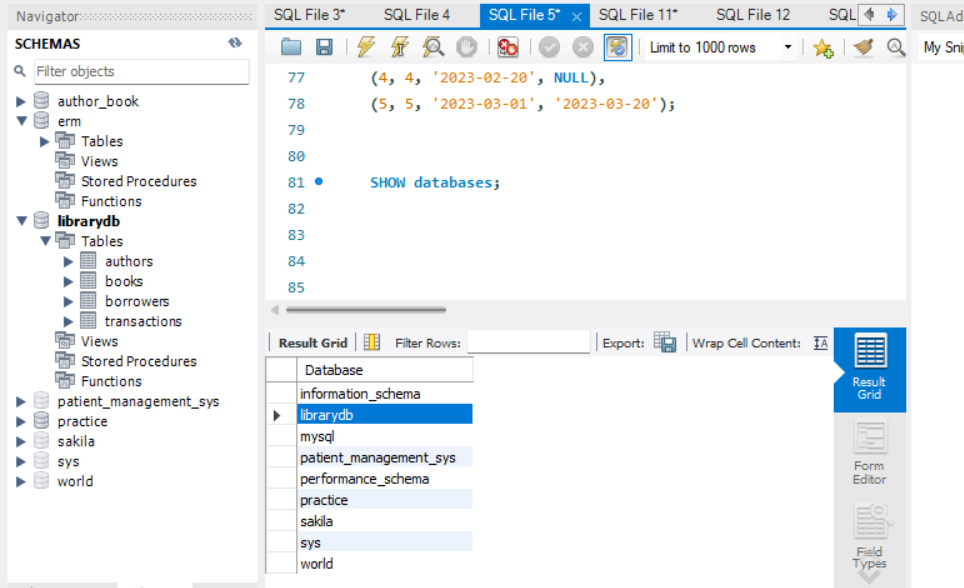
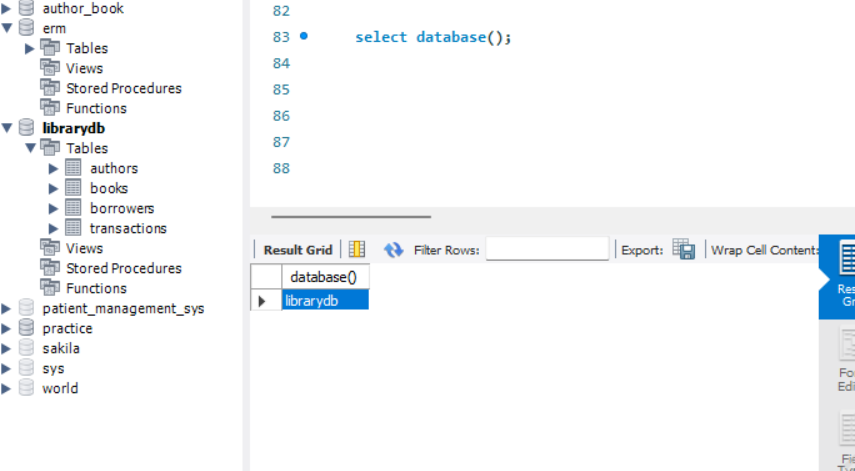
MySQL Project (Library DB) Implementation

1. A) Database Creation and Verification.

Running SHOW DATABASES; query to verify the presence of the database(librarydb)

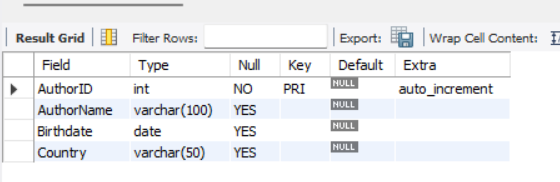


B) SELECT DATABASE();

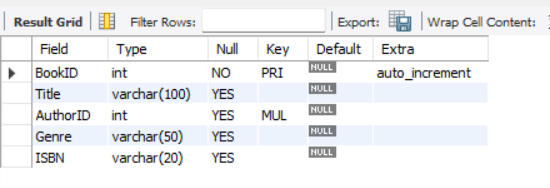


1. Table Structure.

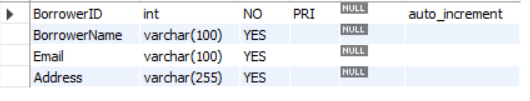
Authors Table(DESCRIBE Authors;)



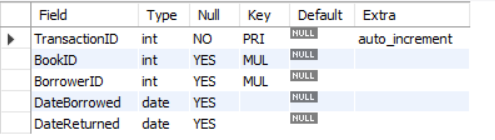
Books Table(DESCRIBE Books;)



Borrowers Table(DESCRIBE Borrowers;)

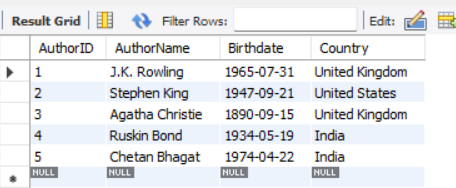


Transactions Table( DESCRIBE Transactions);

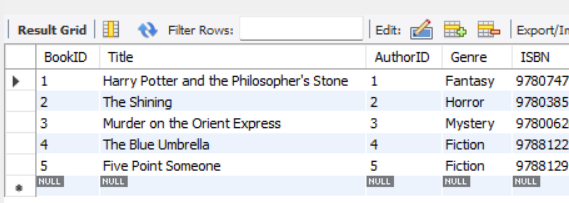


1. Data Insertion and Basic Queries

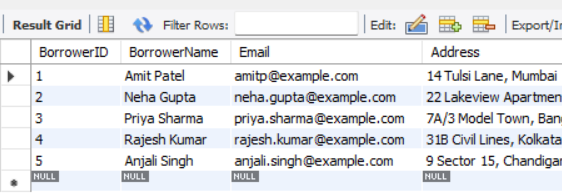
Authors Table (SELECT \* FROM Authors;)



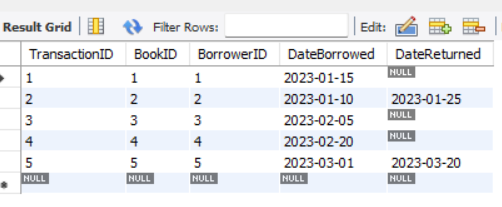
Books Table (SELECT \* FROM Books;)



Borrowers Table(SELECT \* FROM Borrowers;)



Transactions Table (SELECT \* FROM Transactions;)

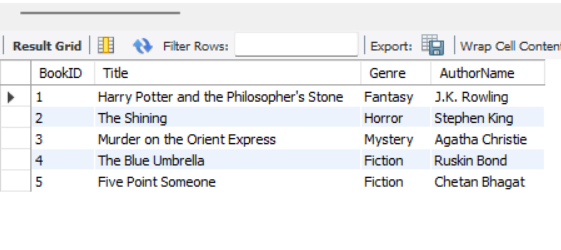


1. Advanced Queries and Functions.
2. Query to join Books and Authors tables to get book details with author names

SELECT Books.BookID ,Books.Title, Books.Genre, Authors.AuthorName

FROM Books

INNER JOIN Authors ON Books.AuthorID = Authors.AuthorID;

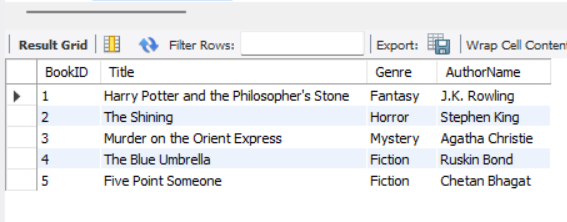


1. Query to demonstrate aliasing of table names for clearer queries

SELECT B.BookID, B.Title, B.Genre, A.AuthorName

FROM Books AS B

INNER JOIN Authors AS A ON B.AuthorID = A.AuthorID;



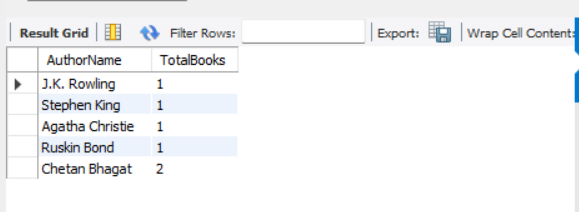
1. Example query using aggregate function (Count the number of books)

Before executing this query I will add a book under Chetan Bhagat

INSERT INTO Books (Title, AuthorID, Genre, ISBN)

VALUES ('2 States', 5, 'Fiction', '9788129135509');

Then executing the following query



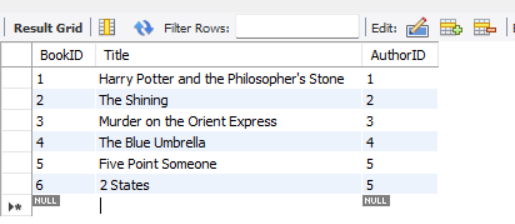
1. Validating Normalization Principles

1NF: Checking for atomic values in all the tables:

This we could check from step3 where we can see each column holds individual values and each row represents unique record without containing repeated groups within a single column. No violation of 1NF

2NF:Checking for 2NF

SELECT BookID, Title, AuthorID FROM Books;



Here We can notice that the primary key BookID is unique .

And the AuthorID references are proper

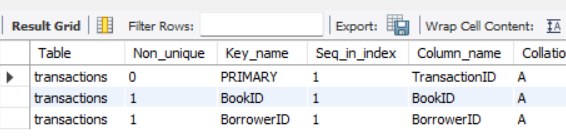
Every Title (nonkey) corresponds to a unique book in the Books Table.

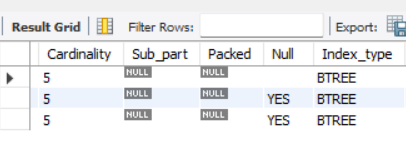
For 3NF we can take the same table, As we can see there is no transient relation between these non-key columns, So no violation

1. Database Exploration and Inspection
2. Examining table index

For transactions table

SHOW INDEX FROM transactions;





1. We can get the details about foreign key constraint on the table, For suppose I want to know on Books table

SELECT

CONSTRAINT\_NAME,

COLUMN\_NAME,

REFERENCED\_TABLE\_NAME,

REFERENCED\_COLUMN\_NAME

FROM

INFORMATION\_SCHEMA.KEY\_COLUMN\_USAGE

WHERE

TABLE\_SCHEMA = 'librarydb’

AND TABLE\_NAME = 'Books'

AND REFERENCED\_TABLE\_NAME=’Authors’;

