Library Management System

Ravi Teja Vanama

<https://github.com/rvanama17/Database-Project-1>

05/01/2022

**Project Proposal**

The purpose of the library management system is to run the library efficiently and at low cost. A fully automated system streamlines all tasks related to library operations. Book purchases, cataloging, indexing, distribution records, and inventory management activities are taken over by the software. Such software eliminates the need for repetitive manual labor and minimizes the possibility of error.

Library management system software helps reduce operating costs. Manually maintaining the library is labor intensive and requires a huge amount of paperwork. Automated systems reduce the need for labor and office supplies. This reduces operating costs. The system saves time for both users and librarians. With just one click, you can search for books available in the library. Librarians can easily answer questions about book availability. Adding, deleting, or editing a database is a simple process. It's easy to add new members or terminate existing memberships.

Library book inventory checks and reviews can be done in hours. Automated systems save a considerable amount of time compared to manual systems.

Library control structures are designed to manipulate the motion of books and hold data of the individuals in a library. The software program answer is designed primarily based totally at the device requirements, the human beings involved, the content material of the operation and the interest to be completed.

The device requirement in library control makes a speciality of the opportunity of look for books through title, writer or difficulty through the member. They ought to be capable of discover a book bodily through the precise identity code and the rack variety for every book. The device ought to offer info at the books held through the individuals. The device ought to restriction the variety of books that may be taken and the variety of days that a book may be saved for. The device ought to generate fines while due from the member.

The subsequent step makes a speciality of the capabilities of the librarian, the member and the device. Managing books through the librarian, looking for books through the individuals and notifications despatched through the device are certain in a case diagram.

The 1/3 step withinside the layout of the library control device software program is primarily based totally at the distinctive components of a library. The call of the library, the book info, member info, club cards, book reservations, book lending, cataloging, fines, book racks and notifications are consolidated as a category diagram.

The subsequent step similarly breaks down the capabilities into unique sports which can be completed in a library. An interest diagram primarily based totally at the number one sports – sorting out a book, returning a book and renewing a book – is drawn.

The very last degree is coding for every interest withinside the case, magnificence and interest diagrams. This is the maximum vital feature withinside the layout of the library control device software program.

Library control device gitHub, the software program created the usage of Python, shops information on a log file. The software program structures are evolved in this platform.

Book

|  |  |
| --- | --- |
| BookID | INT(PK) |
| BookTitle | Varchar(50) |
| Author | Varchar(50) |
| Publisher | Varchar(50) |
| PublishDate | Date |

User

|  |  |
| --- | --- |
| UserID | INT(PK) |
| UserName | Varchar(100) |
| Gender | Varchar(5) |
| Email | Varchar(50) |
| ContactAddress | Varchar(50) |
| BookID | INT (FK) |

Student

|  |  |
| --- | --- |
| StudentID | Int(PK) |
| StudentName | Varchar(50) |
| StudentGender | Varchar(5) |
| StudentEmail | Varchar(50) |
| StudentAddress | Varchar(50) |
| UserID | Int (FK) |

Transaction

|  |  |
| --- | --- |
| TranID | Int(PK) |
| TranName | Varchar(50) |
| BorrowID | Varchar(50) |
| BookID | Int(FK) |
| TranDate | Date |

Report

|  |  |
| --- | --- |
| ReportID | Int (PK) |
| TranID | Int(FK) |
| UserID | Int(FK) |
| ReportDate | Date |

**DATA SOURCES:**

**Book.csv**

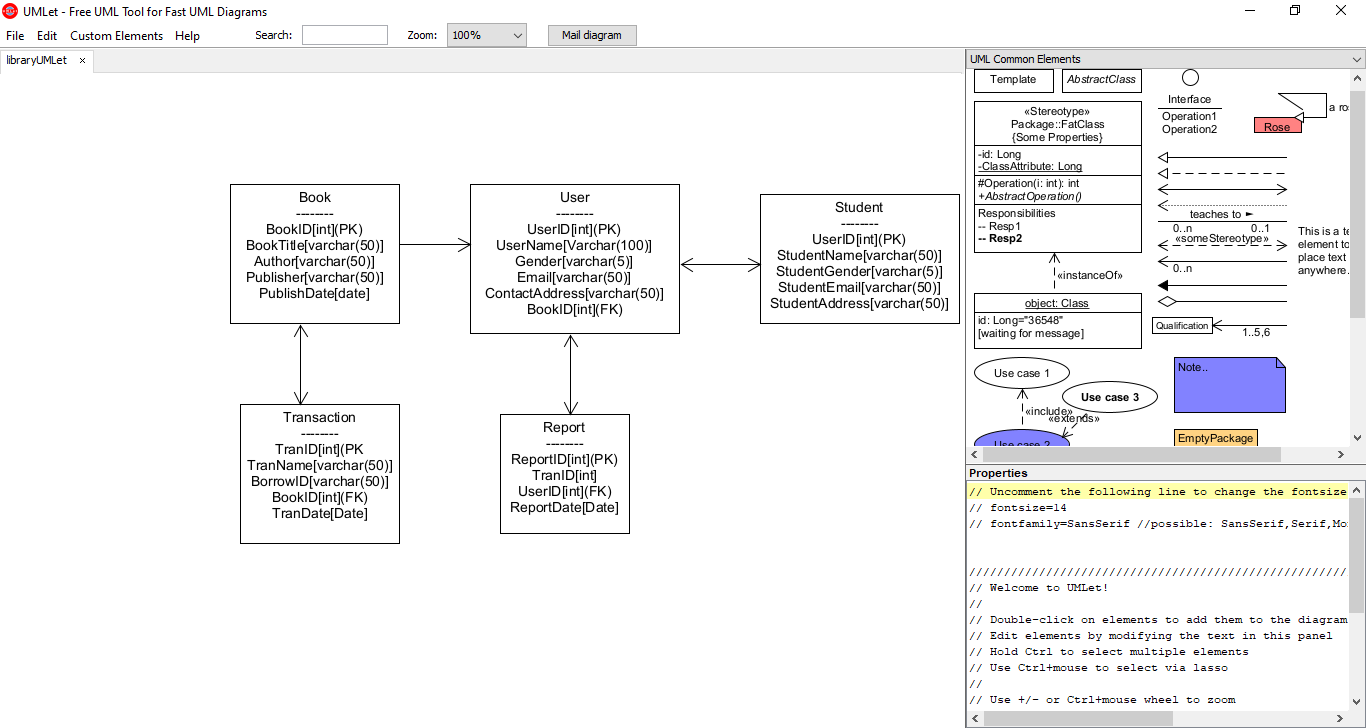
**User.csv**

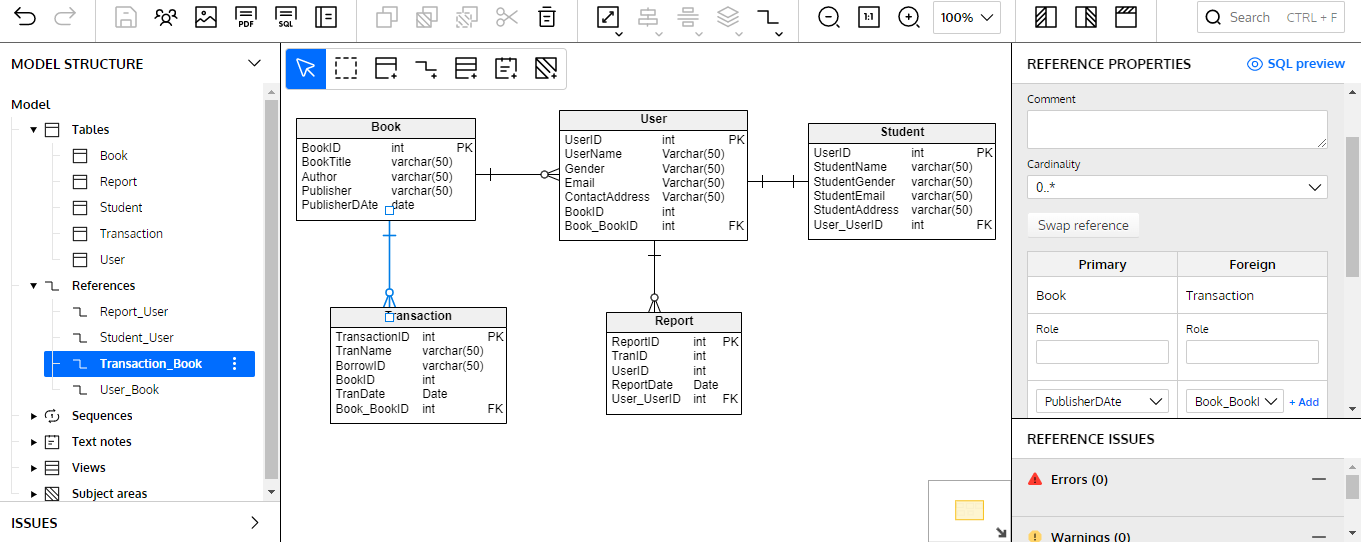
**Student.csv**

**Transaction.csv**

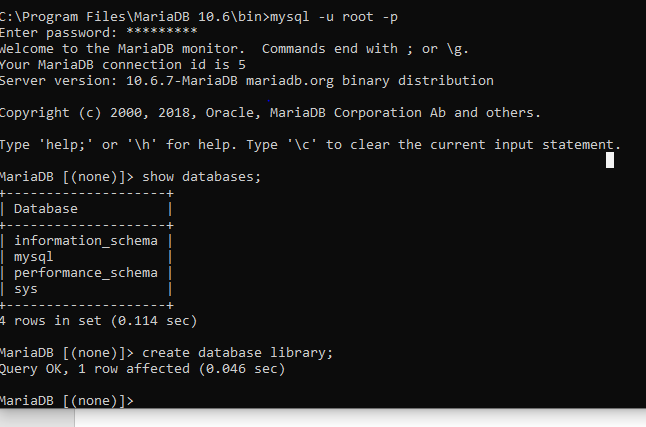
**Report.csv**

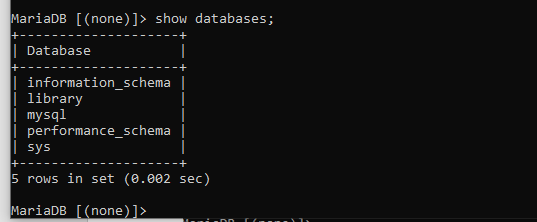
**ER Diagrams**

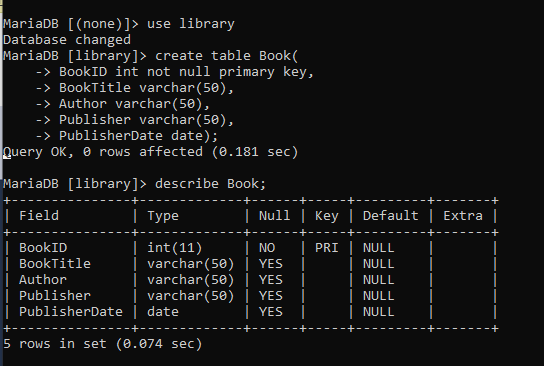


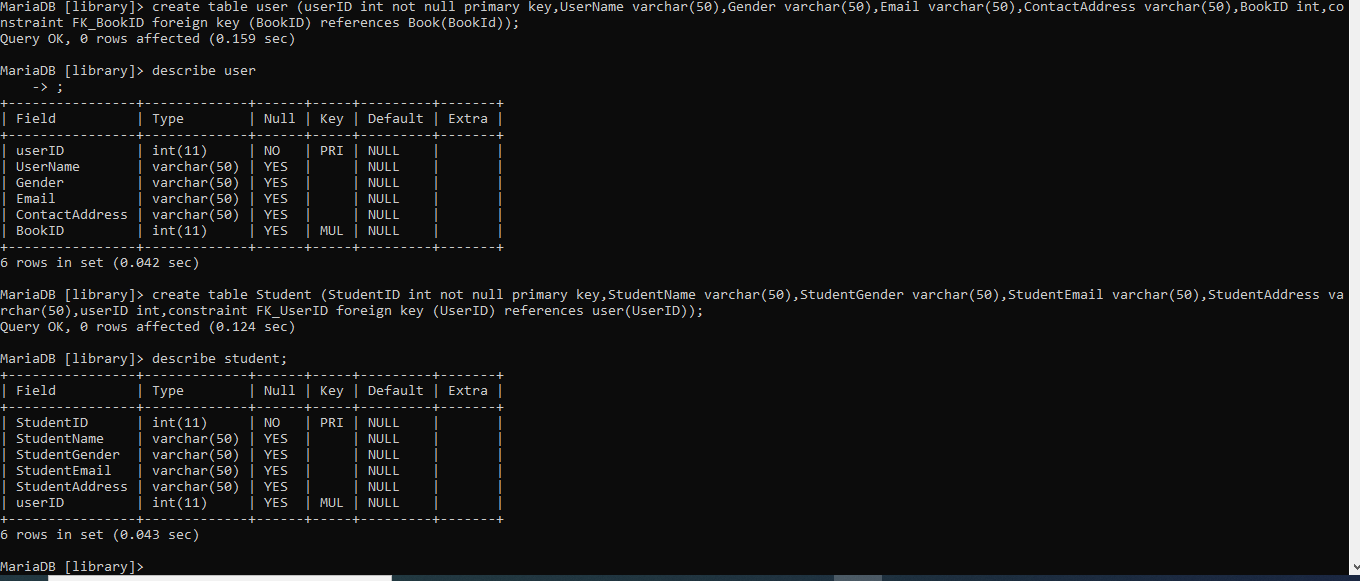


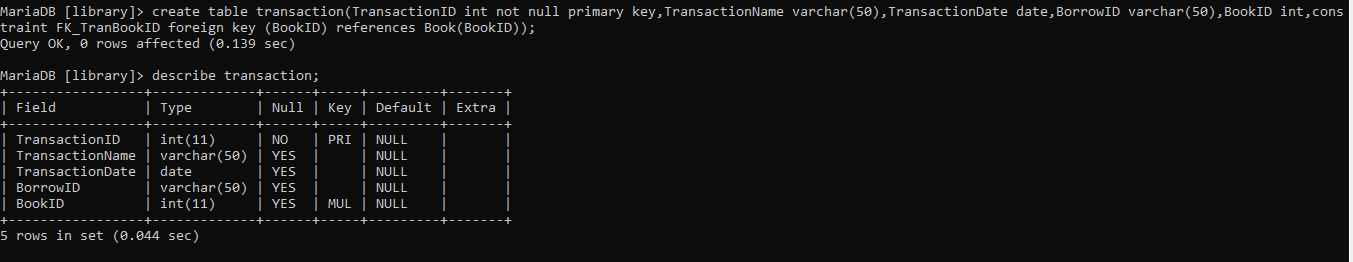
**Project Part 4**

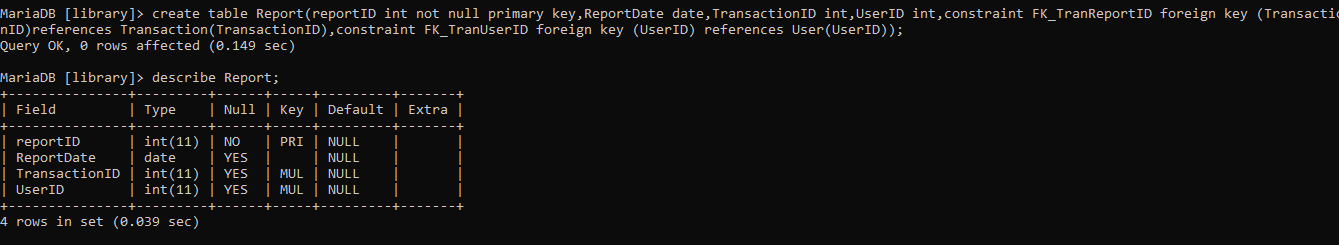
**Creation of Database Using MariaDB:**

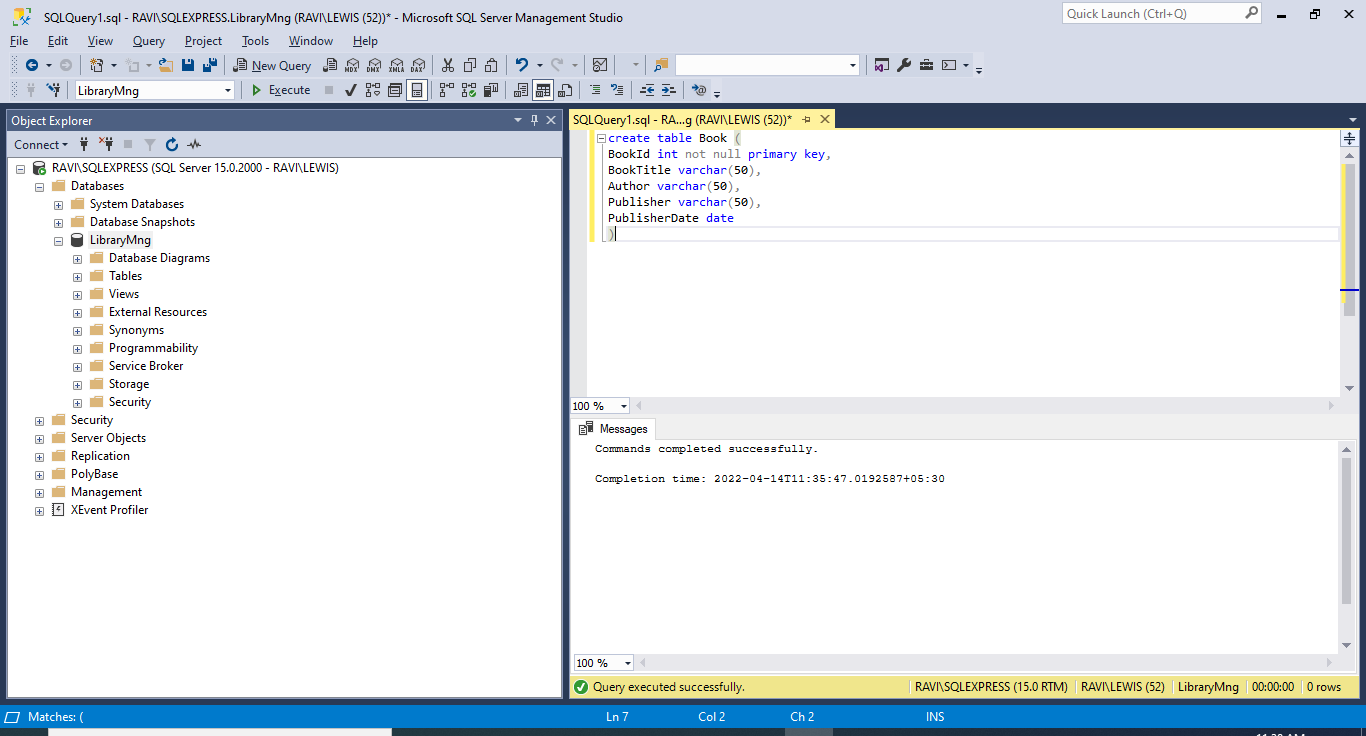


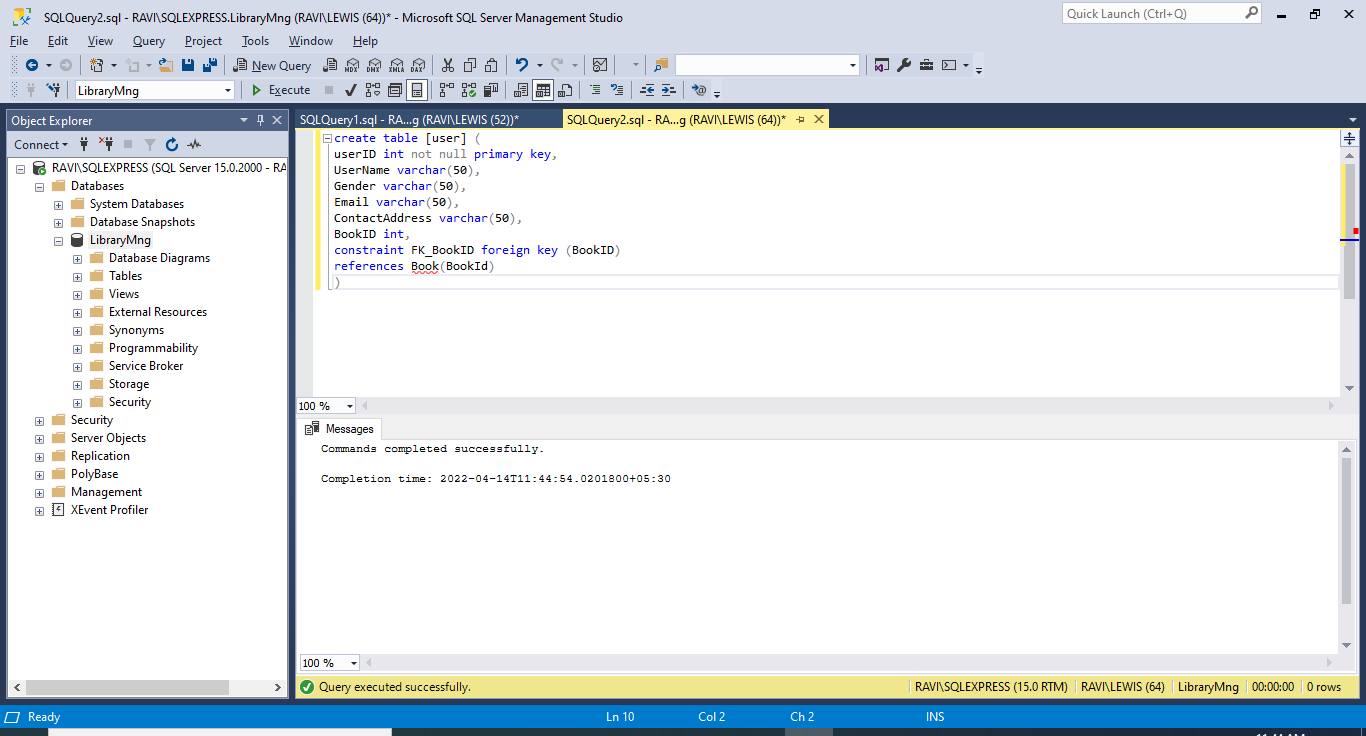


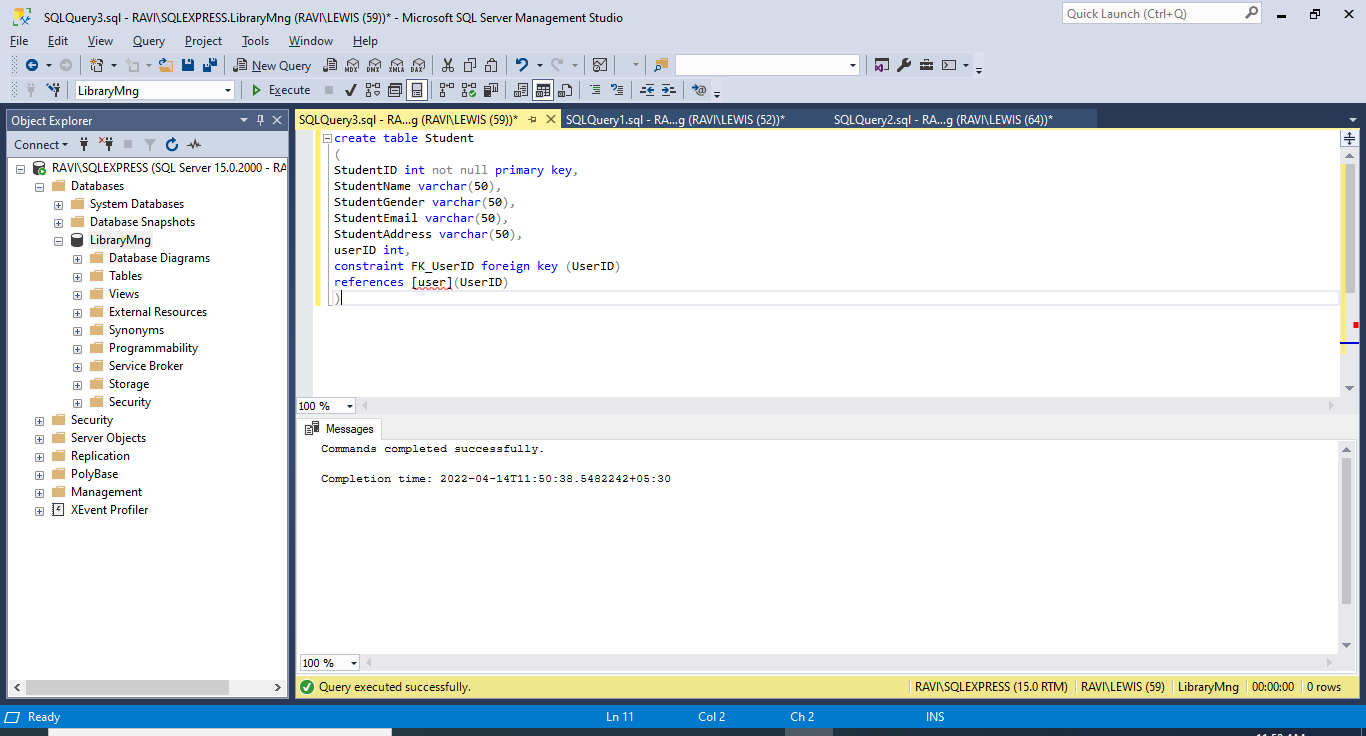


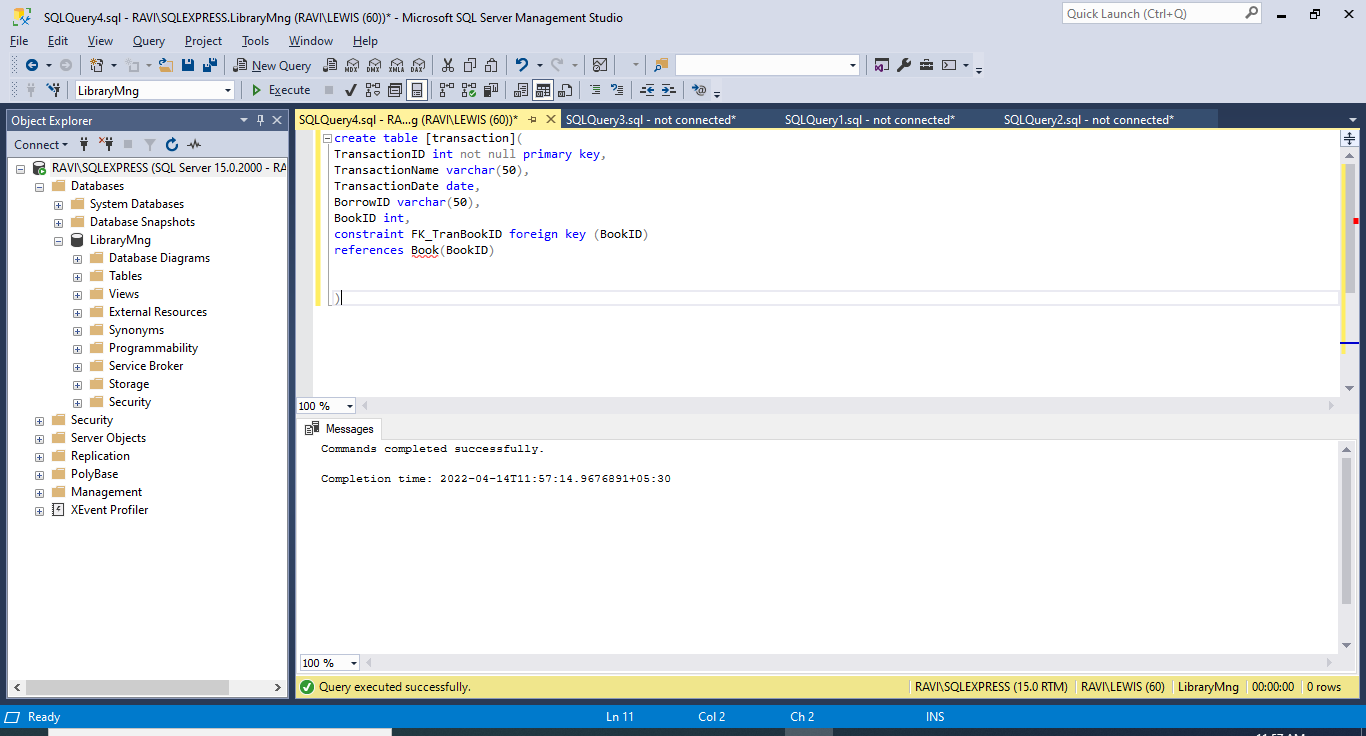


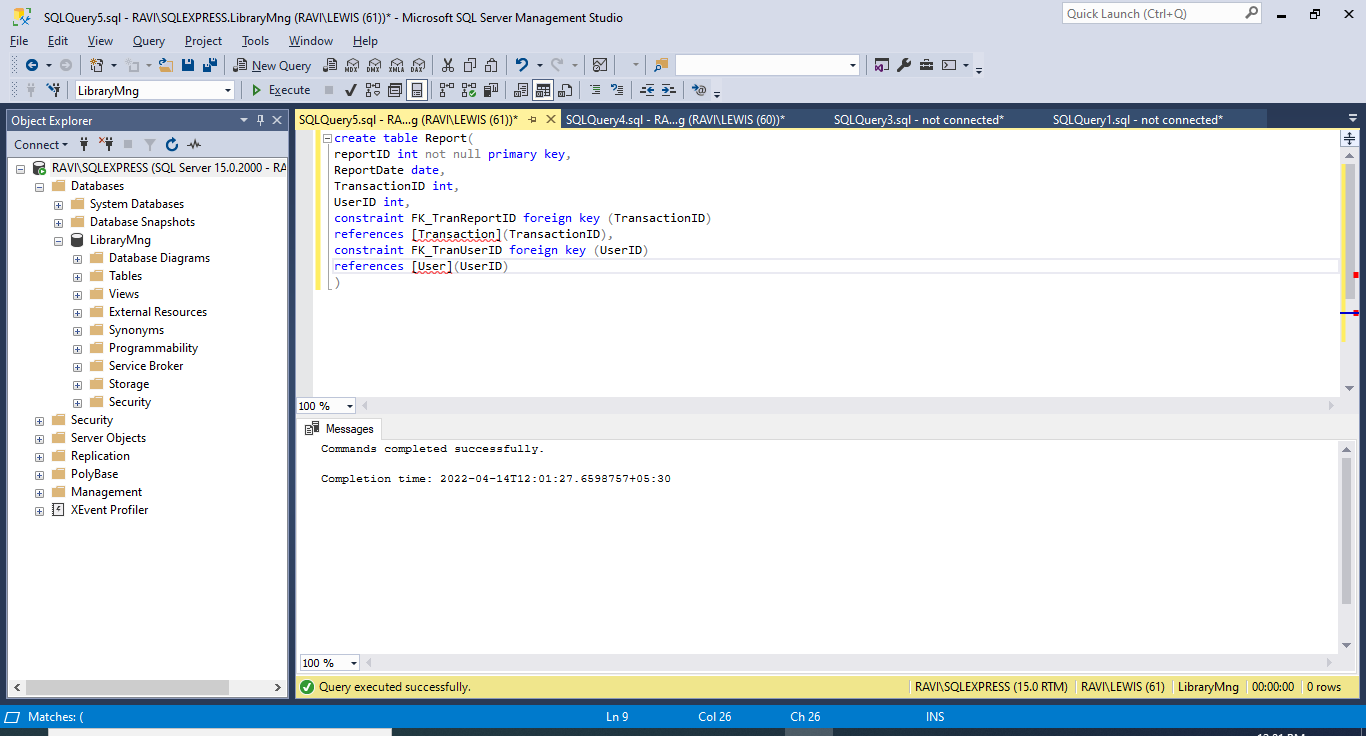




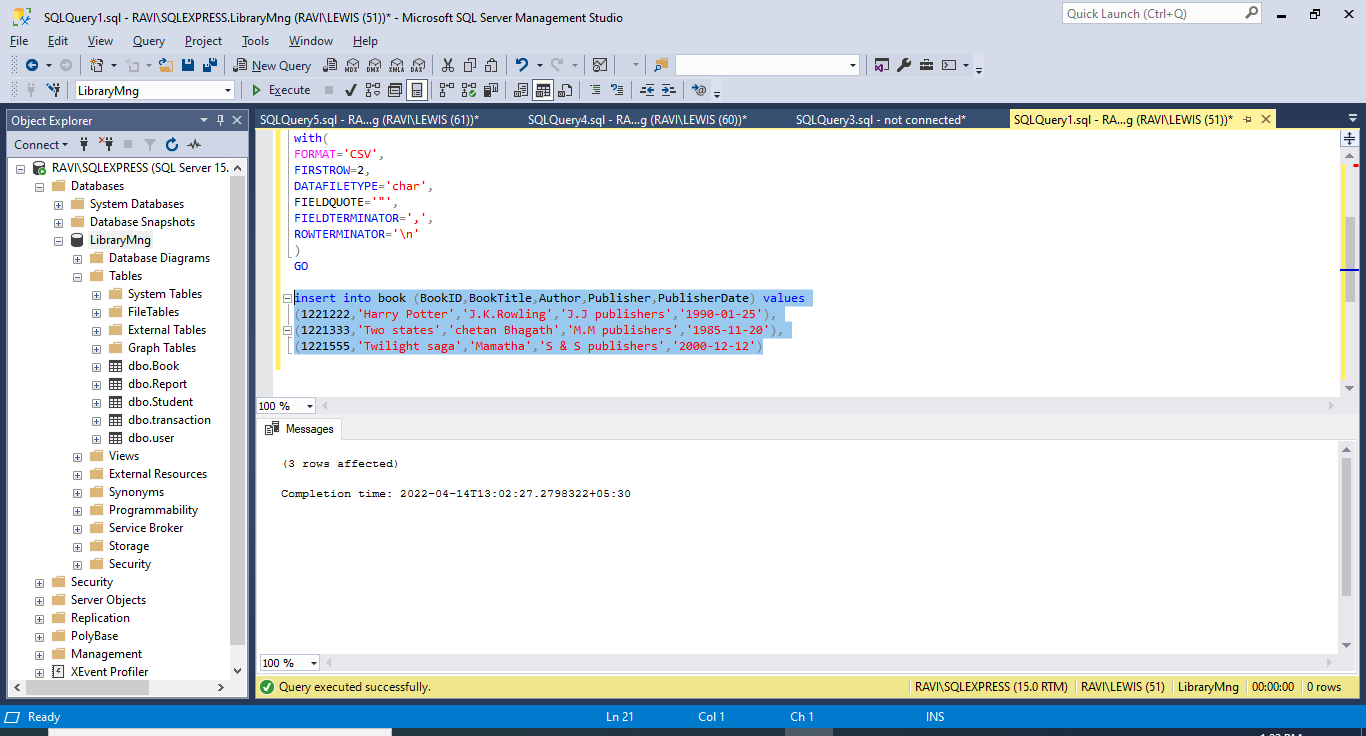


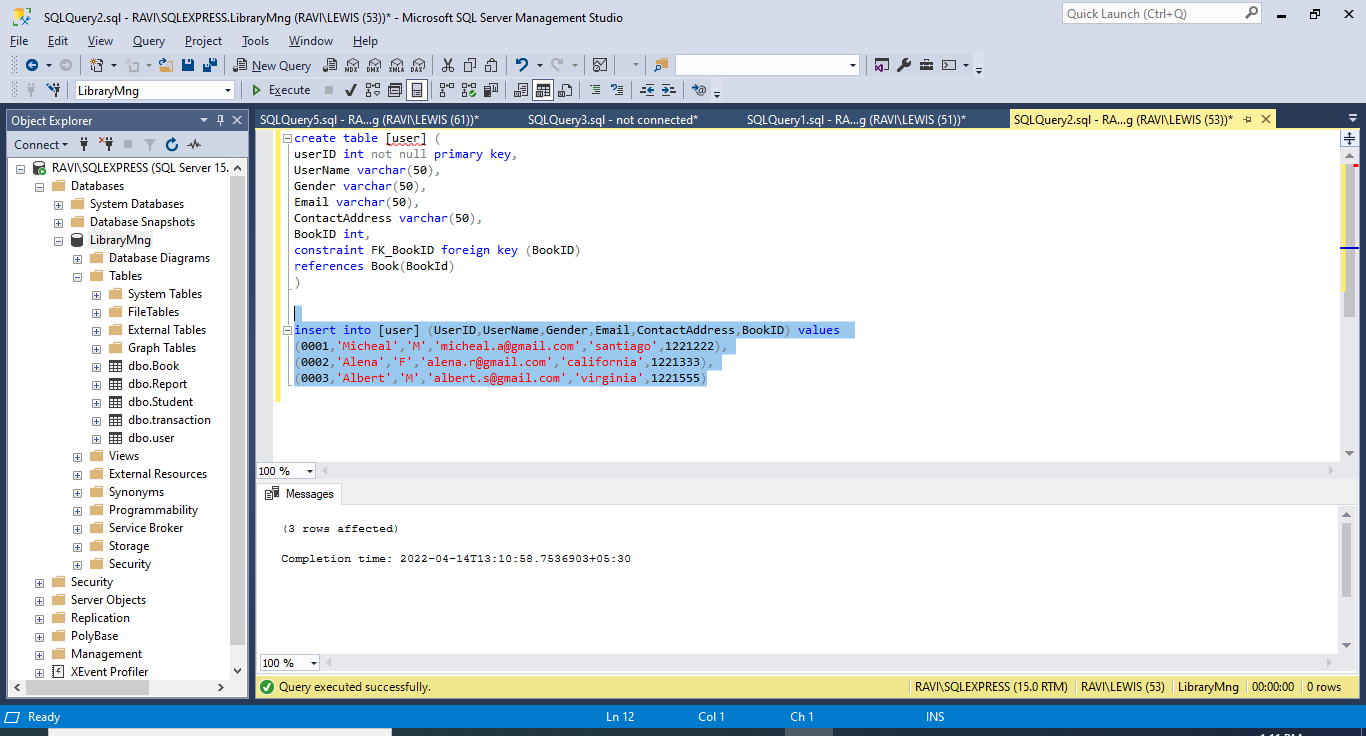


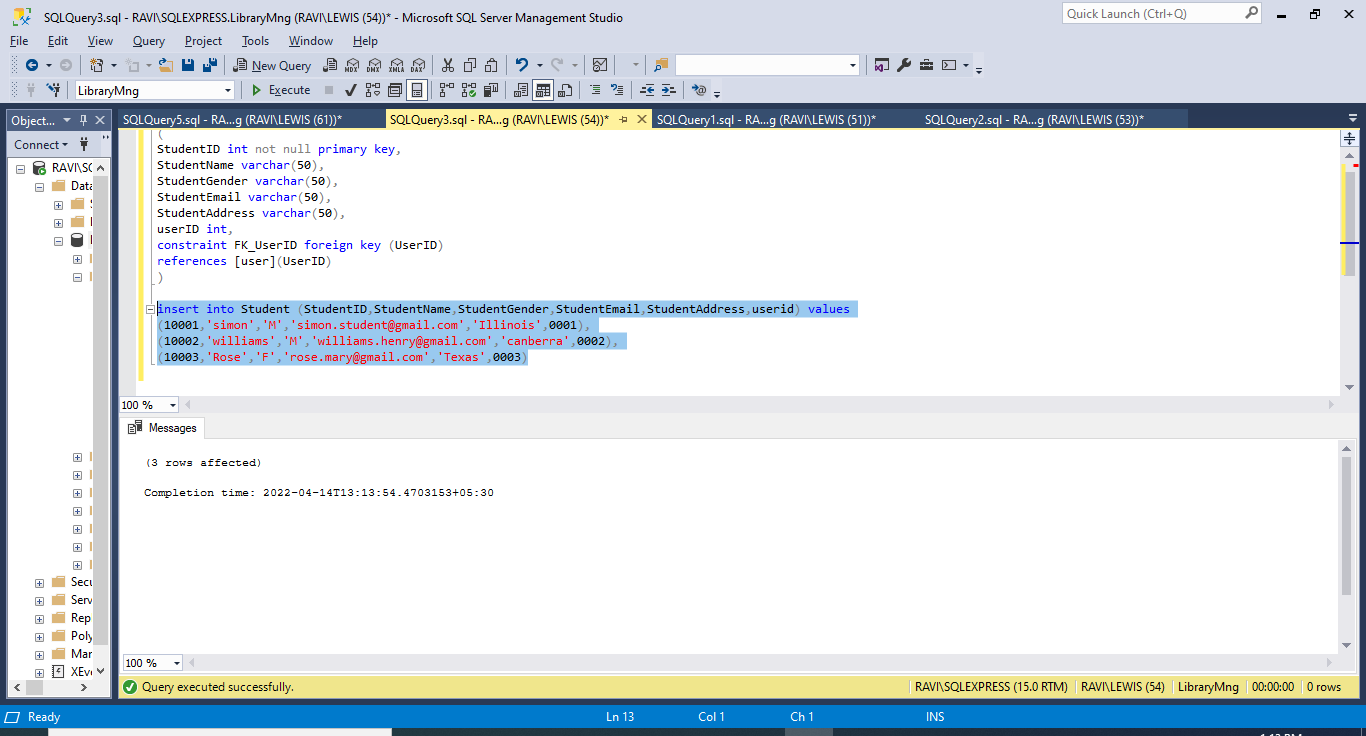


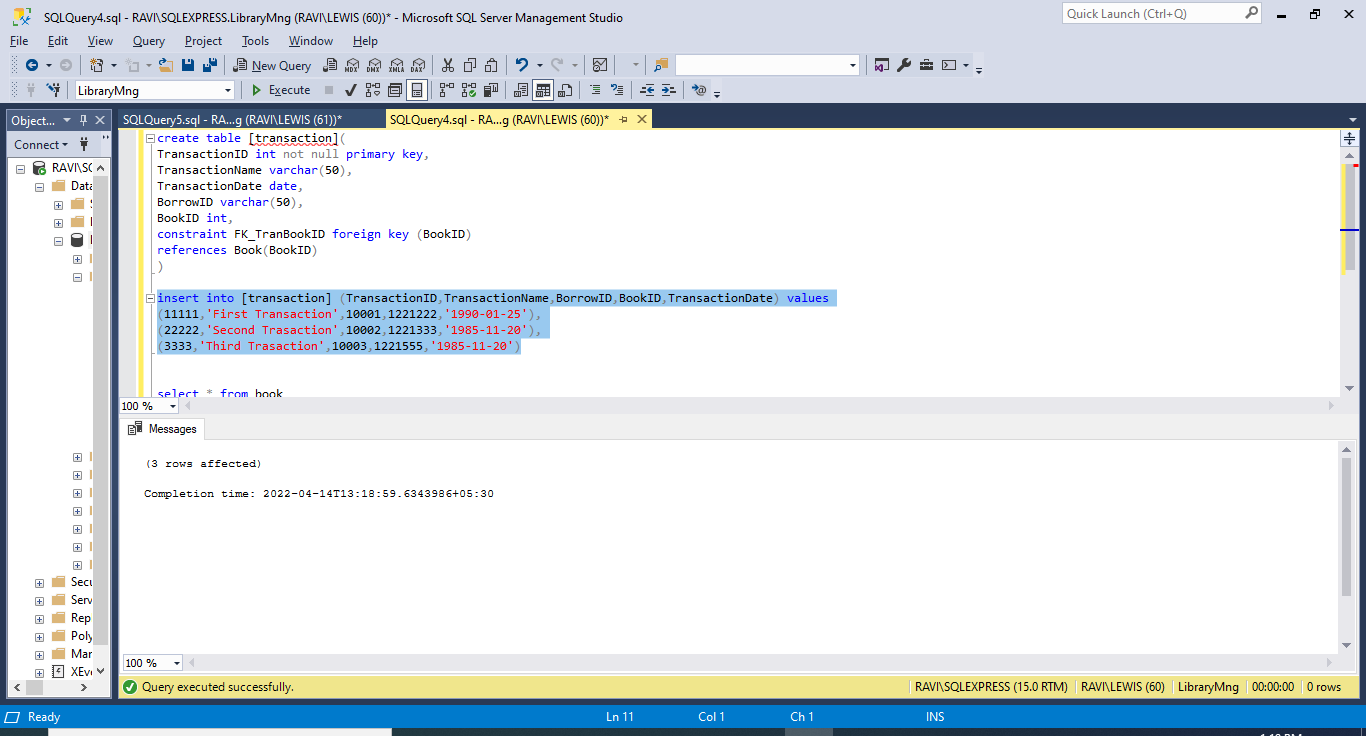


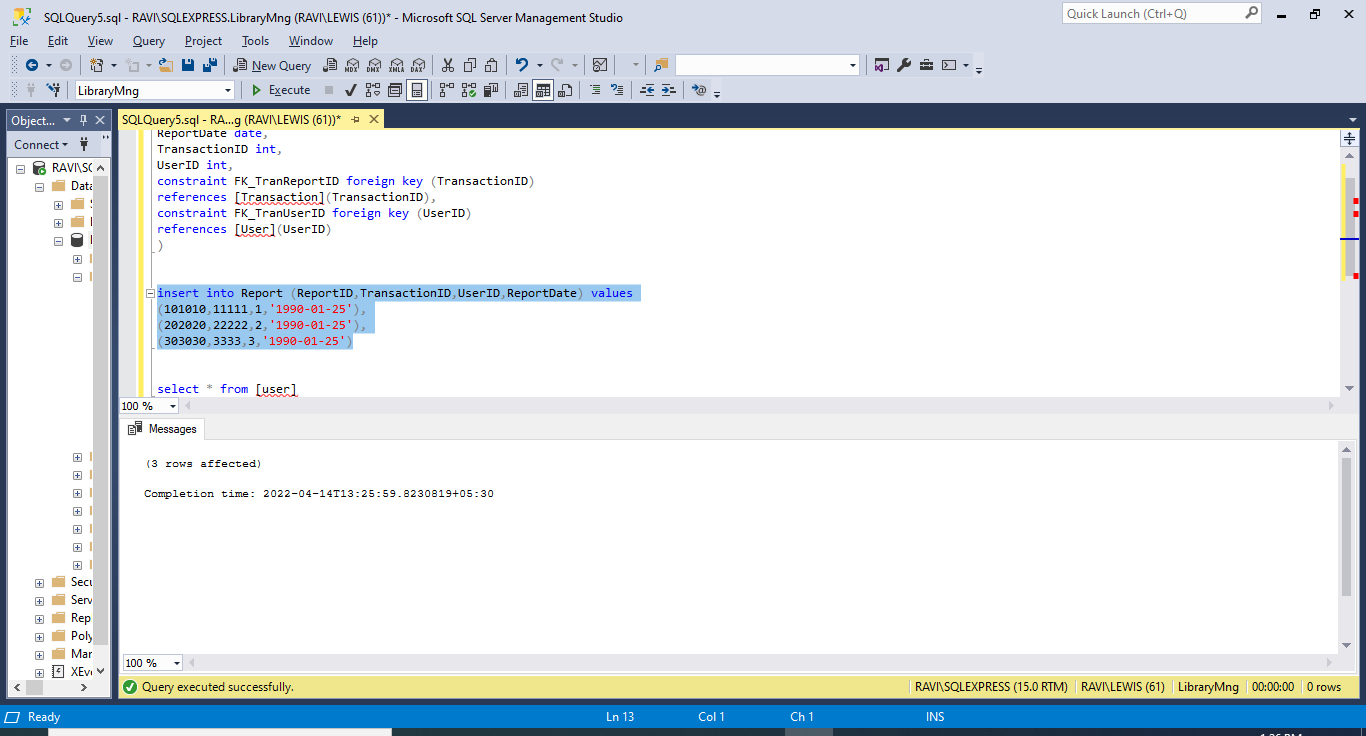
**Inserting values from .csv files to the appropriate tables:**

.

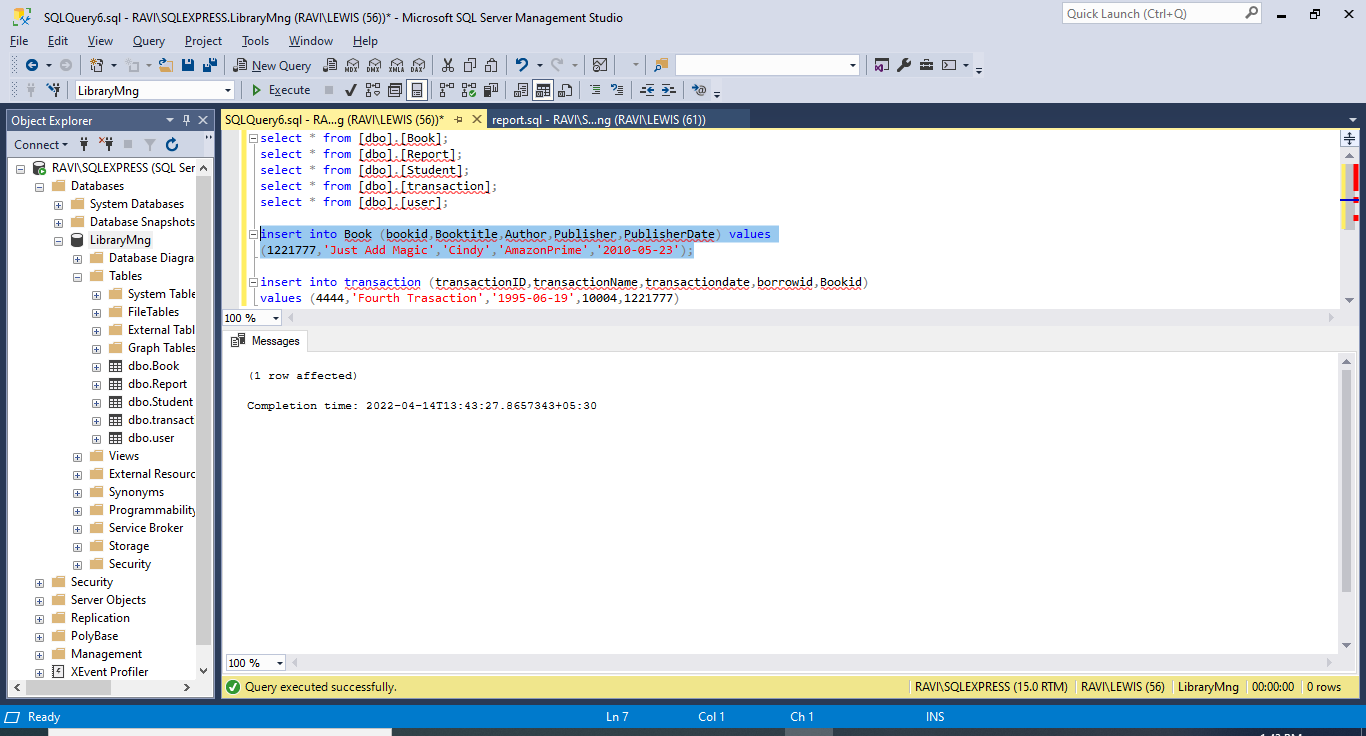


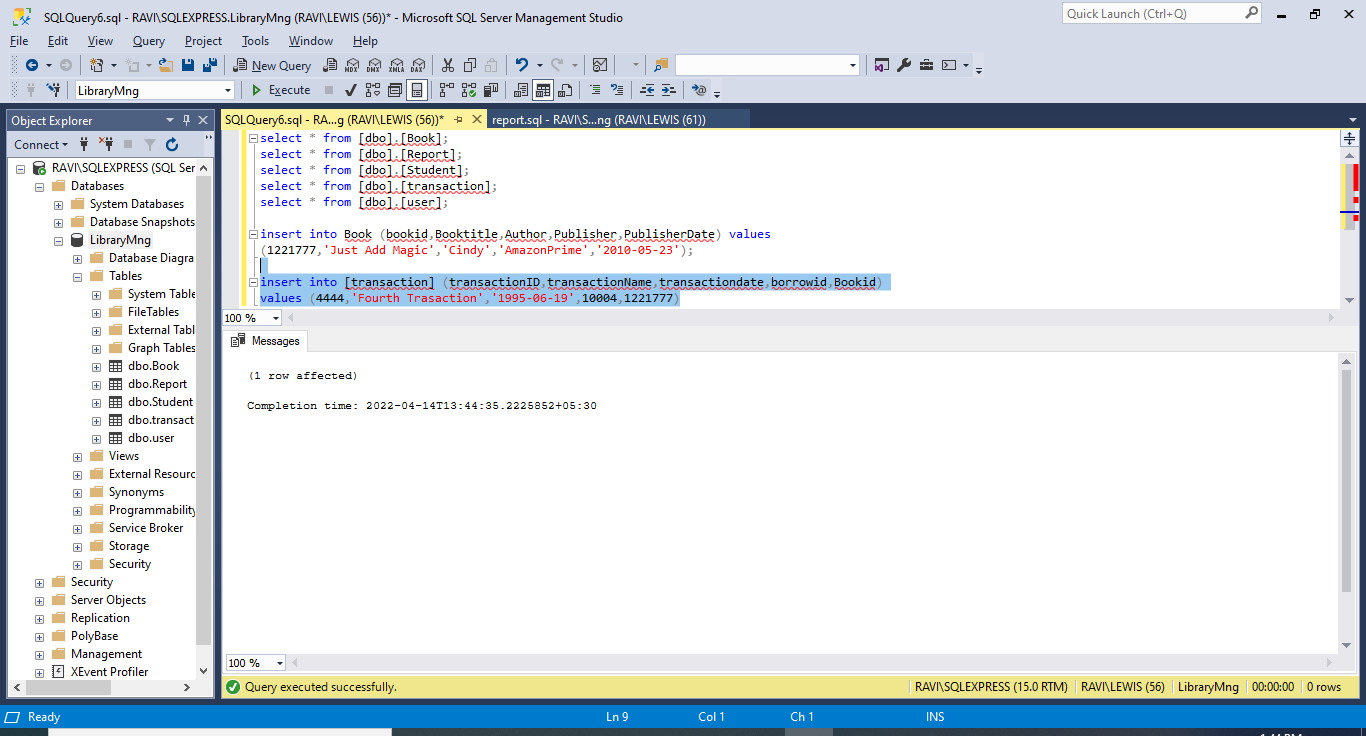


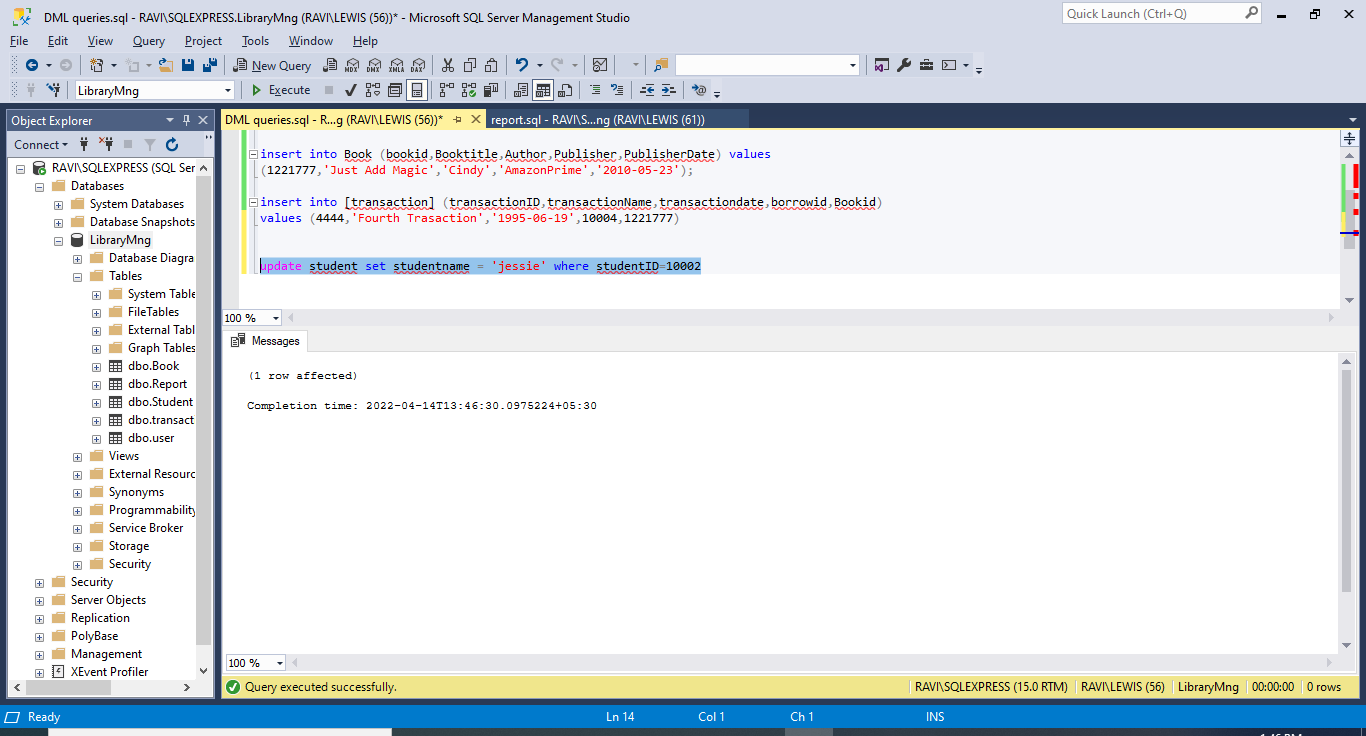


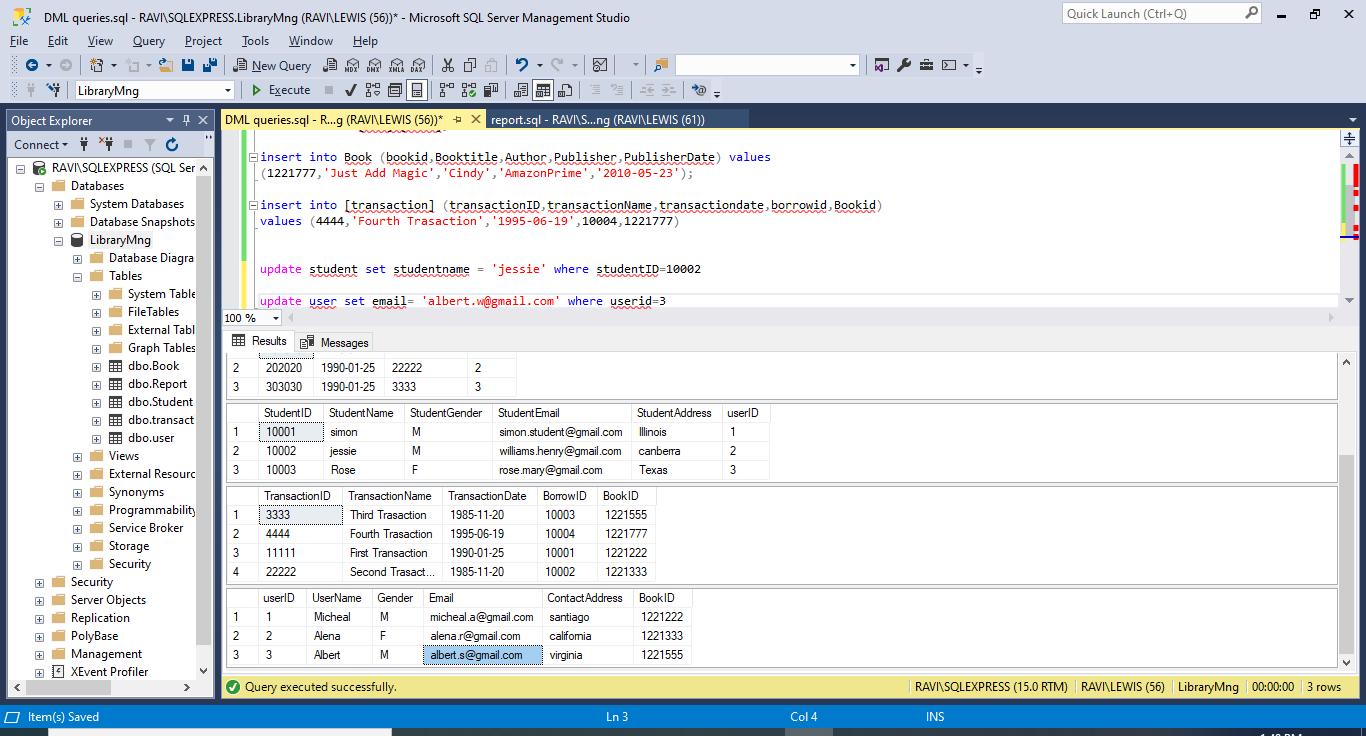


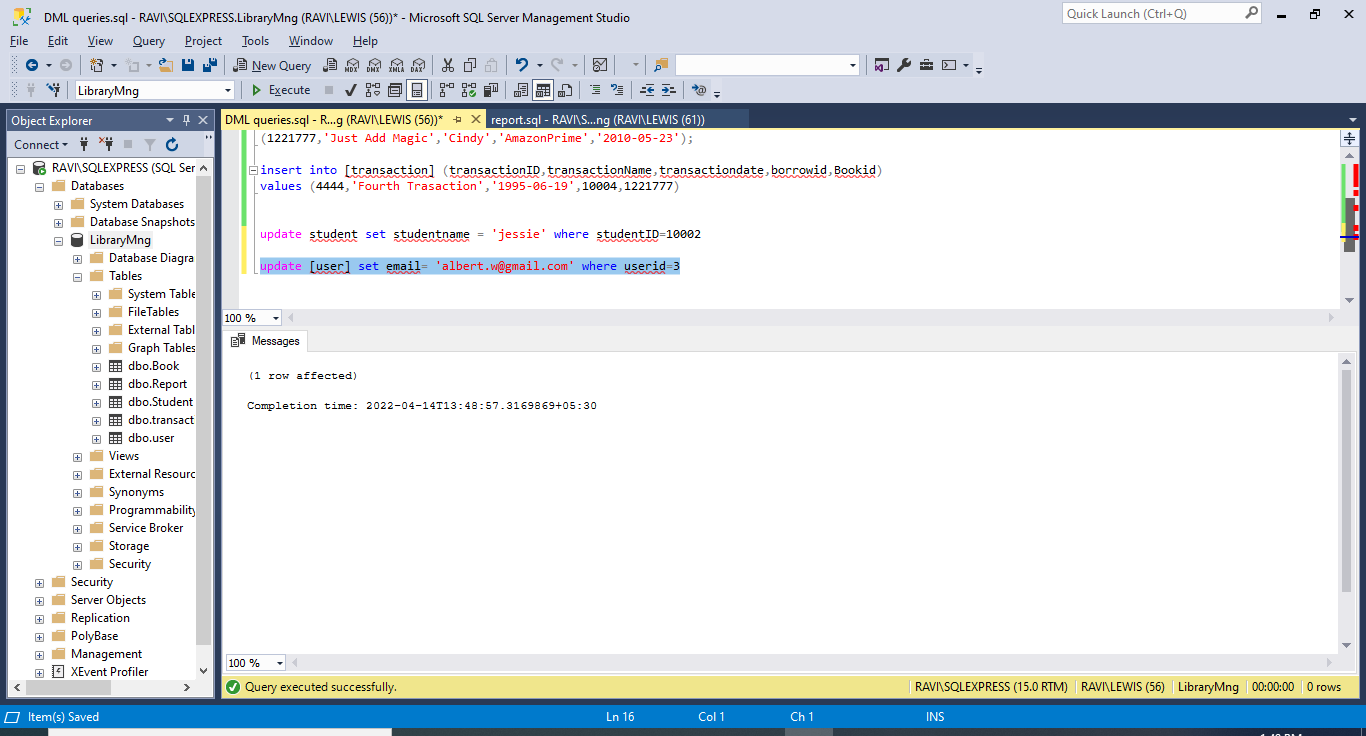
**Performing DML statements as mentioned and performed testing:**

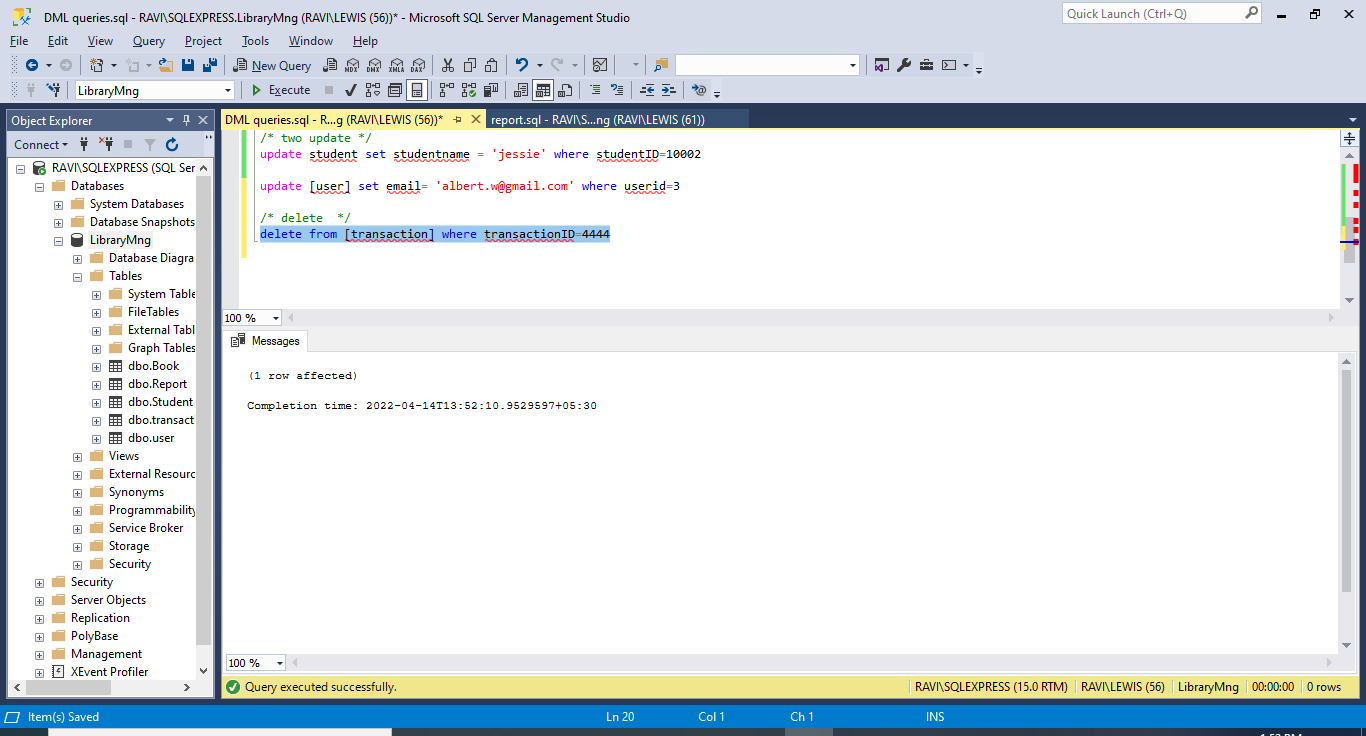


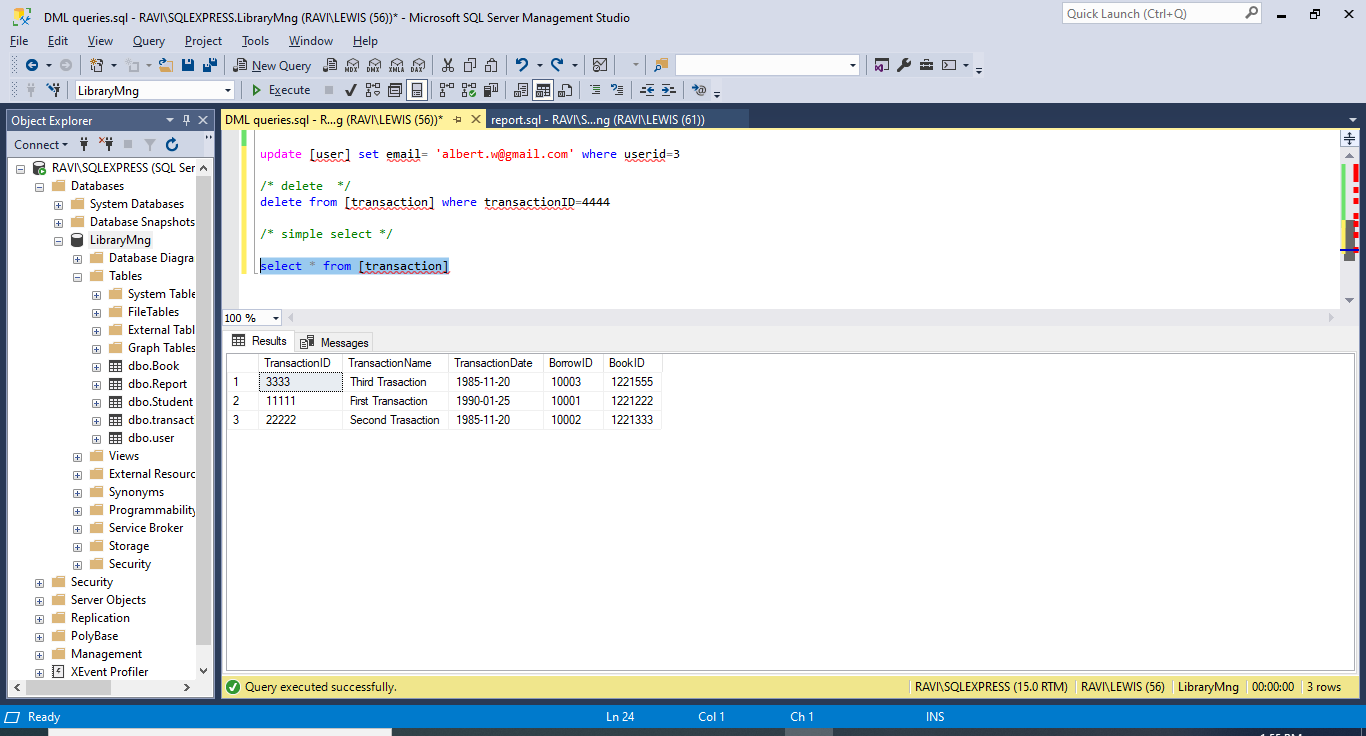


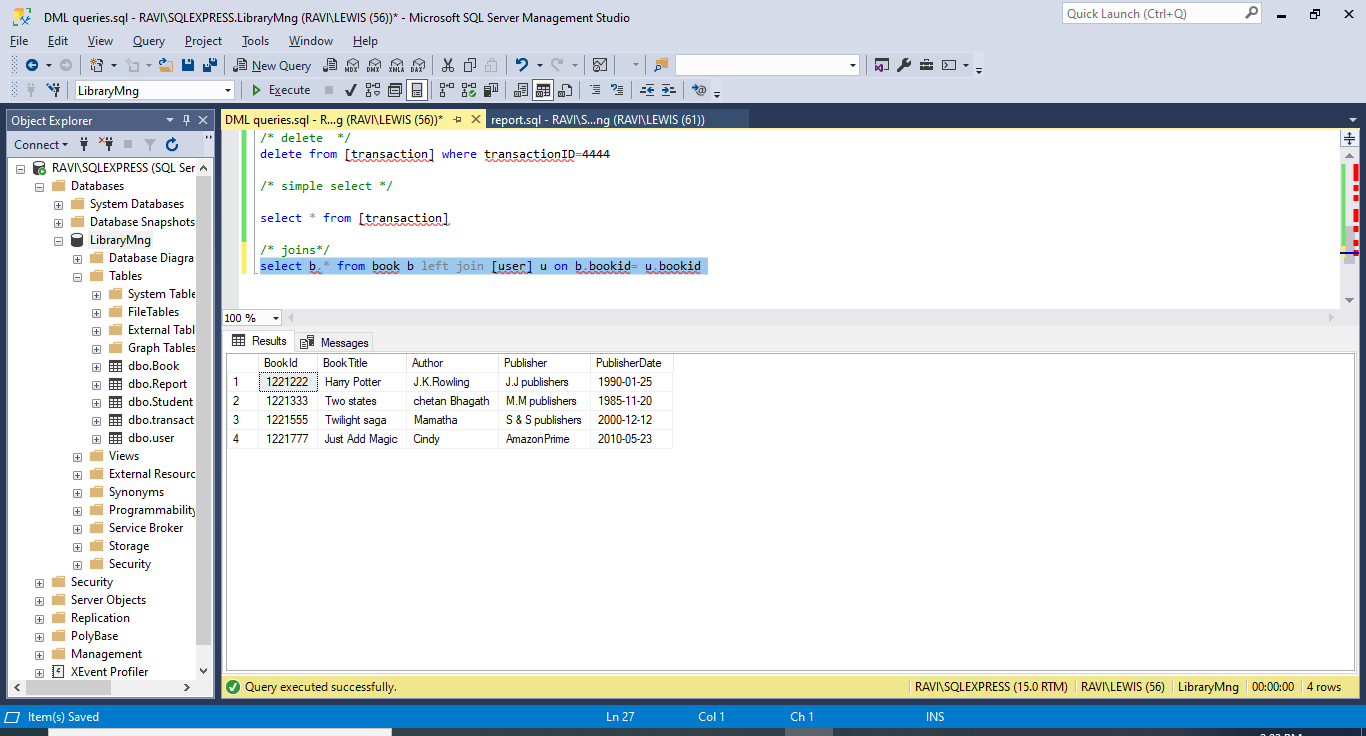


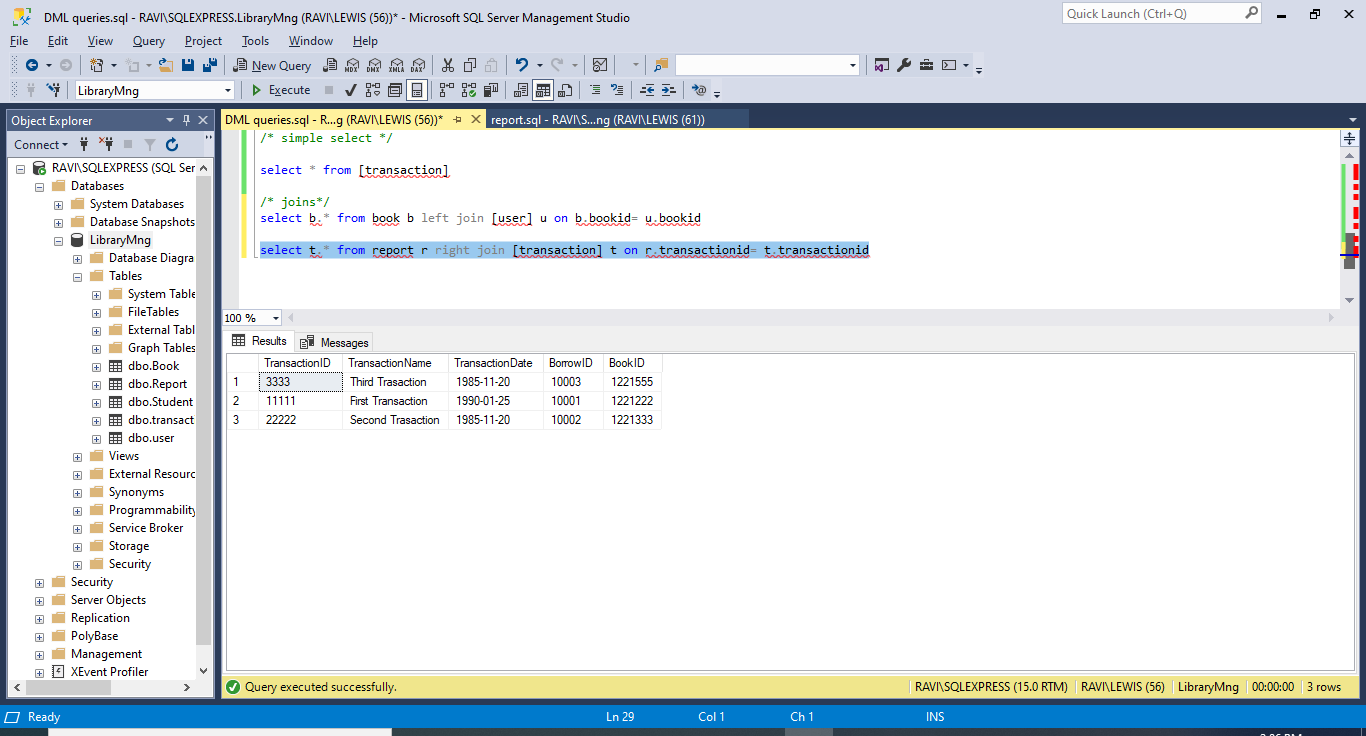


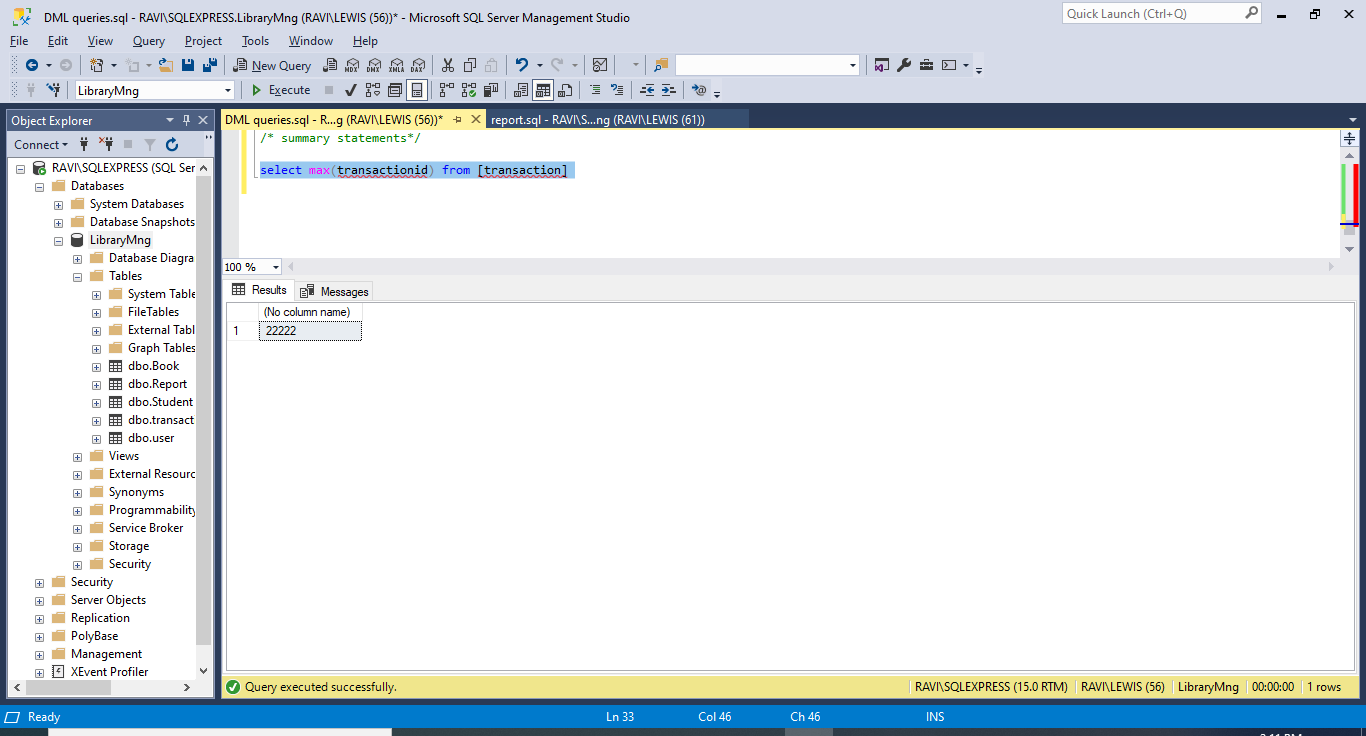


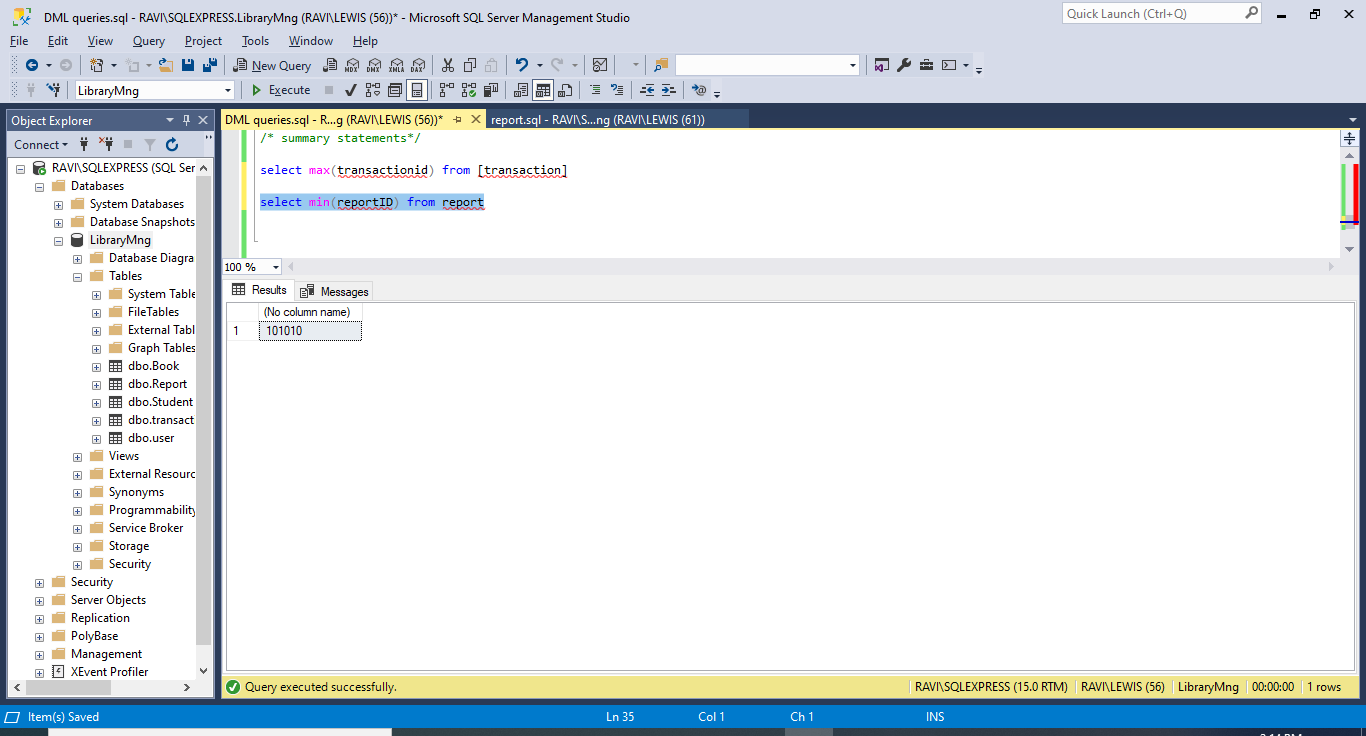


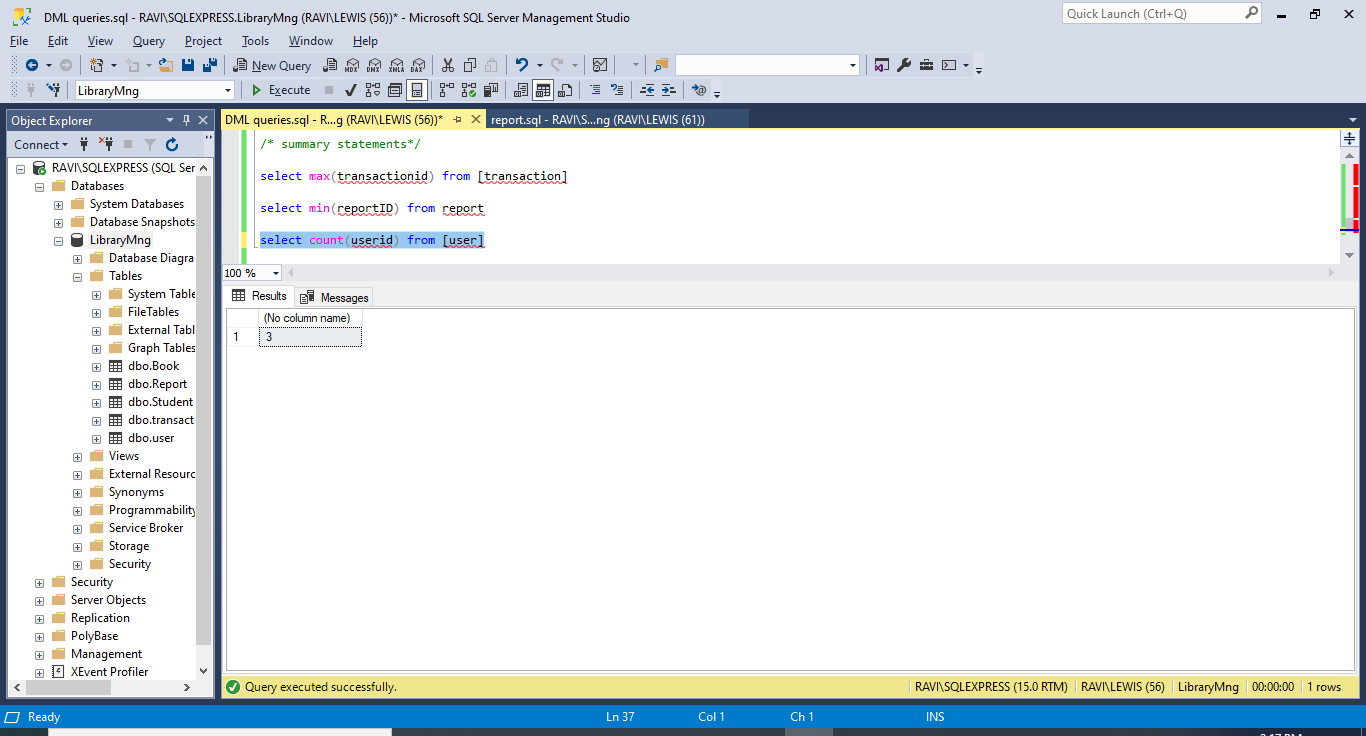


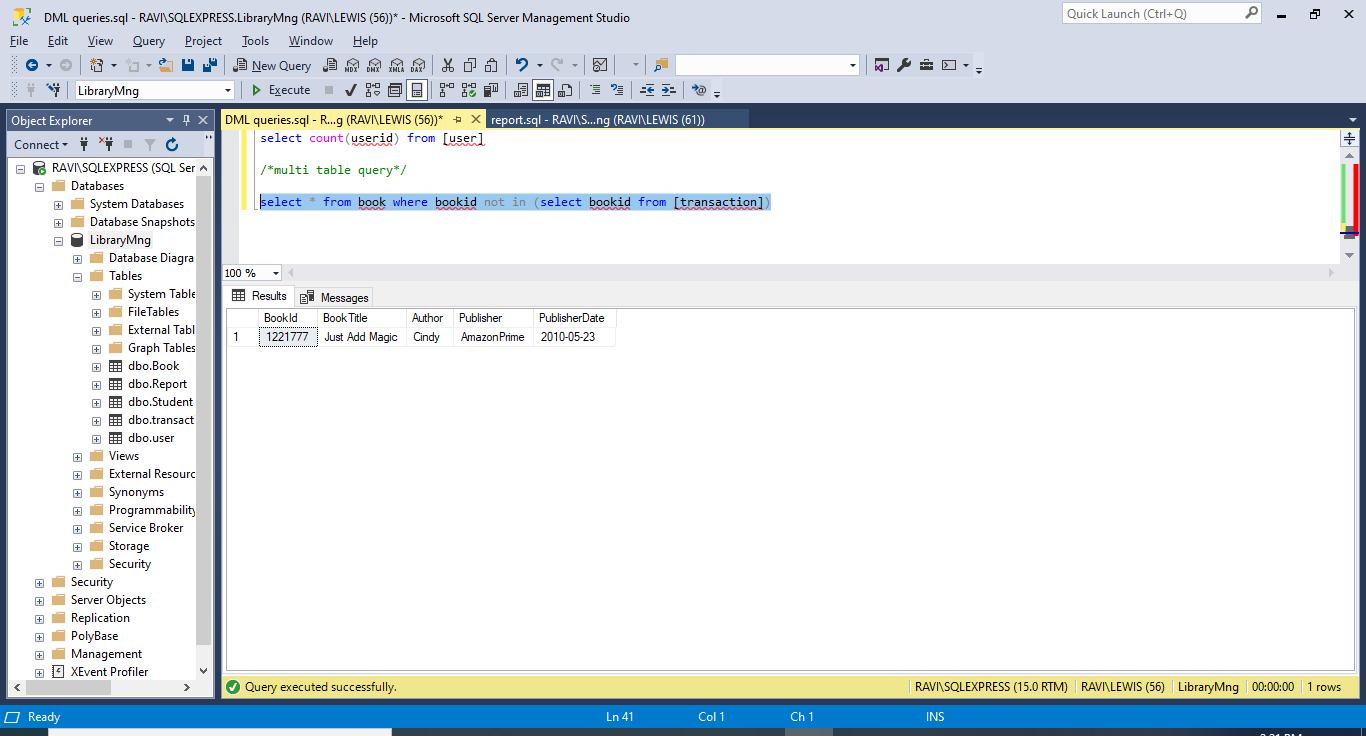












Project Part 5:

Indexes:

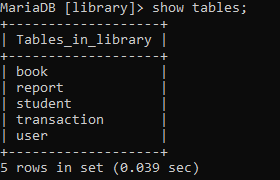
Before going to indexes we need to identify what are slow queries. MySQL slow query log is a file that records all long-running queries submitted to the database. To check if it's enabled and location show variables like "%slow\_query%";

We can use EXPLAIN to get additional information on slow queries. Based on the information we can create and drop Indexes based on result of slow queries.

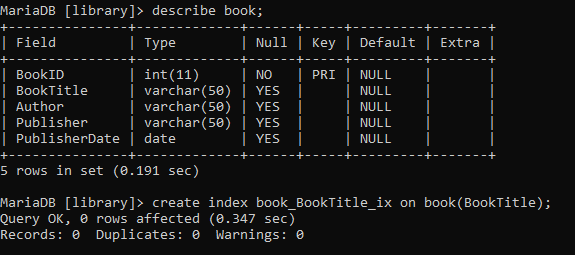
An index provides a way for a DBMS to locate information more quickly.

The below is the information of tables in my “Library”database.

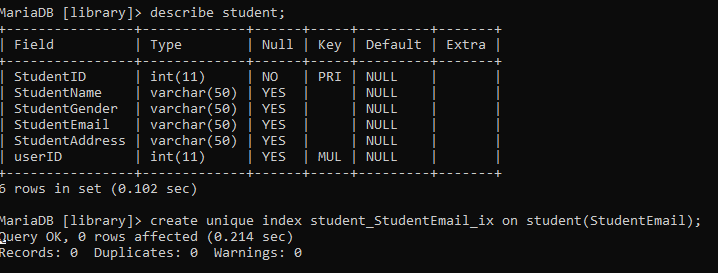




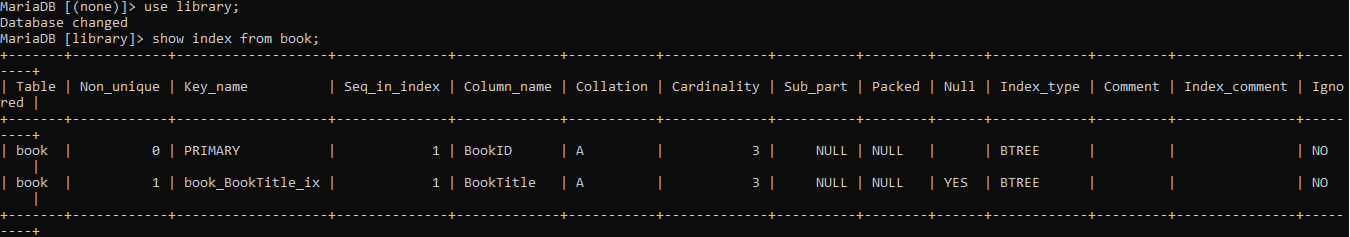
I would like to create an index on the “Book” table using column “BookTitle”. I am choosing this column because it contains a large number of distinct values.

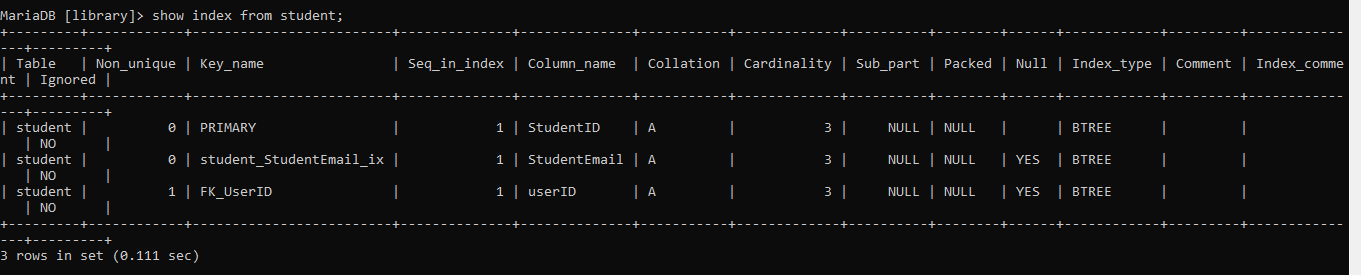


Later I am creating an index on “student” table, “StudentEmail” column. I a creating index on this specific column because this column is updated infrequently .



Displaying index:



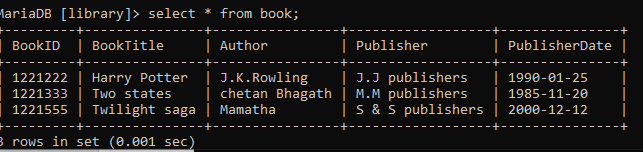


Database monitoring tool like Windows Performance Monitor (perfmon) can be used to check the performance improvement.

Views:

A view is a SELECT statement that's stored in the database as an object. Here I am creating two views.

One from Book table.



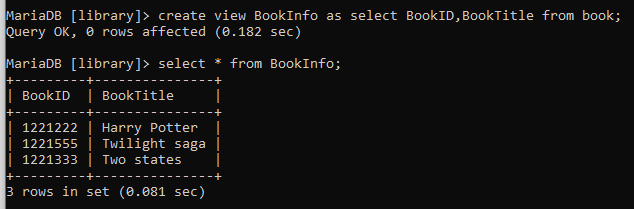
I am creating a vertical view by selecting only regularly used columns “BookID”,”BookTitle” from the table “Book”.

VIEW:

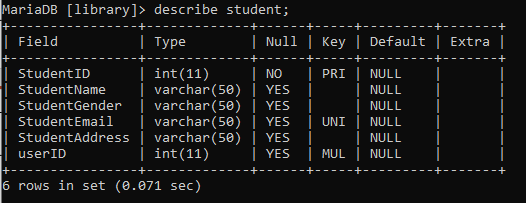
Create view BookInfo as select BookID,BookTitle from book;

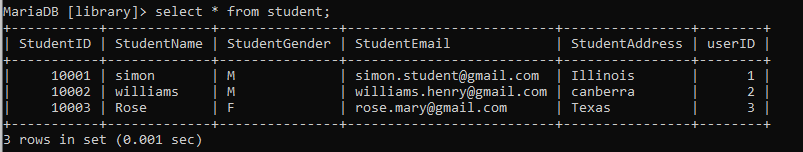
Displaying the view:

Select \* from BookInfo;



I am creating the second view on “student” table.





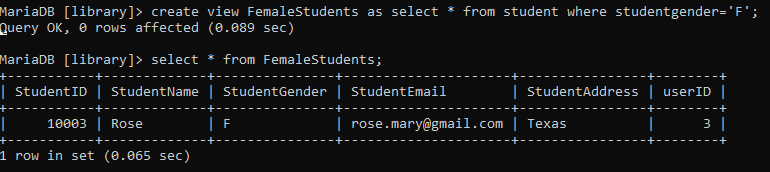
I want to separate “Female”,”Male” students. So, a horizontal view is created from “student” table where gender=”F”.

View:

Create view FemaleStudents as select \* from student where studentgender=’F’;

Also displayed the view.

Select \* from FemaleStudents;



The views here are used to “Reduce the Complexity”. The View on ‘Book’ table is created to get simple info like ‘BookID’, ‘BookTitle’. So that the visitors to the library can grab a book easily by viewing this data. The view on the ‘Student’ table is used to get information on the “Female” students regularly visiting the library.