**Database Design**

<https://www.knowledgefactory.net/2022/06/java-spring-boot-mini-project-library-management-system-free-download.html>

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

* Admin
* Author
* Publisher
* Genre
* Book
* Users
* Book Fines

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**

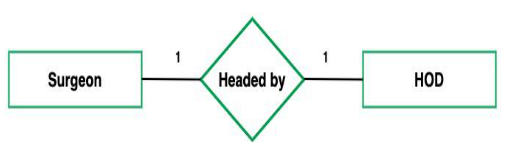
* For each entity, list the attributes (properties or fields) that describe it.
* These attributes will become columns in the corresponding database table.

**Identify Relationships**

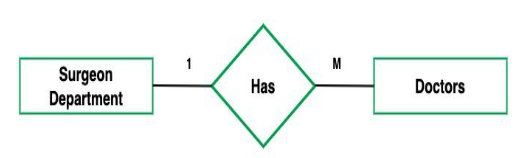
* Determine how entities are related to each other. There are three types of relationships: one-to-one (1:1), one-to-many (1:N), and many-to-many (N:M).
* Represent these relationships using lines connecting the entities.

Let’s see a few examples of relationships:

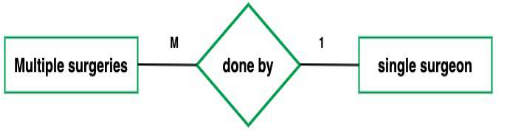
**One to One**



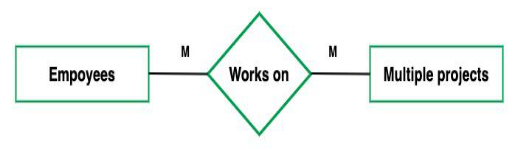
**One to Many**



**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.

* **Admin:**

**Attributes**

Id

Username

Password

* **Users**

**attributes**:

Id

Username

Password

* **Book**

**Attributes**:

id

title;

authorId;

publisherId;

genreIds;

quantity;

**Relationships**:

Multiple Book have a Authour (Many-to-One)

Multiple Book have a publisher (Many-to-One)

A Book can have multiple Genre (Many-to-Many)

* **Author**

**Attributes**:

ID (Primary Key)

Name

Biography

**Relationships**:

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

* **Publisher**
* **attributes**:

Id

Name

BookIds

* **Relationships**:

A Publisher can have multiple Books (One-to-Many)

* **Genre**
* **attributes**:

Id

Name

BookIds

* **Relationships**:

Multiple Publisher can have multiple Books (Many-to-Many)

* **BookFines**
* **attributes**:

Id

amount

paid

* **Relationships**:

A book can have a Book fine (One-to-One)

* **Users**
* **attributes**:

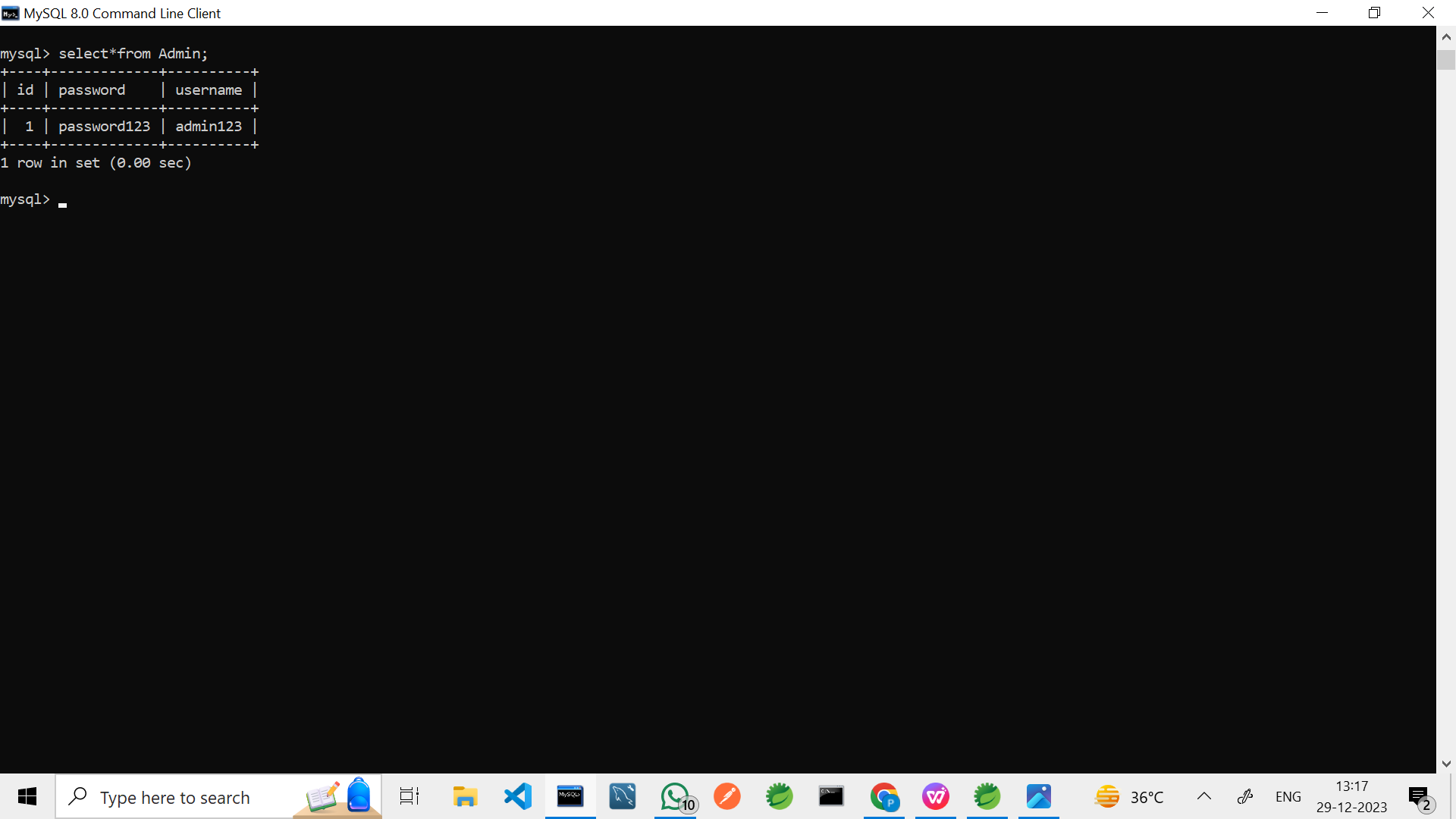
Id

Username

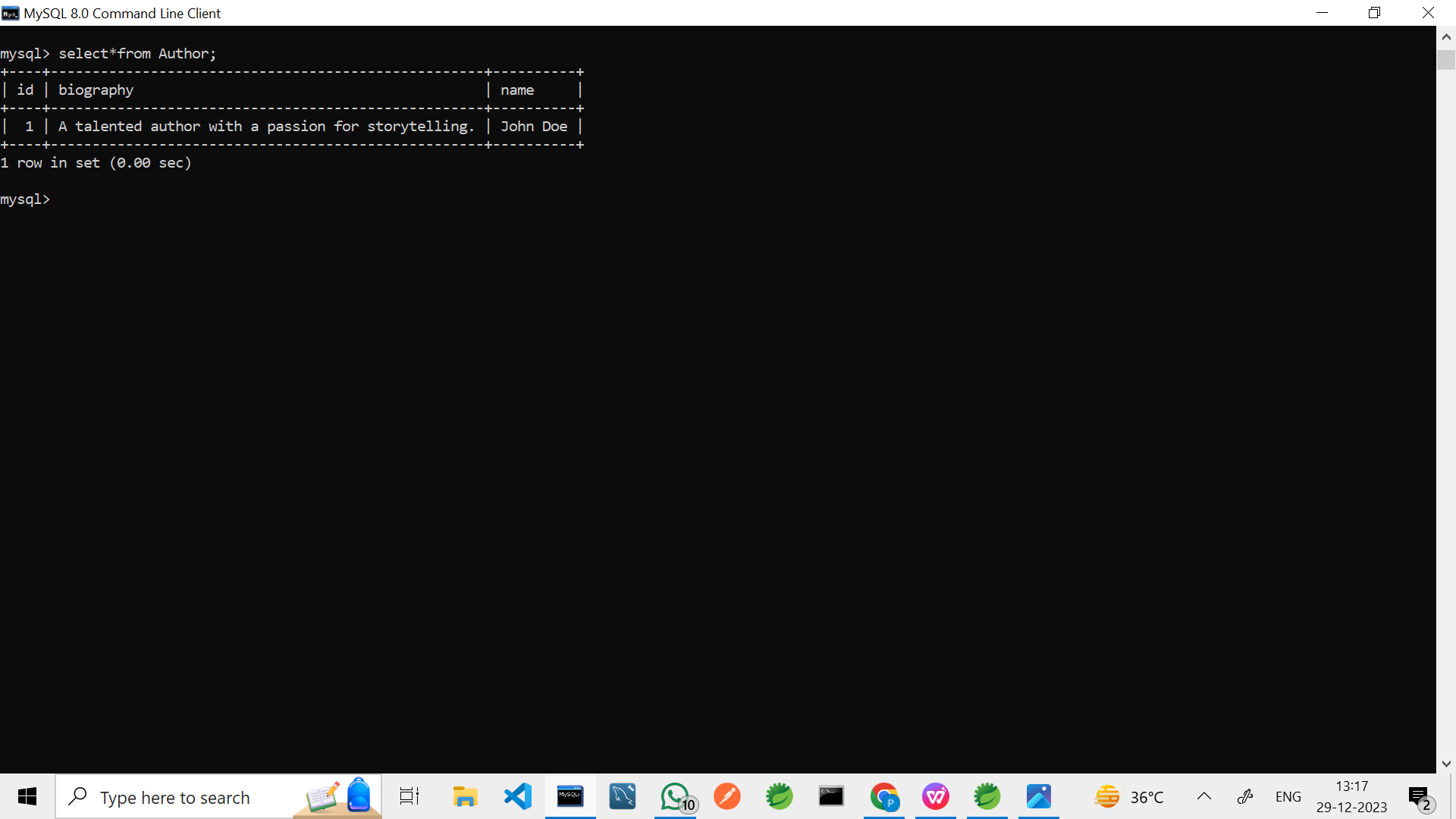
Password

* **Table Structure**

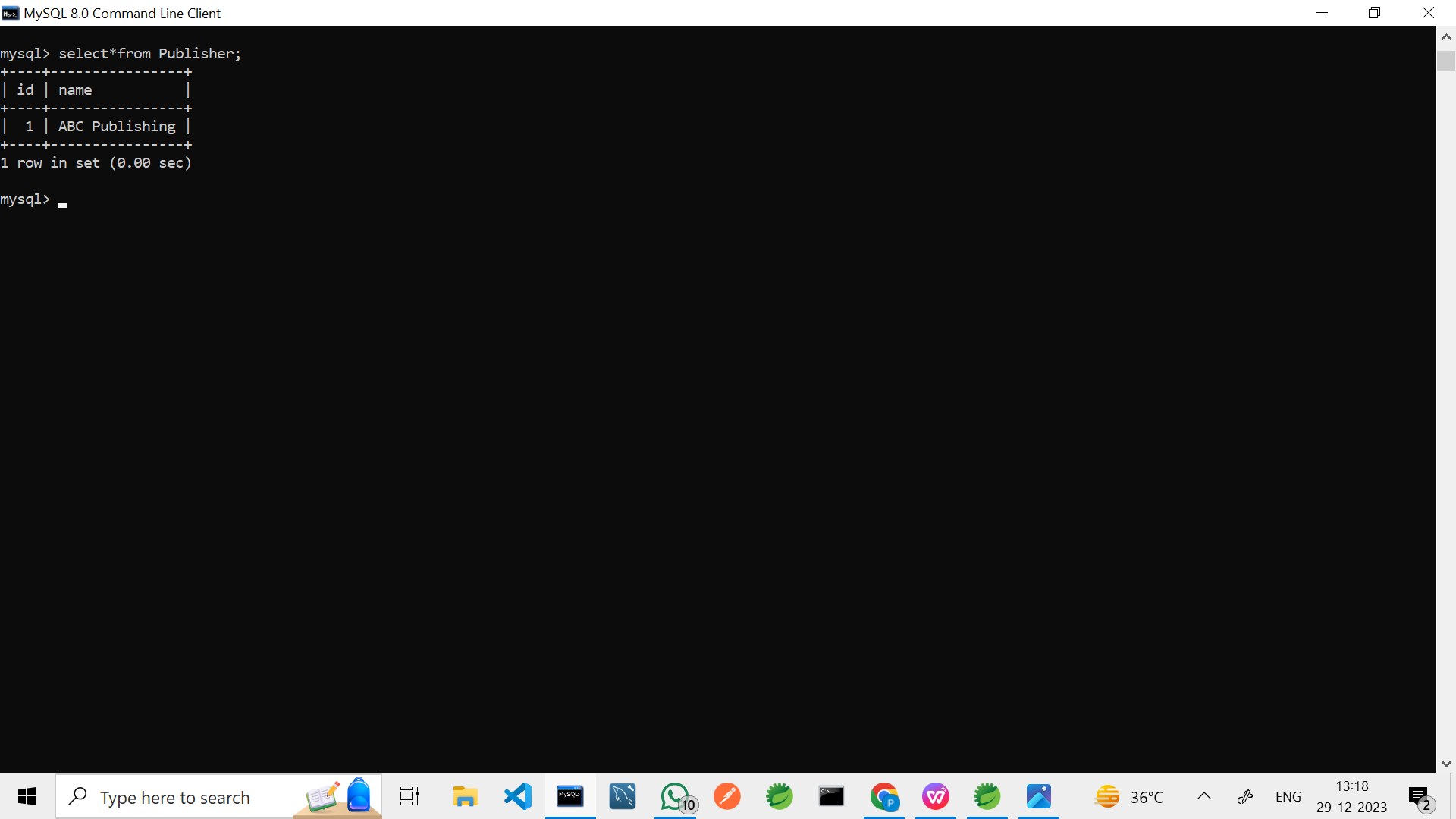
1. Admin



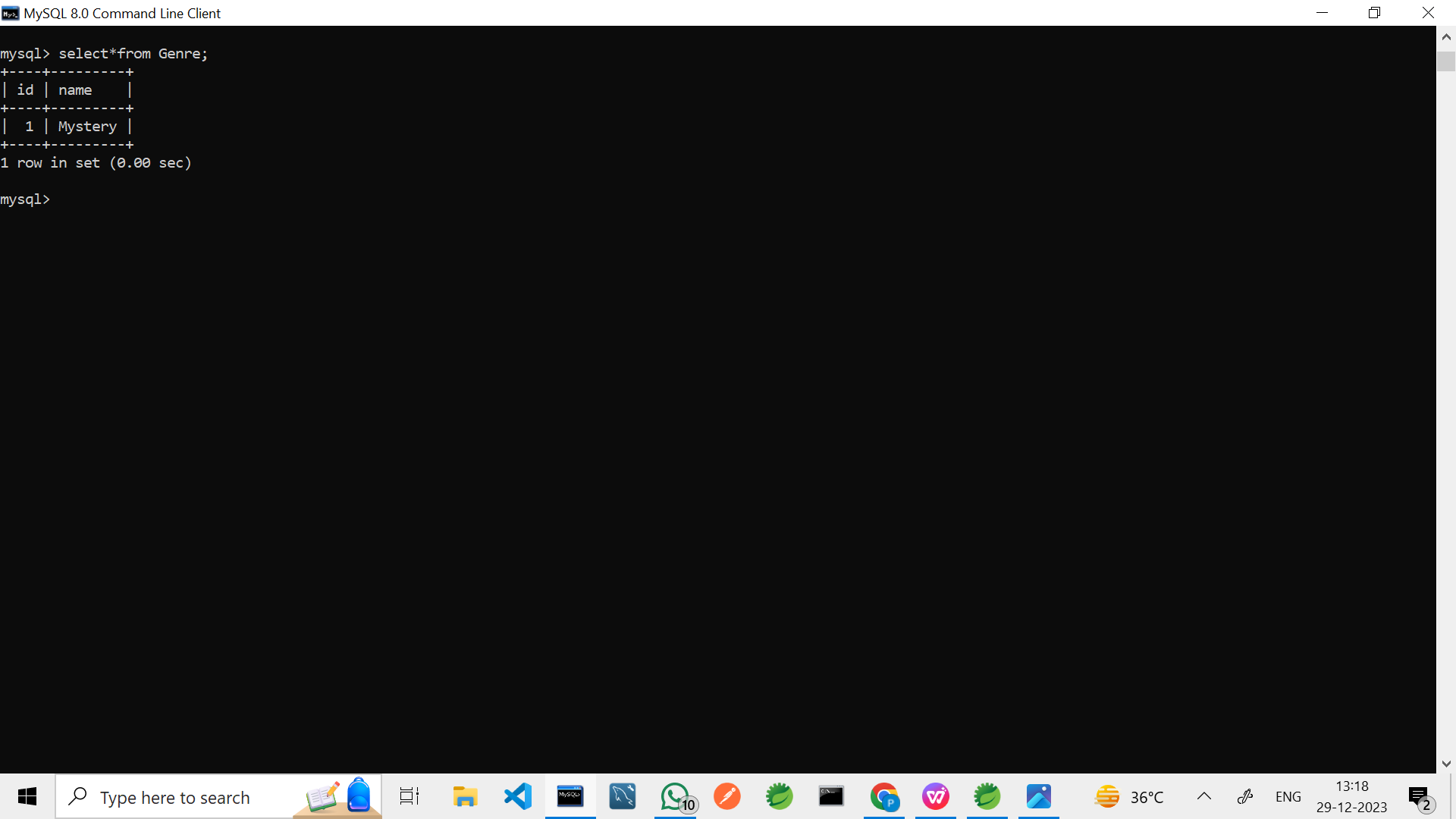
1. Author



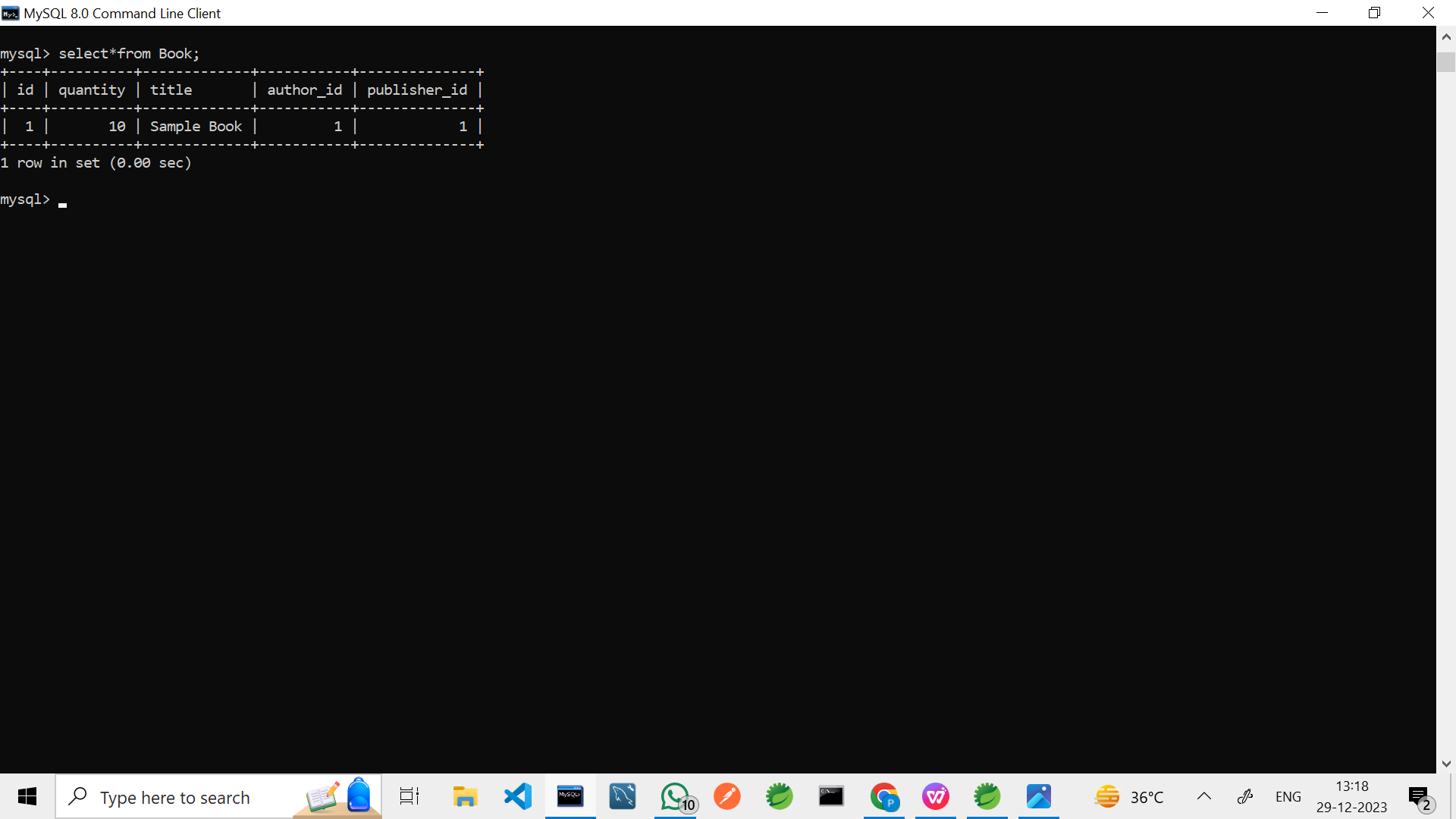
1. Publisher



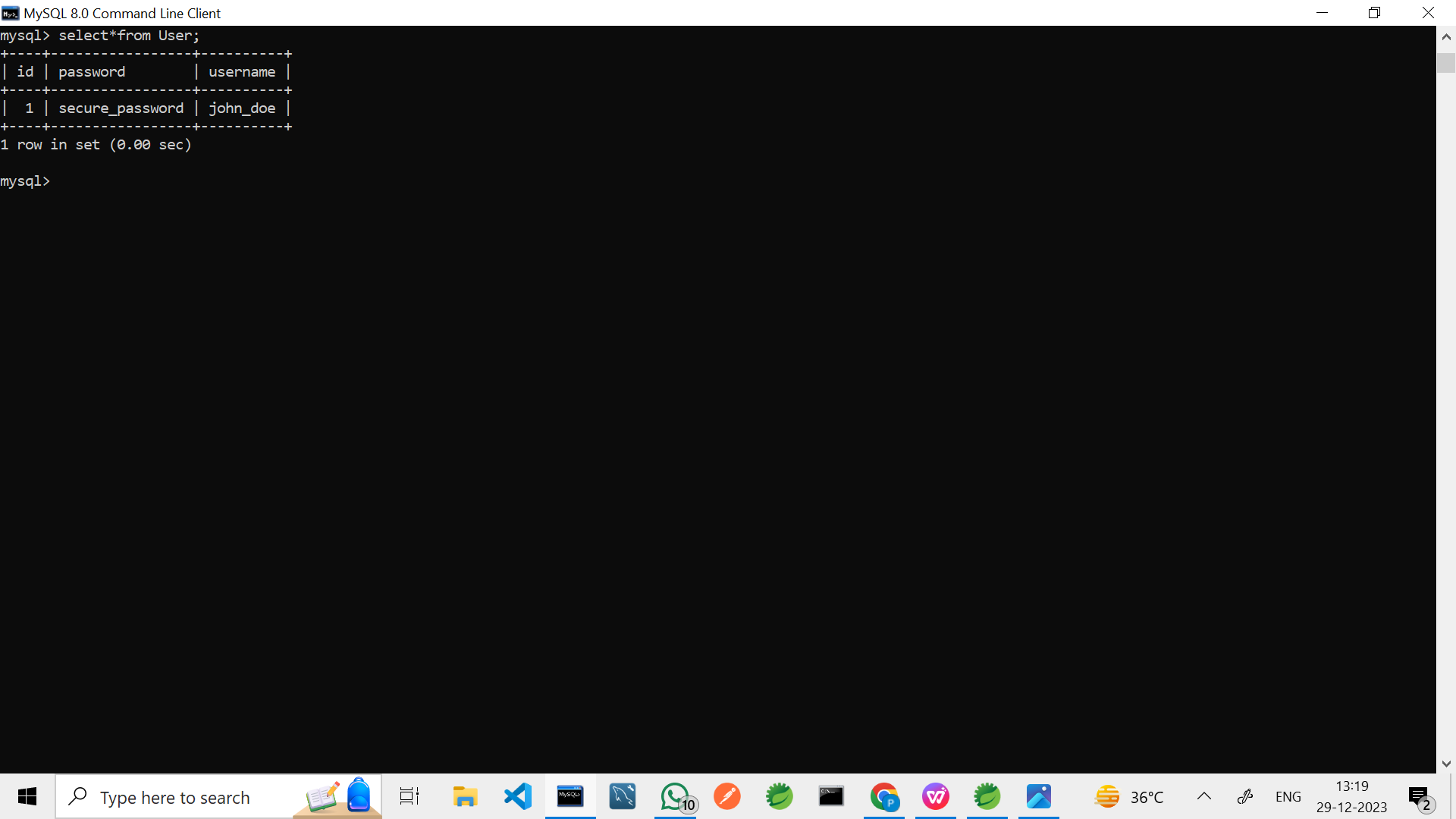
1. Genre



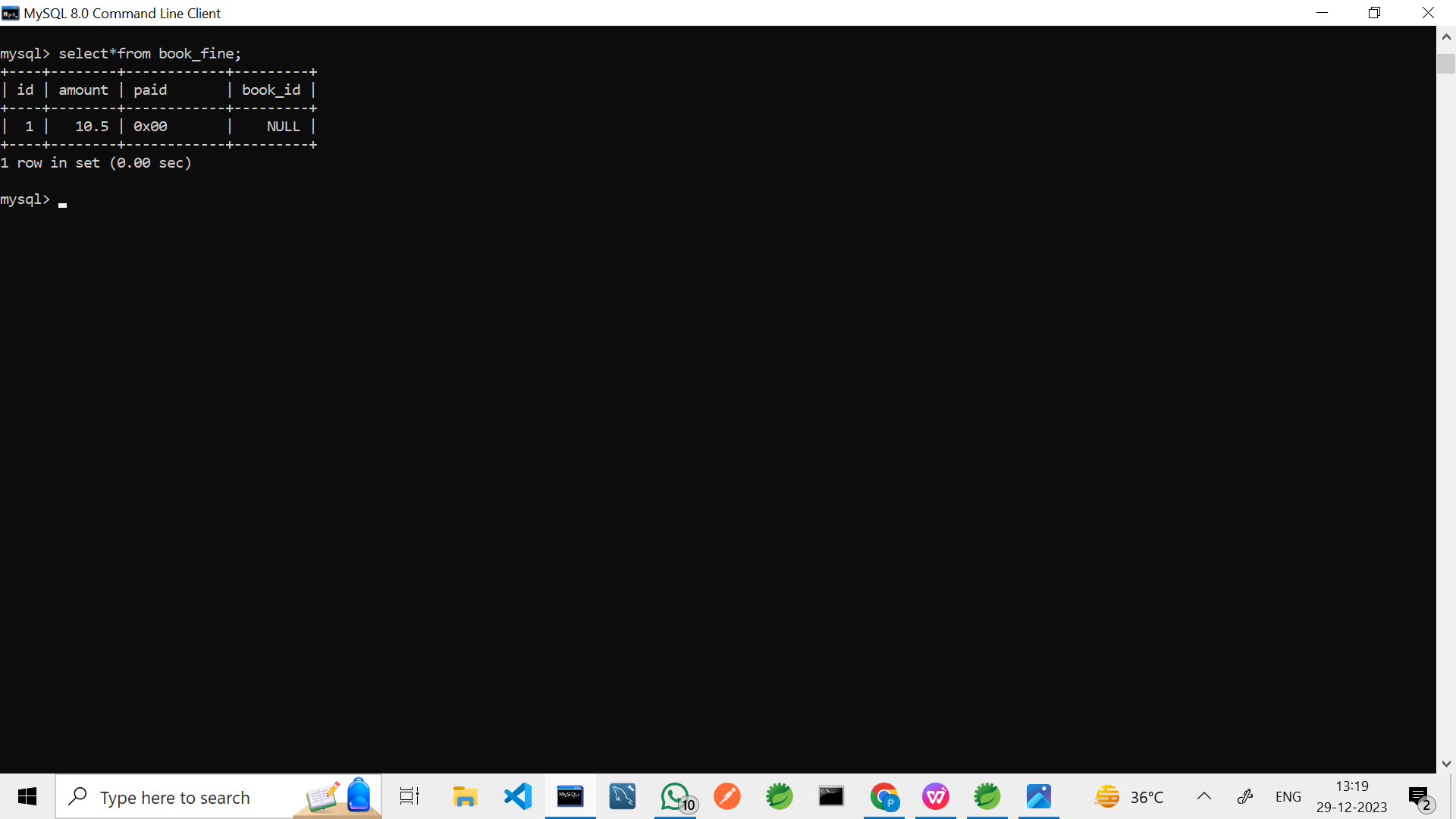
1. Book



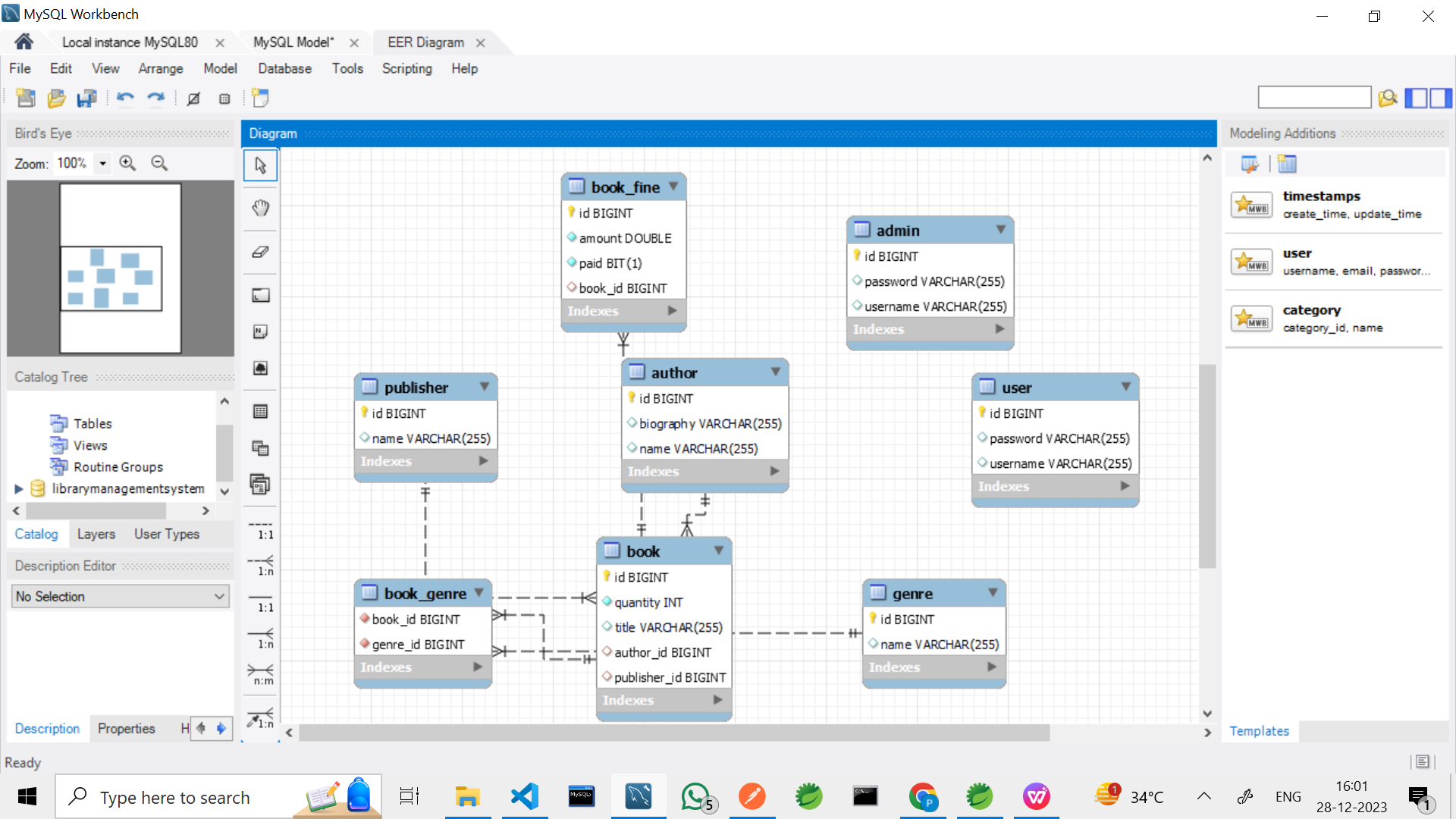
1. Users



1. Book Fines



**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 librarymanagement\_ystem;*

**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 **Admin** (

id INT PRIMARY KEY,

username VARCHAR(255),

password VARCHAR(255);

CREATE TABLE **Author** (

id INT PRIMARY KEY,

biography VARCHAR(50),

name VARCHAR(50) );

CREATE TABLE **Publisher** (

id INT PRIMARY KEY,

name VARCHAR(255));

CREATE TABLE **Genre** (

id INT PRIMARY KEY,

name VARCHAR(255));

CREATE TABLE **Book** (

id INT PRIMARY KEY,

quantity VARCHAR(50),

title VARCHAR(50),

FOREIGN KEY (author\_id) REFERENCES Publisher(author\_id) );

FOREIGN KEY (publisher\_id) REFERENCES Publisher(publisher\_id) );

CREATE TABLE **User** (

id INT PRIMARY KEY,

username VARCHAR(255),

password VARCHAR(255);

CREATE TABLE **book\_fine** (

id INT PRIMARY KEY,

amount INT PRIMARY KEY,

paid INT PRIMARY KEY,

FOREIGN KEY (book\_id) REFERENCES book(book\_id) );

* **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';

* **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 publisher with id 1*

*UPDATE Publisher*

*id= 1*

*name=’xyz’;*