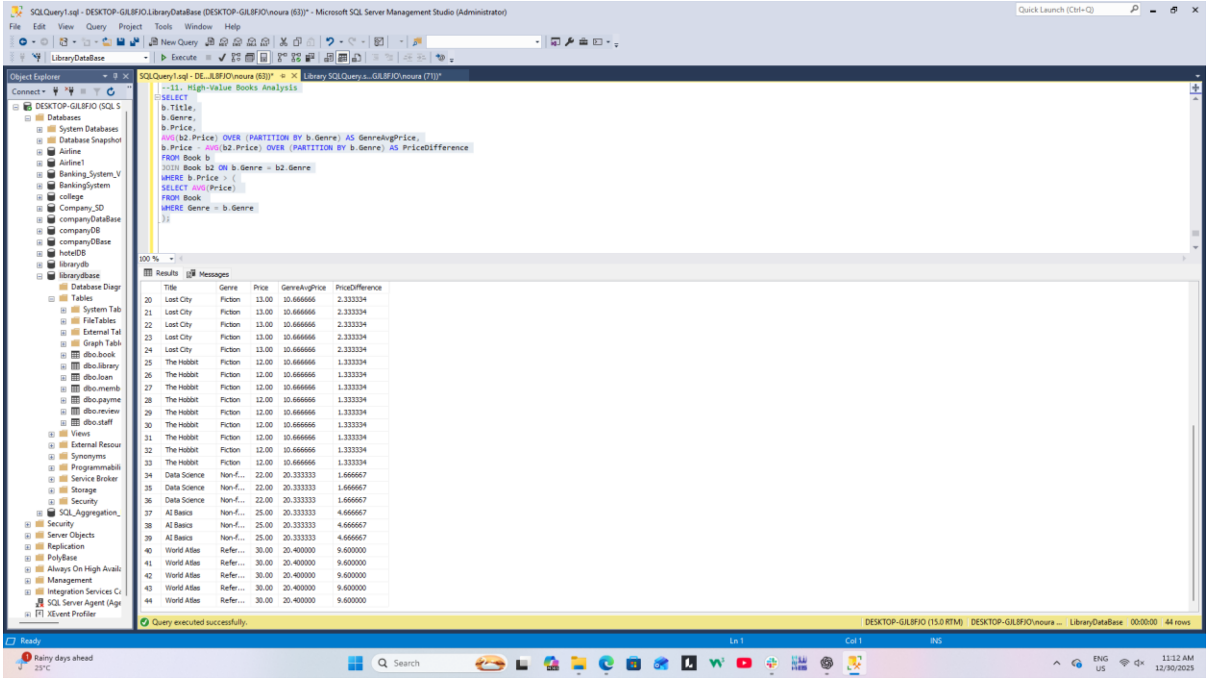
SQLQuery1.sql - DESKTOP-GJL8FJO.LibraryDataBase (DESKTOP-GJL8FJO\noura (63))* - Microsoft SQL Server Management Studio (Administrator)

```sql
--11. High-Value Books Analysis
SELECT
b.Title,
b.Genre,
b.Price,
AVG(b2.Price) OVER (PARTITION BY b.Genre) AS GenreAvgPrice,
b.Price - AVG(b2.Price) OVER (PARTITION BY b.Genre) AS PriceDifference
FROM Book b
JOIN Book b2 ON b.Genre = b2.Genre
WHERE b.Price > (
SELECT AVG(Price)
FROM Book
WHERE Genre = b.Genre
);
```

	Title	Genre	Price	GenreAvgPrice	PriceDifference
1	Fairy Tales	Children	8.00	7.666666	0.333334
2	Fairy Tales	Children	8.00	7.666666	0.333334
3	Fairy Tales	Children	8.00	7.666666	0.333334
4	Science Fun	Children	9.00	7.666666	1.333334
5	Science Fun	Children	9.00	7.666666	1.333334
6	Science Fun	Children	9.00	7.666666	1.333334
7	Mystery Island	Fiction	11.00	10.666666	0.333334
8	Mystery Island	Fiction	11.00	10.666666	0.333334
9	Mystery Island	Fiction	11.00	10.666666	0.333334
10	Mystery Island	Fiction	11.00	10.666666	0.333334
11	Mystery Island	Fiction	11.00	10.666666	0.333334
12	Mystery Island	Fiction	11.00	10.666666	0.333334
13	Mystery Island	Fiction	11.00	10.666666	0.333334
14	Mystery Island	Fiction	11.00	10.666666	0.333334
15	Mystery Island	Fiction	11.00	10.666666	0.333334
16	Lost City	Fiction	13.00	10.666666	2.333334
17	Lost City	Fiction	13.00	10.666666	2.333334
18	Lost City	Fiction	13.00	10.666666	2.333334
19	Lost City	Fiction	13.00	10.666666	2.333334
20	Lost City	Fiction	13.00	10.666666	2.333334
21	Lost City	Fiction	13.00	10.666666	2.333334
22	Lost City	Fiction	13.00	10.666666	2.333334
23	Lost City	Fiction	13.00	10.666666	2.333334
24	Lost City	Fiction	13.00	10.666666	2.333334
25	The Hobbit	Fiction	12.00	10.666666	1.333334

Query executed successfully.

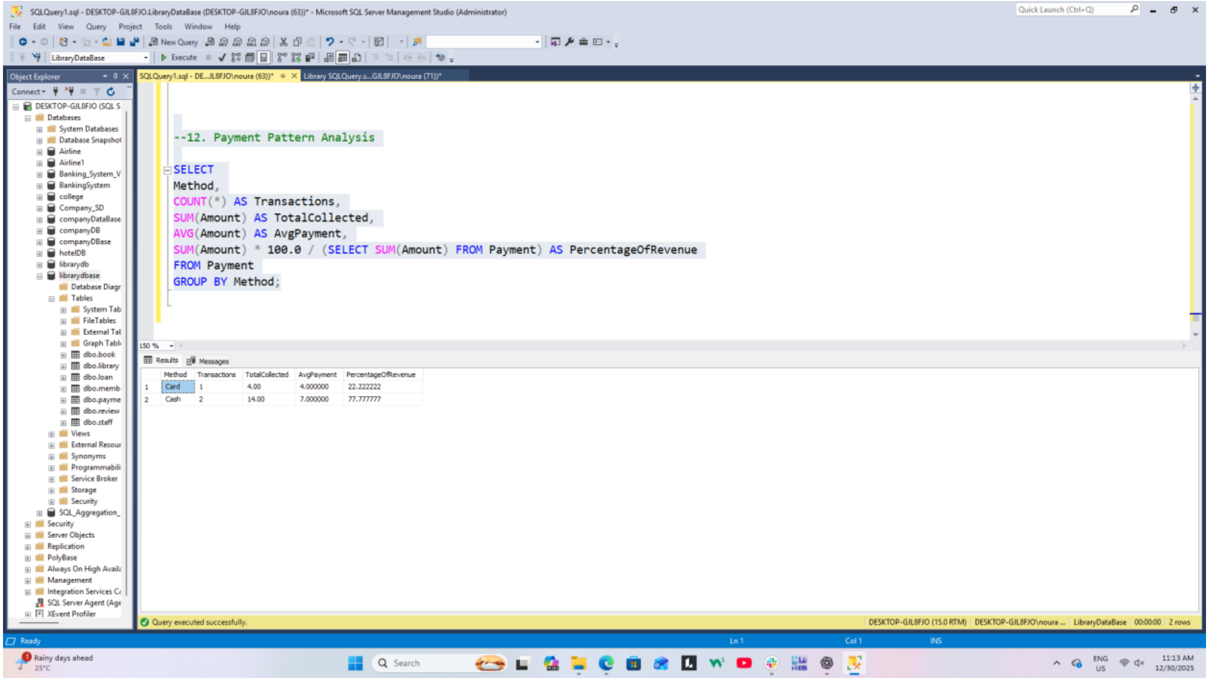
DESKTOP-GJL8FJO (15.0 RTM) | DESKTOP-GJL8FJO\noura ... | LibraryDataBase | 00:00:00 | 44 rows



```sql
--11. High-Value Books Analysis
SELECT
b.Title,
b.Genre,
b.Price,
AVG(b2.Price) OVER (PARTITION BY b.Genre) AS GenreAvgPrice,
b.Price - AVG(b2.Price) OVER (PARTITION BY b.Genre) AS PriceDifference
FROM Book b
JOIN Book b2 ON b.Genre = b2.Genre
WHERE b.Price > (
SELECT AVG(Price)
FROM Book
WHERE Genre = b.Genre
);
```

	Title	Genre	Price	GenreAvgPrice	PriceDifference
20	Lost City	Fiction	13.00	10.666666	2.333334
21	Lost City	Fiction	13.00	10.666666	2.333334
22	Lost City	Fiction	13.00	10.666666	2.333334
23	Lost City	Fiction	13.00	10.666666	2.333334
24	Lost City	Fiction	13.00	10.666666	2.333334
25	The Hobbit	Fiction	12.00	10.666666	1.333334
26	The Hobbit	Fiction	12.00	10.666666	1.333334
27	The Hobbit	Fiction	12.00	10.666666	1.333334
28	The Hobbit	Fiction	12.00	10.666666	1.333334
29	The Hobbit	Fiction	12.00	10.666666	1.333334
30	The Hobbit	Fiction	12.00	10.666666	1.333334
31	The Hobbit	Fiction	12.00	10.666666	1.333334
32	The Hobbit	Fiction	12.00	10.666666	1.333334
33	The Hobbit	Fiction	12.00	10.666666	1.333334
34	Data Science	Non-f...	22.00	20.333333	1.666667
35	Data Science	Non-f...	22.00	20.333333	1.666667
36	Data Science	Non-f...	22.00	20.333333	1.666667
37	AI Basics	Non-f...	25.00	20.333333	4.666667
38	AI Basics	Non-f...	25.00	20.333333	4.666667
39	AI Basics	Non-f...	25.00	20.333333	4.666667
40	World Atlas	Refer...	30.00	20.400000	9.600000
41	World Atlas	Refer...	30.00	20.400000	9.600000
42	World Atlas	Refer...	30.00	20.400000	9.600000
43	World Atlas	Refer...	30.00	20.400000	9.600000
44	World Atlas	Refer...	30.00	20.400000	9.600000

Query executed successfully.



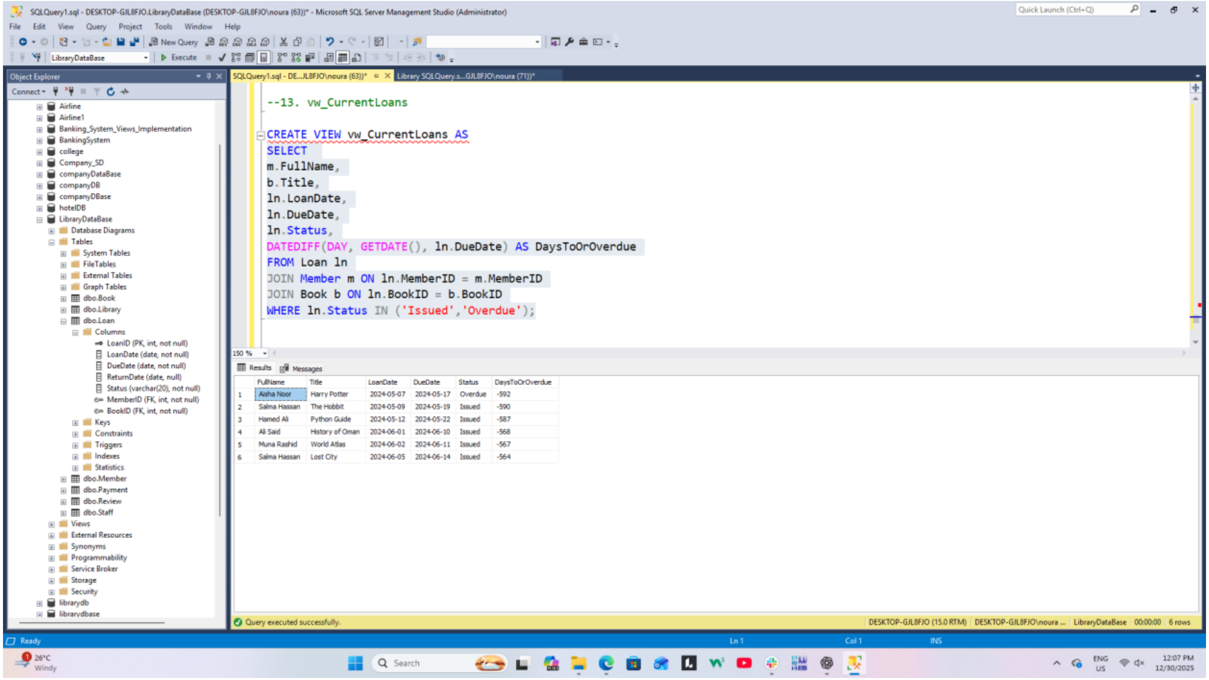
```sql
--12. Payment Pattern Analysis

SELECT
Method,
COUNT(*) AS Transactions,
SUM(Amount) AS TotalCollected,
AVG(Amount) AS AvgPayment,
SUM(Amount) * 100.0 / (SELECT SUM(Amount) FROM Payment) AS PercentageOfRevenue
FROM Payment
GROUP BY Method;
```

	Method	Transactions	TotalCollected	AvgPayment	PercentageOfRevenue
1	Card	1	4.00	4.000000	22.222222
2	Cash	2	14.00	7.000000	77.777777

Query executed successfully.

section 3: Views Creation



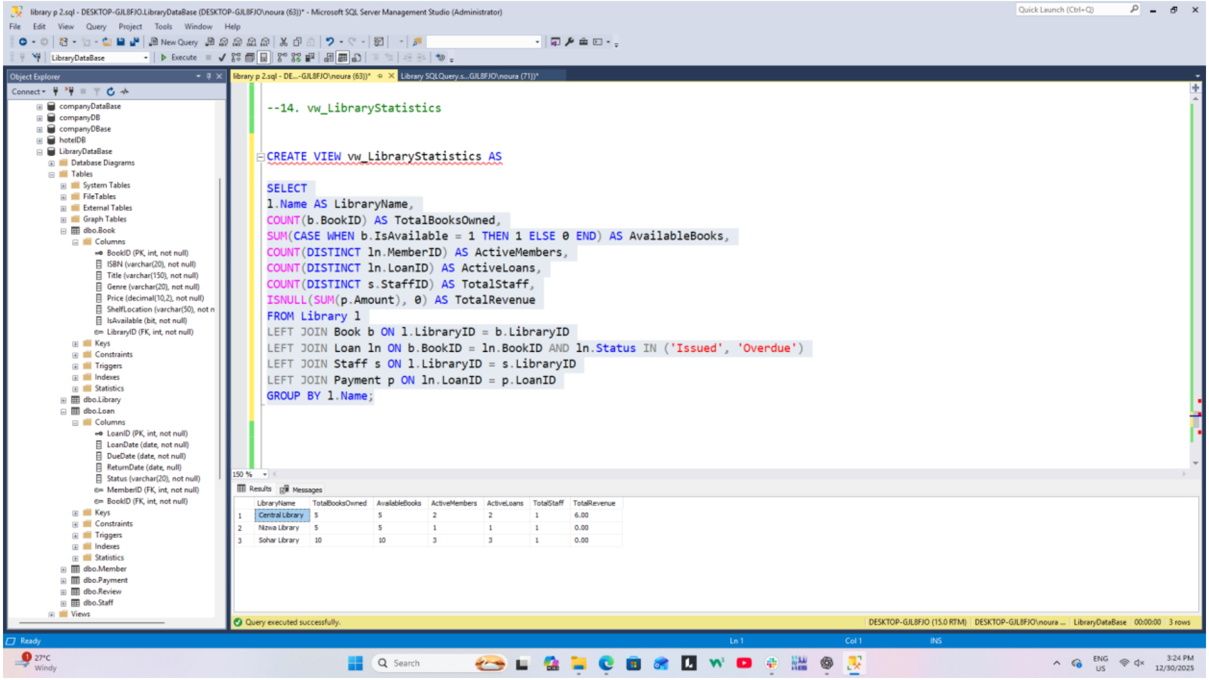
```sql
--13. vw_CurrentLoans

CREATE VIEW vw_CurrentLoans AS
SELECT
m.FullName,
b.Title,
ln.LoanDate,
ln.DueDate,
ln.Status,
DATEDIFF(DAY, GETDATE(), ln.DueDate) AS DaysToOrOverdue
FROM Loan ln
JOIN Member m ON ln.MemberID = m.MemberID
JOIN Book b ON ln.BookID = b.BookID
WHERE ln.Status IN ('Issued','Overdue');
```

	FullName	Title	LoanDate	DueDate	Status	DaysToOrOverdue
1	Aisha Noor	Harry Potter	2024-05-07	2024-05-17	Overdue	-592
2	Salma Hassan	The Hobbit	2024-05-09	2024-05-19	Issued	-590
3	Hamed Ali	Python Guide	2024-05-12	2024-05-22	Issued	-587
4	Ali Said	History of Oman	2024-06-01	2024-06-10	Issued	-568
5	Muna Rashid	World Atlas	2024-06-02	2024-06-11	Issued	-567
6	Salma Hassan	Lost City	2024-06-05	2024-06-14	Issued	-564

Query executed successfully.

Q14- vw_LibraryStatistics should show library name, total books owned by the library, number of available books, total active members(members who have at least one loan from this library's books), active loans (loans of books belonging to this library), total staff working at the library, total revenue from fines (from loans of this library's books).



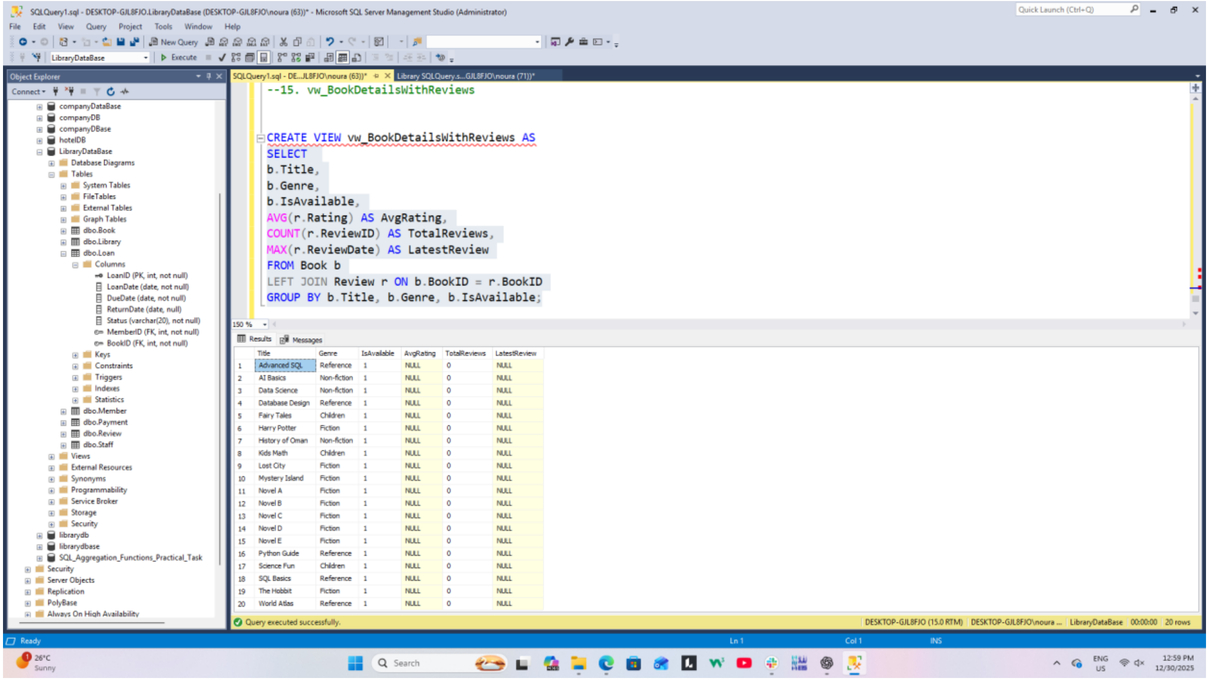
```sql
--14. vw_LibraryStatistics


CREATE VIEW vw_LibraryStatistics AS

SELECT
l.Name AS LibraryName,
COUNT(b.BookID) AS TotalBooksOwned,
SUM(CASE WHEN b.IsAvailable = 1 THEN 1 ELSE 0 END) AS AvailableBooks,
COUNT(DISTINCT ln.MemberID) AS ActiveMembers,
COUNT(DISTINCT ln.LoanID) AS ActiveLoans,
COUNT(DISTINCT s.StaffID) AS TotalStaff,
ISNULL(SUM(p.Amount), 0) AS TotalRevenue
FROM Library l
LEFT JOIN Book b ON l.LibraryID = b.LibraryID
LEFT JOIN Loan ln ON b.BookID = ln.BookID AND ln.Status IN ('Issued', 'Overdue')
LEFT JOIN Staff s ON l.LibraryID = s.LibraryID
LEFT JOIN Payment p ON ln.LoanID = p.LoanID
GROUP BY l.Name;
```

	LibraryName	TotalBooksOwned	AvailableBooks	ActiveMembers	ActiveLoans	TotalStaff	TotalRevenue
1	Central Library	5	5	2	2	1	6.00
2	Nizwa Library	5	5	1	1	1	0.00
3	Sohar Library	10	10	3	3	1	0.00

Query executed successfully.



```sql
--15. vw_BookDetailsWithReviews


CREATE VIEW vw_BookDetailsWithReviews AS
SELECT
b.Title,
b.Genre,
b.IsAvailable,
AVG(r.Rating) AS AvgRating,
COUNT(r.ReviewID) AS TotalReviews,
MAX(r.ReviewDate) AS LatestReview
FROM Book b
LEFT JOIN Review r ON b.BookID = r.BookID
GROUP BY b.Title, b.Genre, b.IsAvailable;
```

	Title	Genre	IsAvailable	AvgRating	TotalReviews	LatestReview
1	Advanced SQL	Reference	1	NULL	0	NULL
2	AI Basics	Non-fiction	1	NULL	0	NULL
3	Data Science	Non-fiction	1	NULL	0	NULL
4	Database Design	Reference	1	NULL	0	NULL
5	Fairy Tales	Children	1	NULL	0	NULL
6	Harry Potter	Fiction	1	NULL	0	NULL
7	History of Oman	Non-fiction	1	NULL	0	NULL
8	Kids Math	Children	1	NULL	0	NULL
9	Lost City	Fiction	1	NULL	0	NULL
10	Mystery Island	Fiction	1	NULL	0	NULL
11	Novel A	Fiction	1	NULL	0	NULL
12	Novel B	Fiction	1	NULL	0	NULL
13	Novel C	Fiction	1	NULL	0	NULL
14	Novel D	Fiction	1	NULL	0	NULL
15	Novel E	Fiction	1	NULL	0	NULL
16	Python Guide	Reference	1	NULL	0	NULL
17	Science Fun	Children	1	NULL	0	NULL
18	SQL Basics	Reference	1	NULL	0	NULL
19	The Hobbit	Fiction	1	NULL	0	NULL
20	World Atlas	Reference	1	NULL	0	NULL

Query executed successfully.

DESKTOP-GJL8FJO (15.0 RTM) | DESKTOP-GJL8FJO\noura ... | LibraryDataBase | 00:00:00 | 20 rows
Ready
Ln 1
Col 1
INS
26°C
Sunny
Search
ENG
US
12:59 PM
12/30/2025