Database Project | part_2

Section 1: Complex Queries with Joins

1. Library Book Inventory Report

Display library name, total number of books, number of available books, and number of books currently on loan for each library.

```
----------------------------------------------------------------------------------------++PART-2
-----SECTION 1:------

---1. Library Book Inventory Report

select l.Library_Name,count(b.Book_ID) AS TotalBooks,
SUM(CASE WHEN B.Book_IsAvailable = 1 THEN 1 ELSE 0 END) AS Available_Books,
SUM(CASE WHEN o.Loan_Status = 'Issued'  THEN 1 ELSE 0 END) AS Books_On_Loan
from  Books b
Join  Libraries l on b.Library_ID = l.Library_ID
JOIN  Loan o on b.Book_ID = o.Book_ID
Group by l.Library_Name

Select * from Books
Select * from Libraries
Select * from Loan
```

89 %

Results | Messages

	Library_Name	TotalBooks	Available_Books	Books_On_Loan
1	Central Library	10	10	5
2	City Library	5	5	2

2. Active Borrowers Analysis

List all members who currently have books on loan (status = 'Issued' or 'Overdue').
Show member name, email, book title, loan date, due date, and current status

```sql
---2. Active Borrowers Analysis

select m.Member_FullName,m.Member_Email,b.Book_Title,o.LoanDate,o.DueDate,o.Loan_Status
from Loan o
Join Members m on m.Member_ID = o.Member_ID
Join Books b on b.Book_ID = o.Book_ID
where o.Loan_Status in ('Issued','Overdue')
```

89 %

Results Messages

	Member_FullName	Member_Email	Book_Title	LoanDate	DueDate	Loan_Status
1	Ali Ahmed	ali@email.com	Introduction to Programming	2024-03-01	2024-04-15	Issued
2	Omar Khaled	omar@email.com	Harry Potter	2024-04-10	2024-05-20	Overdue
3	Anoud	Anoud@example.com	Networking	2024-05-01	2024-05-10	Issued
4	Sara Mohammed	saraa@email.com	Database Systems	2024-05-03	2024-05-13	Issued
5	Omar Khaled	omarr@email.com	Harry Potter	2024-05-04	2024-05-14	Overdue
6	Mona Said	mona2@email.com	Advanced SQL	2024-05-06	2024-05-16	Issued
7	Yousef Nasser	yousef3@email.com	Cyber Security 101	2024-05-07	2024-05-17	Overdue
8	Aisha Salem	aisha4@email.com	Networking Basics	2024-05-08	2024-05-18	Issued
9	Fatma Ahmed	fatma6@email.com	Fairy Tales	2024-05-10	2024-05-20	Issued
10	Nasser Hamed	nasser7@email.com	Short Stories	2024-05-11	2024-05-21	Overdue
11	Noor Rashid	noor8@email.com	Mystery Night	2024-05-12	2024-05-22	Issued

3. Overdue Loans with Member Details

Retrieve all overdue loans showing member name, phone number, book title, library name, days overdue (calculated as difference between current date and due date), and any fines paid for that loan

```sql
---3. Overdue Loans with Member Details
select m.Member_FullName,b.Book_Title,Lb.Library_Name,
Datediff(day,L.LoanDate,L.DueDate) AS Days_Overdue,
ISNULL(P.Amount,0) AS Fines_Paid

 FROM Loan L
JOIN Members M ON L.Member_ID = M.Member_ID
JOIN Books B ON L.Book_ID = B.Book_ID
JOIN Libraries Lb ON B.Library_ID = Lb.Library_ID
JOIN Payment P ON L.Loan_ID = P.Loan_ID
WHERE L.Loan_Status = 'Overdue'

Select * from Payment
```

89 %

Results Messages

	Member_FullName	Book_Title	Library_Name	Days_Overdue	Fines_Paid
1	Omar Khaled	Harry Potter	City Library	40	45.75
2	Yousef Nasser	Cyber Security 101	Central Library	10	9.00
3	Nasser Hamed	Short Stories	City Library	10	45.75

4. Staff Performance Overview

For each library, show the library name, staff member names, their positions, and count of books managed at that library

```
---4. Staff Performance Overview
select l.Library_Name,ss.staff_fullName,ss.Position,
count (b.Book_ID) AS Books_Managed
from staff_library ss
JOIN Libraries l ON ss.Library_ID = l.Library_ID
JOIN Books B ON l.Library_ID = b.Library_ID
GROUP BY l.Library_Name, ss.staff_fullName, ss.Position


select * from staff_library
select * from Books
select * from Libraries
```

9 %

Results Messages

	Library_Name	staff_fullName	Position	Books_Managed
1	Central Library	Anoud	Assistant Librarian	25
2	Central Library	Saleh	Librarian	25
3	City Library	Mohammed	Manager	16

5. Book Popularity Report

Display books that have been loaned at least 3 times. Include book title, ISBN, genre, total number of times loaned, and average review rating (if any reviews exist)

```
---5. Book Popularity Report -----comment(Because I do not have books which were loaned more than 1,
---   I changed the question from 3 to at least 1 to see the result)

select b.Book_Title,b.ISBN,b.Book_Genre,
count(l.Loan_ID) AS Total_Loaned,AVG(R.Rating) as Avg_Review_Rating
from Books b
Join Reviews R ON b.Book_ID = R.Book_ID
Join Loan l ON b.Book_ID = l.Book_ID
Group by b.Book_Title,b.ISBN,b.Book_Genre
Having count(l.Loan_ID) >= 3
```

89 %

Results Messages

	Book_Title	ISBN	Book_Genre	Total_Loaned	Avg_Review_Rating
1	Introduction to Programming	0131103	Reference	1	5
2	Harry Potter	043190	Children	1	3
3	Database Systems	321380	Non-fiction	1	4

Book Popularity Report: Joined Books, Loan, and Reviews to count loans and get average ratings per book, using HAVING to filter popular books.

6. Member Reading History

Create a query that shows each member's complete borrowing history including: member name, book titles borrowed (including currently borrowed and previously returned), loan dates, return dates, and any reviews they left for those books

```sql
--- 6. Member Reading History
select m.Member_FullName,b.Book_Title,l.LoanDate,l.ReturnDate,r.Rating,r.Comments
from Members m
JOIN Loan l ON m.Member_ID = l.Member_ID
JOIN Books b ON l.Book_ID = b.Book_ID
left JOIN Reviews r ON m.Member_ID = r.Member_ID
AND B.Book_ID = R.Book_ID
```

89 %

Results Messages

	Member_FullName	Book_Title	LoanDate	ReturnDate	Rating	Comments
1	Ali Ahmed	Introduction to Programming	2024-03-01	NULL	5	Excellent book, highly recommend!
2	Sara Mohammed	Database Systems	2024-01-01	2024-03-09	4	Good read, but a bit lengthy.
3	Omar Khaled	Harry Potter	2024-04-10	2024-04-22	3	Average content, could be better.
4	Anoud	Networking	2024-05-01	NULL	NULL	NULL
5	Ali Ahmed	Introduction to Programming	2024-05-02	2024-05-12	NULL	NULL
6	Sara Mohammed	Database Systems	2024-05-03	NULL	NULL	NULL
7	Omar Khaled	Harry Potter	2024-05-04	2024-05-15	NULL	NULL
8	Hassan Ali	SQL Basics	2024-05-05	2024-05-14	NULL	NULL
9	Mona Said	Advanced SQL	2024-05-06	NULL	NULL	NULL
10	Yousef Nasser	Cyber Security 101	2024-05-07	2024-05-18	NULL	NULL
11	Aisha Salem	Networking Basics	2024-05-08	NULL	NULL	NULL
12	Khalid Omar	Data Analysis	2024-05-09	2024-05-19	NULL	NULL
13	Fatma Ahmed	Fairy Tales	2024-05-10	NULL	NULL	NULL
14	Nasser Hamed	Short Stories	2024-05-11	2024-05-22	NULL	NULL
15	Noor Rashid	Mystery Night	2024-05-12	NULL	NULL	NULL

Member Reading History: Joined `Members`, `Loan`, and `Books`, left-joined `Reviews` to list each member's loans with any review details.

7. Revenue Analysis by Genre

Calculate total fine payments collected for each book genre. Show genre name, total number of loans for that genre, total fine amount collected, and average fine per loan.

```
----7. Revenue Analysis by Genre
select b.Book_Genre,count(l.Loan_ID) AS Total_Loans,SUM(isnull(P.Amount,0)) as Total_Amounts,
avg(isnull(P.Amount,0)) as Avg_Fine_Per_Loan
from Books b
Join Loan l on b.Book_ID = l.Book_ID
Join Payment p on l.Loan_ID = p.Loan_ID
GROUP BY B.Book_Genre
```

89 %

Results Messages

	Book_Genre	Total_Loans	Total_Amounts	Avg_Fine_Per_Loan
1	Children	3	104.00	34.666666
2	Fiction	2	91.50	45.750000
3	Non-fiction	4	36.00	9.000000
4	Reference	3	37.50	12.500000

Revenue Analysis by Genre: Joined Books, Loan, and Payment, using aggregates (COUNT, SUM, AVG) grouped by genre to summarize total loans and revenue

Section 2: Aggregate Functions and Grouping

8. Monthly Loan Statistics

Generate a report showing the number of loans issued per month for the current year. Include month name, total loans, total returned, and total still issued/overdue

```
--- 8. Monthly Loan Statistics

select
DATENAME(month,LoanDate) AS Month_Name,
count(*) AS Total_Loans,
sum(Case when Loan_Status ='Returned' THEN 1 else 0 END) AS Total_returned,

sum(case when Loan_Status IN ('Issued','Overdue') THEN 1 else 0 END) AS Total_Issued_OrOverdue

FROM Loan

WHERE YEAR(LoanDate) = YEAR(GETDATE())
GROUP BY DATENAME(month,LoanDate) , month(LoanDate)

select * from Loan
----- Because I didn't have any loan in 2025
```

89 %

Results | Messages

Month_Name	Total_Loans	Total_returned	Total_Issued_OrOverdue

9. Member Engagement Metrics

For each member, calculate: total books borrowed, total books currently on loan, total fines paid, and average rating they give in reviews. Only include members who have borrowed at least one book

```sql
--- 9. Member Engagement Metrics
select m.Member_FullName,
count(l.Loan_ID) as Total_Books_Borrowed,

sum(case when l.Loan_Status in ('Issued','Overdue') then 1 else 0 end) as Total_Books_on_Loans,
sum(isnull(p.Amount,0)) AS Total_Fines_Paid,
avg(r.Rating) AS Avg_Rating
from Loan l
Join Members m on m.Member_ID = l.Member_ID
Join Payment p on l.Loan_ID = p.Loan_ID
Join Reviews r on m.Member_ID = r.Member_ID
Group by m.Member_FullName
having count(l.Loan_ID) >= 1

select * from Members
```

89 %

Results Messages

	Member_FullName	Total_Books_Borrowed	Total_Books_on_Loans	Total_Fines_Paid	Avg_Rating
1	Ali Ahmed	3	3	37.50	5
2	Omar Khaled	1	1	45.75	3
3	Sara Mohammed	2	0	18.00	4

10. Library Performance Comparison

Compare libraries by showing: library name, total books owned, total active members (members with at least one loan), total revenue from fines, and average books per member

```sql
--- 10. Library Performance Comparison
select l.Library_Name,
count(b.Book_ID) AS Total_Books_Owned,

count(DISTINCT case when lo.Loan_ID IS NOT NULL THEN lo.Member_ID END) AS Total_Active_Members,
sum(isnull(p.Amount,0)) as Total_Fines_Paid,

cast(count(b.Book_ID) * 1.0 / nullif(count(DISTINCT lo.Member_ID),0)AS DECIMAL(10,2)) as Avg_Books

from Libraries l
Join books b on l.Library_ID = b.Library_ID
LEFT JOIN Loan lo ON b.Book_ID = lo.Book_ID
LEFT JOIN Payment p ON lo.Loan_ID = p.Loan_ID
Group by l.Library_Name
```

89 %

Results Messages

	Library_Name	Total_Books_Owned	Total_Active_Members	Total_Fines_Paid	Avg_Books_Per_Member
1	Central Library	29	10	73.50	2.90
2	City Library	17	5	195.50	3.40
3	Uni Library	6	0	0.00	NULL

11. High-Value Books Analysis

Identify books priced above the average book price in their genre. Show book title, genre, price, genre average price, and difference from average

```sql
--- 11. High-Value Books Analysis
SELECT
    b.Book_Title,
    b.Book_Genre,
    b.Book_Price,
    AVG(b2.Book_Price) AS Genre_Avg_Price,
    b.Book_Price - AVG(b2.Book_Price) AS Price_Difference
FROM Books b
JOIN Books b2 ON b.Book_Genre = b2.Book_Genre
GROUP BY b.Book_ID, b.Book_Title, b.Book_Genre, b.Book_Price
HAVING b.Book_Price > AVG(b2.Book_Price)
```

89 %

Results Messages

	Book_Title	Book_Genre	Book_Price	Genre_Avg_Price	Price_Difference
1	Harry Potter	Children	45.75	17.000000	28.750000
2	Harry Potter	Children	45.75	17.000000	28.750000
3	Fantasy World	Fiction	12.00	10.833333	1.166667
4	Mystery Night	Fiction	11.00	10.833333	0.166667
5	Fantasy World	Fiction	12.00	10.833333	1.166667
6	Mystery Night	Fiction	11.00	10.833333	0.166667
7	Cloud Computing	Non-fiction	21.00	18.000000	3.000000
8	AI Fundamentals	Non-fiction	22.50	18.000000	4.500000
9	Data Analysis	Non-fiction	20.00	18.000000	2.000000
10	Modern Physics	Non-fiction	23.00	18.000000	5.000000
11	Data Analysis	Non-fiction	20.00	18.000000	2.000000
12	Modern Physics	Non-fiction	23.00	18.000000	5.000000
13	Cloud Computing	Non-fiction	21.00	18.000000	3.000000
14	AI Fundamentals	Non-fiction	22.50	18.000000	4.500000
15	Python Programming	Reference	19.00	14.058823	4.941177
16	Java Basics	Reference	17.00	14.058823	2.941177
17	Advanced SQL	Reference	15.00	14.058823	0.941177
18	Networking	Reference	15.00	14.058823	0.941177
19	Advanced SQL	Reference	15.00	14.058823	0.941177
20	Python Programming	Reference	19.00	14.058823	4.941177
21	Java Basics	Reference	17.00	14.058823	2.941177

12. Payment Pattern Analysis

Group payments by payment method and show: payment method, number of transactions, total amount collected, average payment amount, and percentage of total revenue

```sql
--- 12. Payment Pattern Analysis
select payment_Methed,
count(Payment_ID) as Transactions_Type,
sum(Amount) as Total_Amount,
avg(Amount) as Avg_payment_Amount,
cast(sum(Amount) * 100 / (select sum(Amount) from Payment) AS DECIMAL(5,2)) as Prnctage_of_Total_Revenue
from Payment
Group by payment_Methed

select * from Payment
```

89 %

Results Messages

	payment_Methed	Transactions_Type	Total_Amount	Avg_payment_Amount	Prnctage_of_Total_Revenue
1	Bank Transfer	4	183.00	45.750000	68.03
2	Cash	4	50.00	12.500000	18.59
3	Credit Card	4	36.00	9.000000	13.38

Section 3: Views Creation

13. vw_CurrentLoans

A view that shows all currently active loans (status 'Issued' or 'Overdue') with member details, book details, loan information, and calculated days until due (or days overdue)

```sql
--- 13. vw_CurrentLoans

CREATE VIEW vw_CurrentLoans AS
SELECT
    l.Loan_ID,m.Member_FullName,m.Member_Email,b.Book_Title,b.Book_Genre,lb.Library_Name,
    l.LoanDate,
    l.DueDate,
    l.Loan_Status,
    CASE
        WHEN l.Loan_Status = 'Overdue' THEN DATEDIFF(DAY, l.DueDate, GETDATE())
        ELSE DATEDIFF(DAY, GETDATE(), l.DueDate)
    END AS Days_Until_Due_Or_Overdue
FROM Loan l
JOIN Members m ON l.Member_ID = m.Member_ID
JOIN Books b ON l.Book_ID = b.Book_ID
JOIN Libraries lb ON b.Library_ID = lb.Library_ID
WHERE l.Loan_Status IN ('Issued','Overdue')

select * from vw_CurrentLoans
```

89 %

Results Messages

	Loan_ID	Member_FullName	Member_Email	PhoneNumber	Book_Title	Book_Genre	Library_Name	LoanDate	DueDate	Loan_Status	Days_Until_Due_Or_Overd
1	51	Ali Ahmed	ali@email.com	NULL	Introduction to Programming	Reference	Central Library	2024-03-01	2024-04-15	Issued	-625
2	53	Omar Khaled	omar@email.com	NULL	Harry Potter	Children	City Library	2024-04-10	2024-05-20	Overdue	590
3	54	Anoud	Anoud@example.com	NULL	Networking	Reference	Central Library	2024-05-01	2024-05-10	Issued	-600
4	56	Sara Mohammed	saraa@email.com	NULL	Database Systems	Non-fiction	Central Library	2024-05-03	2024-05-13	Issued	-597
5	57	Omar Khaled	omar@email.com	NULL	Harry Potter	Children	City Library	2024-05-04	2024-05-14	Overdue	596
6	59	Mona Said	mona2@email.com	NULL	Advanced SQL	Reference	Central Library	2024-05-06	2024-05-16	Issued	-594
7	60	Yousef Nasser	yousef3@email.com	NULL	Cyber Security 101	Non-fiction	Central Library	2024-05-07	2024-05-17	Overdue	593
8	61	Aisha Salem	aisha4@email.com	NULL	Networking Basics	Non-fiction	Central Library	2024-05-08	2024-05-18	Issued	-592
9	63	Fatma Ahmed	fatma6@email.com	NULL	Fairy Tales	Children	City Library	2024-05-10	2024-05-20	Issued	-590
10	64	Nasser Hamed	nasser7@email.com	NULL	Short Stories	Fiction	City Library	2024-05-11	2024-05-21	Overdue	589
11	65	Noor Rashid	noor8@email.com	NULL	Mystery Night	Fiction	City Library	2024-05-12	2024-05-22	Issued	-588

14. vw_LibraryStatistics

A comprehensive view showing library-level statistics including total books, available books, total members, active loans, total staff, and total revenue from fines

```sql
--- 14. vw_LibraryStatistics

CREATE VIEW vw_LibraryStatistics AS

SELECT lb.Library_ID,lb.Library_Name,
COUNT( b.Book_ID) AS Total_Books,
Sum(case when b.Book_IsAvailable = 1 then 1 else 0 end) as Available_Books,
count(m.Member_ID) AS Total_Members,
count(case when l.Loan_Status in ('Issued','Overdue') then 1 else 0 end) as Active_Loans,
count(s.staff_ID) as Total_Saff,
sum(ISNULL(p.Amount,0)) as Total_revenue

from Libraries lb
Join Books b on lb.Library_ID = b.Library_ID
JOIN Loan l ON b.Book_ID = l.Book_ID
JOIN Members m ON l.Member_ID = m.Member_ID
JOIN staff_library s ON lb.Library_ID = s.Library_ID
JOIN Payment p ON l.Loan_ID = p.Loan_ID
GROUP BY lb.Library_ID, lb.Library_Name

select * from vw_LibraryStatistics
```

89 %

Results Messages

	Library_ID	Library_Name	Total_Books	Available_Books	Total_Members	Active_Loans	Total_Saff	Total_revenue
1	1	Central Library	14	14	14	14	14	147.00
2	2	City Library	5	5	5	5	5	195.50

15. vw_BookDetailsWithReviews

A view combining book information with aggregated review data (average rating, total reviews, latest review date) and current availability status.

```sql
--- 15. vw_BookDetailsWithReview

CREATE VIEW vw_BookDetailsWithReviews AS

select b.Book_ID,b.Book_Title,b.ISBN,b.Book_Genre,b.Book_Price,b.ShelfLocation,b.Book_IsAvailable,
count(r.Review_ID) as Total_Review,
avg(r.Rating) AS Avg_Rating,
MAX(r.ReviewDate) AS Latest_Review_Date

from Books b
JOIN Reviews r ON b.Book_ID = r.Book_ID
Group by b.Book_ID, b.Book_Title, b.ISBN, b.Book_Genre, b.Book_Price, b.ShelfLocation, b.Book_IsAvailable

select * from vw_BookDetailsWithReviews
```

89 %

Results | Messages

	Book_ID	Book_Title	ISBN	Book_Genre	Book_Price	ShelfLocation	Book_IsAvailable	Total_Review	Avg_Rating	Latest_Review_Date
1	1	Introduction to Programming	0131103	Reference	12.50	A-12	1	1	5	2025-12-01
2	2	Database Systems	321380	Non-fiction	9.00	B-05	1	1	4	2025-12-05
3	3	Harry Potter	043190	Children	45.75	C-01	1	1	3	2025-12-10

Section 4: Stored Procedures

16. sp_IssueBook

Input Parameters: MemberID, BookID, DueDate

Functionality:

- Check if book is available
- Check if member has any overdue loans
- If validations pass, create a new loan record and update book availability
- Return appropriate success or error message

```
SELECT 'Success: Book issued successfully.' AS Message
END

EXEC sp_IssueBook
    @MemberID = 1,
    @BookID = 3,
    @DueDate = '2025-01-20'

    SELECT * FROM Loan --- TO CHECK
```

89 %

Results Messages

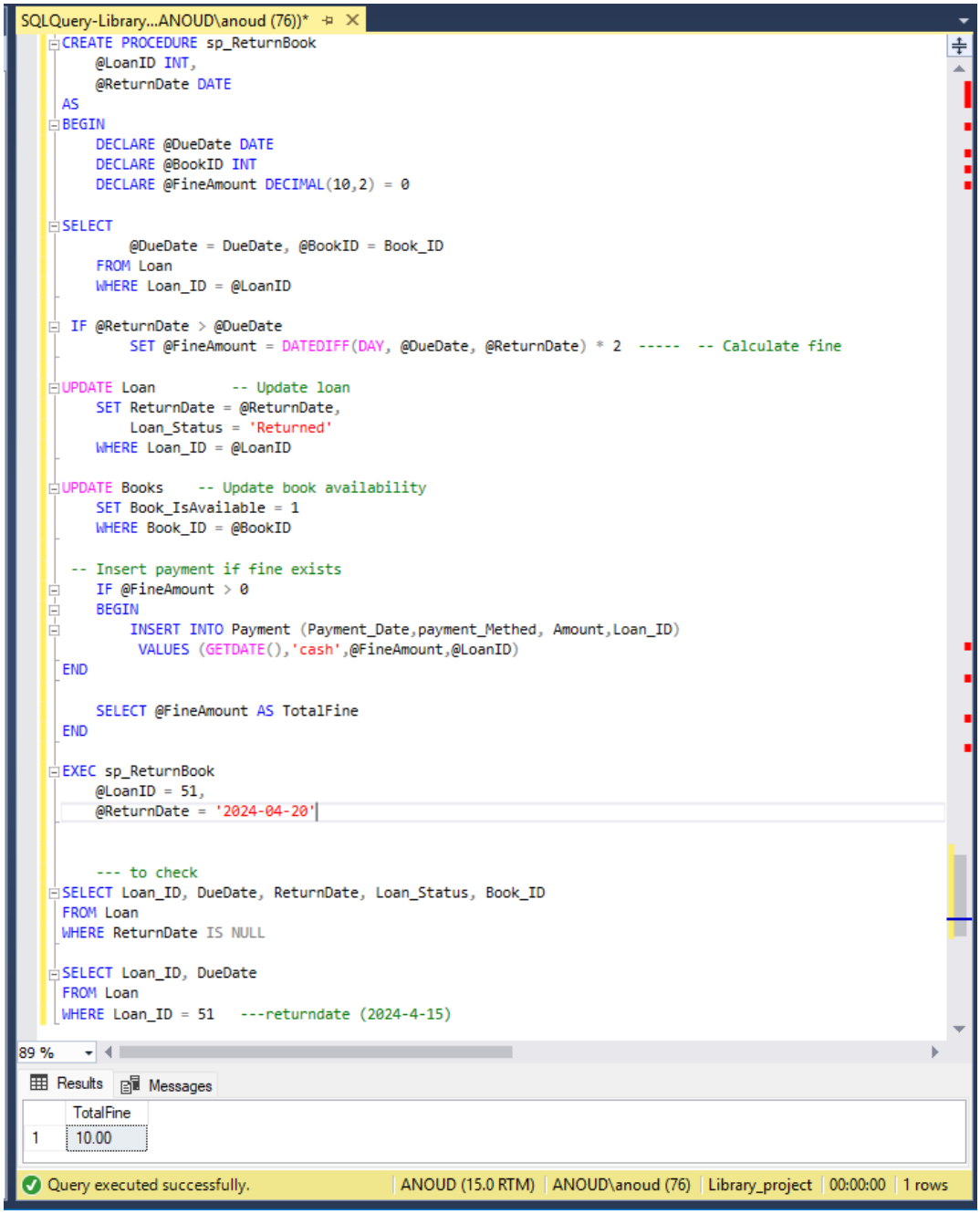
	Loan_ID	LoanDate	DueDate	ReturnDate	Loan_Status	Book_ID	Member_ID
1	51	2024-03-01	2024-04-15	NULL	Issued	1	1
2	52	2024-01-01	2024-08-10	2024-03-09	Returned	2	2
3	53	2024-04-10	2024-05-20	2024-04-22	Overdue	3	3
4	54	2024-05-01	2024-05-10	NULL	Issued	1011	405
5	55	2024-05-02	2024-05-12	2024-05-12	Returned	1013	413
6	56	2024-05-03	2024-05-13	NULL	Issued	1014	414
7	57	2024-05-04	2024-05-14	2024-05-15	Overdue	1015	415
8	58	2024-05-05	2024-05-15	2024-05-14	Returned	1016	416
9	59	2024-05-06	2024-05-16	NULL	Issued	1017	417
10	60	2024-05-07	2024-05-17	2024-05-18	Overdue	1018	418
11	61	2024-05-08	2024-05-18	NULL	Issued	1019	419
12	62	2024-05-09	2024-05-19	2024-05-19	Returned	1020	420
13	63	2024-05-10	2024-05-20	NULL	Issued	1021	421
14	64	2024-05-11	2024-05-21	2024-05-22	Overdue	1022	422
15	65	2024-05-12	2024-05-22	NULL	Issued	1023	423
16	66	2025-12-31	2025-01-20	NULL	Issued	3	1

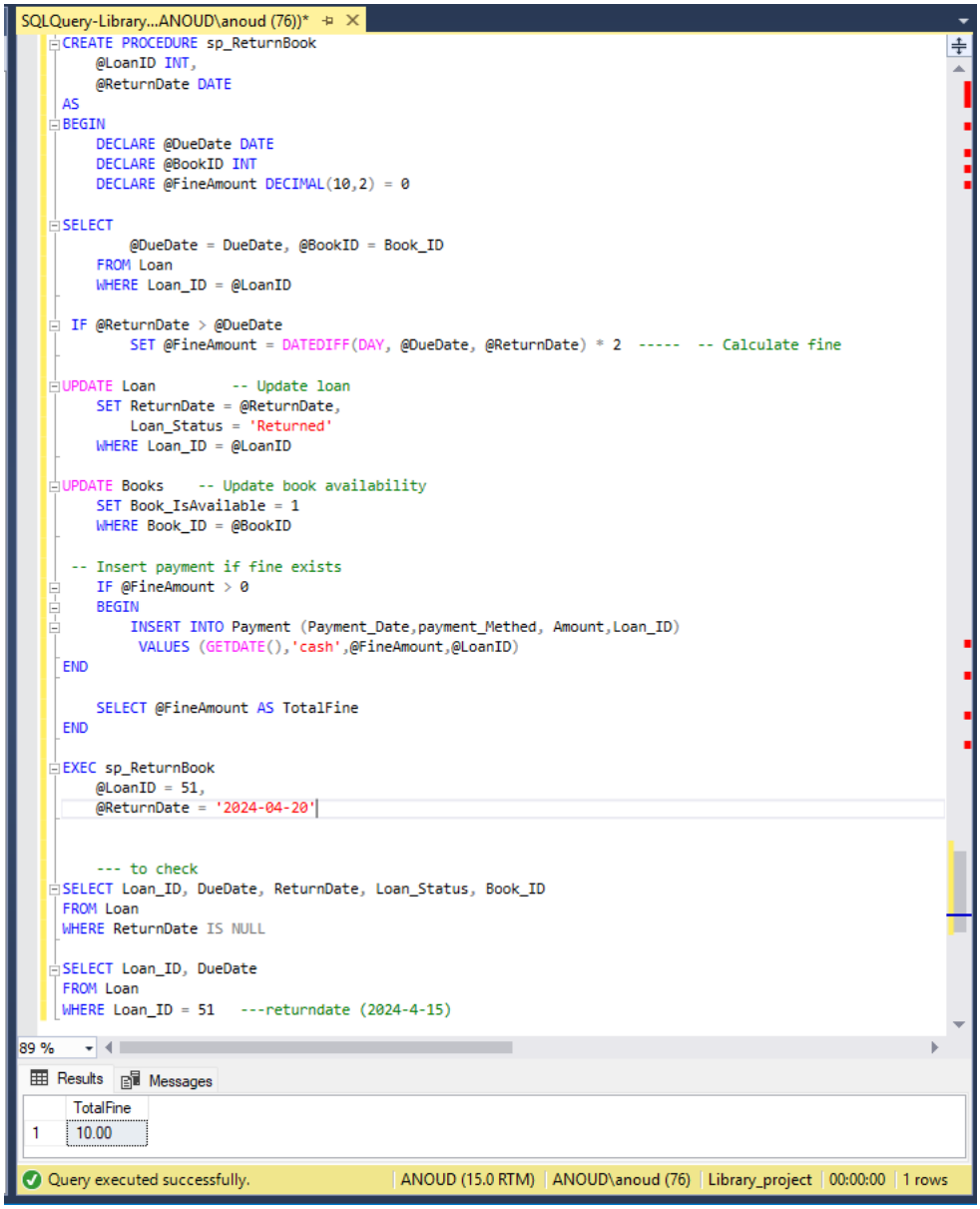
17. sp_ReturnBook

Input Parameters: LoanID, ReturnDate

Functionality:

- Update loan status to 'Returned' and set return date
- Update book availability to TRUE
- Calculate if there's a fine (e.g., $2 per day overdue)
- If fine exists, automatically create a payment record with 'Pending' status
- Return total fine amount (if any)


SQLQuery-Library...ANOUD\anoud (76))*
CREATE PROCEDURE sp_ReturnBook
@LoanID INT,
@ReturnDate DATE
AS
BEGIN
DECLARE @DueDate DATE
DECLARE @BookID INT
DECLARE @FineAmount DECIMAL(10,2) = 0
SELECT
@DueDate = DueDate, @BookID = Book_ID
FROM Loan
WHERE Loan_ID = @LoanID
IF @ReturnDate > @DueDate
SET @FineAmount = DATEDIFF(DAY, @DueDate, @ReturnDate) * 2 ----- -- Calculate fine
UPDATE Loan -- Update loan
SET ReturnDate = @ReturnDate,
Loan_Status = 'Returned'
WHERE Loan_ID = @LoanID
UPDATE Books -- Update book availability
SET Book_IsAvailable = 1
WHERE Book_ID = @BookID
-- Insert payment if fine exists
IF @FineAmount > 0
BEGIN
INSERT INTO Payment (Payment_Date,payment_Methed, Amount,Loan_ID)
VALUES (GETDATE(),'cash',@FineAmount,@LoanID)
END
SELECT @FineAmount AS TotalFine
END
EXEC sp_ReturnBook
@LoanID = 51,
@ReturnDate = '2024-04-20'
--- to check
SELECT Loan_ID, DueDate, ReturnDate, Loan_Status, Book_ID
FROM Loan
WHERE ReturnDate IS NULL
SELECT Loan_ID, DueDate
FROM Loan
WHERE Loan_ID = 51 ---returndate (2024-4-15)
89 %
Results
Messages
TotalFine
1
10.00
Query executed successfully.
ANOUD (15.0 RTM)
ANOUD\anoud (76)
Library_project
00:00:00
1 rows


SQLQuery-Library...ANOUD\anoud (76))*
CREATE PROCEDURE sp_ReturnBook
@LoanID INT,
@ReturnDate DATE
AS
BEGIN
DECLARE @DueDate DATE
DECLARE @BookID INT
DECLARE @FineAmount DECIMAL(10,2) = 0
SELECT
@DueDate = DueDate, @BookID = Book_ID
FROM Loan
WHERE Loan_ID = @LoanID
IF @ReturnDate > @DueDate
SET @FineAmount = DATEDIFF(DAY, @DueDate, @ReturnDate) * 2 ----- -- Calculate fine
UPDATE Loan -- Update loan
SET ReturnDate = @ReturnDate,
Loan_Status = 'Returned'
WHERE Loan_ID = @LoanID
UPDATE Books -- Update book availability
SET Book_IsAvailable = 1
WHERE Book_ID = @BookID
-- Insert payment if fine exists
IF @FineAmount > 0
BEGIN
INSERT INTO Payment (Payment_Date,payment_Methed, Amount,Loan_ID)
VALUES (GETDATE(),'cash',@FineAmount,@LoanID)
END
SELECT @FineAmount AS TotalFine
END
EXEC sp_ReturnBook
@LoanID = 51,
@ReturnDate = '2024-04-20'
--- to check
SELECT Loan_ID, DueDate, ReturnDate, Loan_Status, Book_ID
FROM Loan
WHERE ReturnDate IS NULL
SELECT Loan_ID, DueDate
FROM Loan
WHERE Loan_ID = 51 ---returndate (2024-4-15)
89 %
Results
Messages
TotalFine
1
10.00
Query executed successfully.
ANOUD (15.0 RTM)
ANOUD\anoud (76)
Library_project
00:00:00
1 rows

18. sp_GetMemberReport
Input Parameters: MemberID
Output: Multiple result sets showing:
• Member basic information
• Current loans (if any)
• Loan history with return status
• Total fines paid and any pending fines
• Reviews written by the member

```
    SELECT *        ----loan register
    FROM Loan
    WHERE Member_ID = @MemberID;

                    --- total loan
    SELECT SUM(P.Amount) AS TotalFines
    FROM Payment P
    JOIN Loan L ON P.Loan_ID = L.Loan_ID
    WHERE L.Member_ID = @MemberID;

                    ---- reviews
    SELECT *
    FROM Reviews
    WHERE Member_ID = @MemberID;
END

EXEC sp_GetMemberReport @MemberID = 1

SELECT Member_ID, Member_FullName
FROM Members
--- Member_ID = 1


SELECT *
FROM Members
WHERE Member_ID = 1

---The procedure was executed successfully and returned multiple result sets including member details,
---current loans, loan history, total fines, and reviews.
```

89 %

Results Messages

	Member_ID	Member_FullName	Member_Email	Membership_StartDate	Phone_Number	PhoneNumber
1	1	Ali Ahmed	ali@email.com	2024-01-15	NULL	NULL

	Loan_ID	LoanDate	DueDate	ReturnDate	Loan_Status	Book_ID	Member_ID
1	66	2025-12-31	2025-01-20	NULL	Issued	3	1

	Loan_ID	LoanDate	DueDate	ReturnDate	Loan_Status	Book_ID	Member_ID
1	51	2024-03-01	2024-04-15	2024-04-20	Returned	1	1
2	66	2025-12-31	2025-01-20	NULL	Issued	3	1

	TotalFines
1	47.50

	Review_ID	Rating	Comments	ReviewDate	Book_ID	Member_ID
1	1	5	Excellent book, highly recommend!	2025-12-01	1	1

19. sp_MonthlyLibraryReport
Input Parameters: LibraryID, Month, Year
Output: Comprehensive report showing:
• Total loans issued in that month
• Total books returned in that month
• Total revenue collected
• Most borrowed genre
• Top 3 most active members (by number of loans)

```sql
SELECT COUNT(*) AS TotalLoans        ---- Total loans issued
FROM Loan L
JOIN Books B ON L.Book_ID = B.Book_ID
WHERE B.Library_ID = 1
  AND MONTH(L.LoanDate) = 3
  AND YEAR(L.LoanDate) = 2024


  SELECT TOP 1 B.Book_Genre, COUNT(*) AS BorrowCount  --- Most borrowed genre
FROM Loan L
JOIN Books B ON L.Book_ID = B.Book_ID
WHERE B.Library_ID = 1
GROUP BY B.Book_Genre
ORDER BY BorrowCount DESC

SELECT TOP 3 M.Member_FullName, COUNT(*) AS LoanCount   --- Top 3 most active members
FROM Loan L
JOIN Members M ON L.Member_ID = M.Member_ID
JOIN Books B ON L.Book_ID = B.Book_ID
WHERE B.Library_ID = 1
GROUP BY M.Member_FullName
ORDER BY LoanCount DESC



---The monthly library report was validated by executing the procedure for different months and comparing each result
--set with equivalent aggregate queries.
---The outputs accurately reflected loan activity, returns, revenue, popular genres, and member activity.
```

89 %

Results Messages

	TotalLoans
1	1

	Book_Genre	BorrowCount
1	Reference	5

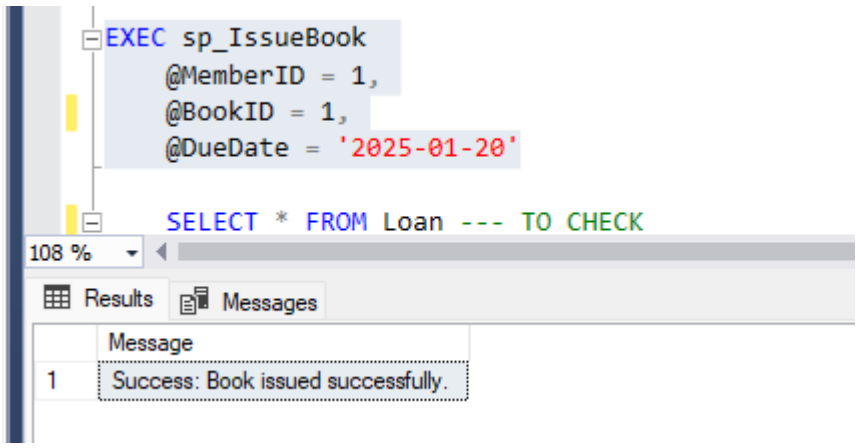
	Member_FullName	LoanCount
1	Ali Ahmed	2
2	Sara Mohammed	2
3	Hassan Ali	1

Query executed successfully. | ANOUD (15.0 RTM) | ANOUD\anoud (76) | Library_project | 00:00:00 | 5 rows

Testing Evidence

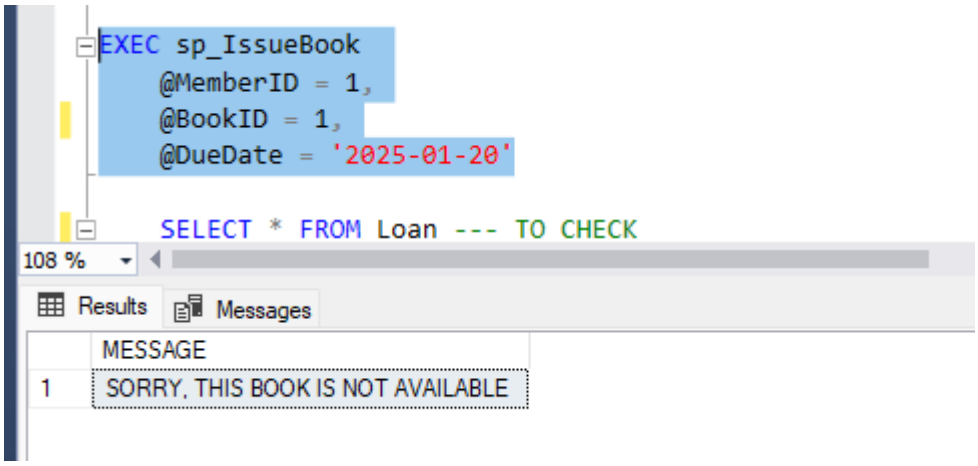
Section_4 / question 16

Add first time :



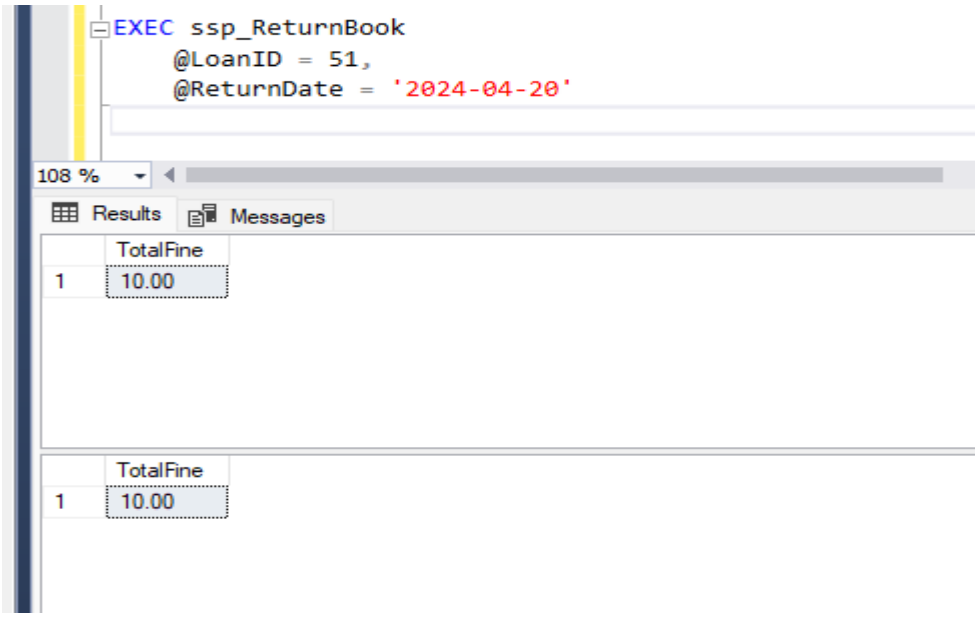
When i want to add again , msg show

Test Case – Validation Error (Book not available)



Section_4 / question 17

Test Case 1 – Overdue Return (Fine Created)



Test Case 2 – On-Time Return (No Fine)