# 1.首先需要安装MySQL数据库,可以使用压缩版的,需要把它安装为服务

# 2.安装.net用的MySQL链接器

## 1)新建一个项目,如Mysqldemo,下载c#.net控制台项目

|  |
| --- |
|  |

## 2)在项目文件上面点击右键->管理Nuget包

|  |
| --- |
|  |

## 3)在弹出的界面中点击浏览选项卡,在下面选择MySQL.Data包,点击右边的安装按钮

|  |
| --- |
|  |

## 4)然后下面就开始安装

|  |
| --- |
|  |

## 5)此时会弹出一个预览更改对话框,点击应用按钮

|  |
| --- |
|  |

## 6)然后会弹出一个许可证协议对话框,点击”我接受”

|  |
| --- |
|  |

## 7)然后下面会继续安装

|  |
| --- |
|  |

## 8)当下面的信息被清空,点击已安装选项卡,发现已经安装好了

|  |
| --- |
|  |

## 9)退出nuget工具界面,点击项目里面的引用,发现有MySQL.Data和一些工具

|  |
| --- |
|  |

# 3.使用MySQL

## 1)打开一个命令行窗口,输入mysql –uroot –p

|  |
| --- |
|  |

## 2)输入show databases;显示所有数据库

|  |
| --- |
|  |

## 3)选择company数据库,输入命令use company;

|  |
| --- |
|  |

## 4)查看这个数据库有哪些表,输入show tables;

|  |
| --- |
|  |

## 5)需要查看这个数据库的employee数据表的数据,输入select \* from employee;

|  |
| --- |
|  |

# 4.好我们使用c#代码来操作这个company数据库的employee数据表

MySql.Data.dll提供以下8个类：

* MySqlConnection: 连接MySQL服务器数据库。
* MySqlCommand:执行一条sql语句。
* MySqlDataReader: 包含sql语句执行的结果，并提供一个方法从结果中阅读一行。
* MySqlTransaction: 代表一个SQL事务在一个MySQL数据库。
* MySqlException: MySQL报错时返回的Exception。
* MySqlCommandBuilder: Automatically generates single-table commands used to reconcile changes made to a DataSet with the associated MySQL database.
* MySqlDataAdapter: Represents a set of data commands and a database connection that are used to fill a data set and update a MySQL database.
* MySqlHelper: Helper class that makes it easier to work with the provider.

## 1)导入数据库操作组件

|  |
| --- |
|  |

## 2)测试连接,定义一个字符串保存数据库的连接字符串,然后利用这个字符串创建一个连接对象,然后我们编写一个InitDB函数来连接数据库,然后在main方法里面调用这个函数,代码如下

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Data;  using MySql.Data.MySqlClient;  namespace Mysqldemo  {  internal class Program  {  public static string connetStr = "server=127.0.0.1;port=3306;user=root;password=root; database=company;";  public static MySqlConnection conn = new MySqlConnection(connetStr);  static void InitDB()  {  try  { if(conn.State == ConnectionState.Closed)  conn.Open();//打开通道，建立连接，可能出现异常,使用try catch语句  Console.WriteLine("已经建立连接");  //在这里使用代码对数据库进行增删查改  }  catch (MySqlException ex)  {  switch (ex.Number)  {  case 0:  Console.WriteLine("Cannot connect to server. Contact administrator");  break;  case 1045:  Console.WriteLine("Invalid username/password, please try again");  break;  }  }  finally  {  conn.Close();  }  }  static void Main(string[] args)  {  InitDB();  }  }  } |

## 3)运行程序,发现连接成功

|  |
| --- |
|  |

## 4)添加查询方法.需要先把InitDB的代码修改一下

|  |
| --- |
|  |

## 5)为了方便程序显示,我们把数据表的内容修改如下

|  |
| --- |
|  |

## 6)我们编写一个显示所有数据的函数ShowAll,代码如下

|  |
| --- |
| = |

### 运行程序,效果如下

|  |
| --- |
|  |

## 7)然后我们根据性别来查找员工,写一个GetEmpByGender(string gender)方法

|  |
| --- |
|  |

### 效果

|  |
| --- |
|  |

## 8)然后我们做一个获取高工资的函数GetHigerSal

|  |
| --- |
|  |
|  |

### 效果

|  |
| --- |
|  |

## 9) 然后我们做一个获取低工资的函数GetLowerSal

|  |
| --- |
|  |

### 效果

|  |
| --- |
|  |

## 10)实现新增功能函数AddNew,代码如下

|  |
| --- |
|  |

### 运行程序,效果如下

|  |
| --- |
|  |

### 然后到数据库里面查看,发现新增成功

|  |
| --- |
|  |

## 11)实现一个Update函数来继续修改数据

|  |
| --- |
|  |

### 效果:所有高于3000的工资都给改了

|  |
| --- |
|  |

## 12)然后我们做一个删除记录的函数DeleteById,代码如下

|  |
| --- |
|  |

### 测试,我们把id=56的员工删除

|  |
| --- |
|  |

### 效果

|  |
| --- |
|  |

### 然后我们到数据库里面查看

|  |
| --- |
|  |

### 发现id为56的员工被删除

# 这一节的学习到此为止,完整的代码如下

## Program.cs

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Data;  using MySql.Data.MySqlClient;  using System.Web;  using System.Runtime.Remoting.Messaging;  using System.Reflection;  namespace Mysqldemo  {  internal class Program  {  public static string connetStr = "server=127.0.0.1;port=3306;user=root;password=root; database=company;";  public static MySqlConnection conn = new MySqlConnection(connetStr);  static void InitDB()  {  if (conn.State == ConnectionState.Closed)  conn.Open();//打开通道，建立连接，可能出现异常,使用try catch语句  Console.WriteLine("已经建立连接");    }  //查询所有记录,操作company数据库中的employee数据表  static void ShowAll()  {  InitDB();  string sqlstr = "select \* from employee";  MySqlCommand cmd = new MySqlCommand(sqlstr, conn);  cmd.CommandType = CommandType.Text;  MySqlDataReader reader = cmd.ExecuteReader();  Console.WriteLine("id\tname\tgender\tsalary");  while (reader.Read())  {  Console.WriteLine($"{reader.GetInt32("id")}\t{reader.GetString("name")}\t{reader.GetString("gender")}" +  $"\t{reader.GetInt32("salary")}");  }  }  // 查询所有指定性别的员工  static void GetEmpByGender(string gender)  {  InitDB();  string sqlstr = "select \* from employee where gender = @param";  MySqlCommand cmd = new MySqlCommand(sqlstr, conn);  cmd.Parameters.AddWithValue("param", gender);  MySqlDataReader reader = cmd.ExecuteReader();  Console.WriteLine("id\tname\tgender\tsalary");  while (reader.Read())  {  Console.WriteLine($"{reader.GetInt32("id")}\t{reader.GetString("name")}\t{reader.GetString("gender")}" +  $"\t{reader.GetInt32("salary")}");  }  }  //获取高工资员工  static void GetHigherSal(int sal)  {  InitDB();  string sqlstr = "select \* from employee where salary > @param";  MySqlCommand cmd = new MySqlCommand(sqlstr, conn);  cmd.Parameters.AddWithValue("param", sal);  MySqlDataReader reader = cmd.ExecuteReader();  Console.WriteLine("id\tname\tgender\tsalary");  while (reader.Read())  {  Console.WriteLine($"{reader.GetInt32("id")}\t{reader.GetString("name")}\t{reader.GetString("gender")}" +  $"\t{reader.GetInt32("salary")}");  }  }  //获取低工资员工  static void GetLowerSal(int sal)  {  InitDB();  string sqlstr = "select \* from employee where salary < @param";  MySqlCommand cmd = new MySqlCommand(sqlstr, conn);  cmd.Parameters.AddWithValue("param", sal);  MySqlDataReader reader = cmd.ExecuteReader();  Console.WriteLine("id\tname\tgender\tsalary");  while (reader.Read())  {  Console.WriteLine($"{reader.GetInt32("id")}\t{reader.GetString("name")}\t{reader.GetString("gender")}" +  $"\t{reader.GetInt32("salary")}");  }  }  //新增  static void AddNew()  {  InitDB();  string sqlstr = "insert into employee(name,depId,gender,salary) values('Marcia',2,'female',2900)";  MySqlCommand cmd = new MySqlCommand(sqlstr, conn);  int result = cmd.ExecuteNonQuery();  Console.WriteLine("受影响记录数{0}",result);  }  //修改数据  static void UpdateData()  {  InitDB();  string sqlstr = "update employee set salary=3200 where salary > 3000";  MySqlCommand cmd = new MySqlCommand(sqlstr, conn);  int result = cmd.ExecuteNonQuery();  Console.WriteLine("受影响记录数{0}", result);  }  //删除数据  static void DeleteById(int id)  {  InitDB();  string sqlstr = "delete from employee where id=@id ";  MySqlCommand cmd = new MySqlCommand(sqlstr, conn);  cmd.Parameters.AddWithValue("id", id);  int result = cmd.ExecuteNonQuery();  Console.WriteLine("受影响记录数{0}", result);  }  static void Main(string[] args)  {  //InitDB();  //ShowAll();  //GetEmpByGender("female");  //GetHigherSal(3000);  // GetLowerSal(3500);  //AddNew();  //UpdateData();  DeleteById(56);  }  }  } |