**Database Design**

This guide will help the student to create a database on the Library Management System. It will help to manage the below functionalities.

* Book
* Author
* Member (Library Member)
* Loan (Book Loan)
* Librarian
* Publisher

We will use MySQL as the DBMS to create the database and its related operations.

**1. Introduction to MySQL**

MySQL is an open-source relational database management system (RDBMS) that is widely used for various applications. It offers features like scalability, flexibility, performance, and reliability.MySQL's key features include:

* Scalability: Capable of handling large amounts of data and concurrent connections.
* Flexibility: Supports various data types and storage engines.
* Performance: Optimized for speed and efficiency.
* Reliability: Known for its stability and robustness.

**2. Installation of MySQL**

MySQL can be installed on various operating systems, including Windows, macOS, and Linux. Here are the general steps to install MySQL:

**Windows:**

* Download the MySQL installer from the official website.

<https://dev.mysql.com/downloads/installer/>

* Run the installer and follow the on-screen instructions.
* Choose the installation type (Typical, Complete, or Custom). Recommended Custom.
* Set a root password for the MySQL server.

**3. E-R Diagram (ERD)**

An Entity-Relationship Diagram (ERD) is a visual representation of the data model that shows the entities, attributes, relationships between entities, and cardinality. ERDs are commonly used in database design to help developers and stakeholders understand the structure and relationships within a database.

**Identify Entities**

* Start by identifying the main entities in your system. These are the objects or concepts about which you want to store data.
* Each entity should correspond to a table in your database.

**Define Attributes**

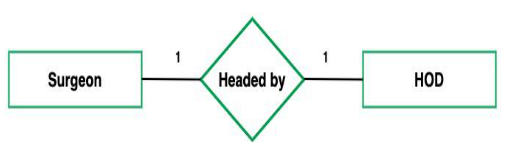
* For each entity, list the attributes (properties or fields) that describe it.
* These attributes will become columns in the corresponding database table.

**Identify Relationships**

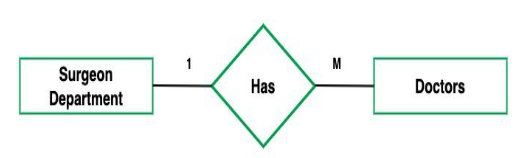
* Determine how entities are related to each other. There are three types of relationships: one-to-one (1:1), one-to-many (1:N), and many-to-many (N:M).
* Represent these relationships using lines connecting the entities.

Let’s see a few examples of relationships:

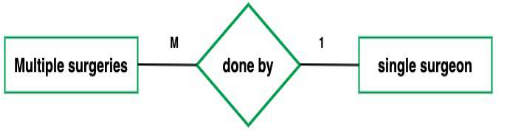
**One to One**



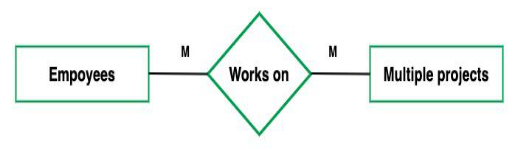
**One to Many**



**Many to One**



**Many to Many**



**Cardinality Notation**

Cardinality represents the number of times an entity of an entity set participates in a relationship set. Or we can say that the cardinality of a relationship is the number of tuples (rows) in a relationship.

* Use notation (such as Crow's Foot Notation or Chen Notation) to indicate the cardinality of each relationship.
* Cardinality describes how many instances of one entity are related to how many instances of another entity.
* Common notations include:
* One (1)
* Zero or one (0..1)
* Many (N)
* Zero or many (0..N)

**Optional:**

**Add Attributes and Constraints**

* Include additional information in your ERD, such as primary keys, foreign keys, and constraints (e.g., unique constraints).

**Create the Diagram**

* Use specialized diagramming software or tools (e.g., Lucidchart, draw.io, or even pen and paper) to create your ERD.

**Refine and Review:**

* Review your ERD with stakeholders and team members to ensure it accurately represents the data model and relationships. Make any necessary refinements.

Let’s identify the entities of the Library management system

Book

Author

Member (Library Member)

Loan (Book Loan)

Librarian

Publisher

\*\*\* Now let’s identify the attributes and relationships of each entity for the Student Management System.

**Book**

**Attributes**:

BookID (Primary Key)

Title

Author

ISBN

Genre

...

**Relationships**:

A Book can have multiple Reservations (One-to-Many)

**Author**

**Attributes**:

AuthorID (Primary Key)

FirstName

LastName

BirthDate

Nationality

...

**Relationships**:

An Author can have multiple Books (One-to-Many)

**Member (Library Member)**

* **Attributes:**

MemberID (Primary Key)

FirstName

LastName

Email

Phone

Address

**Relationships:**

A Member can have multiple Reservations (One-to-Many)

**BookLoan**

* **Attributes:**

LoanID (Primary Key)

BorrowerID (Foreign Key)

BookID (Foreign Key)

LoanDate

DueDate

ReturnDate

FineAmount

**Relationships**:

Each Book Loan is associated with one Borrower (Many-to-One)

Each Book Loan is associated with one Book (Many-to-One)

**LibrarianServices**

* **Attributes**:

ServiceID (Primary Key)

ServiceName

Description

Cost

PublisherID (Foreign Key)

* **Relationships**:

A Librarian Service belongs to one Publisher (Many-to-One).

**Publisher**

* **attributes**:

PublisherID (Primary Key)

PublisherName

PublisherAddress

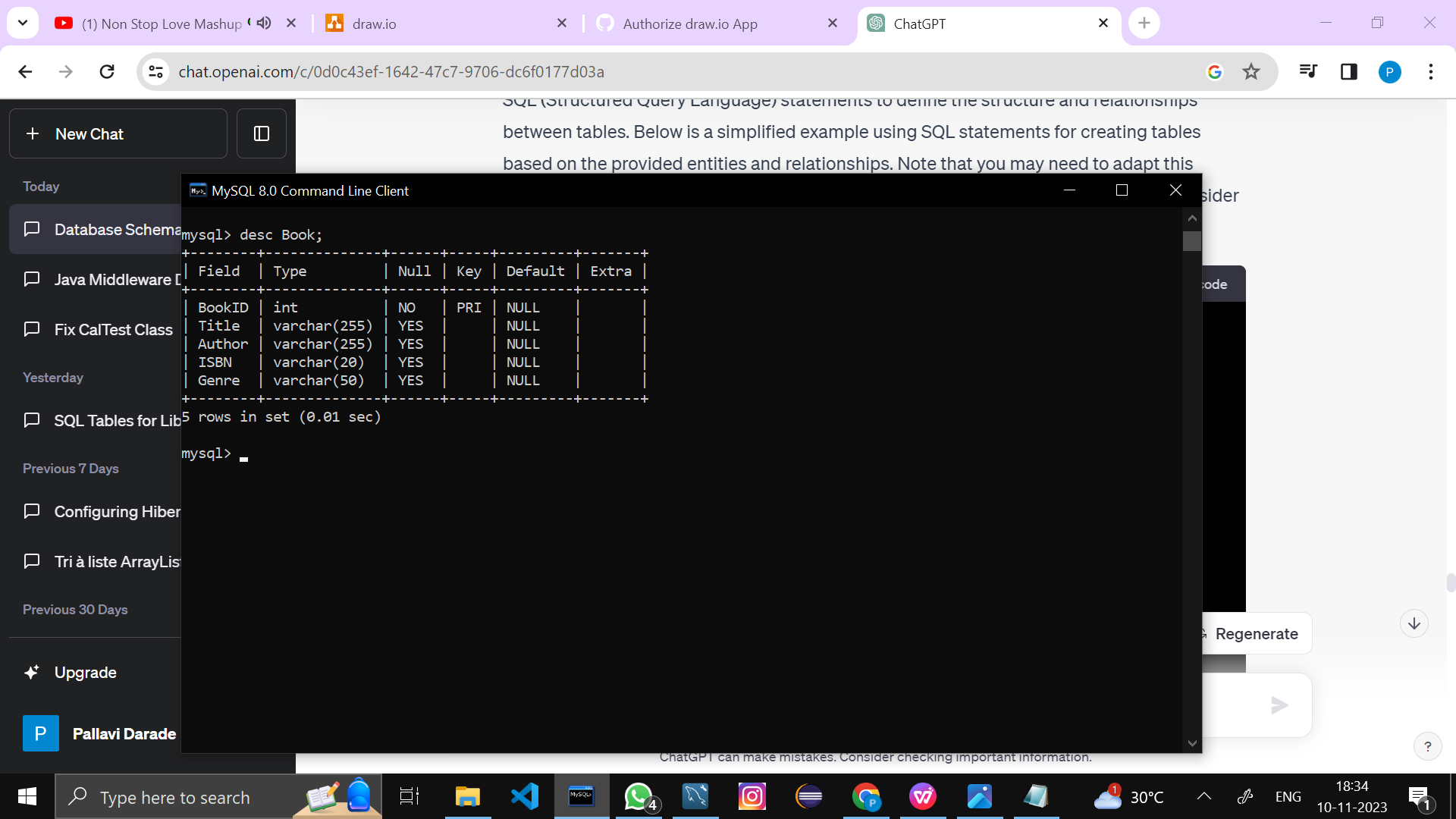
PublisherPhone

* **Relationships**:

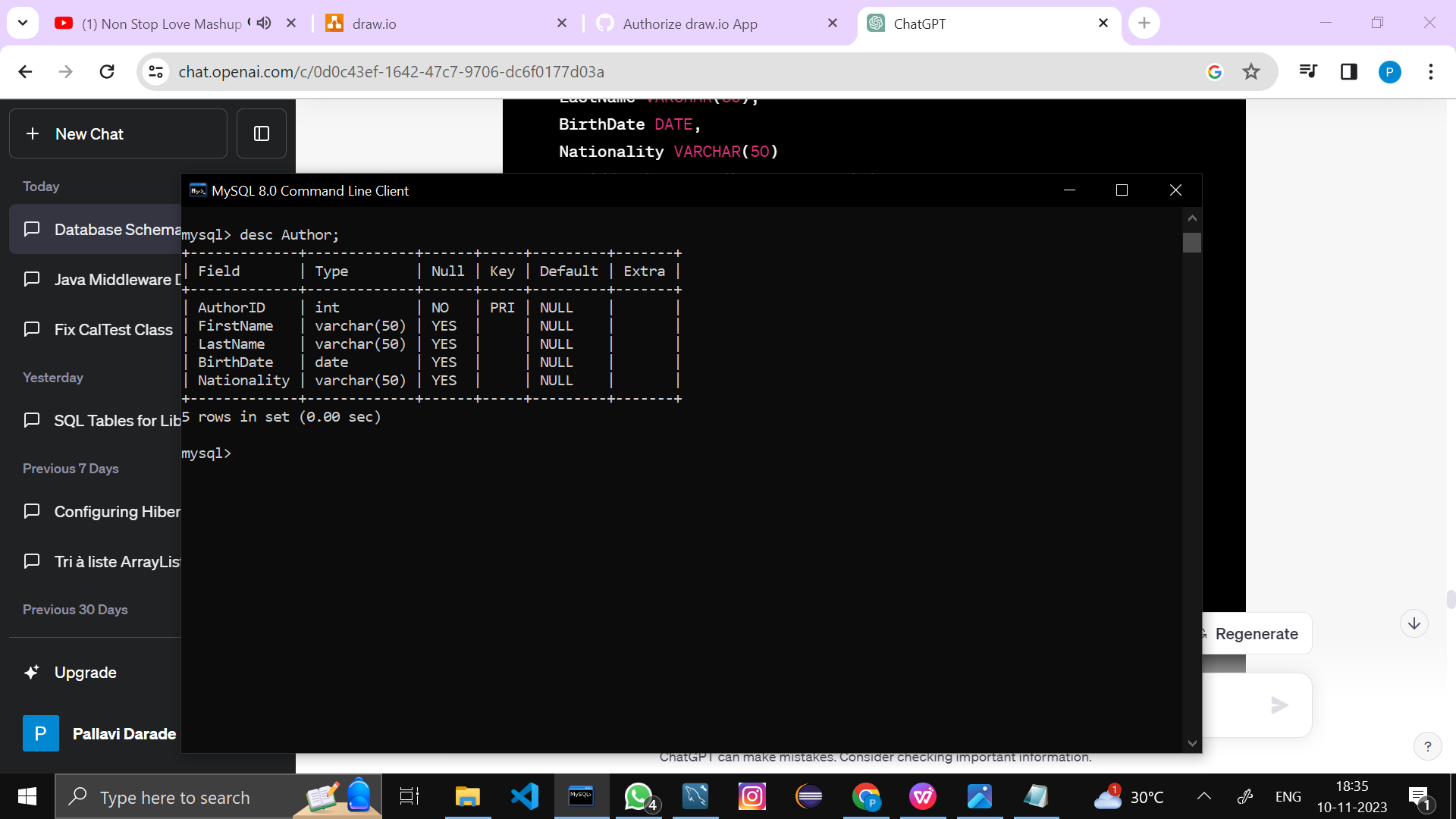
A Publisher can have multiple Librarian Services (One-to-Many)

**Table Structure**

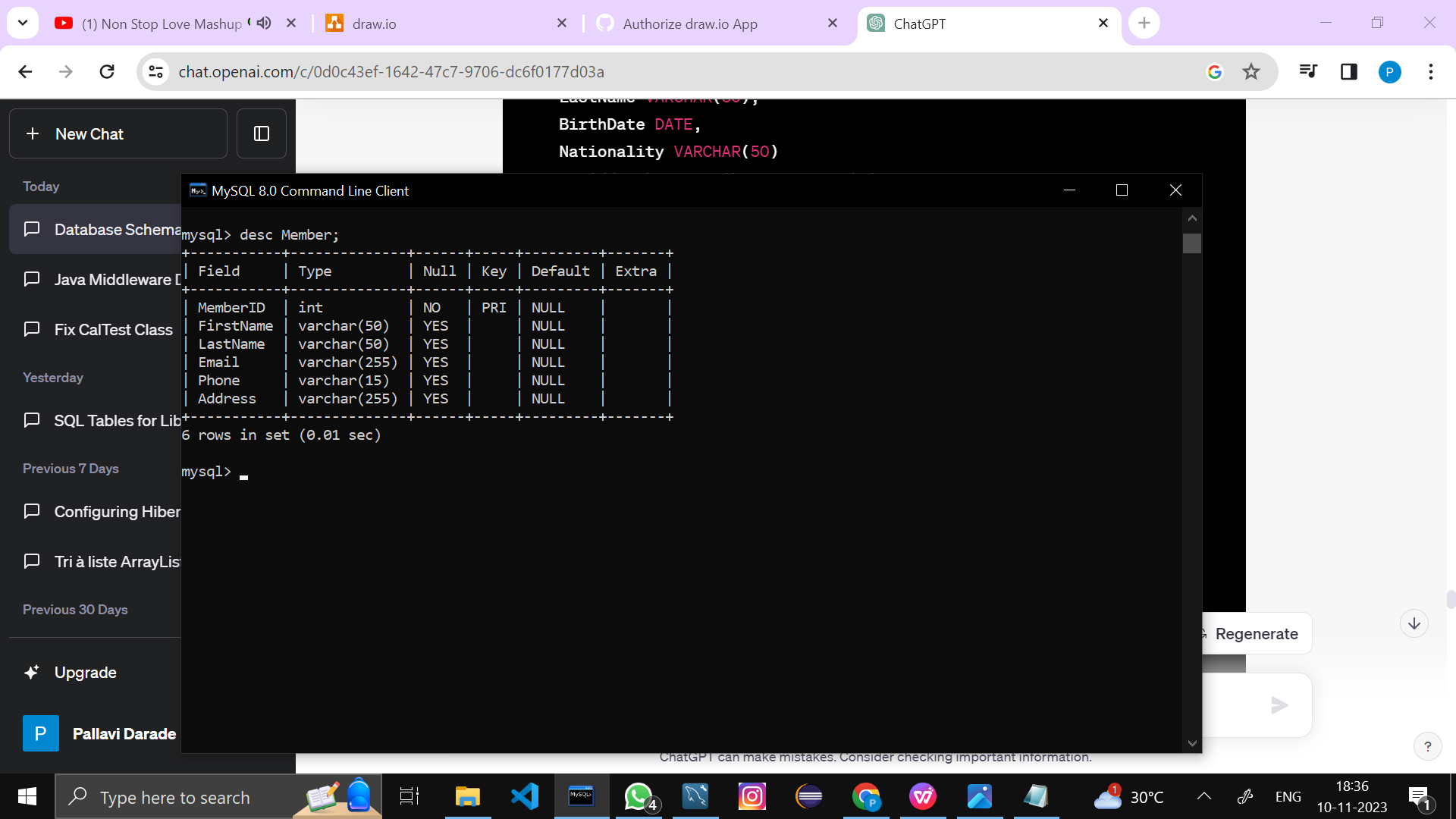
1. Book



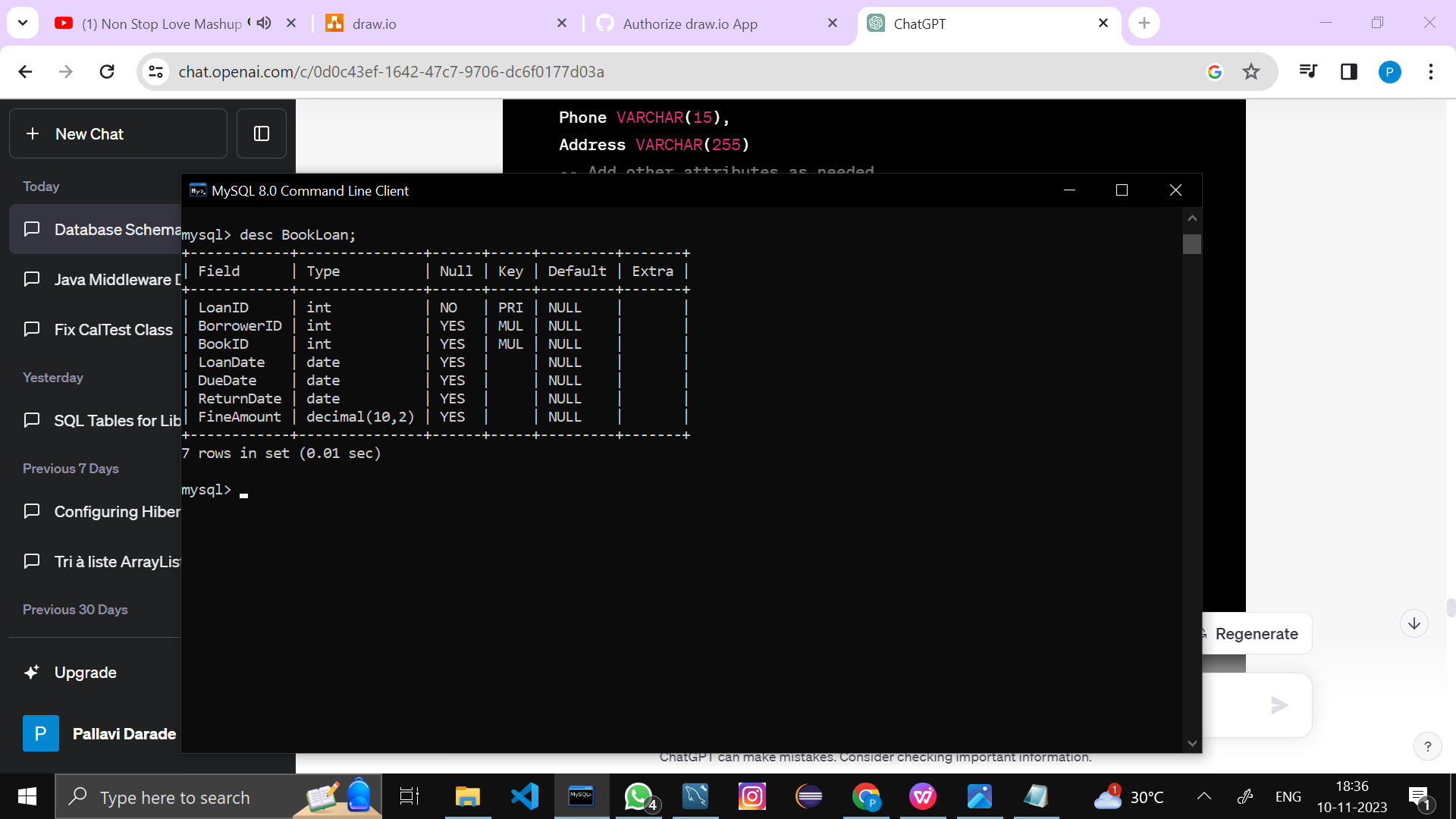
1. Author



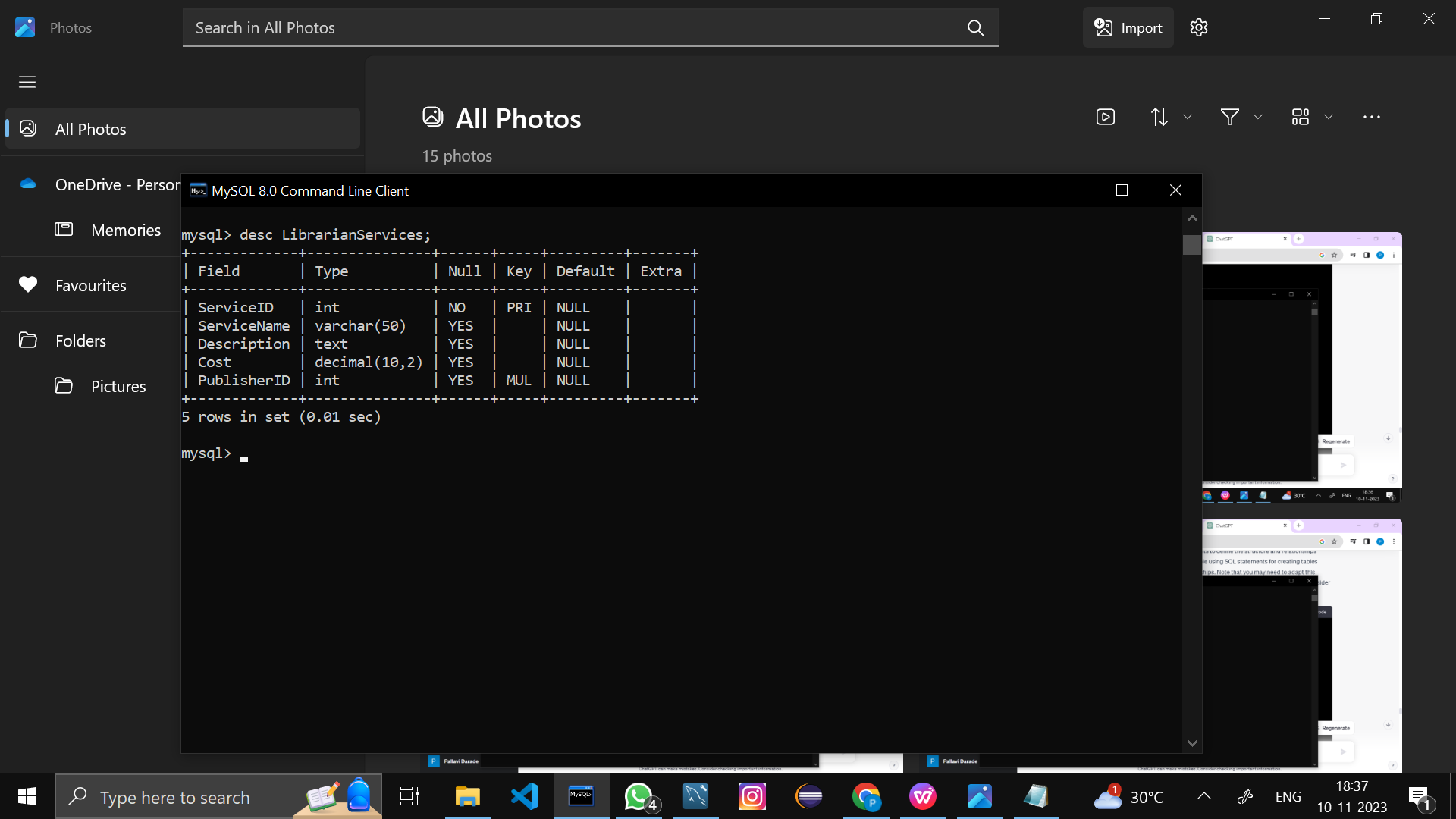
1. Member (Library Member)



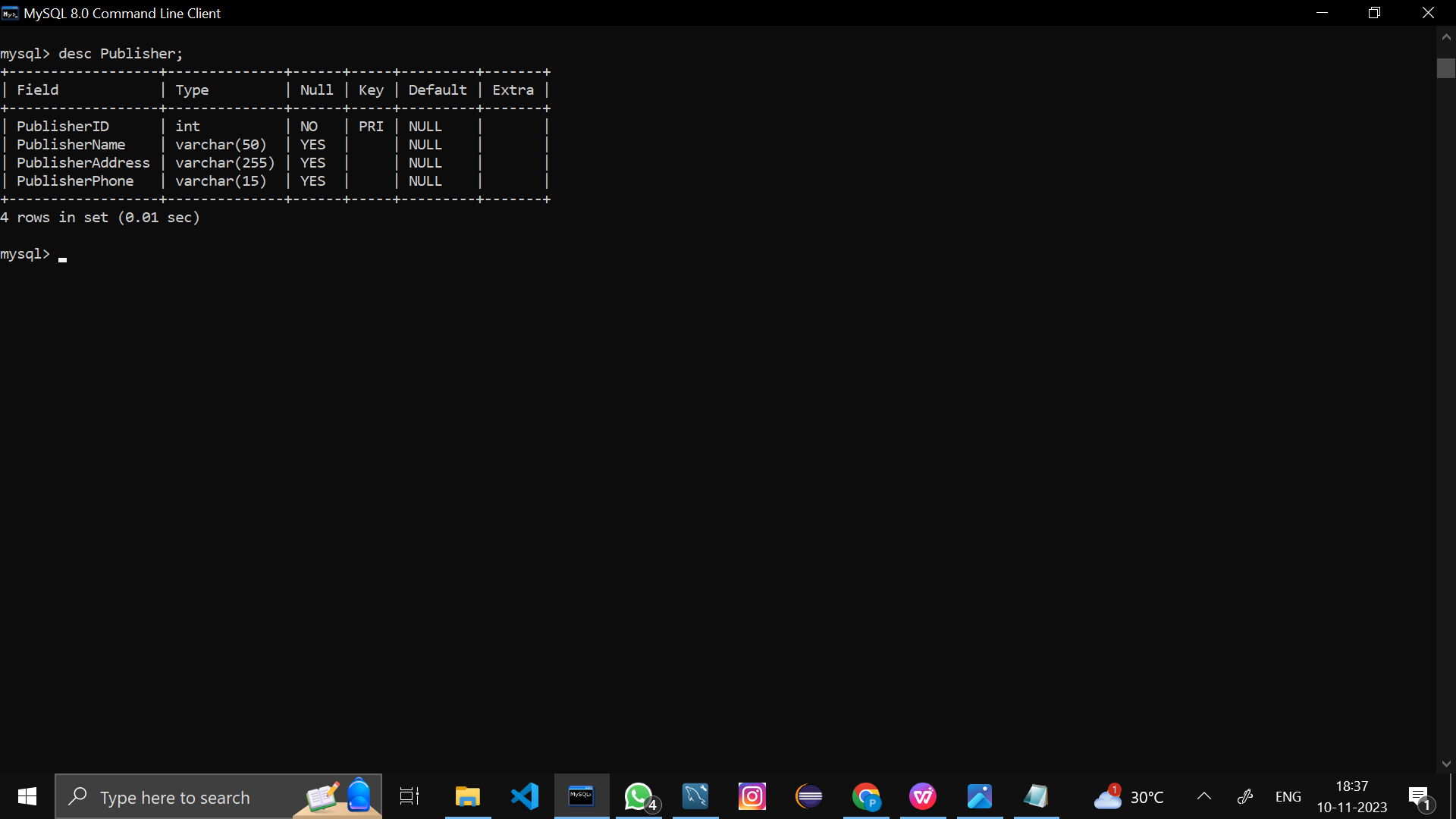
1. Loan (Book Loan)



1. Librarian

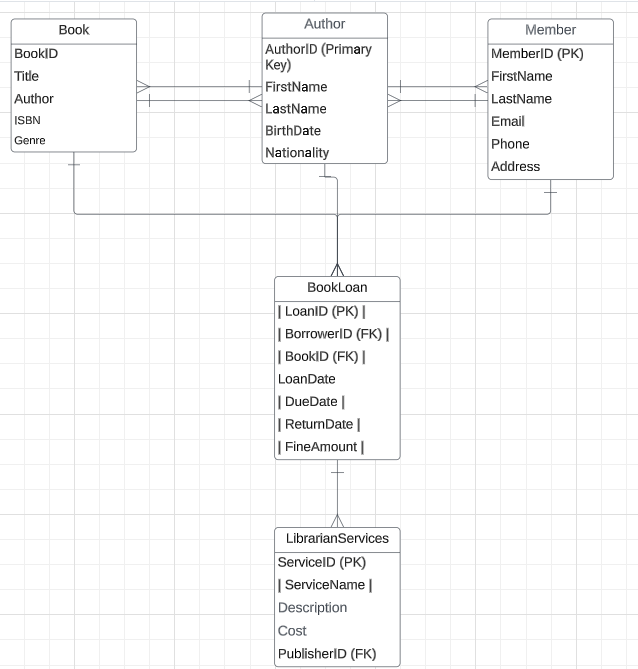


1. Publisher



Now, let’s create the ER diagram to visually represent the entities and relationships.

**ERD Diagram**

****

**In this ERD:**

Entities are represented by rectangles.

Attributes are listed within the rectangles.

Primary keys (PK) are indicated.

Foreign keys (FK) are shown where there are relationships.

Relationships are depicted with lines connecting the related entities.

**4. Creating a Database**

Using MySQL server, create a new database for your student management system. You can do this with SQL commands or through the graphical interface.

*CREATE DATABASE library\_management\_system;*

**5. Using a Database**

Before performing any operations on a database, you need to select it using the USE statement:

*USE library\_management\_system;*

1. **Creating the tables for each entity**

CREATE TABLE **Book** (

BookID INT PRIMARY KEY,

Title VARCHAR(255),

Author VARCHAR(255),

ISBN VARCHAR(20),

Genre VARCHAR(50) );

CREATE TABLE **Author** (

AuthorID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

BirthDate DATE,

Nationality VARCHAR(50) );

CREATE TABLE **Member** (

MemberID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(255),

Phone VARCHAR(15),

Address VARCHAR(255));

CREATE TABLE **Publisher** (

PublisherID INT PRIMARY KEY,

PublisherName VARCHAR(50),

PublisherAddress VARCHAR(255),

PublisherPhone VARCHAR(15) );

CREATE TABLE **LibrarianServices** (

ServiceID INT PRIMARY KEY,

ServiceName VARCHAR(50),

Description TEXT,

Cost DECIMAL(10, 2),

PublisherID INT,

FOREIGN KEY (PublisherID) REFERENCES Publisher(PublisherID) );

CREATE TABLE **BookLoan** (

LoanID INT PRIMARY KEY,

BorrowerID INT, BookID INT,

LoanDate DATE,

DueDate DATE,

ReturnDate DATE,

FineAmount DECIMAL(10, 2), FOREIGN KEY (BorrowerID) REFERENCES Member(MemberID),

FOREIGN KEY (BookID) REFERENCES Book(BookID) );

**7. Insert records**

Add data to your tables to work with. This step helps you test your database.

-- Insert records into Book table

INSERT INTO Book (BookID, Title, Author, ISBN, Genre)

VALUES

(1, 'The Great Gatsby', 'F. Scott Fitzgerald', '978-3-16-148410-0', 'Fiction'),

(2, 'To Kill a Mockingbird', 'Harper Lee', '978-0-06-112008-4', 'Fiction'),

(3, '1984', 'George Orwell', '978-0-452-28423-4', 'Dystopian'),

(4, 'The Catcher in the Rye', 'J.D. Salinger', '978-0-316-76948-0', 'Fiction'),

(5, 'Harry Potter and the Sorcerer\'s Stone', 'J.K. Rowling', '978-0-7475-3269-6', 'Fantasy');

-- Insert records into Author table

INSERT INTO Author (AuthorID, FirstName, LastName, BirthDate, Nationality)

VALUES

(1, 'F. Scott', 'Fitzgerald', '1896-09-24', 'American'),

(2, 'Harper', 'Lee', '1926-04-28', 'American'),

(3, 'George', 'Orwell', '1903-06-25', 'British'),

(4, 'J.D.', 'Salinger', '1919-01-01', 'American'),

(5, 'J.K.', 'Rowling', '1965-07-31', 'British');

-- Insert records into Member table

INSERT INTO Member (MemberID, FirstName, LastName, Email, Phone, Address)

VALUES

(1, 'John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Main St'),

(2, 'Jane', 'Smith', 'jane.smith@example.com', '987-654-3210', '456 Oak St'),

(3, 'Alice', 'Johnson', 'alice.j@example.com', '555-1234', '789 Pine St'),

(4, 'Bob', 'Williams', 'bob.w@example.com', '555-5678', '101 Maple St'),

(5, 'Eva', 'Jones', 'eva.j@example.com', '555-4321', '202 Cedar St');

-- Insert records into BookLoan table

INSERT INTO BookLoan (LoanID, BorrowerID, BookID, LoanDate, DueDate, ReturnDate, FineAmount)

VALUES

(1, 1, 1, '2023-01-01', '2023-01-15', '2023-01-10', 5.00),

(2, 2, 2, '2023-02-01', '2023-02-15', NULL, NULL),

(3, 3, 3, '2023-03-01', '2023-03-15', NULL, NULL),

(4, 4, 4, '2023-04-01', '2023-04-15', NULL, NULL),

(5, 5, 5, '2023-05-01', '2023-05-15', NULL, NULL);

-- Insert records into LibrarianServices table

INSERT INTO LibrarianServices (ServiceID, ServiceName, Description, Cost, PublisherID)

VALUES

(1, 'Library Card Registration', 'Register for a new library card', 2.00, 1),

(2, 'Book Check-In/Check-Out', 'Borrow or return a book', 0.50, 2),

(3, 'Late Fee Payment', 'Pay fines for overdue books', 1.00, 1),

(4, 'Book Recommendation', 'Get personalized book recommendations', 0.75, 2),

(5, 'Library Event Registration', 'Sign up for upcoming library events', 0.50, 1);

-- Insert records into Publisher table

INSERT INTO Publisher (PublisherID, PublisherName, PublisherAddress, PublisherPhone)

VALUES

(1, 'Random House', '123 Publishing St', '555-1234'),

(2, 'Penguin Books', '456 Literature Ave', '555-5678'),

(3, 'Houghton Mifflin', '789 Book Lane', '555-9876'),

(4, 'Scholastic', '101 Education Blvd', '555-5432'),

(5, 'HarperCollins', '202 Reading Road', '555-6789');

**8. Select records**

Write SQL queries to retrieve and manage data.

For example:

**Retrieve all Book:**

*Select \* FROM Book;*

**Retrieve specific Author**

SELECT Title, Author

FROM Book

WHERE Genre = 'Fiction';

**9. Update records**

Write SQL statements to update record(s) when needed. For example:

**Update the email address for the member with MemberID 1 in the Member table:**

*-- Update the email address for the member with MemberID 1*

*UPDATE Member*

*SET Email = 'new.email@example.com'*

*WHERE MemberID = 1;*