University Library Database Management System

Part 3

\_\_\_\_\_\_\_\_\_

Yadav, Dilip - 1002023632

Makaraju, Chinmayee - 1002091569



Database Systems

Fall 2023

Supervisor: Ranjan Dash

The University of Texas at Arlington

**Nov 19, 2023**

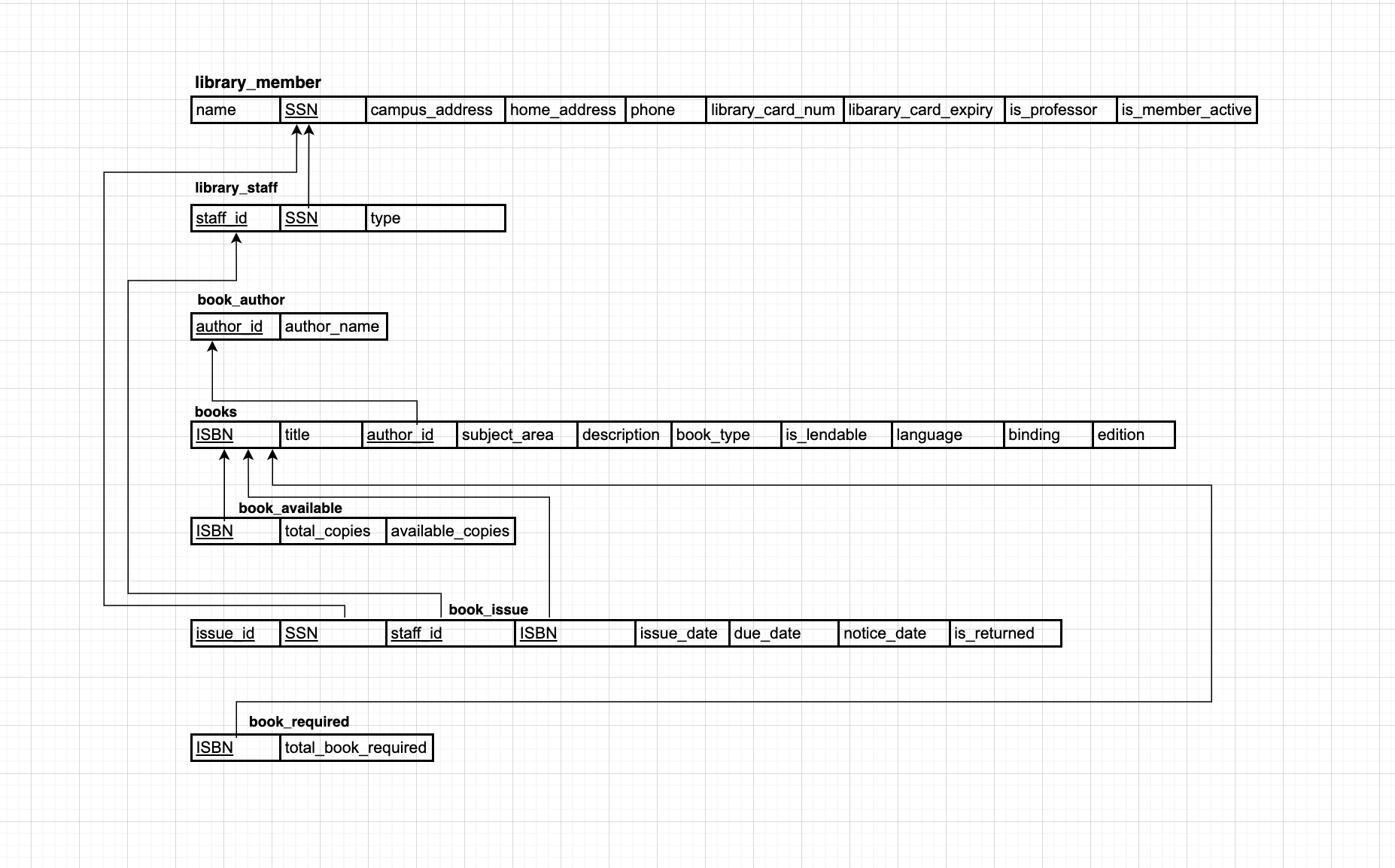
Table of Contents

[Schema Diagram 3](#_heading=h.v4nz8qys4nkp)

[Approach 4](#_heading=h.tp5q1givv0qq)

[Project Specifications: 5](#_heading=h.vb1f1byykf65)

## **Schema Diagram**



## **Approach**

1. Loading Initial Data

* Approach: Used a Python script to load initial data into the database tables.
* Method: Utilized CSV files for data storage and wrote Python code for data insertion.
* Implementation: The script uses the cx\_Oracle library to connect to the Oracle database. The data insertion process is designed to easily reload data during debugging.

2. Data Retrieval Queries

* Approach: Executed raw SQL queries to retrieve and print all entered data.
* Method: Used SQL/Plus for querying the database.
* Output Format: Organized data output with appropriate headings (e.g., Authors, Book Title, Subject area) for readability.
* Implementation: Created and ran SQL queries to fetch data, and spooled the results to files for easy review.

3. Weekly Borrowing Activity Report

* Objective: Prepare a report for weekly Borrowing activity by Subject area, Author, number of copies, and days loaned out.
* Approach: Wrote a raw SQL query to gather the required information.
* Output Format: Designed the report to include relevant details for each borrowing activity.
* Implementation: Executed the SQL query, spooled the output to a file, and reviewed the data.

4. Database Update Transactions

* Transactions:
  + Add information about a new member.
  + Add all information about a new Book.
  + Add all information about a new Borrow (loan).
  + Handle the return of a book and print a return receipt.
  + Renew the membership.
* Programming Language: Utilized Python for implementing the database update transactions.
* Implementation: Incorporated queries for each transaction within a Python script, interacting with the Oracle database.

5. Triggers

* Triggers:
  + Notify a member about the outstanding overdue book.
  + Notify a member about his membership renewal.
* Approach: Implemented raw trigger queries in SQL.
* Execution: Ran the trigger queries on Oracle's SQL Plus for the desired database actions.

6. User-Friendly Interface

* Interface Type: Employed a Python script to provide a command-line interface for user interaction.
* Method: Designed methods within the Python script to receive input and perform necessary actions for each transaction.

## 

## 

## **Project Specifications:**

Hardware Specifications

* Device: Macbook
* Processor: Intel Core i5, 2 GHz Quad-Core
* RAM: 16 GB

Software Specifications

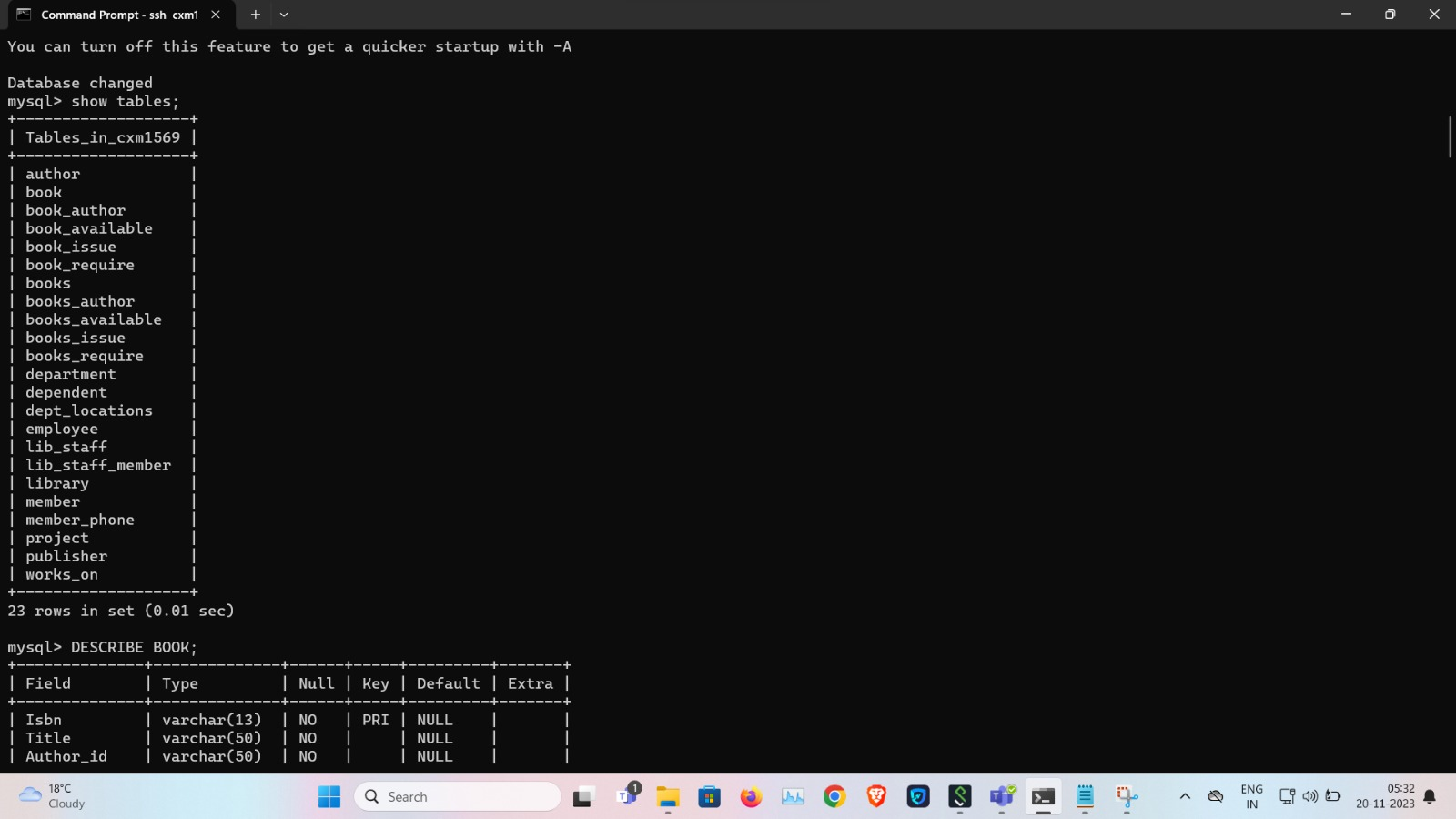
* Programming Language: Python
* Database Management System: Oracle SQLPlus

**Part – 3:**

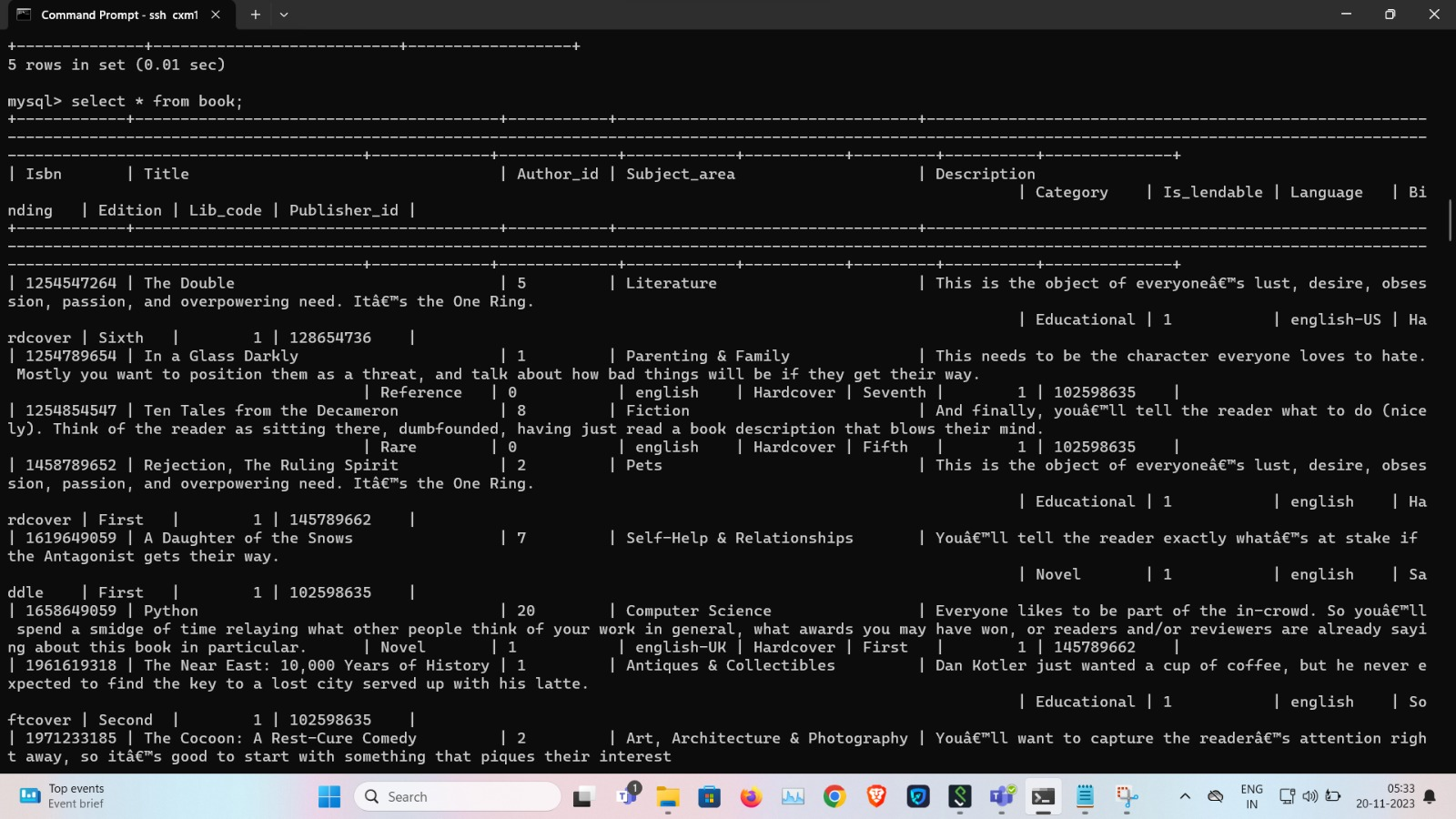
Loading of data, displaying the tables and data from the library database, and performing Update Transactions and Retrieval queries:

**Q1: Loading of data from .csv file to MYSQL Tables using Python Script:**

* In the "DB Project\_2" directory, there exists a subfolder named "CSE\_5330\_005\_databaseSystems\_Project2\_Part3." Within this subfolder, you'll find a Python script titled "Records\_Insertion\_Library\_Management.py." This script is designed for loading data from a .csv file into MYSQL tables that were created in Parts 1 and 2 of Project 2.
* Additionally, there's another subfolder named "Data" containing project data, specifically 12 .csv files. These files correspond to the 12 tables created in the initial parts of Project 2, which focus on Library Management.
* To run the script successfully, you'll need to have the "mysql-connector-python-8.0.30-windows-x86-64bit" package installed, and Python version 3.7 is required. The script establishes a connection, and data is inserted into the tables.
* After the insertion of records, snapshots are taken before and after the data insertion process. These snapshots are then spooled and stored in a folder named "CSE\_5330\_005\_Project 2\_Part 3\_Spool File – Output."

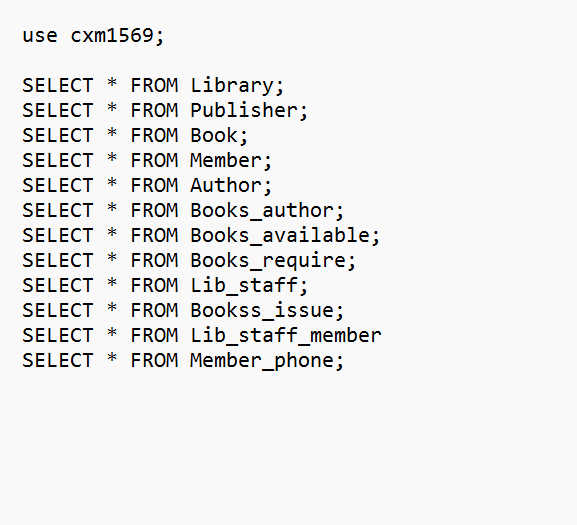


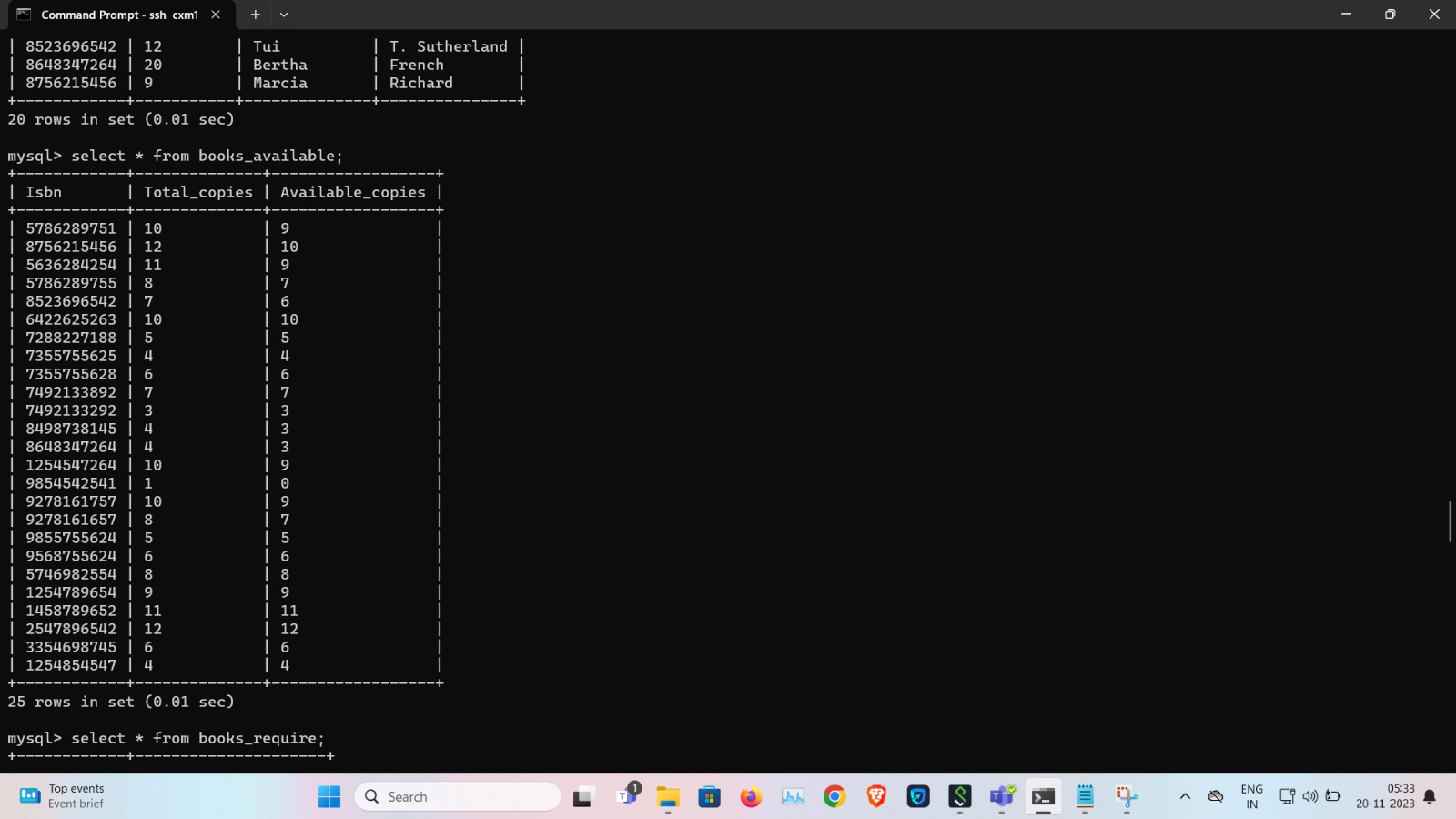
* The script utilizes file spooling to capture the output, generating two text files named "OUTPUT\_Tables\_before\_loading\_data.lst" and "OUTPUT\_Tables\_after\_inserting\_data.lst." These files serve the purpose of distinguishing the state of the tables before and after the insertion of records through the Python script.
* To facilitate the transfer of both text files, FileZilla is employed to move them from the Omega server. The destination folder for these files is specified as the 'CSE\_5330\_005\_Project 2\_Part 3\_Spool File – Output' subfolder.
* A snapshot retrieved from the Omega server showcases the records that have been successfully inserted.



**Q2: Queries to retrieve and print the data from the MYSQL database for Library management:**

Within the directory labeled 'CSE\_5330\_005\_Project\_2\_Part\_3\_Queries,' there is a file named 'Problem\_2.txt.' This file contains queries crafted to retrieve and display the data that has been loaded, as demonstrated in the previous description.



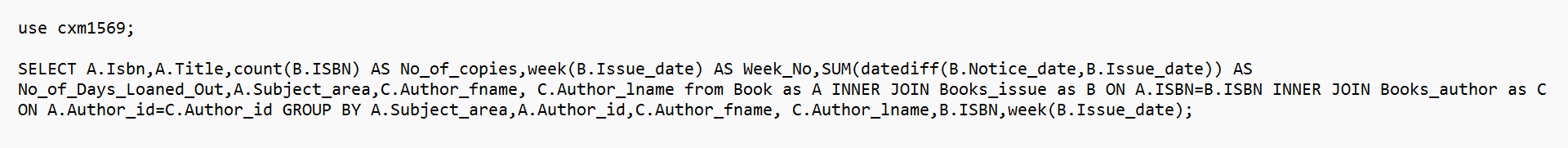


The results of executing the queries from the 'Problem\_2.txt' file are documented through file spooling, generating a text file named 'OUTPUT2\_Part3.lst.' To facilitate the transfer of this text file, FileZilla is employed, and it is moved from the Omega server to the 'CSE\_5330\_005\_Project 2\_Part 3\_Spool File – Output' subfolder.

**Q3: Query to retrieve and print a report for weekly Borrowing activity by Subject area, by Author, number of copies, and number of days loaned out:**

In the directory labeled 'CSE\_5330\_005\_Project\_2\_Part\_3\_Queries,' there exists a file named 'Problem\_3.txt.' This file contains queries designed to retrieve and display the loaded data, as detailed earlier.

Furthermore, the specified query is employed within a Java program to generate a weekly borrowing activity report. This functionality is integrated into the Java program and is triggered upon user selection to generate the report. A detailed discussion about this aspect of the program in Java is provided in the subsequent sections of the document.



The results of executing the queries from the 'Problem\_3.txt' file are documented through file spooling, creating a text file named 'OUTPUT3\_Part3.lst.' To facilitate the transfer of this text file, FileZilla is employed, and it is moved from the Omega server to the 'CSE\_5330\_005\_Project 2\_Part 3\_Spool File – Output' subfolder.

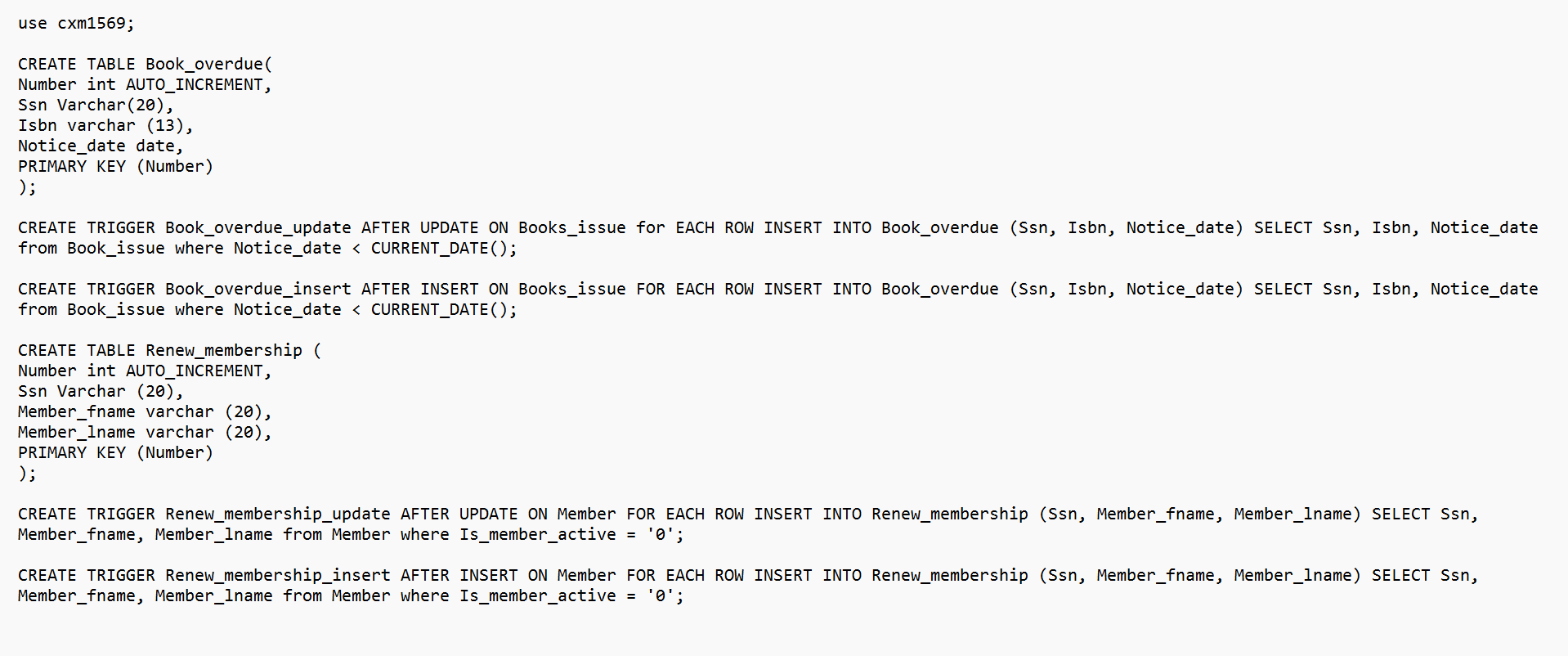
**Q4: Transactions to be performed on the Library database - Adding New member/book, Borrow/return book:**

The programming language utilized is Java. To ensure optimal user readability and ease of navigating the final output, it is recommended to use IntelliJ IDE.

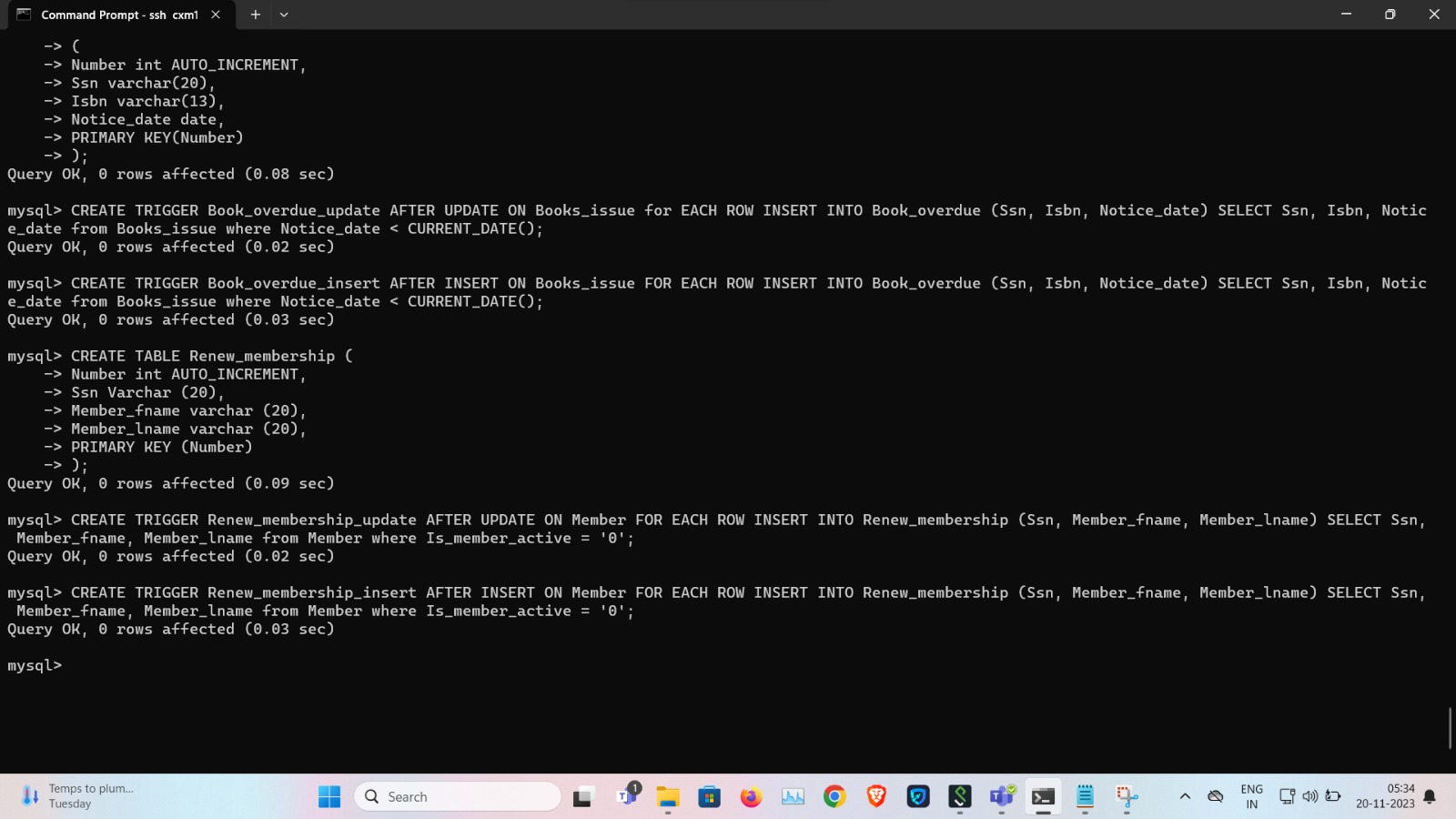
Certain prerequisites need to be met for the project:

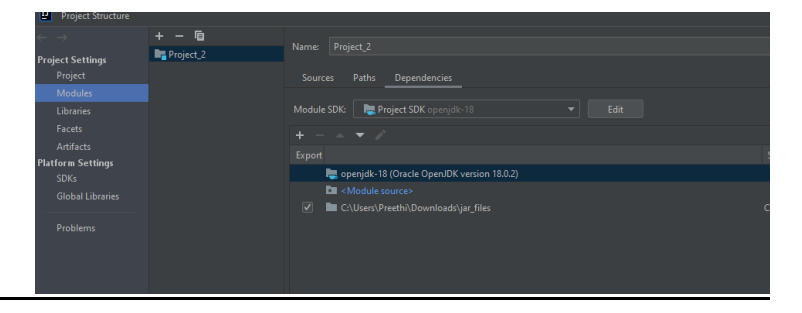
1. IntelliJ IDE is required for enhanced readability and efficient lookup of the final output.
2. The project relies on the "mysql-connector-java-5.1.48" library.
3. JDK for Java is necessary, and any latest version or specifically version 11 is suitable.

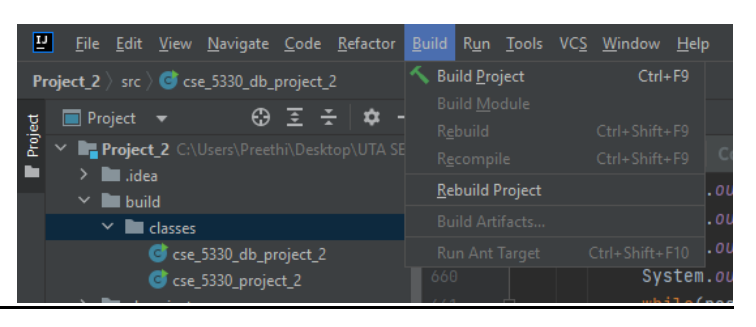
It is advised to configure the JDK and the JAR file as modules under the Project Structure for proper setup.

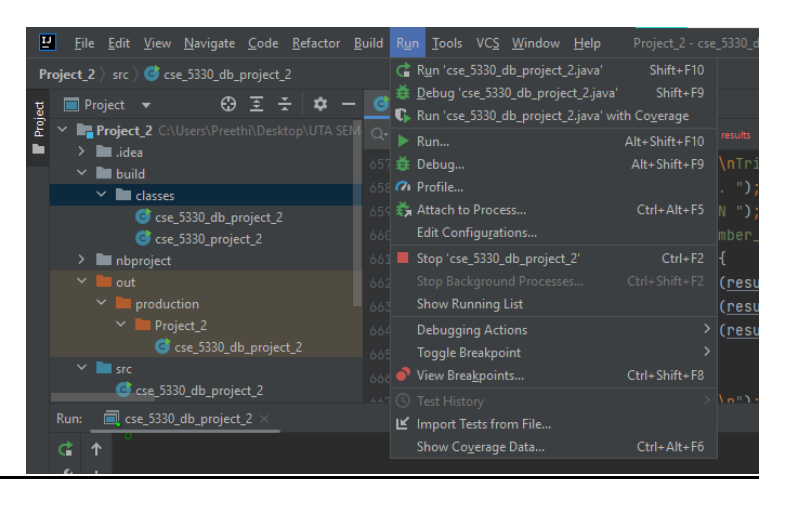


After completing the initial setup, proceed to build the project and execute the file "cse\_5330\_db\_project2.java," following the steps illustrated in the snapshots below.









The project encompasses six main operations or executions:

a) Utilizing the database

b) Printing and presenting tables from the library database

c) Generating a weekly borrowing activity report

d) Executing new transactions on the library database, involving the following sub-operations:

* Adding new members to the database.
* Adding new books to the database.
* Borrowing a book from the database.
* Returning a book to the database.
* Renewing the membership of library members.

e) Triggering the execution of actions, including:

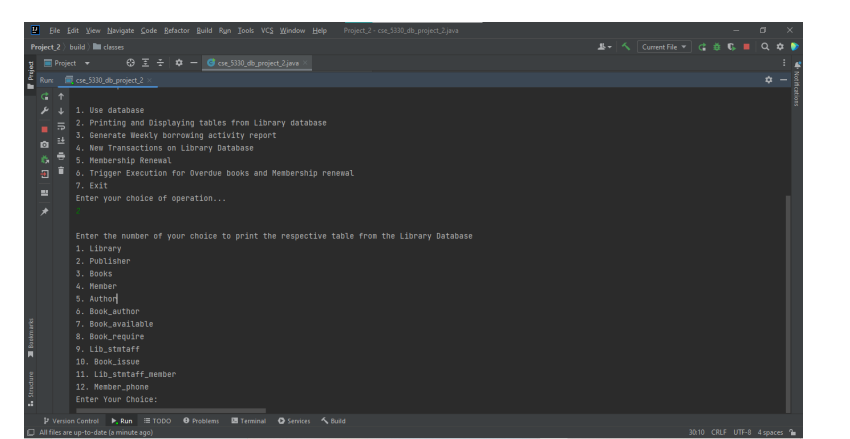
* Notifying about outstanding overdue for borrowed books.
* Notifying members about membership renewal.

a) Database Utilization:

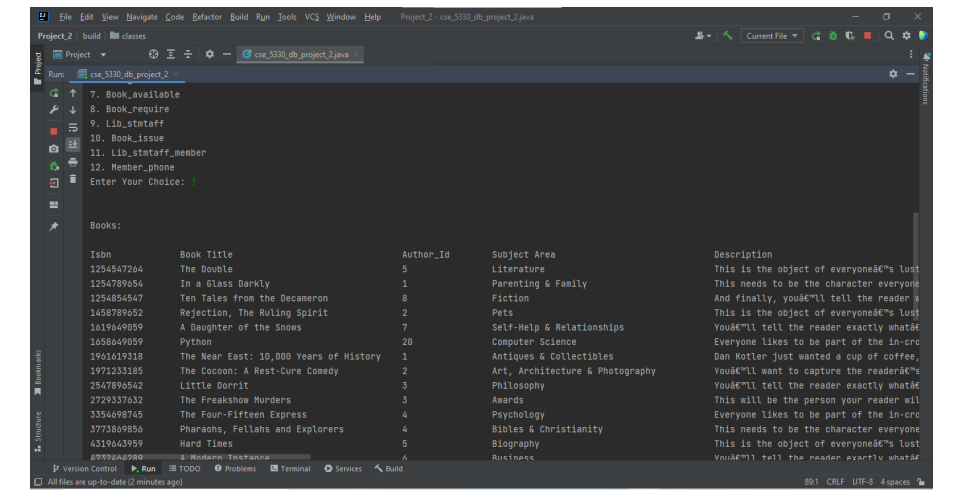
In this phase, the command-line interface is employed to obtain user input, and actions are executed according to the choices provided, as illustrated in the snapshot below.

**b) Part 3 - Problem 2) - To print and display the data within the created tables for Library management:**

The tables, which include the 12 tables created in Part 1 and Part 2 of Project 2, are presented. The command-line interface is then utilized to gather user input, and actions are carried out according to the choices available, as demonstrated in the snapshot below.



Subsequently, the tables are enumerated along with their respective data, as depicted in the provided snapshot. For example, opting for option 3 from the list signifies a request for data from the Books table. The ensuing snapshot illustrates the data contained within the Books table.



**c) Part 3 - Problem 3) - Generate a report for weekly Borrowing activity by Subject area, Author, number of copies, and number of days loaned out:**

For example, if the user chooses option 3 from the presented functions for Library Management, option 3 is specifically designed to produce a report detailing the weekly borrowing activity by subject area, author, the number of copies, and the duration the items are loaned out. The following snapshot showcases the generated weekly borrowing activity report for the library.

**d) Part 3 - Problem 4) - Transactions to be performed on the Library database - Adding New member/book, Borrow/return book:**

**4.1 To add a new member to the library database:**

The transactions are executed as follows:

* The user selects option 4 to initiate transactions on the library database, as captured in the provided snapshot.
* Within this section, the user inputs option 1 under transaction operations, signifying the addition of a new member to the library database, as depicted in the accompanying snapshot.
* The program utilizes the command-line interface to obtain user input and successfully adds the new member to the database, as illustrated in the subsequent snapshot.

**4.2 To add a new book to the library database:**

In this scenario, the user provides input for option 2 under transaction operations, indicating the addition of a new book to the library database, as depicted in the provided snapshot. Subsequently, the program utilizes the command-line interface to gather user input and successfully adds the specified book to the database, as illustrated in the following snapshot.

**4.3 To borrow a boom from the library:**

In this instance, the user inputs option 3 under transaction operations, indicating the intention to borrow a book from the library database, as presented in the provided snapshot. The program then employs the command-line interface to receive user input, and it verifies the availability of the requested book. If the book is unavailable, a message is displayed indicating that the book cannot be borrowed by the member.

**4.4 To return a boom from the library:**

In this scenario, the user selects option 4 under transaction operations, expressing the intent to return a book to the library database, as indicated in the provided snapshot. The program then utilizes the command-line interface to collect user input, facilitating the successful return of the book to the database, as demonstrated in the subsequent snapshot. During the book return process, a receipt containing the return details is generated for the user.

**4.5 Renew Membership:**

The user selects option 5 to initiate the membership renewal operation on the library database, as depicted in the provided snapshot. Within this operation, the user inputs the Social Security Number (SSN), a unique identifier for library members. If the system identifies the member as an active participant in the library, the membership is successfully renewed for the next six months, as illustrated in the accompanying snapshot.

**Q5: Triggers execution to notify members of the library on overdue of books and membership renewal:**

The user enters option 6 to activate the notification process for members with overdue books and membership renewal in the library database, as seen in the provided snapshot. Triggers have been established for these specific scenarios, and when insert or update operations occur in the Book\_issue table and renew\_membership table, the triggers are triggered. This, in turn, initiates the notification process, alerting members accordingly, as highlighted in the snapshot.

When a new member or book is added through the Transaction operation, and subsequently, when option 6 is chosen for Trigger execution, a report is generated to notify members about overdue books and those who need to renew their membership, provided they are active members of the library. The process and outcome are depicted in the following snapshot.