**新建项目**

**计算机学院（软件学院）实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **学生姓名** | 徐可可 | **学号** | 171530425 | **实验成绩** |  |
| **专业** | 软件工程 | **班级** | 软工四班 | **实验日期** | 19年12月25日 |
| **课程名称** | C#高级程序设计 | | | **任课教师** | 彭伟国 |
| **实验名称** | MySQL数据库编程 | | | **实验序号** | 7 |
| **实验地点** | S409 | **实验台号** | 25 | **指导教师** | 彭伟国 |
| **一、实验目的及要求**  1. 掌握MySQL中SQL语句的使用。  2. 能够使用C#程序操作MySQL数据库。 | | | | | |
| **二、实验内容（或实验原理、实验拓扑）**  1. 创建一个数据库，在数据库中创建3个表：student、course、score。 student表中有以下字段：id（主键）、name（指学生名）、sex、grade，course表中有以下字段：id（主键）、name（指课程名）、desc（课程描述 类型varchar），score表中有以下字段：s\_id（外键）、c\_id（外键）、sc\_num，s\_id和c\_id组合为复合主键。写出创建 student、course、score三个表的语句；写出sql语句：查询哪些人的某门课成绩（如chinese）大于这门课的平均成绩。  2. C#连接Mysql数据库有5个步骤，分别是：拿到数据库连接、拿到发送的sql语句、执行sql语句、拿到执行结果、关闭资源。写出两个函数，bool FindStudentByName(string name)实现按名字查询数据库中的数据，bool Save(Student s)实现向数据表student中插入一条数据记录。 | | | | | |
| **三、实验设备与环境**  1. 高配笔记本：Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz  、16G内存、48TB分布式硬盘、Windows 10 1909政府版、Visual Studio 2019 企业版、.NET Framework 4.7.2  2. 实验室电脑：英特尔 酷睿2 双核 T7700 @ 2.40GHz、3G内存、128 G虚拟硬盘、Windows 7 旗舰版 32位 SP1、Microsoft Visual Studio Ultimate 2012、.NET Framework 4.5 | | | | | |
| **四、实验设计方案（包括实验步骤、设计思想、算法描述或开发流程等）** | | | | | |

|  |
| --- |
| **五、实验结果（包括设计效果、测试数据、运行结果等）**  在这里插入图片描述 |
| **六、实验小结（包括收获、心得体会、注意事项、存在问题及解决办法、建议等）**  如果使用ado.net链接mysql数据库则只需要引用  MySql.Data.dll即可，并不需要安装mysql-connector-net驱动程序；  如果使用EF的话需要安装mysql-connector-net驱动程序和mysql-for-visualstudio 这个vs里面链接数据源的插件，否则EF无法使用VS视图模型的浏览和创建以及更新数据库实体；    其中mysql-connector-net驱动程序安装目录包含了  MySql.Data.dll；  MySql.Data.Entity.EF5.dll；  MySql.Data.Entity.EF6.dll；  MySql.Fabric.Plugin.dll；  MySql.Web.dll； |

|  |
| --- |
| **七、附录（包括作品、流程图、源程序及命令清单等）**  **建库.sql**  drop database if exists zhang;  CREATE DATABASE IF NOT EXISTS zhang;  use zhang;  -- ----------------------------  -- Table structure for student  -- ----------------------------  DROP TABLE IF EXISTS `student`;  CREATE TABLE `student`  (    `id` int(11) NOT NULL,    `name` varchar(15) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,    `sex` varchar(8) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NULL DEFAULT NULL,    `grade` int(11) NOT NULL,    PRIMARY KEY (`id`) USING BTREE  ) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = Dynamic;  SET FOREIGN\_KEY\_CHECKS = 1;  -- create database zhang;  -- use zhang;  -- show tables;  -- -- (表和类一一对应)  -- create table student(  -- id int primary key,  -- name varchar(15) not null,  -- sex varchar(8),  -- grade int not null  -- );  **Program.cs**  using System;  using System.Collections.Generic;  using MySql.Data.MySqlClient;  namespace U28D第十五天  {  class Program  {  //先建表 再建类    //连接数据库5个步骤：  //1 拿到数据库连接  //2 拿到发送语句  //3 执行sql语句  //4 拿到执行结果  //5 关闭资源  static void Main(string[] args)  {  ////string url = "User ID=root;Password=myPassword;Host=localhost;Port=3306;Database=zhang;";//zhang 为自己定义的数据库  string url = "User ID=zj175;Password=i,@mc0c0@my;Host=coco56.top;Port=6603;Database=zhang;";//Password=密码为空什么都不写 localhost本地 ID=root用户  MySqlConnection conn = new MySqlConnection(url);//创建数据库连接  conn.Open();//数据库连接打开  string sql = "truncate table student;";//创建sql语句  MySqlCommand comd = new MySqlCommand(sql, conn);//发送sql语句给数据库服务器    sql = "insert into student values(18,'ppp','man',4)";//创建sql语句  comd = new MySqlCommand(sql, conn);//发送sql语句给数据库服务器  //MySqlDataReader reader = comd.ExecuteReader();//执行sql语句 MySqlDataReader类似于迭代器  //while (reader.Read())//判断是否有下一个  //{  // Console.WriteLine(reader.GetInt32(0));//拿到ID  // Console.WriteLine(reader.GetString(1));//拿到名字  // Console.WriteLine(reader.GetString(2));//拿到性别  // Console.WriteLine("---------");  //}  ////1.hasNext CurrentValue  //int a = comd.ExecuteNonQuery();//表示受影响的行数  //Console.WriteLine(a);//执行sql语句受影响的行数 不能执行第2次  //public static bool FindStudentByName(string name)函数  //bool b = FindStudentByName("shunv");  //Console.WriteLine(b);  //public static bool Save(Student s)函数  //Student s = new Student(15, "5555", "man", 3);  //bool b = Save(s);//Save(s)--->s在Save(）圆括号里面 ORM---->object relation Mapping对象关系映射  //Console.WriteLine(b);  // public static Student FindStudentById(int id)函数  //Student s = FindStudentById(8);  //Console.WriteLine(s.Id + " " + s.Name + " " + s.Sex + " " + s.Grade);  //List<Student> list = FindStudents(80);  //Console.WriteLine(list.Count);  //for (int i = 0; i < list.Count; i++)  //{  // Console.WriteLine(list[i].Id + " " + list[i].Name + " " + list[i].Sex + " " + list[i].Grade);  //}  conn.Close();  Console.WriteLine("\nPress any key to quit.");  Console.ReadKey();  }  public static bool FindStudentByName(string name)  {  string url = "User ID=root;Password=;Host=localhost;Port=3306;Database=zhang;";//Password=密码为空什么都不写 localhost本地 ID=root用户  MySqlConnection conn = new MySqlConnection(url);//创建数据库连接  conn.Open();//数据库连接打开  string sql = "select \* from student where name='" + name + "'";  Console.WriteLine(sql);//打印输出sql语句  MySqlCommand comd = new MySqlCommand(sql, conn);//发送sql语句给数据库服务器  MySqlDataReader reader = comd.ExecuteReader();//执行sql语句 MySqlDataReader类似于迭代器  return reader.Read();  //接下来在“U28D第十五天”项目上鼠标右键添加“Student”类  }  public static bool Save(Student s)  {  bool b = false;  string url = "User ID=root;Password=;Host=localhost;Port=3306;Database=zhang;";  MySqlConnection conn = new MySqlConnection(url);//创建数据库连接  conn.Open();//数据库连接打开  //string sql = "insert into student values(" + s.Id + ",'" + s.Name + "','" + s.Sex + "'," + s.Grade + ")";//创建sql语句 这种写法太痛苦  //sql语句的简单写法是一下两行  string sql = "insert into student values({0},'{1}','{2}',{3})";  sql= string.Format(sql, s.Id, s.Name, s.Sex, s.Grade);//F12转到定义，Format函数有params object[] args（可变参数）  Console.WriteLine(sql);  MySqlCommand comd = new MySqlCommand(sql, conn);// 发送sql语句给数据库服务器  int a = comd.ExecuteNonQuery();//执行sql语句  if (a==1)  {  return b= true;  }  conn.Close();//关闭资源  return b;  }  public static Student FindStudentById(int id)  {  Student s = null;  string url = "User ID=root;Password=;Host=localhost;Port=3306;Database=zhang;";  MySqlConnection conn = new MySqlConnection(url);//创建数据库连接  conn.Open();//数据库连接打开  string sql = "select \* from student where id="+id;  Console.WriteLine(sql);    MySqlCommand comd = new MySqlCommand(sql, conn);//发送sql语句  MySqlDataReader read = comd.ExecuteReader();  if (read.Read())  {  //如何得到Student类的对象呢？  s=new Student();//找不到就不需要开辟内存  s.Id = read.GetInt32(0);  s.Name = read.GetString(1);  s.Sex = read.GetString(2);  s.Grade = read.GetInt32(3);  }  conn.Close();//关闭资源  return s;  }  public static List<Student> FindStudents(double d)  {  List<Student> list = new List<Student>();  //总分的平均分大于80的学生信息  //List <Student>= null;  string url = "User ID=root;Password=;Host=localhost;Port=3306;Database=zhang;";  MySqlConnection conn = new MySqlConnection(url);//创建数据库连接  conn.Open();//数据库连接打开  //select avg(sc\_num) from score group by c\_id 拿到每一个人的平均分  //select s\_id,avg(sc\_num) a from score group by s\_id having a>80  //select id,name,sex,grade from student,(select s\_id,avg(sc\_num) a from score group by s\_id having a>80) t where student.id=t.s\_id;  string sql = "select id,name,sex,grade from student,(select s\_id,avg(sc\_num) a from score group by s\_id having a>"+d+") t where student.id=t.s\_id;";  Console.WriteLine(sql);  MySqlCommand comd = new MySqlCommand(sql, conn);//发送sql语句  MySqlDataReader read = comd.ExecuteReader();  //Console.WriteLine(read.Read());//不能增加这条语句 要不然少输出一条信息  while (read.Read())  {  Student s = new Student();  s.Id = read.GetInt32(0);  s.Name = read.GetString(1);  s.Sex = read.GetString(2);  s.Grade = read.GetInt32(3);  list.Add(s);  }  conn.Close();//关闭资源  return list;  }  }  } |