# 创建项目和数据库

## 注意:access2016以上版本的数据库是无法使用.net Framework项目来操作的,即使了配置了数据连接,测试连接成功,编程的时候还会报错:没有在此计算机上面注册Microsoft.ACE.OLEDB.12.0,即使了安装了access2016或者access数据库引擎,它还是这个报错.而且就算了找到这个dll用regsrv32注册也会失败

# 使用.net8(基于.netcore)来操作access数据库

## 1.新建一个项目,取名netcore-access-db

|  |
| --- |
|  |

## 1.2在项目的输出文件夹也就是bin\Debug\net8.0下面新建一个数据库文件mygoods.accdb,里面有一个goods数据表

|  |
| --- |
|  |

## 1.3. 点击vs中的工具->连接到数据库,会弹出下面的窗口

|  |
| --- |
|  |

## 1.4保持默认,点击继续,会弹出下面的窗口,第一项保持默认,数据库选择我们刚刚创建的数据库,

|  |
| --- |
|  |

### 点击打开,数据库路径就添加到了文本框中

|  |
| --- |
|  |

### 然后点击测试连接按钮,发现连接成功

|  |
| --- |
|  |

## 1.5.点击高级按钮,会弹出一个新窗口

|  |
| --- |
|  |

## 1.6 .点击2次确定退出数据库配置工具,然后可以看到vs显示了我们和数据库

|  |
| --- |
|  |

## 1.6.选中最下面的连接字符串,把它复制粘贴到一个文件文件里面暂时保存

|  |
| --- |
|  |

## 注意:1.3-1.6步骤可以省略,因为它没有什么实际作用,只是用来获取链接字符串和可视化查看数据库的

## 2.在项目名称上面点击右键->管理Nuget程序包,会弹出一个窗口,选择浏览选项卡,然后输入OLEDB,选择第一个,点击右边的安装按钮

|  |
| --- |
|  |

## 3.然后下面就开始安装,然后会弹出一个对话框,点击应用按钮

|  |
| --- |
|  |

## 4.然后程序就继续安装,完成后的界面如下

|  |
| --- |
|  |

## 5.然后导入这个组件

|  |
| --- |
|  |

## 6.我们先添加一个AddNew方法用来新增数据到数据库,为了方便操作,我们把原来的数据都删除了,新增数据的代码如下

|  |
| --- |
| static void AddNew()  {  try  {  InitDB();  int result = 0;  int id =100;  string name = "coca cola";  string gtype = "600ml";  string brand = "fenza";  bool tax = true;  int price = 1000;  string sqlstr = "insert into goods(id,name,type,brand,taxable,price) values(@id,@name,@gtype,@brand,@tax,@price);";  //string sqlstr = "insert into goods(id,name,type,brand,taxable,price) values(105 ,'Fanta','600ml','Fenza',true,1200);";  //创建命令  //创建命令  OleDbCommand cmd = new OleDbCommand(sqlstr, conn);  cmd.Parameters.AddWithValue("id", id);  cmd.Parameters.AddWithValue("name", name);  cmd.Parameters.AddWithValue("gtype", gtype);  cmd.Parameters.AddWithValue("brand", brand);  cmd.Parameters.AddWithValue("tax", tax);  cmd.Parameters.AddWithValue("price", price);  result = cmd.ExecuteNonQuery();  Console.WriteLine($"受影响的行数:{result}");  }  catch (Exception ex)  {  Console.WriteLine(ex.Message);  }  finally  {  conn?.Close();  }  } |

### 运行程序,效果如下

|  |
| --- |
|  |

## 7.此时会有一个现象,当你取窗口access数据库,你看不到数据,刷新了,还是看不到

|  |
| --- |
|  |

## 8.然后你尝试运行相同的代码,会得到一个错误,大概意思是数据已经存在,了需要修改数据在插入

|  |
| --- |
|  |

## 9.上网找了很多网站,都说是因为.net把数据库拷贝到一个地方,其实我的项目没有这个问题我的数据库就是建立在exe文件所在路径呀.经过多次尝试,终于找到解决办法,就是因为我们的权限不够无法查看数据库,我们需要在access数据库上面点击右键->获取所有管理员权限,如图

|  |
| --- |
|  |

## 10,然后刷新数据库,数据出来了

|  |
| --- |
|  |

### 这个坑一定要小心

## 11.为了方便程序我们多插入几条数据到数据库中

|  |
| --- |
|  |

## 12.其实vs里面的数据库路径服务器的连接是没有用的,它的唯一用途是获取链接字符串,不过在实际开发中,数据库的路径是不能够写死的需要动态变化,我们可以把这个数据库链接删除.然后我们尝试调用程序代码添加数据,发现可以添加

|  |
| --- |
|  |
|  |
|  |

## 13.我们添加一个显示所有记录的函数ShowAll,然后在Main方法里面调用,代码如下

|  |
| --- |
|  |

### 效果如下

|  |
| --- |
|  |

## 14.然后我们来做条件查询函数QueryByTerm,代码如下,然后在Main函数里面调用这个函数

|  |
| --- |
|  |

### 效果

|  |
| --- |
|  |

### 可以到数据库里面验证一下,是对的

|  |
| --- |
|  |

## 15.我们来做一个条件更新方法UpdateByTerm,然后在Main函数里面调用这个函数

|  |
| --- |
|  |

### 运行程序,效果如下

|  |
| --- |
|  |

### 查看数据库,发现是对的

|  |
| --- |
|  |

## 16.然后我们来做删除,写一个DeleteByTerm函数,然后在Main函数里面调用这个函数

|  |
| --- |
|  |

### 运行程序,效果如下

|  |
| --- |
|  |

### 查看数据库,发现价钱小于600元的记录被删除了

|  |
| --- |
|  |

# 这一节的学习到此为止,完整代码如下,注意这是一个.net core项目也就是没有.net framework的那个.

## Program.cs

|  |
| --- |
| using System.Data;  using System.Data.OleDb;  namespace netcore\_access\_db  {  internal class Program  {  public static OleDbConnection? conn = null;  static void InitDB()  {  //conn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=F:\Projects\_f\dotnet-projs\csharp-basic-access-crud\bin\Debug\mygoods.accdb");  string connstr = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + Environment.CurrentDirectory + @"\mygoods.accdb";  conn = new OleDbConnection(connstr);  if (conn.State == ConnectionState.Closed)  {  conn.Open();  }  }  static void AddNew()  {  try  {  InitDB();  int result = 0;  int id =400;  string name = "atun";  string gtype = "500g";  string brand = "sardinar";  bool tax = true;  int price = 900;  string sqlstr = "insert into goods(id,name,type,brand,taxable,price) values(@id,@name,@gtype,@brand,@tax,@price);";  //string sqlstr = "insert into goods(id,name,type,brand,taxable,price) values(105 ,'Fanta','600ml','Fenza',true,1200);";  //创建命令  //创建命令  OleDbCommand cmd = new OleDbCommand(sqlstr, conn);  cmd.Parameters.AddWithValue("id", id);  cmd.Parameters.AddWithValue("name", name);  cmd.Parameters.AddWithValue("gtype", gtype);  cmd.Parameters.AddWithValue("brand", brand);  cmd.Parameters.AddWithValue("tax", tax);  cmd.Parameters.AddWithValue("price", price);  result = cmd.ExecuteNonQuery();  Console.WriteLine($"受影响的行数:{result}");  }  catch (Exception ex)  {  Console.WriteLine(ex.Message);  }  finally  {  conn?.Close();  }  }  //显示所有记录  static void ShowAll()  {  InitDB();  string sql = "select name,type,price from goods;";  OleDbCommand cmd = new OleDbCommand(sql, conn);  cmd.CommandType = CommandType.Text;  OleDbDataReader reader = cmd.ExecuteReader();  while (reader.Read())  {  Console.WriteLine($"name:{reader["name"]},\ttype:{reader["type"]},\tprice:{reader["price"]}");  }  conn?.Close();  }  //条件查询  static void QueryByTerm(string term)  {  InitDB();  string sql = term;  OleDbCommand cmd = new OleDbCommand(sql, conn);  cmd.CommandType = CommandType.Text;  OleDbDataReader reader = cmd.ExecuteReader();  while (reader.Read())  {  Console.WriteLine($"name:{reader["name"]},\ttype:{reader["type"]},\ttaxable:{reader["taxable"]}," +  $"\tprice:{reader["price"]}");  }  conn?.Close();  }  //条件更新  static void UpdateByTerm(string term)  {  InitDB();  string sql = term;  OleDbCommand cmd = new OleDbCommand(sql, conn);  cmd.CommandType = CommandType.Text;  int result = cmd.ExecuteNonQuery();  Console.WriteLine($"受影响的行数:{result}");  conn?.Close();  }  //按条件删除  static void DeleteByTerm(string term)  {  InitDB();  string sql = term;  OleDbCommand cmd = new OleDbCommand(sql, conn);  cmd.CommandType = CommandType.Text;  int result = cmd.ExecuteNonQuery();  Console.WriteLine($"受影响的行数:{result}");  conn?.Close();  }  static void Main(string[] args)  {  // AddNew();  //ShowAll();  //QueryByTerm("select name,type,taxable,price from goods where taxable=true;");  // QueryByTerm("select name,type,taxable,price from goods where price<900;");  //UpdateByTerm("update goods set price=900 where price>950;");  DeleteByTerm("delete from goods where price<600");  }  }  } |

# 扩展:可以把这个项目改为面向对象,也就是使用类的形式来做,也可以学习网上一些orm框架