LibraryDB – Short Documentation

---

1. Project Overview

LibraryDB is a small library management database created in SQL Server. It tracks: - Books, Authors, and Categories - Members and their borrow transactions - Audit logs for insert/update/delete actions

The project includes: - Tables – 6 tables - Sample Data – minimal data for testing - Stored Procedures – borrow, return, search books - Functions – calculate overdue days, list available books - Triggers – automatically log borrow/return actions

---

1. Table Explanation

Table Name	Purpose	Key Columns & Constraints
Books	Stores all books in the library	BookID (PK, identity), Title, AuthorID (FK), CategoryID (FK), PublishedYear, AvailableCopies
Authors	Stores author details	AuthorID (PK, identity), AuthorName, Country
Categories	Stores book categories	CategoryID (PK, identity), CategoryName (unique)
Members	Stores library members	MemberID (PK, identity), MemberName, Email (unique), JoinDate
Borrow_Transactions	Tracks which member borrowed which book	BorrowID (PK, identity), BookID (FK), MemberID (FK), BorrowDate, ReturnDate
Audit_Log	Tracks insert/update/ delete actions	LogID (PK, identity), ActionType, TableName, Description, CreatedAt

---

1. Query Explanation

2. List all books with authors and categories

```
select b.Title, a.AuthorName, c.CategoryName
from Books b
join Authors a on b.AuthorID = a.AuthorID
join Categories c on b.CategoryID = c.CategoryID;
```

3. Find overdue books (>30 days)

```
select bt.BorrowID, b.Title, m.MemberName, bt.BorrowDate
from Borrow_Transactions bt
join Books b on bt.BookID = b.BookID
join Members m on bt.MemberID = m.MemberID
where bt.ReturnDate is null
  and dbo.fn_overdue_days(convert(date, bt.BorrowDate, 105)) > 0;
```

4. Top borrowed books

```
select b.Title, count(bt.BookID) as TotalBorrowed,
       rank() over(order by count(bt.BookID) desc) as BorrowRank
from Borrow_Transactions bt
join Books b on bt.BookID = b.BookID
group by b.Title;
```

5. Members borrowing more than 3 books

```
select m.MemberName, count(bt.BookID) as TotalBorrowed
from Members m
join Borrow_Transactions bt on m.MemberID = bt.MemberID
group by m.MemberName
having count(bt.BookID) > 3;
```

6. Available books – Table-Valued Function

```
select * from dbo.fn_available_books();
```

---

1. Stored Procedures

Procedure Name	Purpose
sp_borrowbook @BookID, @MemberID	Records a borrow transaction, updates available copies, logs action
sp_returnbook @TransactionID	Marks a book as returned, updates available copies, logs action
sp_searchbooks @Keyword	Searches books by title, author, or category

1. Functions

Function Name	Type	Purpose
fn_overdue_days(@BorrowDate)	Scalar	Returns overdue days (>30) for borrowed books
fn_available_books()	Table-Valued	Returns all currently available books

1. Triggers

Trigger Name	Event	Purpose
trg_borrow_insert	AFTER INSERT on Borrow_Transactions	Adds entry in Audit_Log for borrow action
trg_return_update	AFTER UPDATE on Borrow_Transactions	Updates AvailableCopies and logs return action

1. How to Execute the Project

2. Create the database

```
create database LibraryDB;
use LibraryDB;
```

3. Create tables (Authors, Categories, Books, Members, Borrow_Transactions, Audit_Log)
4. Insert sample data – 10 books, 5 members, 5 authors, 5 categories, 5 borrow records
5. Create functions, stored procedures, and triggers
6. Test queries

Examples: - List all books:

```
select * from Books;
```

- Borrow a book:

```
exec sp_borrowbook @BookID = 301, @MemberID = 400;
```

- Return a book:

```
exec sp_returnbook @TransactionID = 5001;
```

- Check available books:

```
select * from dbo.fn_available_books();
```

- Find overdue books:

```
select * from Borrow_Transactions
where dbo.fn_overdue_days(convert(date, BorrowDate, 105)) > 0;
```