Library Management System

Documentation Document

1. Explanation of Approach for Complex Queries

Query 5: Book Popularity Report

Objective:
Identify books loaned at least 3 times and show popularity metrics.

Approach:

- Used `JOIN` between **Book**, **Loan**, and **Review** tables.
- Applied `COUNT(Loan_Id)` to calculate how many times each book was loaned.
- Used `HAVG(Rating)` to calculate average review rating (if reviews exist).
- Applied `GROUP BY` on book attributes.
- Used `HAVING COUNT(*) >= 3` to filter popular books only.

Why this works:
`GROUP BY` aggregates loan records per book, and `HAVING` filters based on total loans.

Query 6: Member Reading History

Objective:
Show full borrowing history for each member, including returned and current loans and reviews.

Approach:

- Joined **Members → Loan → Book** to retrieve borrowing details.
- Used `LEFT JOIN` with **Review** so members appear even if they didn't review a book.
- Included both `LoanDate` and `ReturnDate` to reflect full history.

Why this works:

`LEFT JOIN` ensures no data is lost when optional information (reviews) is missing.

Query 7: Revenue Analysis by Genre

Objective:

Analyze fines collected per genre.

Approach:

- Joined **Book → Loan → Payment** tables.
- Grouped results by `Genre`.
- Used:
  - `COUNT(Loan_Id)` → number of loans
  - `SUM(Amount)` → total fines
  - `AVG(Amount)` → average fine per loan

Why this works:

Grouping by genre allows financial performance comparison across book categories.

2. Screenshots of Query Results

1. Library Book Inventory Report

Results | Messages

	LibraryName	TotalBook	AvailableBooks	BooksOnLoan
1	Muscat Library	7	3	4
2	Salalah Library	6	3	3
3	Sohar Library	7	0	2

2. Active Borrowers Analysis

Results Messages

	MemberName	Email	BookTitle	LoanDate	DueDate	CurrentStatus
1	Aliya Rashid	aliya@gmail.com	SQL Basics	2025-06-01	2025-06-15	Issued
2	Sara Al-Hinai	sara@gmail.com	Oman History	2025-06-03	2025-06-17	Overdue
3	Khalid Al-Badi	khalid@gmail.com	Web Development	2025-06-05	2025-06-19	Issued
4	Aisha Al-Raisi	aisha@gmail.com	Python Guide	2025-06-06	2025-06-20	Issued
5	Noor Al-Sabti	noor@gmail.com	Harry Potter	2025-06-08	2025-06-22	Issued
6	Maha Al-Lawati	maha@gmail.com	Short Stories	2025-06-10	2025-06-24	Overdue
7	Aliya Rashid	aliya@gmail.com	Startup Basics	2025-06-12	2025-06-26	Issued
8	Sara Al-Hinai	sara@gmail.com	Children Tales	2025-06-14	2025-06-28	Issued
9	Sara Al-Hinai	sara@gmail.com	Math for Kids	2025-06-15	2025-06-29	Issued

3. Revenue Analysis by Genre

Results Messages

	Genre	TotalLoans	TotalFinesCollected	AvgFinePerLoan
1	Non-Fiction	2	15.00	7.500000

4. High-Value Books Analysis

Results Messages

	Title	Genre	Price	GenreAvgPrice	DifferenceFromAvg
1	Digital Marketing	Business	26.00	24.000000	2.000000
2	Math for Kids	Children	14.00	13.000000	1.000000
3	Database Design	Education	25.00	24.166666	0.833334
4	Web Development	Education	28.00	24.166666	3.833334
5	Python Guide	Education	32.00	24.166666	7.833334
6	Harry Potter	Fiction	30.00	26.750000	3.250000
7	Lord of the Rings	Fiction	35.00	26.750000	8.250000
8	Arabic Literature	Fiction	27.00	26.750000	0.250000
9	Advanced SQL	Non-Fiction	30.00	22.250000	7.750000

5. Payment Pattern Analysis

	Method	Transactions	TotalAmount	AvgPayment	RevenuePercentage
1	Card	1	10.00	10.000000	66.666666
2	Cash	1	5.00	5.000000	33.333333

3. Stored Procedures – Edge Case Handling

sp_IssueBook

Edge Cases Handled:

- Prevents issuing books that are not available
- Prevents issuing if member has overdue loans
- Uses transaction control to avoid partial updates

Result:
Ensures business rules are enforced before issuing books.

sp_ReturnBook

Edge Cases Handled:

- Handles early, on-time, and late returns
- Calculates fines dynamically ($2 per overdue day)
- Automatically creates a pending payment if fine exists

Result:
Ensures accurate fine calculation and payment tracking.

sp_GetMemberReport

Edge Cases Handled:

- Members with no current loans
- Members with no fines
- Members with no reviews

Result:
Procedure safely returns empty result sets without errors.

sp_MonthlyLibraryReport

Edge Cases Handled:

- Months with no activity
- Libraries with no payments
- Uses grouped queries to avoid incorrect totals

Result:
Returns NULL or empty values gracefully when data does not exist.

4. Assumptions Made

- Each book has a boolean field `IsAvailable`
- Fine rate is fixed at **$2 per overdue day**
- Payment method `'Pending'` indicates unpaid fines
- A member can borrow multiple books
- Reviews are optional
- Only valid foreign keys are inserted
- SQL Server database environment

3. Testing Evidence:

16. sp_IssueBook

```
--Execute
--Success
EXEC sp_IssueBook 4, 1032, '2025-12-01'
--Error
EXEC sp_IssueBook 4, 1032, '2025-12-01'
```

17. sp_ReturnBook

```
--execute
--success
EXEC sp_ReturnBook 11, '2025-12-05'
--(No fine)
EXEC sp_ReturnBook 7, '2025-12-05'
```

18. sp_GetMemberReport

```
]--execute:
 --successful
 EXEC sp_GetMemberReport 4
 --Error
 EXEC sp_GetMemberReport 999
 EXEC sp_helptext sp_GetMemberReport
```

--19. sp_MonthlyLibraryReport

```
--execute:
--successful
EXEC sp_MonthlyLibraryReport 1, 6, 1995
--output no data /null
EXEC sp_MonthlyLibraryReport @LibraryID= 1, @Month=1, @Year=2020

EXEC sp_helptext sp_MonthlyLibraryReport
```