

G E O R G E M A S O N

U N I V E R S I T Y



2 0 2 4 S P R I N G C S 5 0 4

P R O J E C T

L I B R A R Y
M A N A G E M E N T
S Y S T E M

A M I T C H O W D A R Y K A M M A

G 0 1 4 7 4 8 2 2

Introduction

A Database Management System (DBMS) is a system software which is used for organizing, managing, and retrieving data in databases varying from small to large databases. Every data which is created is often stored in a database, making this a reason for everyone ranging from individuals to companies to use DBMS. In addition, DBMS also has features of storage, data manipulation, security, giving access to the right people in a database by View, and maintaining data integrity.

The making of library management system is a very important step as it holds so much of data regarding the books and authors it is very vital for it to show accurate and clear results so the user using the system understands well. Employees entering the data into the system must also understand the guidelines of the database well. By this the final product will leave a good impact on users and will give a good experience to the users.

Scope of the Project (Library Management System)

The primary objective in this project being Library Management System is to develop a trusted and user-friendly database platform that efficiently and precisely manages the fundamental activities of a library. The purpose of the system is to store all the book titles and the respective authors and their details, storing of the borrowing activity of members in the library with the help of staff in the library, organizing the records, allocating the books to specific genres, cataloguing library materials. This project report will display the Entity-Relationship Diagram, Relational Schema, Creation of tables, insertion of data into the tables, querying and manipulating the Library Management Database/System, and the use of extended features such as stored procedures and triggers.

Entities and Relationships

For the library management system, we have a total of 8 tables of data where each table of data represents different types of data. These tables are referred as ‘entities’ in the Entity Relationship Diagram. The eight entities are Author, Authorship, Borrow, Catalog, Genre, Material, Member, and Staff. Below are the brief descriptions of the entities and what data they hold.

Author - This entity has the details of the Author such as the name, author ID, birth date and nationality.

Authorship – Authorship entity contains data of the materials created by the author and stores the material ID of the books. With the combination of Author ID and Material ID a unique authorship ID is stored.

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

Borrow – The data of the materials which are borrowed by members is stored here and has data of all the dates related to the entity and the required IDs of various entities.

Catalog - represents a list of library resources along with details about their location and availability.

Genre – This holds data of the types of genres the library has and the respected ID of each Genre. Description is also given for each Genre.

Material – All the books/materials which are present in the library are stored here. Additionally, all the needed IDs are stored such as Borrow ID, Genre ID, and Catalog ID.

Member – All the users or customers of the library is stored in this table with the details of Name, Material ID, Join date and contact information.

Staff – This entity represents the working officials who manage the resources in the library, and come in middle for the borrowing activity of members. This entity has data such as Staff ID, Name, Contact information, hire date, and job title.

Software requirements for ER diagram and Schema

The software I used for drawing/building my ER diagram and Relational Schema is a web-based application called draw.io. It offered me a wide range of shapes specifically made for drawing Entity Relationship diagrams.

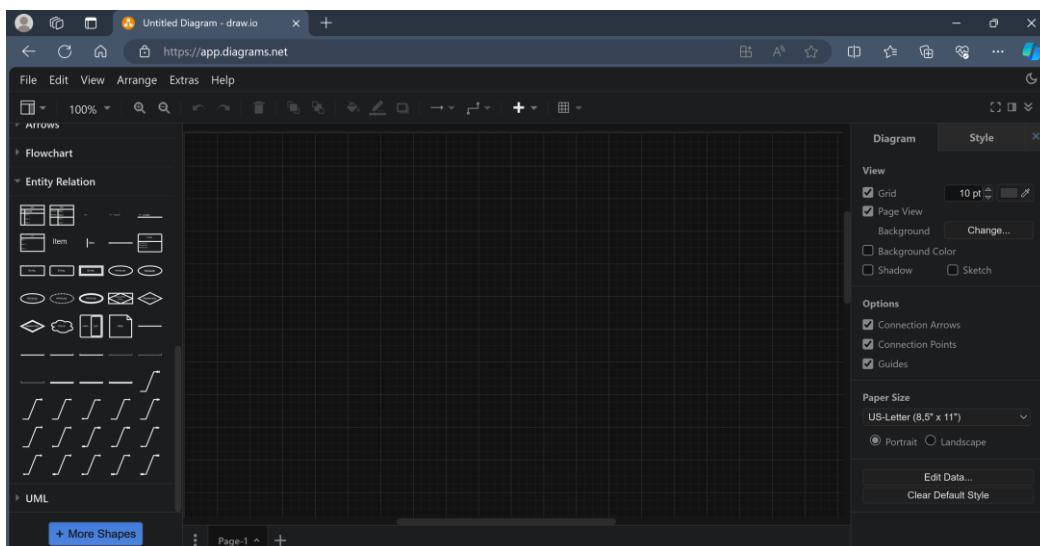


Figure 1- Displaying the user interface of draw.io

Entity Relationship Diagram (ERD)

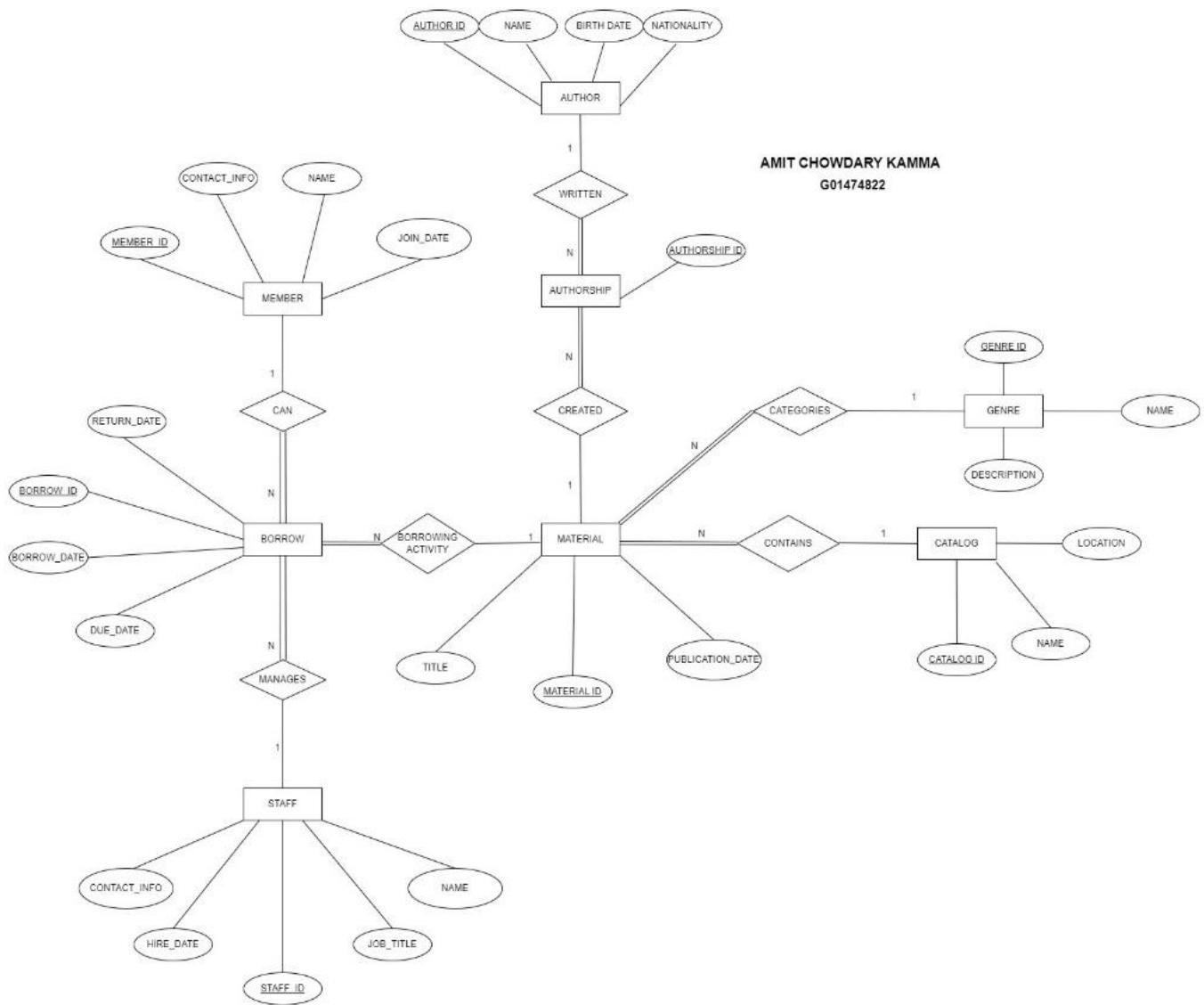


Figure 2 – Entity Relationship Diagram (ERD) of Library Management System

Entities

Entities are represented as rectangles as shown above in the diagram.

Material – This entity has the attributes of Material ID, Title, and Publication Date.

Catalog – This entity has the attributes Catalog ID, Name, and Location.

Genre - This entity has the attributes Genre ID, Name, and Description.

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

Borrow - This entity has the attributes Borrow ID, Borrow Date, Due Date, and Return Date.

Author - This entity has the attributes Author ID, Name, Birth Date, and Nationality.

Authorship - This entity has the attributes Authorship ID.

Member - This entity has the attributes Member ID, Contact Info, Name, and Join Date.

Staff - This entity has the attributes Staff ID, Contact Info, Hire Date, Job Title, and Name.

Relationships

Relationships are represented as a diamond which comes in between of two entities like shown above in the figure 2. It shows the activity or relation between two entities. Below are the seven relationships present in the Entity Relationship diagram.

Written – The relationship between Author and Authorship.

Created – The relationship between Authorship and Material.

Categories – The relationship between Material and Genre.

Contains – The relationship between Material and Catalog.

Borrowing Activity – The relationship between Material and Borrow.

Manages – The relationship between Borrow and Staff.

Can – The relationship between Borrow and Member.

Relational Schema

To draw the relational schema, we make use of the ER- Relational Mapping Algorithm. This algorithm ideally consists of 9 steps but in reference to the Entity Relational Diagram (ERD) above we make use of Step1, and step 4. In addition, all the nine steps are checked but in the end the steps which satisfy this case is step 1 and step 4.

Step 1: Mapping of Regular Entity Types (Satisfies)

Step 2: Mapping of Weak Entity Types (Does not Satisfy hence skip the step)

Step 3: Mapping of Binary 1:1 Relationship Types (Does not Satisfy hence skip the step)

Step 4: Mapping of Binary 1: N or N:1 Relationship Types. (Satisfies)

Step 5: Mapping of Binary M:N Relationship Types. (Does not Satisfy hence skip the step)

Step 6: Mapping of Multivalued attributes. (Does not Satisfy hence skip the step)

Step 7: Mapping of N-ary Relationship Types. (Does not Satisfy hence skip the step)

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

Step 8: Options for Mapping Specialization or Generalization. (Does not Satisfy hence skip the step)

Step 9: Mapping of Union Types (Categories). (Does not Satisfy hence skip the step)

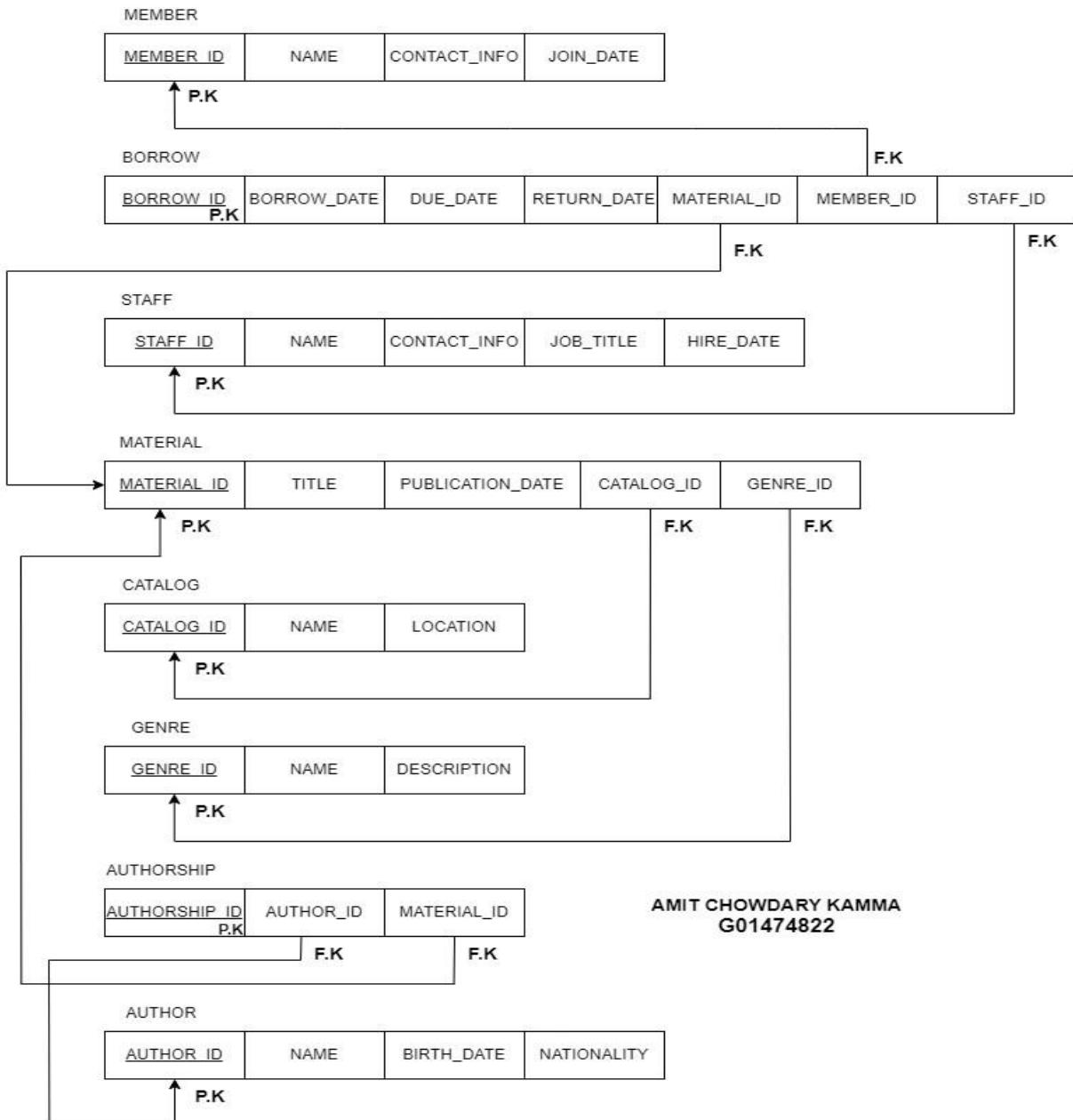


Figure 3 – Relational Schema of Library Management System

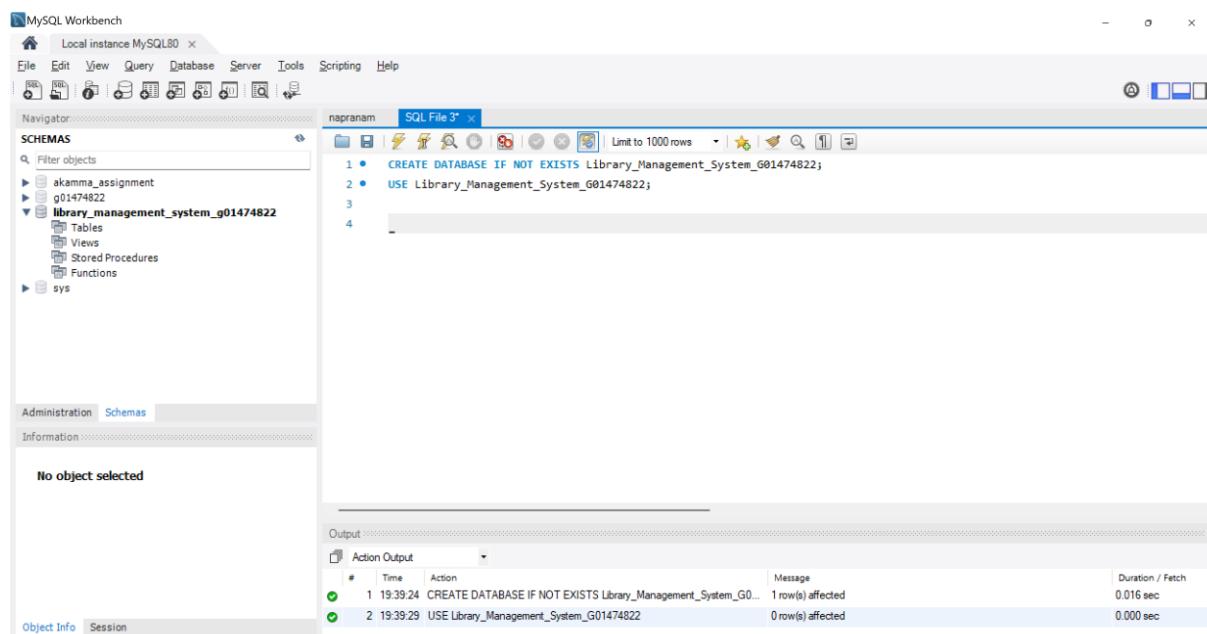
In the relational schema figure above, P.K represents primary key and F.K represents foreign key. In a database table, a primary key provides to uniquely identify a record, whereas a foreign key refers to the primary key of another table to create a relationship between two tables.

Database Implementation

Database Management System Used

For this project of Library Management System, I have used MySQL Workbench. With my previous experiences I found MySQL to be more user friendly and more driven towards security as we have a dedicated root password. Also, MySQL has the biggest number of users around the world which will help as many resources are present which can be of great help.

Creation of Database



Creation of Tables

CREATE TABLE Author(

Author_ID INT PRIMARY KEY,

`Name` VARCHAR(178),

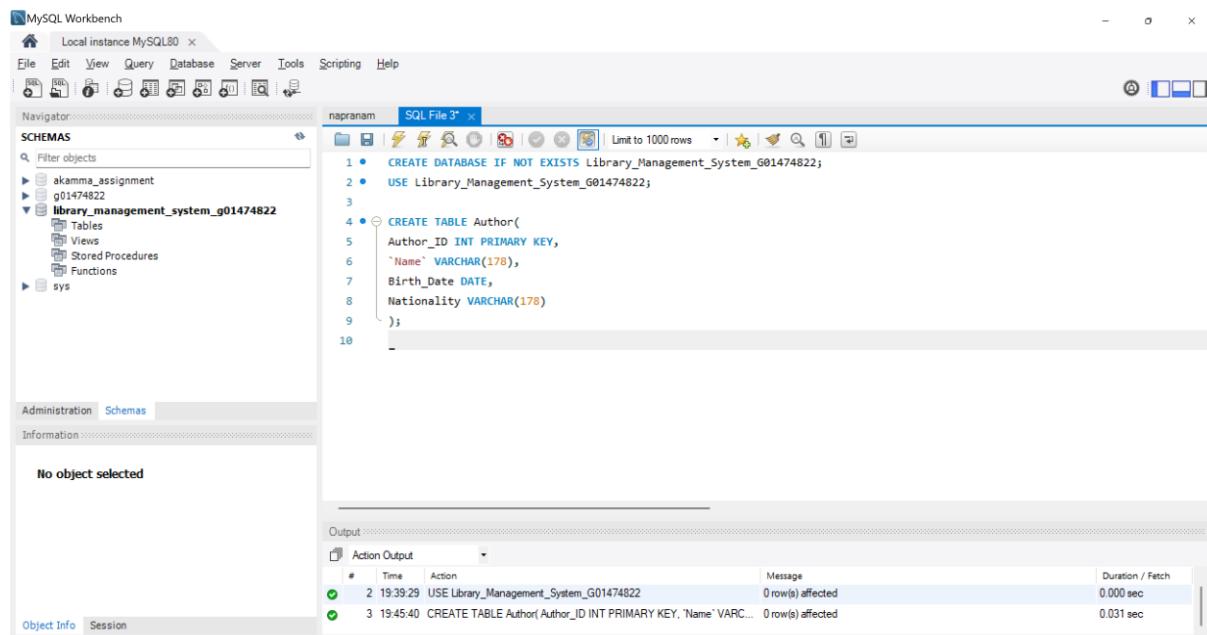
Birth_Date DATE,

Nationality VARCHAR(178)

);

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Shows the database structure with the schema `library_management_system_g01474822` selected.
- SQL Editor:** Titled "SQL File 3*", contains the SQL code for creating the `Author` table:

```
1 • CREATE DATABASE IF NOT EXISTS Library_Management_System_G01474822;
2 • USE Library_Management_System_G01474822;
3
4 • ○ CREATE TABLE Author(
5   Author_ID INT PRIMARY KEY,
6   `Name` VARCHAR(178),
7   Birth_Date DATE,
8   Nationality VARCHAR(178)
9 );
```
- Output Window:** Titled "Action Output", displays the execution results:

#	Time	Action	Message	Duration / Fetch
2	19:39:29	USE Library_Management_System_G01474822	0 row(s) affected	0.000 sec
3	19:45:40	CREATE TABLE Author(Author_ID INT PRIMARY KEY, `Name` VARC... 0 row(s) affected		0.031 sec

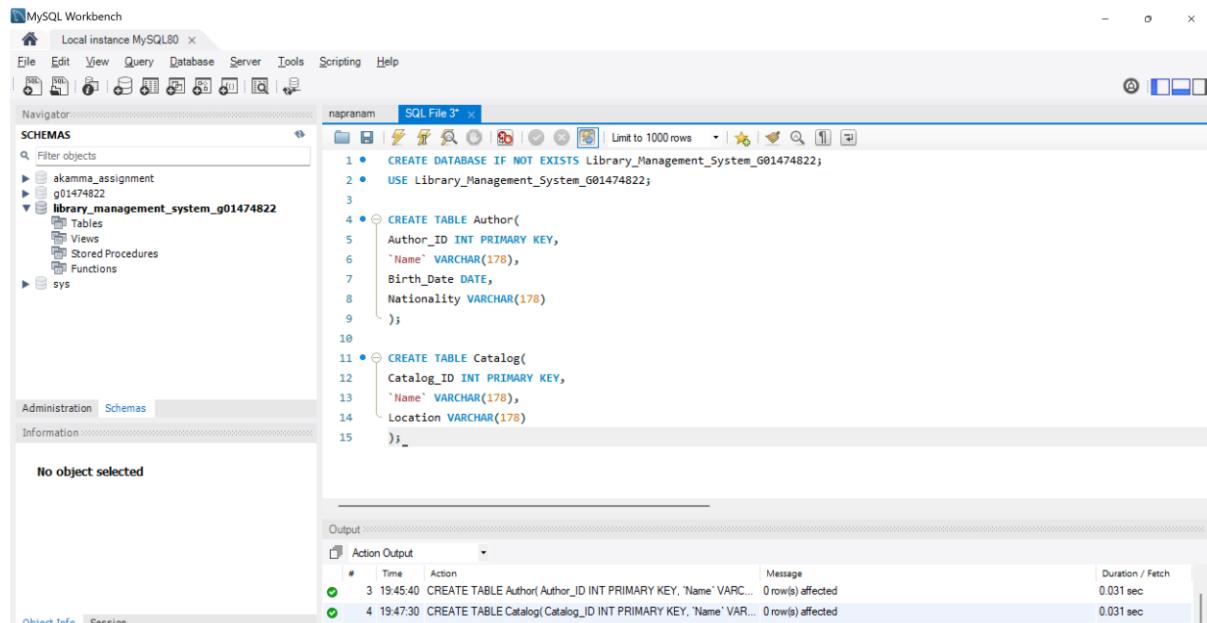
CREATE TABLE Catalog(

Catalog_ID INT PRIMARY KEY,

`Name` VARCHAR(178),

Location VARCHAR(178)

);



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Shows the database structure with the schema `library_management_system_g01474822` selected.
- SQL Editor:** Titled "SQL File 3*", contains the SQL code for creating the `Author` and `Catalog` tables:

```
1 • CREATE DATABASE IF NOT EXISTS Library_Management_System_G01474822;
2 • USE Library_Management_System_G01474822;
3
4 • ○ CREATE TABLE Author(
5   Author_ID INT PRIMARY KEY,
6   `Name` VARCHAR(178),
7   Birth_Date DATE,
8   Nationality VARCHAR(178)
9 );
10
11 • ○ CREATE TABLE Catalog(
12   Catalog_ID INT PRIMARY KEY,
13   `Name` VARCHAR(178),
14   Location VARCHAR(178)
15 );
```
- Output Window:** Titled "Action Output", displays the execution results:

#	Time	Action	Message	Duration / Fetch
3	19:45:40	CREATE TABLE Author(Author_ID INT PRIMARY KEY, `Name` VARC... 0 row(s) affected		0.031 sec
4	19:47:30	CREATE TABLE Catalog(Catalog_ID INT PRIMARY KEY, `Name` VAR... 0 row(s) affected		0.031 sec

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

CREATE TABLE Genre(

Genre_ID INT PRIMARY KEY,

`Name` VARCHAR(178),

Description VARCHAR(178)

);

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the **library_management_system_g01474822** database selected.
- SQL File 3:** Contains the SQL code for creating the **Catalog** and **Genre** tables.
- Output:** Displays the execution log with two entries:

#	Time	Action	Message	Duration / Fetch
4	19:47:30	CREATE TABLE Catalog(Catalog_ID INT PRIMARY KEY, `Name` VARCHAR(178), Location VARCHAR(178))	0 row(s) affected	0.031 sec
5	19:54:30	CREATE TABLE Genre(Genre_ID INT PRIMARY KEY, `Name` VARCHAR(178), Description VARCHAR(178))	0 row(s) affected	0.031 sec

CREATE TABLE Staff(

Staff_ID INT PRIMARY KEY,

`Name` VARCHAR(178),

Contact_Info VARCHAR(178),

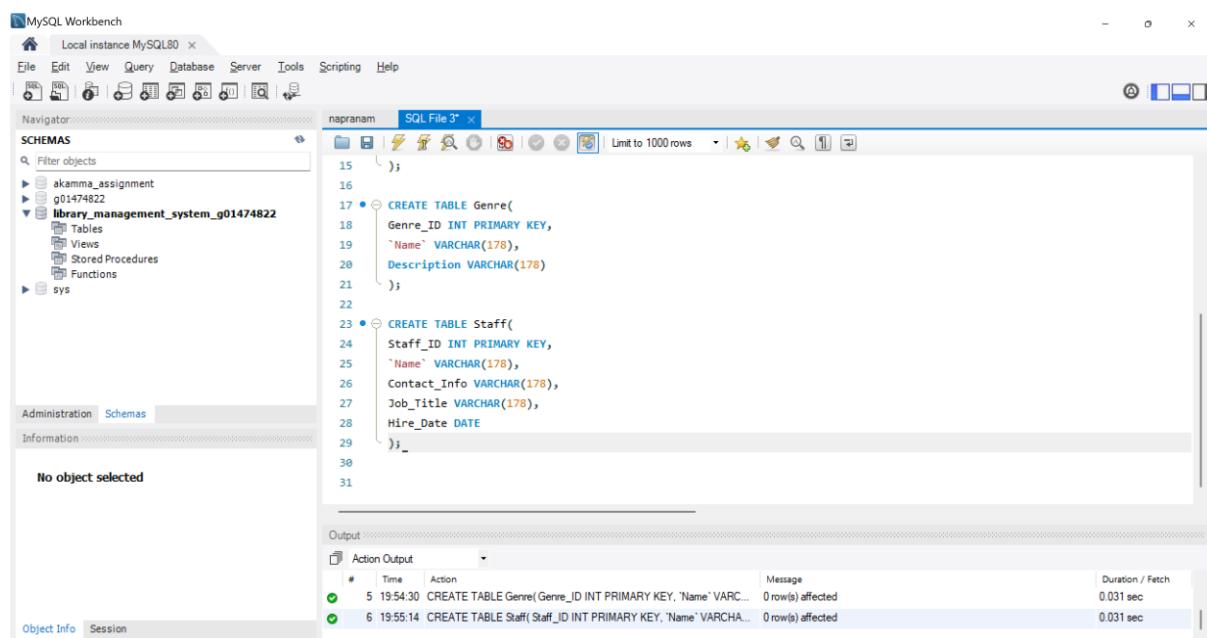
Job_Title VARCHAR(178),

Hire_Date DATE

);

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822



The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** library_management_system_g01474822
- SQL Editor:** napranam SQL File 3* contains the following SQL code:

```
15 );
16
17 • CREATE TABLE Genre(
18   Genre_ID INT PRIMARY KEY,
19   `Name` VARCHAR(178),
20   Description VARCHAR(178)
21 );
22
23 • CREATE TABLE Staff(
24   Staff_ID INT PRIMARY KEY,
25   `Name` VARCHAR(178),
26   Contact_Info VARCHAR(178),
27   Job_Title VARCHAR(178),
28   Hire_Date DATE
29 );
30
31
```
- Output:** Action Output table shows two successful CREATE TABLE operations with 0 rows affected and a duration of 0.031 sec each.

CREATE TABLE `Member`(`

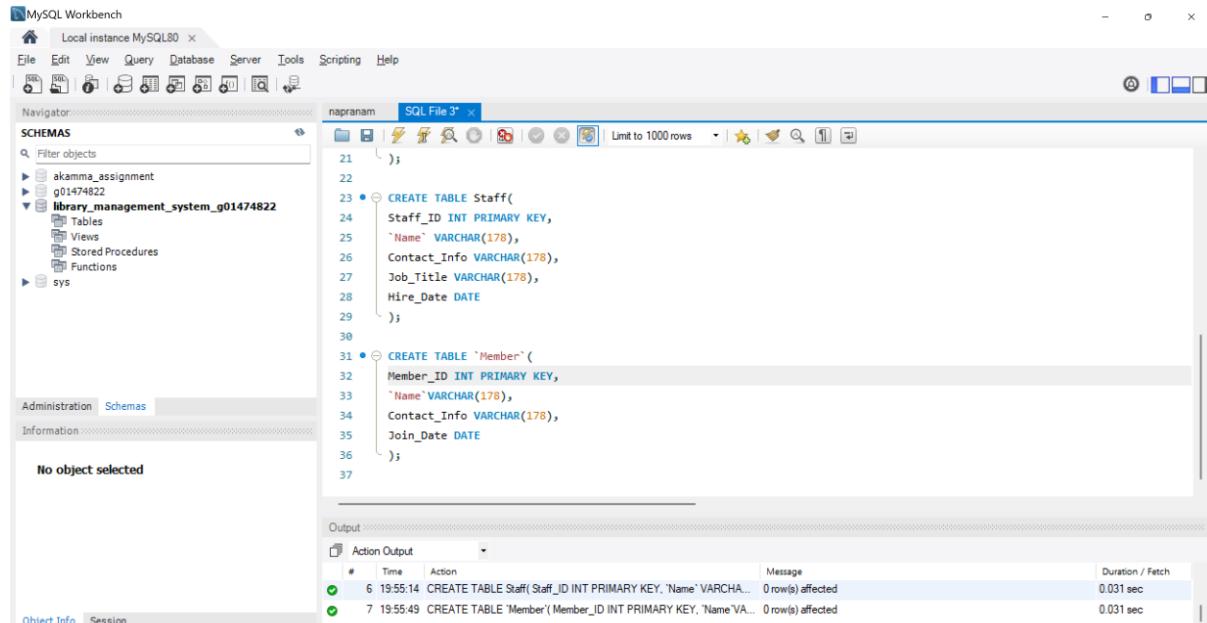
Member_ID INT PRIMARY KEY,

`Name`VARCHAR(178),

Contact_Info VARCHAR(178),

Join_Date DATE

);



The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** library_management_system_g01474822
- SQL Editor:** napranam SQL File 3* contains the following SQL code:

```
21 );
22
23 • CREATE TABLE Staff(
24   Staff_ID INT PRIMARY KEY,
25   `Name` VARCHAR(178),
26   Contact_Info VARCHAR(178),
27   Job_Title VARCHAR(178),
28   Hire_Date DATE
29 );
30
31 • CREATE TABLE `Member`(
32   Member_ID INT PRIMARY KEY,
33   `Name`VARCHAR(178),
34   Contact_Info VARCHAR(178),
35   Join_Date DATE
36 );
37
```
- Output:** Action Output table shows two successful CREATE TABLE operations with 0 rows affected and a duration of 0.031 sec each.

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

CREATE TABLE Material(

Material_ID INT PRIMARY KEY,

Title VARCHAR(178),

Publication_Date DATE,

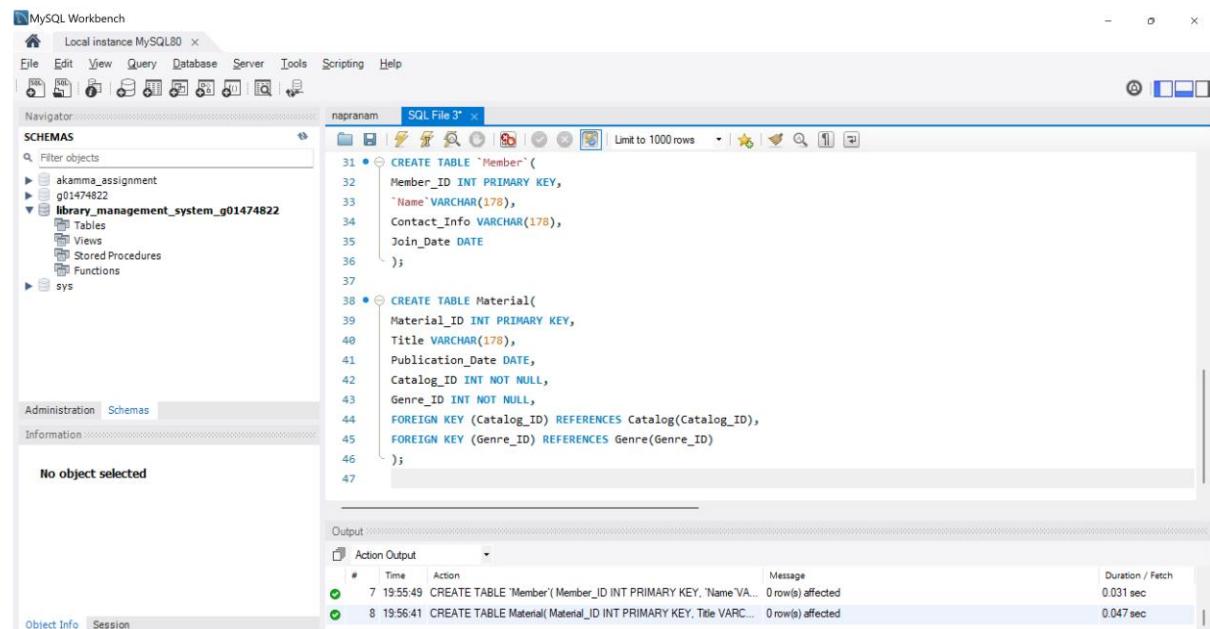
Catalog_ID INT NOT NULL,

Genre_ID INT NOT NULL,

FOREIGN KEY (Catalog_ID) REFERENCES Catalog(Catalog_ID),

FOREIGN KEY (Genre_ID) REFERENCES Genre(Genre_ID)

);



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Shows the database structure. Under the schema "library_management_system_g01474822", there are tables, views, stored procedures, and functions. A table named "Member" is also visible.
- SQL Editor:** Titled "SQL File 3*", contains the SQL code for creating the "Material" table. The code is as follows:

```
31 • CREATE TABLE `Member`(
32     `Member_ID` INT PRIMARY KEY,
33     `Name` VARCHAR(178),
34     `Contact_Info` VARCHAR(178),
35     `Join_Date` DATE
36 );
37
38 • CREATE TABLE Material(
39     Material_ID INT PRIMARY KEY,
40     Title VARCHAR(178),
41     Publication_Date DATE,
42     Catalog_ID INT NOT NULL,
43     Genre_ID INT NOT NULL,
44     FOREIGN KEY (Catalog_ID) REFERENCES Catalog(Catalog_ID),
45     FOREIGN KEY (Genre_ID) REFERENCES Genre(Genre_ID)
46 );
47
```

- Output:** Shows the execution log with two entries:

#	Time	Action	Message	Duration / Fetch
7	19:55:49	CREATE TABLE `Member`(`Member_ID` INT PRIMARY KEY, `Name`VA... 0 row(s) affected	0 rows affected	0.031 sec
8	19:56:41	CREATE TABLE Material(`Material_ID` INT PRIMARY KEY, `Title`V... 0 row(s) affected	0 rows affected	0.047 sec

CREATE TABLE Authorship(

Authorship_ID INT PRIMARY KEY,

Author_ID INT NOT NULL,

Material_ID INT NOT NULL,

FOREIGN KEY (Author_ID) REFERENCES Author(Author_ID),

FOREIGN KEY (Material_ID) REFERENCES Material(Material_ID)

);

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** library_management_system_g01474822 is selected.
- SQL Editor:** Contains the following SQL code:

```
38 • CREATE TABLE Material(
39     Material_ID INT PRIMARY KEY,
40     Title VARCHAR(178),
41     Publication_Date DATE,
42     Catalog_ID INT NOT NULL,
43     Genre_ID INT NOT NULL,
44     FOREIGN KEY (Catalog_ID) REFERENCES Catalog(Catalog_ID),
45     FOREIGN KEY (Genre_ID) REFERENCES Genre(Genre_ID)
46 );
47
48 • CREATE TABLE Authorship(
49     Authorship_ID INT PRIMARY KEY,
50     Author_ID INT NOT NULL,
51     Material_ID INT NOT NULL,
52     FOREIGN KEY (Author_ID) REFERENCES Author(Author_ID),
53     FOREIGN KEY (Material_ID) REFERENCES Material(Material_ID)
54 );
```

- Output:** Shows two log entries:

#	Time	Action	Message	Duration / Fetch
8	19:56:41	CREATE TABLE Material(Material_ID INT PRIMARY KEY, Title VARCHAR...	0 row(s) affected	0.047 sec
9	19:57:31	CREATE TABLE Authorship(Authorship_ID INT PRIMARY KEY, Author...	0 row(s) affected	0.047 sec

CREATE TABLE Borrow(

Borrow_ID INT PRIMARY KEY,

Borrow_Date DATE,

Due_Date DATE ,

Return_Date DATE ,

Material_ID INT NOT NULL,

Member_ID INT NOT NULL,

Staff_ID INT NOT NULL,

FOREIGN KEY (Material_ID) REFERENCES Material(Material_ID),

FOREIGN KEY (Member_ID) REFERENCES `Member`(Member_ID),

FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)

);

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** library_management_system_g01474822 is selected.
- SQL Editor:** Contains the SQL code for creating the Borrow and Authorship tables, including their primary keys, foreign keys, and data types.
- Output Window:** Shows the execution log with two entries indicating the successful creation of the tables.

```
51 Material_ID INT NOT NULL,
52 FOREIGN KEY (Author_ID) REFERENCES Author(Author_ID),
53 FOREIGN KEY (Material_ID) REFERENCES Material(Material_ID)
54 );
55
56 • CREATE TABLE Borrow(
57 Borrow_ID INT PRIMARY KEY,
58 Borrow_Date DATE,
59 Due_Date DATE ,
60 Return_Date DATE ,
61 Material_ID INT NOT NULL,
62 Member_ID INT NOT NULL,
63 Staff_ID INT NOT NULL,
64 FOREIGN KEY (Material_ID) REFERENCES Material(Material_ID),
65 FOREIGN KEY (Member_ID) REFERENCES `Member` (Member_ID),
66 FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
67 );
```

#	Time	Action	Message	Duration / Fetch
9	19:57:31	CREATE TABLE Authorship(Authorship_ID INT PRIMARY KEY, Author...	0 row(s) affected	0.047 sec
10	19:58:10	CREATE TABLE Borrow(Borrow_ID INT PRIMARY KEY, Borrow_Date ...	0 row(s) affected	0.063 sec

Insertion of Tables

INSERT INTO Author

(Author_id, `Name`, Birth_Date, Nationality)

VALUES

(1, 'Jane Austen', '1775-12-16', 'British'),
(2, 'Ernest Hemingway', '1899-07-21', 'American'),
(3, 'George Orwell', '1903-06-25', 'British'),
(4, 'Scott Fitzgerald', '1896-09-24', 'American'),
(5, 'J.K. Rowling', '1965-07-31', 'British'),
(6, 'Mark Twain', '1835-11-30', 'American'),
(7, 'Leo Tolstoy', '1828-09-09', 'Russian'),
(8, 'Virginia Woolf', '1882-01-25', 'British'),
(9, 'Gabriel Márquez', '1927-03-06', 'Colombian'),
(10, 'Charles Dickens', '1812-02-07', 'British'),
(11, 'Harper Lee', '1926-04-28', 'American'),
(12, 'Oscar Wilde', '1854-10-16', 'Irish'),
(13, 'William Shakespeare', '1564-04-26', 'British'),
(14, 'Franz Kafka', '1883-07-03', 'Czech'),

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

(15, 'James Joyce', '1882-02-02', 'Irish'),
(16, 'J.R.R. Tolkien', '1892-01-03', 'British'),
(17, 'Emily Brontë', '1818-07-30', 'British'),
(18, 'Toni Morrison', '1931-02-18', 'American'),
(19, 'Fyodor Dostoevsky', '1821-11-11', 'Russian'),
(20, 'Lucas Piki', '1847-10-16', 'British');

✓ 11 20:00:29 INSERT INTO Author (Author_id, `Name`, Birth_Date, Nationality) VAL... 20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0 0.063 sec

	Author_ID	Name	Birth_Date	Nationality
▶	1	Jane Austen	1775-12-16	British
	2	Ernest Hemingway	1899-07-21	American
	3	George Orwell	1903-06-25	British
	4	Scott Fitzgerald	1896-09-24	American
	5	J.K. Rowling	1965-07-31	British
	6	Mark Twain	1835-11-30	American
	7	Leo Tolstoy	1828-09-09	Russian
	8	Virginia Woolf	1882-01-25	British
	9	Gabriel Márquez	1927-03-06	Colombian
	10	Charles Dickens	1812-02-07	British
	11	Harper Lee	1926-04-28	American
	12	Oscar Wilde	1854-10-16	Irish
	13	William Shakespeare	1564-04-26	British
	14	Franz Kafka	1883-07-03	Czech
	15	James Joyce	1882-02-02	Irish
	16	J.R.R. Tolkien	1892-01-03	British
	17	Emily Brontë	1818-07-30	British
	18	Toni Morrison	1931-02-18	American
	19	Fyodor Dostoevsky	1821-11-11	Russian
	20	Lucas Piki	1847-10-16	British
*	NUL	NUL	NUL	NUL

INSERT INTO Catalog

(Catalog_ID, `Name`, Location)

VALUES

(1, 'Books', 'A1.1'),
(2, 'Magazines', 'B2.1'),
(3, 'E-Books', 'C3.1'),
(4, 'Audiobooks', 'D4.1'),
(5, 'Journals', 'E5.1'),
(6, 'Newspaper', 'F6.1'),
(7, 'Maps', 'G7.1'),

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

(8, 'Novels', 'H8.1'),

(9, 'Sheet Music', 'I9.1'),

(10, 'Educational', 'J10.1');

✓ 13 20:05:55 INSERT INTO Catalog (Catalog_ID, `Name`, Location) VALUES (1, 'B...', 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 0.015 sec

	Catalog_ID	Name	Location
▶	1	Books	A1.1
	2	Magazines	B2.1
	3	E-Books	C3.1
	4	Audiobooks	D4.1
	5	Journals	E5.1
	6	Newspaper	F6.1
	7	Maps	G7.1
	8	Novels	H8.1
	9	Sheet Music	I9.1
	10	Educational	J10.1
*	NULL	NULL	NULL

INSERT INTO Genre

(Genre_ID, `Name`, Description)

VALUES

(1, 'General Fiction', "Literary works with a focus on character and plot development, exploring various themes and human experiences."),

(2, 'Mystery & Thriller', "Suspenseful stories centered around crime, investigation, or espionage with an emphasis on tension and excitement."),

(3, 'Science Fiction & Fantasy', "Imaginative works that explore alternate realities, futuristic concepts, and magical or supernatural elements."),

(4, 'Horror & Suspense', "Stories designed to evoke fear, unease, or dread, often featuring supernatural or psychological elements."),

(5, 'Dystopian & Apocalyptic', "Depictions of societies in decline or collapse, often exploring themes of political and social oppression or environmental disaster."),

(6, 'Classics', "Enduring works of literature that have stood the test of time, often featuring rich language and complex themes."),

(7, 'Historical Fiction', "Fictional stories set in the past, often based on real historical events or figures, and exploring the customs and experiences of that time."),

(8, 'Epic Poetry & Mythology', "Ancient or traditional stories and poems, often featuring heroes, gods, and mythical creatures, and exploring cultural values and beliefs.");

✓ 14 20:07:24 INSERT INTO Genre (Genre_ID, `Name`, Description) VALUES (1, 'Ge...', 8 row(s) affected Records: 8 Duplicates: 0 Warnings: 0 0.000 sec

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

	Genre_ID	Name	Description
▶	1	General Fiction	"Literary works with a focus on character and plot development, exploring various themes and human experiences."
	2	Mystery & Thriller	"Suspenseful stories centered around crime, investigation, or espionage with an emphasis on tension and excitement."
	3	Science Fiction & Fantasy	"Imaginative works that explore alternate realities, futuristic concepts, and magical or supernatural elements."
	4	Horror & Suspense	"Stories designed to evoke fear, unease, or dread, often featuring supernatural or psychological elements."
	5	Dystopian & Apocalyptic	"Depictions of societies in decline or collapse, often exploring themes of political and social oppression or environmental disaster."
	6	Classics	"Enduring works of literature that have stood the test of time, often featuring rich language and complex themes."
	7	Historical Fiction	"Fictional stories set in the past, often based on real historical events or figures, and exploring the customs and experiences of that time."
	8	Epic Poetry & Mythology	"Ancient or traditional stories and poems, often featuring heroes, gods, and mythical creatures, and exploring cultural values and beliefs."
	NULL	NULL	NULL

INSERT INTO Staff

(Staff_ID, `Name`, Contact_Info, Job_Title, Hire_Date)

VALUES

```
(1, 'Amy Green', 'amy.green@email.com', 'Librarian', '2017-06-01'),
(2, 'Brian Taylor', 'brian.taylor@email.com', 'Library Assistant', '2018-11-15'),
(3, 'Christine King', 'christine.king@email.com', 'Library Assistant', '2019-05-20'),
(4, 'Daniel Wright', 'daniel.wright@email.com', 'Library Technician', '2020-02-01');
```

✓ 15 20:09:01 INSERT INTO Staff (Staff_ID, `Name`, Contact_Info, Job_Title, Hire_Date) VALUES (1, 'Amy Green', 'amy.green@email.com', 'Librarian', '2017-06-01'), (2, 'Brian Taylor', 'brian.taylor@email.com', 'Library Assistant', '2018-11-15'), (3, 'Christine King', 'christine.king@email.com', 'Library Assistant', '2019-05-20'), (4, 'Daniel Wright', 'daniel.wright@email.com', 'Library Technician', '2020-02-01') 4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0 0.000 sec

	Staff_ID	Name	Contact_Info	Job_Title	Hire_Date
▶	1	Amy Green	amy.green@email.com	Librarian	2017-06-01
	2	Brian Taylor	brian.taylor@email.com	Library Assistant	2018-11-15
	3	Christine King	christine.king@email.com	Library Assistant	2019-05-20
	4	Daniel Wright	daniel.wright@email.com	Library Technician	2020-02-01
	NULL	NULL	NULL	NULL	NULL

INSERT INTO `Member`

(Member_ID, `Name`, Contact_Info, Join_Date)

VALUES

```
(1, 'Alice Johnson', 'alice.johnson@email.com', '2018-01-10'),
(2, 'Bob Smith', 'bob.smith@email.com', '2018-03-15'),
(3, 'Carol Brown', 'carol.brown@email.com', '2018-06-20'),
(4, 'David Williams', 'david.williams@email.com', '2018-09-18'),
(5, 'Emily Miller', 'emily.miller@email.com', '2019-02-12'),
(6, 'Frank Davis', 'frank.davis@email.com', '2019-05-25'),
(7, 'Grace Wilson', 'grace.wilson@email.com', '2019-08-15'),
(8, 'Harry Garcia', 'harry.garcia@email.com', '2019-11-27'),
(9, 'Isla Thomas', 'isla.thomas@email.com', '2020-03-04'),
(10, 'Jack Martinez', 'jack.martinez@email.com', '2020-07-01'),
```

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

```
(11, 'Kate Anderson', 'kate.anderson@email.com', '2020-09-30'),  
(12, 'Luke Jackson', 'luke.jackson@email.com', '2021-01-18'),  
(13, 'Mia White', 'mia.white@email.com', '2021-04-27'),  
(14, 'Noah Harris', 'noah.harris@email.com', '2021-07-13'),  
(15, 'Olivia Clark', 'olivia.clark@email.com', '2021-10-05'),  
(16, 'Peter Lewis', 'peter.lewis@email.com', '2021-12-01'),  
(17, 'Quinn Hall', 'quinn.hall@email.com', '2022-02-28'),  
(18, 'Rachel Young', 'rachel.young@email.com', '2022-06-17'),  
(19, 'Sam Walker', 'sam.walker@email.com', '2022-09-25'),  
(20, 'Tiffany Allen', 'tiffany.allen@email.com', '2022-12-10');
```

✓ 16 20:09:59 INSERT INTO 'Member' (Member_ID, Name, Contact_Info, Join_Da... 20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0 0.015 sec

	Member_ID	Name	Contact_Info	Join_Date
▶	1	Alice Johnson	alice.johnson@email.com	2018-01-10
	2	Bob Smith	bob.smith@email.com	2018-03-15
	3	Carol Brown	carol.brown@email.com	2018-06-20
	4	David Williams	david.williams@email.com	2018-09-18
	5	Emily Miller	emily.miller@email.com	2019-02-12
	6	Frank Davis	frank.davis@email.com	2019-05-25
	7	Grace Wilson	grace.wilson@email.com	2019-08-15
	8	Harry Garcia	harry.garcia@email.com	2019-11-27
	9	Isla Thomas	isla.thomas@email.com	2020-03-04
	10	Jack Martinez	jack.martinez@email.com	2020-07-01
	11	Kate Anderson	kate.anderson@email.com	2020-09-30
	12	Luke Jackson	luke.jackson@email.com	2021-01-18
	13	Mia White	mia.white@email.com	2021-04-27
	14	Noah Harris	noah.harris@email.com	2021-07-13
	15	Olivia Clark	olivia.clark@email.com	2021-10-05
	16	Peter Lewis	peter.lewis@email.com	2021-12-01
	17	Quinn Hall	quinn.hall@email.com	2022-02-28
	18	Rachel Young	rachel.young@email.com	2022-06-17
	19	Sam Walker	sam.walker@email.com	2022-09-25
*	20	Tiffany Allen	tiffany.allen@email.com	2022-12-10
	NULL	NULL	NULL	NULL

INSERT INTO Material

(Material_ID, Title, Publication_Date, Catalog_ID, Genre_ID)

VALUES

```
(1, 'The Catcher in the Rye', '1951-07-16', 1, 1),  
(2, 'To Kill a Mockingbird', '1960-07-11', 2, 1),  
(3, 'The Da Vinci Code', '2003-04-01', 3, 2),  
(4, 'The Hobbit', '1937-09-21', 4, 3),
```

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

- (5, 'The Shining', '1977-01-28', 5, 4),
(6, 'Pride and Prejudice', '1813-10-18', 3, 1),
(7, 'The Great Gatsby', '1925-04-10', 2, 1),
(8, 'Moby Dick', '1851-10-18', 3, 1),
(9, 'Crime and Punishment', '1866-01-01', 4, 1),
(10, 'The Hitchhikers Guide to the Galaxy', '1979-10-12', 5, 3),
(11, '1984', '1949-06-08', 1, 5),
(12, 'Animal Farm', '1945-08-17', 2, 5),
(13, 'The Haunting of Hill House', '1959-10-17', 3, 4),
(14, 'Brave New World', '1932-08-01', 4, 5),
(15, 'The Chronicles of Narnia: The Lion the Witch and the Wardrobe', '1950-10-16', 5, 3),
(16, 'The Adventures of Huckleberry Finn', '1884-12-10', 6, 1),
(17, 'The Catch-22', '1961-11-10', 7, 1),
(18, 'The Picture of Dorian Gray', '1890-07-01', 8, 1),
(19, 'The Call of Cthulhu', '1928-02-01', 9, 4),
(20, 'Harry Potter and the Philosopher"s Stone', '1997-06-26', 10, 3),
(21, 'Frankenstein', '1818-01-01', 6, 4),
(22, 'A Tale of Two Cities', '1859-04-30', 7, 1),
(23, 'The Iliad', '1750-01-01', 8, 6),
(24, 'The Odyssey', '1725-01-01', 9, 6),
(25, 'The Brothers Karamazov', '1880-01-01', 10, 1),
(26, 'The Divine Comedy', '1320-01-01', 6, 6),
(27, 'The Grapes of Wrath', '1939-04-14', 7, 1),
(28, 'The Old Man and the Sea', '1952-09-01', 8, 1),
(29, 'The Count of Monte Cristo', '1844-01-01', 9, 1),
(30, 'A Midsummer Nights Dream', '1596-01-01', 10, 7),
(31, 'The Tricky Book', '1888-01-01', 10, 7);

✓ 17 20:11:30 INSERT INTO Material (Material_ID, Title, Publication_Date, Catalog_... 31 row(s) affected Records: 31 Duplicates: 0 Warnings: 0

0.000 sec

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

	Material_ID	Title	Publication_Date	Catalog_ID	Genre_ID
▶	1	The Catcher in the Rye	1951-07-16	1	1
	2	To Kill a Mockingbird	1960-07-11	2	1
	3	The Da Vinci Code	2003-04-01	3	2
	4	The Hobbit	1937-09-21	4	3
	5	The Shining	1977-01-28	5	4
	6	Pride and Prejudice	1813-10-18	3	1
	7	The Great Gatsby	1925-04-10	2	1
	8	Moby Dick	1851-10-18	3	1
	9	Crime and Punishment	1866-01-01	4	1
	10	The Hitchhiker's Guide to the Galaxy	1979-10-12	5	3
	11	1984	1949-06-08	1	5
	12	Animal Farm	1945-08-17	2	5
	13	The Haunting of Hill House	1959-10-17	3	4
	14	Brave New World	1932-08-01	4	5
	15	The Chronicles of Narnia: The Lion ...	1950-10-16	5	3
	16	The Adventures of Huckleberry Finn	1884-12-10	6	1

	Material_ID	Title	Publication_Date	Catalog_ID	Genre_ID
	17	The Catch-22	1961-11-10	7	1
	18	The Picture of Dorian Gray	1890-07-01	8	1
	19	The Call of Cthulhu	1928-02-01	9	4
	20	Harry Potter and the Philosopher's ...	1997-06-26	10	3
	21	Frankenstein	1818-01-01	6	4
	22	A Tale of Two Cities	1859-04-30	7	1
	23	The Iliad	1750-01-01	8	6
	24	The Odyssey	1725-01-01	9	6
	25	The Brothers Karamazov	1880-01-01	10	1
	26	The Divine Comedy	1320-01-01	6	6
	27	The Grapes of Wrath	1939-04-14	7	1
	28	The Old Man and the Sea	1952-09-01	8	1
	29	The Count of Monte Cristo	1844-01-01	9	1
	30	A Midsummer Night's Dream	1596-01-01	10	7
	31	The Tricky Book	1888-01-01	10	7
*	NULL	NULL	NULL	NULL	NULL

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

```
INSERT INTO Authorship
(Authorship_ID, Author_ID, Material_ID)
VALUES
(1, 1, 1),
(2, 2, 2),
(3, 3, 3),
(4, 4, 4),
(5, 5, 5),
(6, 6, 6),
(7, 7, 7),
(8, 8, 8),
(9, 9, 9),
(10, 10, 10),
(11, 11, 11),
(12, 12, 12),
(13, 13, 13),
(14, 14, 14),
(15, 15, 15),
(16, 16, 16),
(17, 17, 17),
(18, 18, 18),
(19, 19, 19),
(20, 20, 20),
(21, 1, 21),
(22, 2, 22),
(23, 2, 22),
(24, 3, 23),
(25, 4, 24),
(26, 5, 25),
(27, 6, 26),
(28, 7, 27),
(29, 8, 28),
(30, 19, 28),
```

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

(31, 9, 29),

(32, 10, 30),

(33, 8, 30),

(34, 2, 29);

18 20:13:08 INSERT INTO Authorship (Authorship_ID, Author_ID, Material_ID) VA... 34 row(s) affected Records: 34 Duplicates: 0 Warnings: 0 0.000 sec

	Authorship_ID	Author_ID	Material_ID
▶	1	1	1
	2	2	2
	3	3	3
	4	4	4
	5	5	5
	6	6	6
	7	7	7
	8	8	8
	9	9	9
	10	10	10
	11	11	11
	12	12	12
	13	13	13
	14	14	14
	15	15	15
	16	16	16

	Authorship_ID	Author_ID	Material_ID
	17	17	17
	18	18	18
	19	19	19
	20	20	20
	21	1	21
	22	2	22
	23	2	22
	24	3	23
	25	4	24
	26	5	25
	27	6	26
	28	7	27
	29	8	28
	30	19	28
	31	9	29
	32	10	30
	33	8	30
	34	2	29
✳	NULL	NULL	NULL

INSERT INTO Borrow

(Borrow_ID, Material_ID, Member_ID, Staff_ID, Borrow_Date, Due_Date, Return_Date)

VALUES

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

(1, 1, 1, 1, '2018-09-12', '2018-10-03', '2018-09-30'),
(2, 2, 2, 1, '2018-10-15', '2018-11-05', '2018-10-29'),
(3, 3, 3, 1, '2018-12-20', '2019-01-10', '2019-01-08'),
(4, 4, 4, 1, '2019-03-11', '2019-04-01', '2019-03-27'),
(5, 5, 5, 1, '2019-04-20', '2019-05-11', '2019-05-05'),
(6, 6, 6, 1, '2019-07-05', '2019-07-26', '2019-07-21'),
(7, 7, 7, 1, '2019-09-10', '2019-10-01', '2019-09-25'),
(8, 8, 8, 1, '2019-11-08', '2019-11-29', '2019-11-20'),
(9, 9, 9, 1, '2020-01-15', '2020-02-05', '2020-02-03'),
(10, 10, 10, 1, '2020-03-12', '2020-04-02', '2020-03-28'),
(11, 1, 11, 2, '2020-05-14', '2020-06-04', '2020-05-28'),
(12, 2, 12, 2, '2020-07-21', '2020-08-11', '2020-08-02'),
(13, 3, 13, 2, '2020-09-25', '2020-10-16', '2020-10-15'),
(14, 4, 1, 2, '2020-11-08', '2020-11-29', '2020-11-24'),
(15, 5, 2, 2, '2021-01-03', '2021-01-24', '2021-01-19'),
(16, 6, 3, 2, '2021-02-18', '2021-03-11', '2021-03-12'),
(17, 17, 4, 2, '2021-04-27', '2021-05-18', '2021-05-20'),
(18, 18, 5, 2, '2021-06-13', '2021-07-04', '2021-06-28'),
(19, 19, 6, 2, '2021-08-15', '2021-09-05', '2021-09-03'),
(20, 20, 7, 2, '2021-10-21', '2021-11-11', NULL),
(21, 21, 1, 3, '2021-11-29', '2021-12-20', NULL),
(22, 22, 2, 3, '2022-01-10', '2022-01-31', '2022-01-25'),
(23, 23, 3, 3, '2022-02-07', '2022-02-28', '2022-02-23'),
(24, 24, 4, 3, '2022-03-11', '2022-04-01', '2022-03-28'),
(25, 25, 5, 3, '2022-04-28', '2022-05-19', '2022-05-18'),
(26, 26, 6, 3, '2022-06-22', '2022-07-13', '2022-07-08'),
(27, 27, 7, 3, '2022-08-04', '2022-08-25', '2022-08-23'),
(28, 28, 8, 3, '2022-09-13', '2022-10-04', '2022-09-28'),
(29, 29, 9, 3, '2022-10-16', '2022-11-06', '2022-11-05'),
(30, 30, 8, 3, '2022-11-21', '2022-12-12', '2022-12-05'),
(31, 1, 9, 4, '2022-12-28', '2023-01-18', NULL),
(32, 2, 1, 4, '2023-01-23', '2023-02-13', NULL),
(33, 3, 10, 4, '2023-02-02', '2023-02-23', '2023-02-17'),

Querying and Manipulation

1. Which materials are currently available in the library? If a material is borrowed and not returned, it is not considered as available.

```
SELECT m.Material_ID, m.Title FROM Material AS m  
WHERE NOT EXISTS (  
    SELECT *  
    FROM Borrow AS b  
    WHERE m.Material_ID = b.Material_ID AND b.Return_Date IS NULL  
);
```

Result:

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the `library_management_system_g01474822` database selected.
- SQL Editor:** Displays the SQL query used to find available materials.

```
281  
282 #1. Which materials are currently available in the library?  
283 #If a material is borrowed and not returned, it's not considered as available.  
284  
285 • SELECT m.Material_ID, m.Title FROM Material AS m  
286 ⚡ WHERE NOT EXISTS (  
    SELECT *  
    FROM Borrow AS b  
    WHERE m.Material_ID = b.Material_ID AND b.Return_Date IS NULL  
);
```
- Result Grid:** Shows the results of the query, listing 31 materials with their IDs and titles.

Material_ID	Title
3	The Da Vinci Code
11	1984
12	Animal Farm
13	The Haunting of Hill House
14	Brave New World
15	The Chronicles of Narnia: The Lion the Witch and the Wardrobe
16	The Adventures of Huckleberry Finn
17	The Catch-22
18	The Picture of Dorian Gray
19	The Call of Cthulhu
22	A Tale of Two Cities
23	The Iliad
24	The Odyssey
25	The Brothers Karamazov
26	The Divine Comedy
27	The Grapes of Wrath
28	The Old Man and the Sea
29	The Count of Monte Cristo
30	A Midsummer Night's Dream
31	The Tricky Book
NULL	NULL

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

2. Which materials are currently overdue? Suppose today is 04/01/2023, and show the borrow date and due date of each material.

```
SELECT M.Material_ID, M.Title, B.Borrow_Date, B.Due_Date  
FROM Material AS M, Borrow AS B  
WHERE M.Material_ID = B.Material_ID AND B.Due_Date < '2023-04-01' AND  
B.Return_Date IS NULL ;
```

Result:

The screenshot shows the MySQL Workbench interface. In the top navigation bar, it says "Local instance MySQL80". Below that is a toolbar with various icons. The left sidebar is titled "Navigator" and shows "SCHEMAS" with "library_management_system_g01474822" expanded, revealing "Tables", "Views", "Stored Procedures", and "Functions". The main area has a tab titled "SQL File 3" with the following SQL code:

```
294  
295 • SELECT M.Material_ID, M.Title, B.Borrow_Date, B.Due_Date  
296   FROM Material AS M, Borrow AS B  
297 WHERE M.Material_ID = B.Material_ID AND B.Due_Date < '2023-04-01' AND  
298   B.Return_Date IS NULL ;
```

Below the code is a "Result Grid" table with the following data:

Material_ID	Title	Borrow_Date	Due_Date
20	Harry Potter and the Philosopher's Stone	2021-10-21	2021-11-11
21	Frankenstein	2021-11-29	2021-12-20
1	The Catcher in the Rye	2022-12-28	2023-01-18
2	To Kill a Mockingbird	2023-01-23	2023-02-13
4	The Hobbit	2023-03-01	2023-03-22
5	The Shining	2023-03-10	2023-03-31

At the bottom, there is a "Result 13" panel showing the execution log:

Time	Action	Message	Duration / Fetch
30 20:39:31	SELECT m.Material_ID, m.Title FROM Material AS m WHERE NOT E...	20 row(s) returned	0.000 sec / 0.000 sec
31 21:11:24	SELECT M.Material_ID, M.Title, B.Borrow_Date, B.Due_Date FROM ...	6 row(s) returned	0.016 sec / 0.000 sec

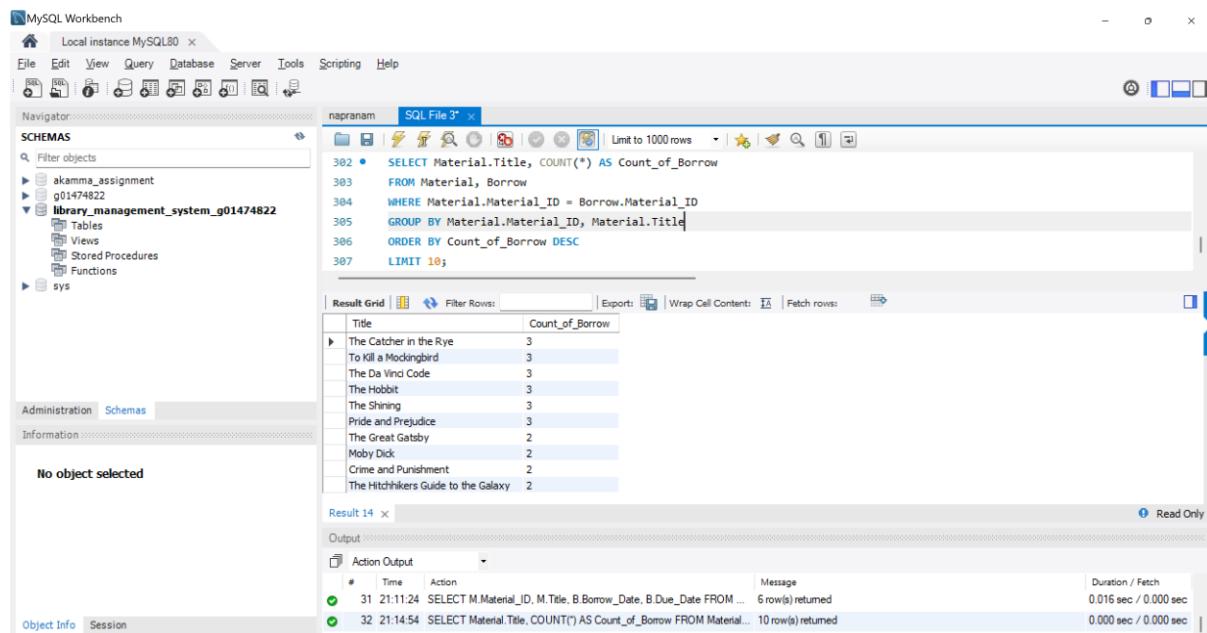
3. What are the top 10 **most borrowed** materials in the library? Show the title of each material and order them based on their **available counts**.

```
SELECT Material.Title, COUNT(*) AS Count_of_Borrow  
FROM Material, Borrow  
WHERE Material.Material_ID = Borrow.Material_ID  
GROUP BY Material.Material_ID, Material.Title  
ORDER BY Count_of_Borrow DESC  
LIMIT 10;
```

Result:

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822



The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'File', 'Edit', 'View', 'Query', 'Database', 'Server', 'Tools', 'Scripting', and 'Help' are visible. The 'Query' tab is active. The 'Navigator' pane on the left lists 'SCHEMAS' with 'library_management_system_g01474822' selected, containing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'Information' pane below it says 'No object selected'. The main workspace contains a SQL editor window titled 'SQL File 3' with the following query:

```
302 • SELECT Material.Title, COUNT(*) AS Count_of_Borrow
303   FROM Material, Borrow
304  WHERE Material.Material_ID = Borrow.Material_ID
305  GROUP BY Material.Material_ID, Material.Title
306  ORDER BY Count_of_Borrow DESC
307  LIMIT 10;
```

Below the SQL editor is a 'Result Grid' table with columns 'Title' and 'Count_of_Borrow'. The data is:

Title	Count_of_Borrow
The Catcher in the Rye	3
To Kill a Mockingbird	3
The Da Vinci Code	3
The Hobbit	3
The Shining	3
Pride and Prejudice	3
The Great Gatsby	2
Moby Dick	2
Crime and Punishment	2
The Hitchhiker's Guide to the Galaxy	2

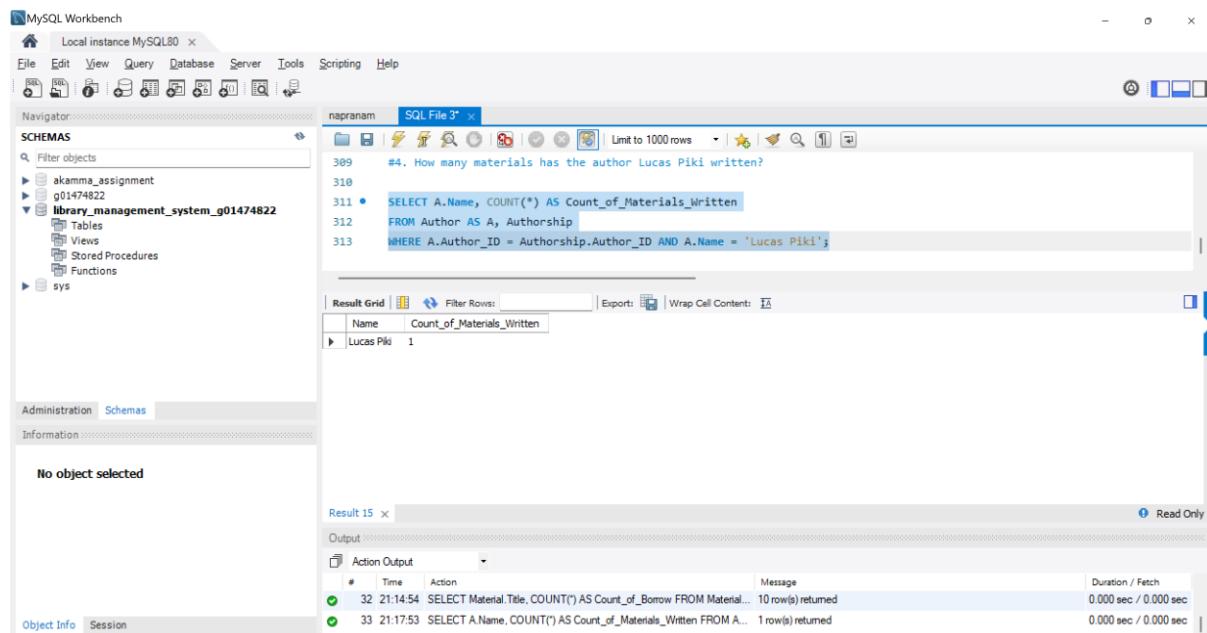
At the bottom of the workspace, the 'Output' pane shows two log entries:

#	Time	Action	Message	Duration / Fetch
31	21:11:24	SELECT M.Material_ID, M.Title, B.Borrow_Date, B.Due_Date FROM ...	6 row(s) returned	0.016 sec / 0.000 sec
32	21:14:54	SELECT Material.Title, COUNT(*) AS Count_of_Borrow FROM Material...	10 row(s) returned	0.000 sec / 0.000 sec

4. How many materials has the author Lucas Piki written?

```
SELECT A.Name, COUNT(*) AS Count_of_Materials_Written
FROM Author AS A, Authorship
WHERE A.Author_ID = Authorship.Author_ID AND A.Name = 'Lucas Piki';
```

Result:



The screenshot shows the MySQL Workbench interface. The 'Query' tab is active. The 'Navigator' pane on the left lists 'SCHEMAS' with 'library_management_system_g01474822' selected, containing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'Information' pane below it says 'No object selected'. The main workspace contains a SQL editor window titled 'SQL File 3' with the following query:

```
309 #4. How many materials has the author Lucas Piki written?
310
311 • SELECT A.Name, COUNT(*) AS Count_of_Materials_Written
312   FROM Author AS A, Authorship
313  WHERE A.Author_ID = Authorship.Author_ID AND A.Name = 'Lucas Piki';
```

Below the SQL editor is a 'Result Grid' table with columns 'Name' and 'Count_of_Materials_Written'. The data is:

Name	Count_of_Materials_Written
Lucas Piki	1

At the bottom of the workspace, the 'Output' pane shows two log entries:

#	Time	Action	Message	Duration / Fetch
32	21:14:54	SELECT Material.Title, COUNT(*) AS Count_of_Borrow FROM Material...	10 row(s) returned	0.000 sec / 0.000 sec
33	21:17:53	SELECT A.Name, COUNT(*) AS Count_of_Materials_Written FROM A...	1 row(s) returned	0.000 sec / 0.000 sec

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

5. How many materials were written by two or more authors?

```
SELECT COUNT(*) AS materials_count
FROM (
    SELECT Material_ID
    FROM Authorship
    GROUP BY Material_ID
    HAVING COUNT(*) >= 2
) AS multiple_authors;
```

Result:

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Navigator' with the 'SCHEMAS' section expanded, showing databases like 'akamma_assignment', 'g01474822', and 'library_management_system_g01474822'. The main area contains a SQL editor window titled 'SQL File 3' with the following code:

```
315  #5. How many materials were written by two or more authors?
316
317 •  SELECT COUNT(*) AS materials_count
318 ◇ FROM (
319     SELECT Material_ID
320     FROM Authorship
321     GROUP BY Material_ID
322     HAVING COUNT(*) >= 2
323 ) AS multiple_authors;
```

Below the SQL editor is a 'Result Grid' table with one row labeled 'materials_count' containing the value '4'. At the bottom of the interface, the 'Output' tab shows the execution log:

Action	Time	Action	Message	Duration / Fetch
33	21:17:53	SELECT A.Name, COUNT(*) AS Count_of_Materials_Written FROM A...	1 row(s) returned	0.000 sec / 0.000 sec
34	21:19:55	SELECT COUNT(*) AS materials_count FROM (SELECT Material_...	1 row(s) returned	0.000 sec / 0.000 sec

6. What are the most popular genres in the library ranked by the total number of borrowed times of each genre?

```
SELECT g.Genre_ID, g.Name AS Genre_Name, COUNT(*) AS Total_Borrowed
FROM Material m
JOIN Genre g ON m.Genre_ID = g.Genre_ID
JOIN Borrow b ON m.Material_ID = b.Material_ID
GROUP BY g.Genre_ID, g.Name
ORDER BY Total_Borrowed DESC;
```

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

Result:

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'File', 'Edit', 'View', 'Query', 'Database', 'Server', 'Tools', 'Scripting', and 'Help' are visible. The 'Navigator' pane on the left lists 'SCHEMAS' with 'akamma_assignment', 'g01474822', and 'library_management_system_g01474822'. The 'Tables' section under the library schema is expanded. The main area contains a SQL editor tab titled 'SQL File 3*' with the following query:

```
325 #6. What are the most popular genres in the library ranked by the total number of borrowed times of each genre?
326
327 • SELECT g.Genre_ID, g.Name AS Genre_Name, COUNT(*) AS Total_Borrowed
328 FROM Material m
329 JOIN Genre g ON m.Genre_ID = g.Genre_ID
330 JOIN Borrow b ON m.Material_ID = b.Material_ID
331 GROUP BY g.Genre_ID, g.Name
332 ORDER BY Total_Borrowed DESC;
```

The 'Result Grid' shows the following data:

Genre_ID	Genre_Name	Total_Borrowed
1	General Fiction	22
3	Science Fiction & Fantasy	6
4	Horror & Suspense	5
2	Mystery & Thriller	3
6	Classics	3
7	Historical Fiction	1

The 'Output' pane at the bottom shows two log entries:

#	Time	Action	Message	Duration / Fetch
34	21:19:55	SELECT COUNT(*) AS materials_count FROM (SELECT Material_ID, COUNT(*) AS Total_Borrowed FROM Borrow WHERE Borrow_Date >= '2020-09-01' AND Borrow_Date <= '2020-10-31'	0.000 sec / 0.000 sec
35	21:21:49	SELECT g.Genre_ID, g.Name AS Genre_Name, COUNT(*) AS Total_Borrowed FROM Material m JOIN Genre g ON m.Genre_ID = g.Genre_ID JOIN Borrow b ON m.Material_ID = b.Material_ID GROUP BY g.Genre_ID, g.Name ORDER BY Total_Borrowed DESC;	6 row(s) returned	0.000 sec / 0.000 sec

7. How many materials had been borrowed from 09/2020-10/2020?

```
SELECT A.Author_ID, A.Name AS Author_Name, M.Title, COUNT(*) AS
      Materials_Borrowed_From_09_2020_10_2020
   FROM Borrow AS B
  JOIN Material AS M ON B.Material_ID = M.Material_ID
  JOIN Authorship ON M.Material_ID = Authorship.Material_ID
  JOIN Author AS A ON Authorship.Author_ID = A.Author_ID
 WHERE B.Borrow_Date >= '2020-09-01' AND B.Borrow_Date <= '2020-10-31'
 GROUP BY A.Author_ID, A.Name, M.Title;
```

Result:

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree, which includes 'skamma_assignment', 'g01474822', and the current database 'library_management_system_g01474822'. The central pane contains a SQL editor window titled 'SQL File 3' with the following query:

```
333
334 #7. How many materials had been borrowed from 09/2020-10/2020?
335
336 • SELECT A.Author_ID, A.Name AS Author_Name, M.Title, COUNT(*) AS Materials_Borrowed_From_09_2020_10_2020
337 FROM Borrow AS B
338 JOIN Material AS M ON B.Material_ID = M.Material_ID
339 JOIN Authorship ON M.Material_ID = Authorship.Material_ID
340 JOIN Author AS A ON Authorship.Author_ID = A.Author_ID
341 WHERE B.Borrow_Date >= '2020-09-01' AND B.Borrow_Date <= '2020-10-31'
342 GROUP BY A.Author_ID, A.Name, M.Title;
```

The 'Result Grid' below the query shows one row of data:

Author_ID	Author_Name	Title	Materials_Borrowed_From_09_2020_10_2020
3	George Orwell	The Da Vinci Code	1

The 'Output' pane at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
35	21:21:49	SELECT g.Genre_ID, g.Name AS Genre_Name, COUNT(*) AS Total_...	6 row(s) returned	0.000 sec / 0.000 sec
36	21:23:55	SELECT A.Author_ID, A.Name AS Author_Name, M.Title, COUNT(*) A...	1 row(s) returned	0.000 sec / 0.000 sec

8. How do you update the “Harry Potter and the Philosopher's Stone” when it is returned on 04/01/2023?

UPDATE Borrow

SET Return_Date = '2023-04-01'

WHERE Material_ID = (SELECT Material_ID FROM Material WHERE Title =
'Harry Potter and the Philosopher"s Stone')

AND Return_Date IS NULL;

```
SELECT B.Borrow_ID, M.Title, B.Return_Date
FROM Borrow AS B
JOIN Material AS M ON B.Material_ID = M.Material_ID
WHERE M.Title = 'Harry Potter and the Philosopher"s Stone';
```

Result:

```
37 21:35:57 UPDATE Borrow SET Return_Date = '2023-04-01' WHERE Material_I... 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.016 sec
```

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

The screenshot shows the MySQL Workbench interface. The left sidebar displays the Navigator with SCHEMAS, showing databases like akamma_assignment, g01474822, and library_management_system_g01474822. The main area has a SQL File 3 tab open with the following query:

```
349 AND Return_Date IS NULL;
350
351 • SELECT B.Borrow_ID, M.Title, B.Return_Date
352 FROM Borrow AS B
353 JOIN Material AS M ON B.Material_ID = M.Material_ID
354 WHERE M.Title = 'Harry Potter and the Philosopher''s Stone';
355
356
357
358
```

The Result Grid shows one row of data:

Borrow_ID	Title	Return_Date
20	Harry Potter and the Philosopher's Stone	2023-04-01

The Output pane shows two log entries:

#	Time	Action	Message	Duration / Fetch
37	21:35:57	UPDATE Borrow SET Return_Date = '2023-04-01' WHERE Material_I...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
38	21:37:16	SELECT B.Borrow_ID, M.Title, B.Return_Date FROM Borrow AS B JO...	1 row(s) returned	0.000 sec / 0.000 sec

9. How do you delete the member Emily Miller and all her related records from the database?

#DELETING FROM BORROW TABLE

DELETE FROM Borrow

```
WHERE Member_ID = ( SELECT Member_ID FROM Member WHERE Name =
    'Emily Miller'
);
```

Result:

```
39 21:43:01 DELETE FROM Borrow WHERE Member_ID = ( SELECT Member_ID ... 3 row(s) affected 0.000 sec
```

SELECT * from BORROW WHERE MEMBER_ID = 5;

Result:

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

The screenshot shows the MySQL Workbench interface. In the Navigator pane, the schema 'library_management_system_g01474822' is selected. A SQL editor tab named 'SQL File 3' contains the following code:

```
355 #9. How do you delete the member Emily Miller and all her related records from the database?
356
357 #DELETING FROM BORROW TABLE
358
359
360 • DELETE FROM Borrow
361 WHERE Member_ID = ( SELECT Member_ID FROM Member WHERE Name = 'Emily Miller'
362 );
363
364 • SELECT * from BORROW WHERE MEMBER_ID = 5;
```

The Result Grid shows a single row of data from the BORROW table:

Borrow_ID	Borrow_Date	Due_Date	Return_Date	Material_ID	Member_ID	Staff_ID
NULL	NULL	NULL	NULL	NULL	NULL	NULL

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
39	21:43:01	DELETE FROM Borrow WHERE Member_ID = (SELECT Member_ID ...	3 row(s) affected	0.000 sec
40	21:43:34	SELECT * from BORROW WHERE MEMBER_ID = 5 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

#DELETING FROM MEMBER TABLE

DELETE FROM Member

WHERE Name = 'Emily Miller';

41 21:44:42 DELETE FROM Member WHERE Name = 'Emily Miller' Error Code: 1175. You are using safe update mode and you tried to up... 0.000 sec

In MySQL, a feature called safe update mode can be kept on or off. In big databases the safe update mode is on so in case of any mistaken deletion MySQL will not allow it to happen and will come as an error shown above.

To Turn off the safe update mode the code required is displayed below, also after deletion of the data it is preferred to have the safe update mode back on for future work.

SET SQL_SAFE_UPDATES = 0;

42 21:45:17 SET SQL_SAFE_UPDATES = 0 0 row(s) affected 0.000 sec

43 21:45:47 DELETE FROM Member WHERE Name = 'Emily Miller' 1 row(s) affected 0.000 sec

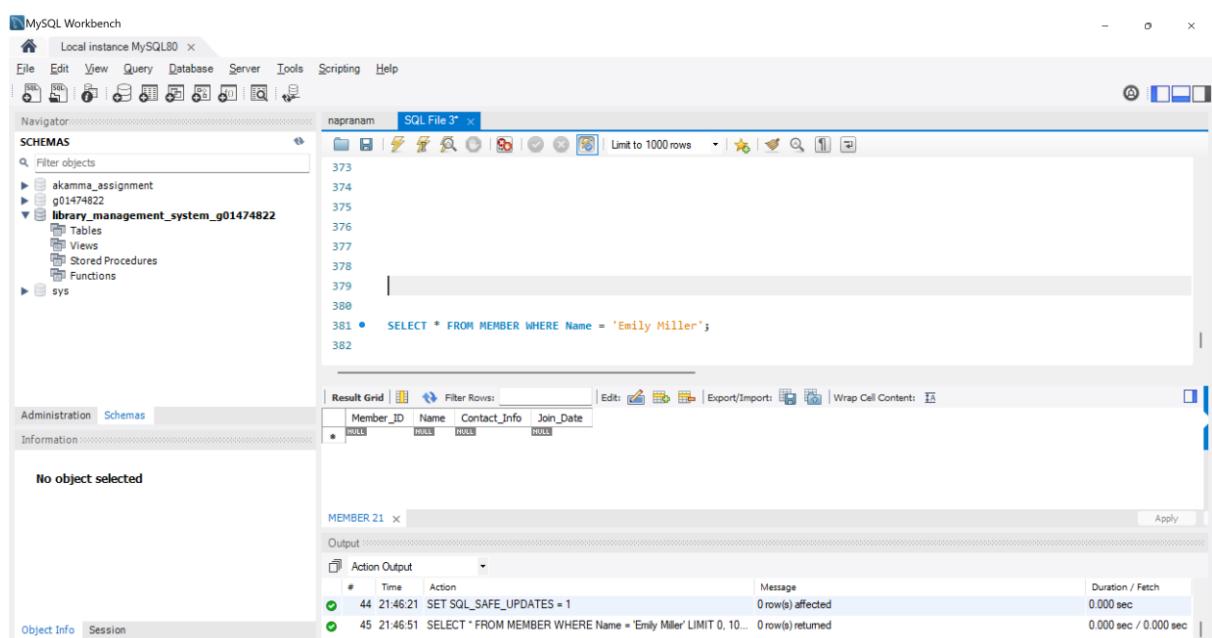
SET SQL_SAFE_UPDATES = 1;

44 21:46:21 SET SQL_SAFE_UPDATES = 1 0 row(s) affected 0.000 sec

SELECT * FROM MEMBER WHERE Name = 'Emily Miller';

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822



10. How do you add the following material to the database? Title: New book

Date: 2020-08-01

Catalog: E-Books

Genre: Mystery & Thriller

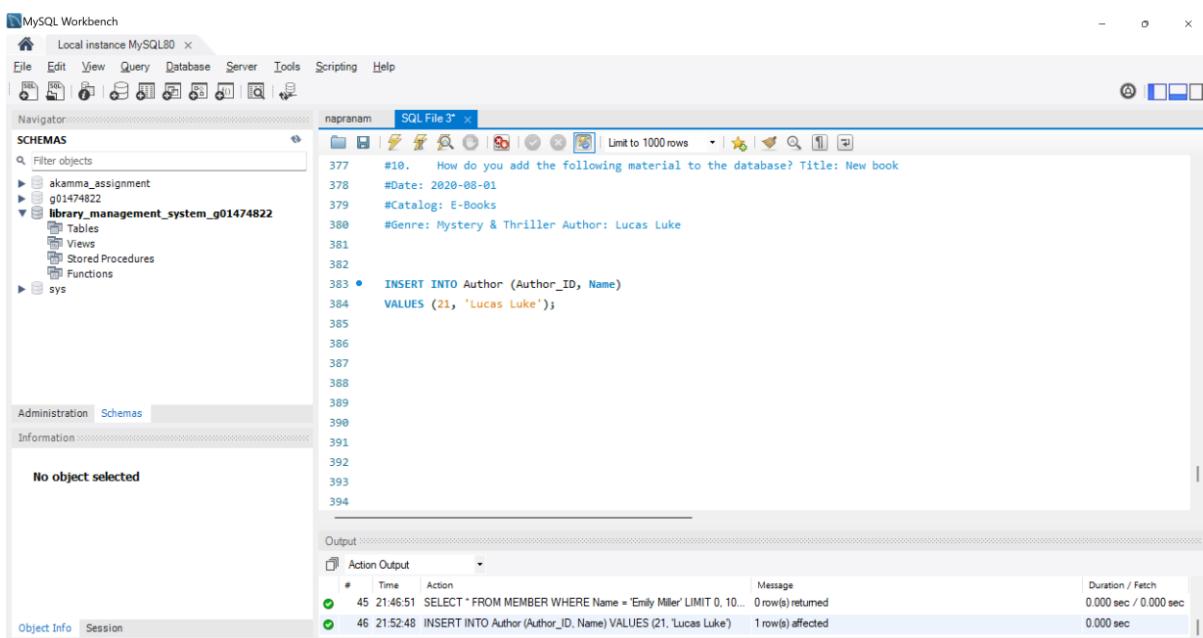
Author: Lucas Luke

```
INSERT INTO Author
    (Author_ID, Name)
VALUES (21, 'Lucas Luke');
```

Result:

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822



The screenshot shows the MySQL Workbench interface. In the Navigator pane, the schema 'library_management_system_g01474822' is selected. The SQL editor tab contains the following SQL code:

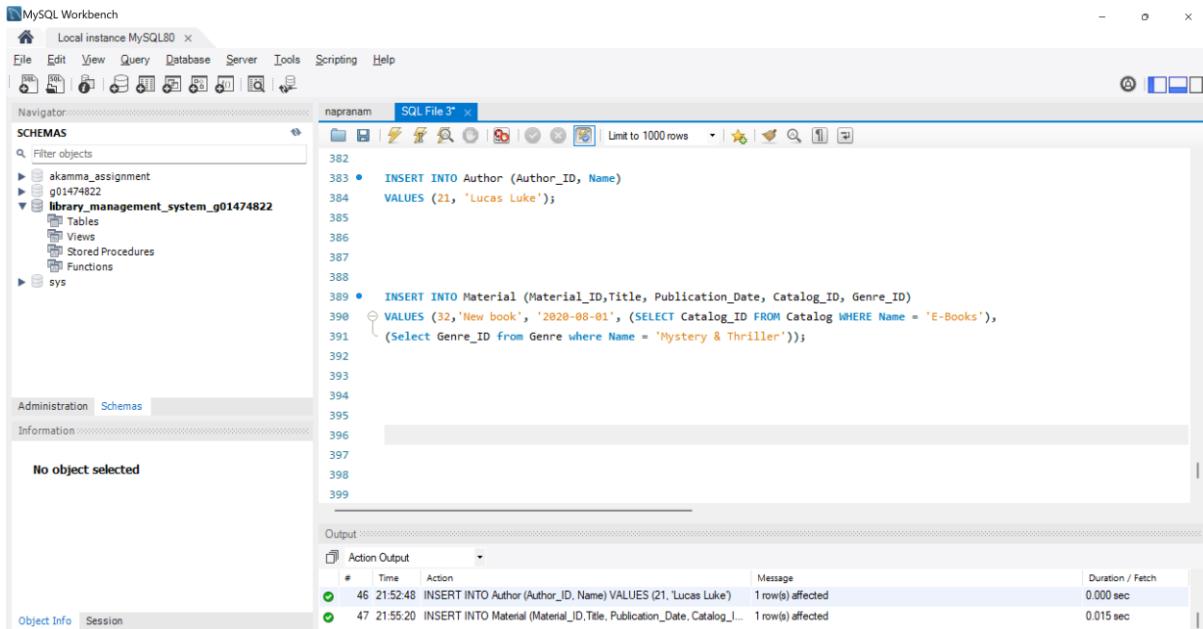
```
377 #10. How do you add the following material to the database? Title: New book
378 #Date: 2020-08-01
379 #Catalog: E-Books
380 #Genre: Mystery & Thriller Author: Lucas Luke
381
382
383 • INSERT INTO Author (Author_ID, Name)
384   VALUES (21, 'Lucas Luke')
385
386
387
388
389
390
391
392
393
394
```

The Output pane shows the results of the executed queries:

#	Time	Action	Message	Duration / Fetch
45	21:46:51	SELECT * FROM MEMBER WHERE Name = 'Emily Miller' LIMIT 0, 10...	0 row(s) returned	0.000 sec / 0.000 sec
46	21:52:48	INSERT INTO Author (Author_ID, Name) VALUES (21, 'Lucas Luke')	1 row(s) affected	0.000 sec

INSERT INTO Material (Material_ID, Title, Publication_Date, Catalog_ID, Genre_ID)
VALUES (32, 'New book', '2020-08-01', (SELECT Catalog_ID FROM Catalog
WHERE Name = 'E-Books'),
(Select Genre_ID from Genre where Name = 'Mystery & Thriller'));

Result:



The screenshot shows the MySQL Workbench interface. In the Navigator pane, the schema 'library_management_system_g01474822' is selected. The SQL editor tab contains the following SQL code:

```
382
383 • INSERT INTO Author (Author_ID, Name)
384   VALUES (21, 'Lucas Luke')
385
386
387
388
389 • INSERT INTO Material (Material_ID, Title, Publication_Date, Catalog_ID, Genre_ID)
390   VALUES (32, 'New book', '2020-08-01', (SELECT Catalog_ID FROM Catalog WHERE Name = 'E-Books'),
391   (Select Genre_ID from Genre where Name = 'Mystery & Thriller'));
392
393
394
395
396
397
398
399
```

The Output pane shows the results of the executed queries:

#	Time	Action	Message	Duration / Fetch
46	21:52:48	INSERT INTO Author (Author_ID, Name) VALUES (21, 'Lucas Luke')	1 row(s) affected	0.000 sec
47	21:55:20	INSERT INTO Material (Material_ID, Title, Publication_Date, Catalog_ID...	1 row(s) affected	0.015 sec

INSERT INTO Authorship (Authorship_ID, Material_ID, Author_ID)
VALUES (35, (SELECT Material_ID FROM Material WHERE Title = 'New book'),
(SELECT Author_ID FROM Author WHERE Name = 'Lucas Luke'));

Result:

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** library_management_system_g01474822
- Tables:** Author, Material, Authorship
- SQL File 3:** Contains the following SQL code:

```
381
382
383 • INSERT INTO Author (Author_ID, Name)
384     VALUES (21, 'Lucas Luke');
385
386
387
388
389 • INSERT INTO Material (Material_ID, Title, Publication_Date, Catalog_ID, Genre_ID)
390     VALUES (32, 'New book', '2020-08-01', (SELECT Catalog_ID FROM Catalog WHERE Name = 'E-Books'),
391             (Select Genre_ID from Genre where Name = 'Mystery & Thriller'));
392
393
394
395 • INSERT INTO Authorship (Authorship_ID, Material_ID, Author_ID)
396     VALUES (35, (SELECT Material_ID FROM Material WHERE Title = 'New book'),
397             (SELECT Author_ID FROM Author WHERE Name = 'Lucas Luke')));
398
```

Action Output:

#	Time	Action	Message	Duration / Fetch
47	21:59:20	INSERT INTO Material (Material_ID, Title, Publication_Date, Catalog_ID, Genre_ID)	1 row(s) affected	0.015 sec
48	21:59:34	INSERT INTO Authorship (Authorship_ID, Material_ID, Author_ID)	1 row(s) affected	0.016 sec

SELECT * FROM Material;

Result:

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** library_management_system_g01474822
- Tables:** Material
- SQL File 3:** Contains the following SQL code:

```
396     VALUES (35, (SELECT Material_ID FROM Material WHERE Title = 'New book'),
397             (SELECT Author_ID FROM Author WHERE Name = 'Lucas Luke'));
398
399 • SELECT * FROM Material;
400 • SELECT * FROM Author;
401 • SELECT * FROM Authorship;
402
403
404
405
```

Result Grid:

Material_ID	Title	Publication_Date	Catalog_ID	Genre_ID
29	The Count of Monte Cristo	1844-01-01	9	1
30	A Midsummer Nights Dream	1596-01-01	10	7
31	The Tricky Book	1888-01-01	10	7
32	New book	2020-08-01	3	2
*	NULL	NULL	NULL	NULL

Action Output:

#	Time	Action	Message	Duration / Fetch
49	22:10:16	select * from Material LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
50	22:10:58	SELECT * FROM Material LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec

Design of Extended Features

1. Alert staff about overdue materials on a daily-basis?

To alert the staff about overdue materials on a daily-basis, we can use stored procedures which is a chunk of code which can be executed multiple times without any modifications to the code.

A system that alerts staff members about overdue materials daily can be implemented by creating a stored procedure and scheduling its execution by creating an event. The code should display the output of BorrowID, Material.Title, Borrow.Due Date, Staff ID, Staff Email.

```
CREATE PROCEDURE Overdue_Materials_To_Alert_Staff()
BEGIN

    SELECT
        Borrow.Borrow_ID,
        Material.Title,
        Borrow.Due_Date,
        Staff.Staff_ID,
        Staff.Email
    FROM
        Borrow, Material, Staff
    WHERE
        Borrow.Material_ID = Material.Material_ID AND Material.Staff_ID =
        Staff.Staff_ID AND Borrow.Return_Date IS NULL
        AND Borrow.Due_Date < CURDATE();
END;
```

#EXECUTION PLAN FOR EVERY DAY (EVENT)

```
CREATE EVENT IF NOT EXISTS Daily_Alerts_Of_Overdue_Materials
ON SCHEDULE EVERY 1 DAY
COMMENT 'Notifying Staff Everyday about the overdue materials.'
DO
BEGIN
    CALL Overdue_Materials_To_Alert_Staff();
END
```

2024 Spring CS504 Project - LIBRARY MANAGEMENT SYSTEM

G01474822

- 2. Automatically deactivate the membership based on the member's overdue occurrence (\geq three times). And reactivate the membership once the member pays the overdue fee.**

As mentioned in the previous question, stored procedures is a chunk of code which can be written once and stored into a stored procedure and can be excused when needed with just calling the procedure.

In the code below, the membership is being managed based on the overdue occurrence. If the member has more than three overdue occurrences his membership status will be 'Inactive' and if it is lesser than 3 it is considered 'Active'. With the help of stored procedures, it can be automated.

```
CREATE PROCEDURE Membership_Overdue_Occurance_Management(IN  
    member_identification_number INT)  
BEGIN  
    DECLARE Number_of_Overdues INT;  
    SELECT COUNT(*)  
        INTO Number_of_Overdues  
        FROM Borrow  
        WHERE Member_ID = memberID AND Return_Date IS NULL AND Due_Date <  
            CURDATE();  
  
    UPDATE Member  
        SET Membership_Status = CASE  
            WHEN Number_of_Overdues >= 3 THEN 'Inactive'  
            ELSE 'Active'  
        END  
        WHERE Member_ID = member_identification_number;  
    END;
```

Learnings

In this project of Library Management System, I have gone through drawing the ER diagram and relation schema where I understood the importance of both while creating tables and querying the database. Using Structured Query Language (SQL) throughout the project and using complex concepts like stored procedures has helped me in understanding it well. By performing the project, I have understood that while creating our tables and insertions of data it should be done under utmost concentration as I had a few problems in the middle of my queries but thankfully I understood my error and problem and fixed it.

Conclusion

In conclusion, the library management system has been built and can be of great help for libraries without a database as it can give insights of how the library is looking in a digital form. With the use of all the queries, the required data can be pulled out of all tables as one and can be done for other types of information required. The extended features will be of great help as it is a real-time feature of alerting the staff member of overdue materials and the memberships being activated and deactivated automatically with the help of stored procedures is also a real time feature.

References

“Diagram Software and Flowchart Maker.” [Www.drawio.com](http://www.drawio.com), www.drawio.com/.