Database II Lab/ 3rd Grade

[Second Semester] and [2026]

[Lab 4]

[24/2/ 2026] [8.30 + 10.30 am]

Instructor Information

Instructor	Email	Hours
Dr. Rasool Hisham Dr. Zainab Namh Dr. Azhar Flaih Assist. Prof. Zahraa Jaaz	[rasool.hisham@nahrainuniv.edu.iq]	[2 Hrs]

Connecting MySQL to C# Windows Forms Application

This lab focuses on the integration of C#, a powerful object-oriented language, with MySQL, a widely-used Relational Database Management System (RDBMS). By combining these technologies, programmers can create data-centric Windows applications.

Lab Objective:

1. To establish a connection between a MySQL database and a C# Windows Forms application using **MySQL Connector/NET**.
2. Use MySqlDataReader to **fetch data** from a MySQL table.
3. Display the retrieved data in a **MessageBox**.

Software Required:

- Visual Studio
- MySQL Server (installed and running)
- MySQL Connector/NET

Connecting a **C# Windows Form Application** to a **MySQL Server** is a standard way to build data-driven desktop software.

To Connect a C# Windows Forms application to a MySQL database we rely on the **MySQL Connector/NET**, a specialized library that facilitates communication between your **.NET applications** and the database server (**MySQL server**). It enables developers to interact with MySQL databases by providing a set of .NET libraries that **handle the communication between the application and the database.**

MySql.Data.dll is the **core library** of **MySQL Connector/NET** that acts as a data provider (driver), enabling .NET applications to connect to and interact with MySQL databases.

The **MySql.Data.MySqlClient** namespace which is the container of MySQL data access classes comes from the **MySQL Connector/NET** which enables C# applications to interact with MySQL databases.

Database II Lab/ 3rd Grade

[Second Semester] and [2026]

[Lab 4]

[24/2/ 2026] [8.30 + 10.30 am]

Instructor Information

Instructor	Email	Hours
Dr. Rasool Hisham Dr. Zainab Namh Dr. Azhar Flaih Assist. Prof. Zahraa Jaaz	[rasool.hisham@nahrainuniv.edu.iq]	[2 Hrs]


```
MySql.Data.dll
    └── MySql.Data.MySqlClient (namespace)
            ├── MySqlConnection
            ├── MySqlCommand
            └── MySqlDataReader
```

The Classes are:

1. **Classes for Database Connection**: It provides classes like **MySqlConnection** for establishing a connection to a MySQL server.
2. **Command and Query Execution**: It includes classes like:
    a. **MySqlCommand:** to execute SQL queries or commands against the database.
    b. **MySqlDataReader**: to read data returned by a query from the MySQL database.

How to Connect MySQL to C#:

1. Install MySQL Connector via NuGet.
2. Add the Namespace using MySql.Data.MySqlClient
3. Define the Connection String: The connection string contains your server address, database name, and login credentials.
4. Open the Connection

   Use a MySqlConnection object within a try-catch block to handle potential errors.

Once you are connected to the MySQL database, you can perform CRUD operations (Create, Read, Update, Delete).

Database II Lab/ 3rd Grade

[Second Semester] and [2026]

[Lab 4]

[24/2/ 2026] [8.30 + 10.30 am]

Instructor Information

Instructor	Email	Hours
Dr. Rasool Hisham Dr. Zainab Namh Dr. Azhar Flaih Assist. Prof. Zahraa Jaaz	[rasool.hisham@nahrainuniv.edu.iq]	[2 Hrs]

Basic CRUD Operations:

To interact with data, use MySqlCommand for queries.

- **Insert Data:** Use ExecuteNonQuery() for actions like INSERT, UPDATE, or DELETE.
- **Read Data:** Use MySqlDataReader and ExecuteReader() to fetch rows from a SELECT query.

Establish the connection: MySqlConnection

- The MySqlConnection class is used to establish a connection between C# program and MySQL database.
- Before executing any queries or commands, you first need to open a connection using this class.

```
var connection = new MySqlConnection("your_connection_string_here");
connection.Open();  // Opens the connection to the MySQL server
```

- **MySqlConnection: A class** from the **MySql.Data.MySqlClient namespace**, used to **establish a connection with a MySQL database**.
- **new MySqlConnection(""):** Creates a new instance of **MySqlConnection** with a proper connection string ("").
- **connection.Open():** This method is used to open the connection to the MySQL database. It should be called before any command or query is executed.

Database II Lab/ 3rd Grade

[Second Semester] and [2026]

[Lab 4]

[24/2/ 2026] [8.30 + 10.30 am]

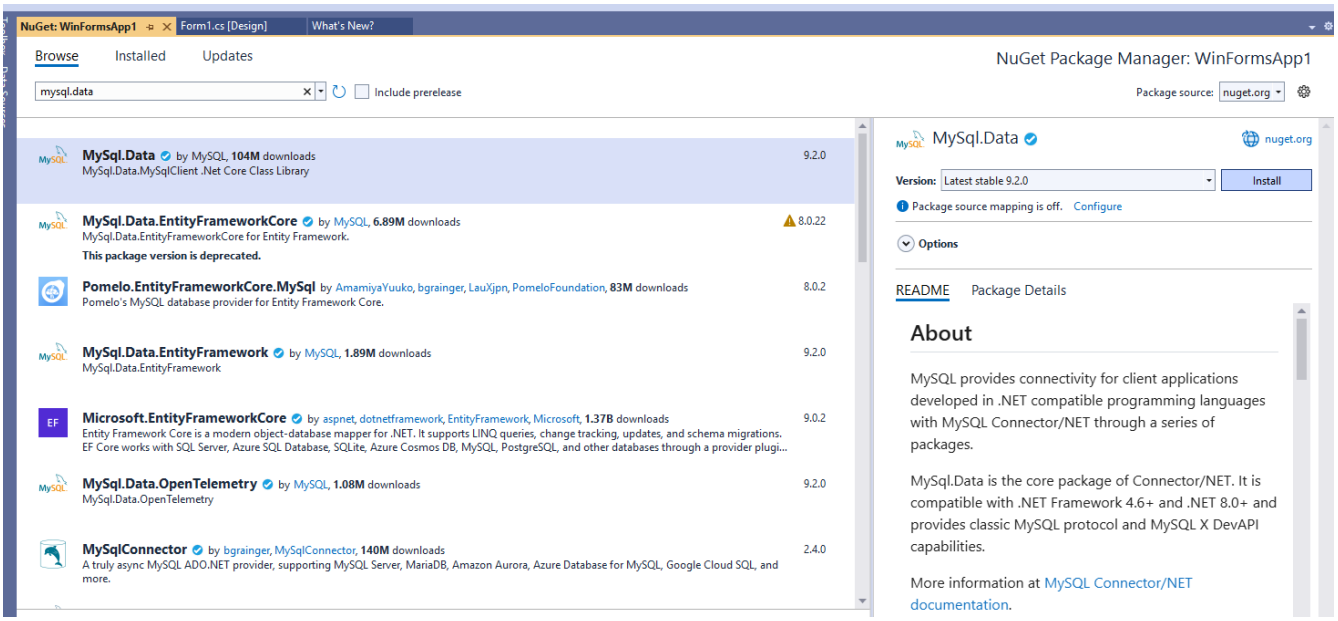
Instructor Information

Instructor	Email	Hours
Dr. Rasool Hisham Dr. Zainab Namh Dr. Azhar Flaih Assist. Prof. Zahraa Jaaz	[rasool.hisham@nahrainuniv.edu.iq]	[2 Hrs]

Task 1: We want to add a **button** in a Windows Forms application C# to connect with MySQL database **company**. The event handler (BtnDB_Click) attempts to establish a connection to a MySQL database using the **MySqlConnection** class from the **MySql.Data.MySqlClient** library.

Step 1: Add MySql.Data.dll (Dynamic Link Library.)

1. Open **Visual Studio** and create a **Windows Forms App** project.
2. Add a reference to **MySql.Data.dll** in your C# project:
   - Go to **Solution Explorer** → Right-click on **References/Dependencies**→ Select **Manage NuGet Packages**.
   - Search for **MySql.Data** and install it.



Instructor Information

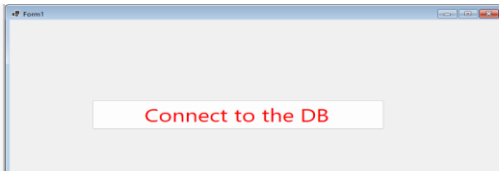
Instructor	Email	Hours
Dr. Rasool Hisham Dr. Zainab Namh Dr. Azhar Flaih Assist. Prof. Zahraa Jaaz	[rasool.hisham@nahrainuniv.edu.iq]	[2 Hrs]

Step 2: Establish Connection in C#

1. Add the **C# Library to Connect MySQL with WinForms**

```
using MySql.Data.MySqlClient; // Import MySQL Library
```

2. Add a button, set the **Text** for example to: **Connect to the DB** while the **Name**: BtnDB.



3. Add the following **connection object** to establish a connection with MySQL. Note that the code is wrapped in **using** and a **try-catch** block for resource management and error handling respectively.

```
string connectionString = @"server=server name; username=user name;
                            password=yourpassword;
                             database=yourdatabse; port=3306";

using (var conn = new MySqlConnection(connectionString))
{
        try
            {
```

MySqlConnection: Creates a connection object to interact with MySQL.

Connection String:

- server=server name or IP address; → Connects to a MySQL server running on the local machine.
- username=user_name; → Connects using the MySQL user.

Instructor Information

Instructor	Email	Hours
Dr. Rasool Hisham Dr. Zainab Namh Dr. Azhar Flaih Assist. Prof. Zahraa Jaaz	[rasool.hisham@nahrainuniv.edu.iq]	[2 Hrs]

- password= yourpassword; → Uses the password " " .
- database= yourdatabase →The database name to connect to
- port=3306; → Uses the default MySQL port (3306).

4. Opening the Connection

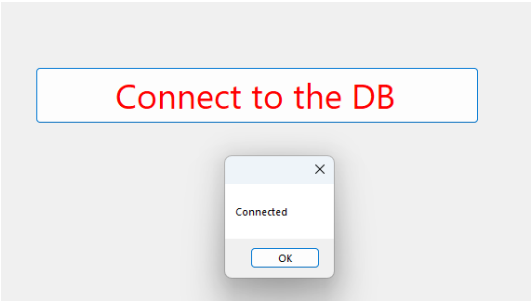
```
connection.Open();
```

- This opens a connection to the MySQL server.
- If the connection is successful, it proceeds to the next step.

5. Success Message

```
MessageBox.Show("Connected");
```

- If the connection is successful, a message box displays **"Connected"**.



6. Closing the Connection

```
connection.Close();
```

- Closes the database connection to free up resources.
- Best practice is to always **close the connection** after use.

Database II Lab/ 3rd Grade

[Second Semester] and [2026]

[Lab 4]

[24/2/ 2026] [8.30 + 10.30 am]

Instructor Information

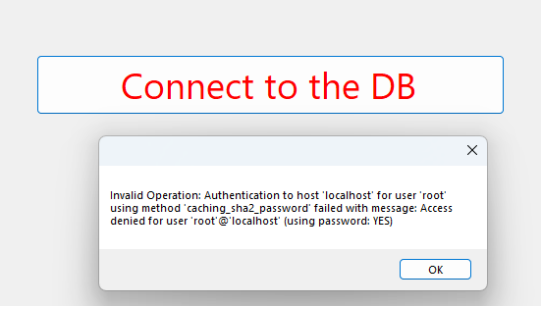
Instructor	Email	Hours
Dr. Rasool Hisham Dr. Zainab Namh Dr. Azhar Flaih Assist. Prof. Zahraa Jaaz	[rasool.hisham@nahrainuniv.edu.iq]	[2 Hrs]

7. Catch Block - Handling Errors

```
catch (Exception ex)
{
   MessageBox.Show("Invalid Operation: " + ex.Message);
}
```

- **Catches exceptions** that may occur when:
  - The MySQL server is **not running**.
  - **Invalid credentials** are provided.
  - **Firewall issues** block the connection.
  - **Network connectivity problems**.

- **Displays the error message** in a message box.



Task 2: We want to **read data records** from the database using a MySqlDataReader. Then **display the query results** in a message box.

Step 1: MySqlCommand : Creating and Configuring the Command

```
using (MySqlCommand cmd = new MySqlCommand())
{
    string sql = "SELECT dno, dname FROM department";
    cmd.CommandType = CommandType.Text;
    cmd.CommandText = sql;
    cmd.Connection = conn;
```

Instructor Information

Instructor	Email	Hours
Dr. Rasool Hisham Dr. Zainab Namh Dr. Azhar Flaih Assist. Prof. Zahraa Jaaz	[rasool.hisham@nahrainuniv.edu.iq]	[2 Hrs]

- **MySqlCommand cmd = new MySqlCommand();**
  - Creates a command object that will be used to execute SQL statements.

- **string sql = "SELECT dno, dname FROM department";**
  - Defines the SQL query to fetch department numbers (dno) and department names (dname) from the department table.

- **cmd.CommandType = CommandType.Text;**
  - Specifies that the command is a simple text query (not a stored procedure).

- **cmd.CommandText = sql;**
  - Assigns the SQL query to the command.

- **cmd.Connection = conn;**
  - Associates the command with the established MySQL connection

Step 2: MySqlDataReader: Executing the Query and Reading the Data

```
using (MySqlDataReader rdr = cmd.ExecuteReader())
{
    while (rdr.Read())
    {
        int dno = rdr.GetInt32("dno");
        string dname = rdr.GetString("dname");
        MessageBox.Show(dname);
    }
}
```

- **cmd.ExecuteReader();**
  - Executes the query and returns a MySqlDataReader, which allows reading data row by row.

Instructor Information

Instructor	Email	Hours
Dr. Rasool Hisham Dr. Zainab Namh Dr. Azhar Flaih Assist. Prof. Zahraa Jaaz	[rasool.hisham@nahrainuniv.edu.iq]	[2 Hrs]

- **while (rdr.Read())**
  - Iterates over each row in the result set.
- **int dno = rdr.GetInt32("dno");**
  - Retrieves the dno (department number) as an integer.
- **string dname = rdr.GetString("dname");**
  - Retrieves the dname (department name) as a string.
- **MessageBox.Show(dname);**
  - Displays the department name in a message box.
- **If there are multiple departments, multiple message boxes will appear.**