**实验1 DBMS的安装和使用**

**姓名：段皞一**

**学号：3190105359**

**专业：计算机科学与技术**

**一 实验目的**

1. 通过安装某个数据库管理系统，初步了解DBMS的运行环境。
2. 了解DBMS交互界面、图形界面和系统管理工具的使用。
3. 搭建实验平台。

**二 实验平台**

1. 操作系统： Windows
2. 数据库管理系统：SQL Server 或MySQL

**三 实验内容和要求**

**3.1认识数据库**

数据库是按照数据结构来组织、存储和管理数据的仓库，是存储在一起的相关数据的集合，其优点主要体现在以下几个方面。其优点主要有以下几个方面：

减少数据的冗余度，节省数据的存储空间；

具有较高的数据独立性和易扩充性；

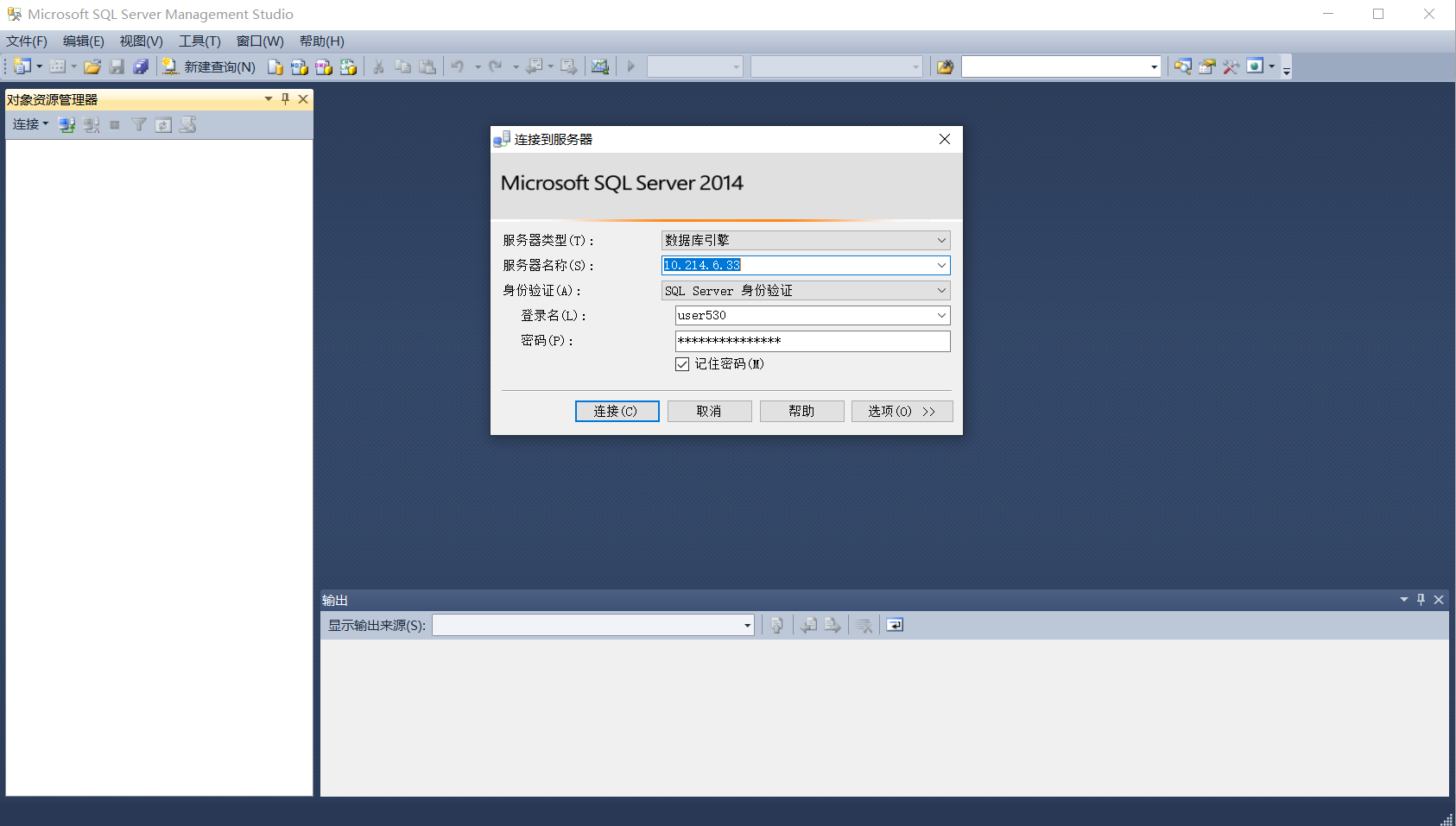
实现数据资源的充分共享。

**3.2根据某个DBMS的安装说明等文档，安装DBMS**

**3.2.1下载安装Microsoft SQL Server 2014**

该版本的DBMS环境在浙江大学正版软件管理系统中即可下载。下载完毕后，需进行一些初步的配置，然后就可通过该平台初步了解DBMS的运行环境。

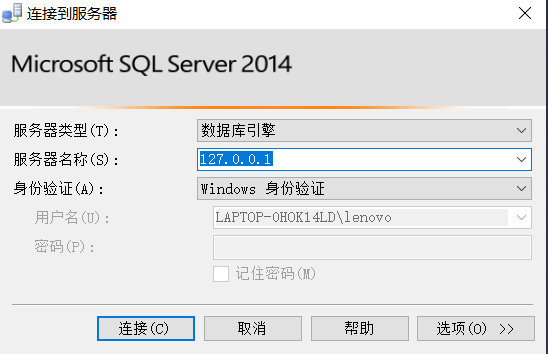
打开Microsoft SQL Server Management Studio,可以出现如下初始界面。



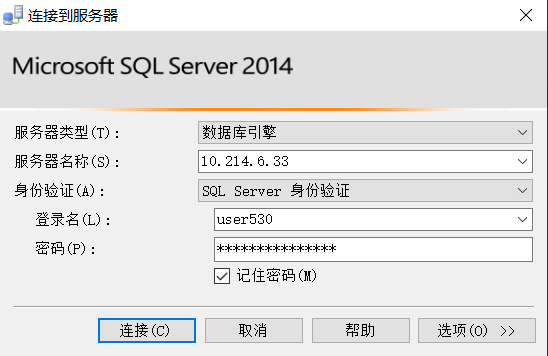
**3.2.2连接到服务器**

需要连接到服务器，才能进行后续的实验。可以连接两种类型的服务器，本地以及外来服务器。

若要链接本地服务器，只需在服务器名称中输入本机的ip地址或者127.0.0.1或者localhost,身份验证选择Windows身份验证，左键点击连接即可进行本地服务器的连接。

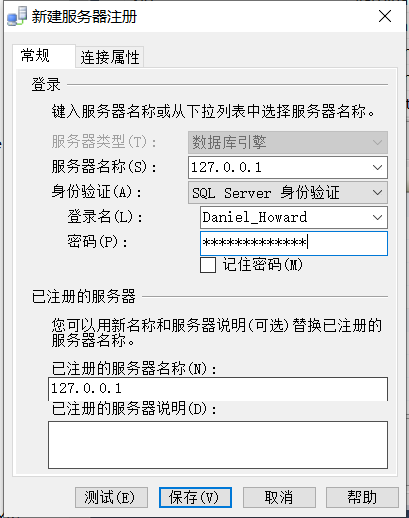


若要连外来的服务器，这也是较为常用的场景，在服务器名称中输入学校的服务器ip地址，并进行SQL Server验证。



**3.3了解DBMS的用户管理**

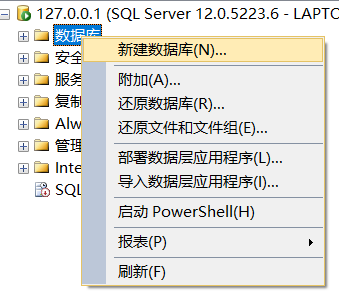
**3.3.1注册**



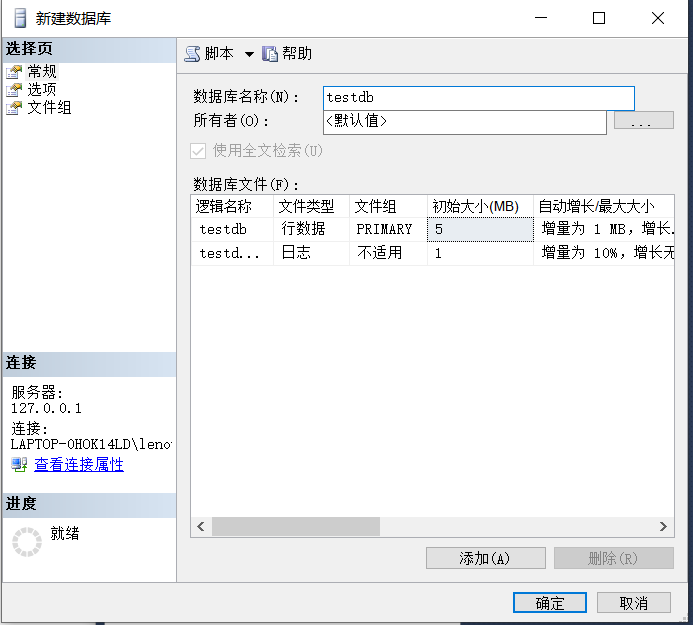
**3.3.2新建数据库**

**3.3.2.1图形界面方法进行数据库的创建**

在“对象资源管理器”中，右键单击数据库，选择“新建数据库”。



接下来在弹出的新建数据库窗口中，可以对数据库进行一系列的初始配置，数据库建成之后也可以打开这个窗口，进行后期的修改与调整。

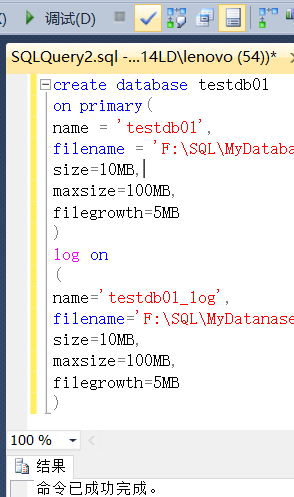


**3.3.2.2脚本方法创建数据库**

单击新建查询按钮，输入以下脚本：

|  |
| --- |
| create database testdb01  on primary(  name = 'testdb01',  filename = 'F:\SQL\MyDatabase\testdb01.mdf',  size=10MB,  maxsize=100MB,  filegrowth=5MB  )  log on  (  name='testdb01\_log',  filename='F:\SQL\MyDatanase\testdb01\_log.ldf',  size=10MB,  maxsize=100MB,  filegrowth=5MB  ) |

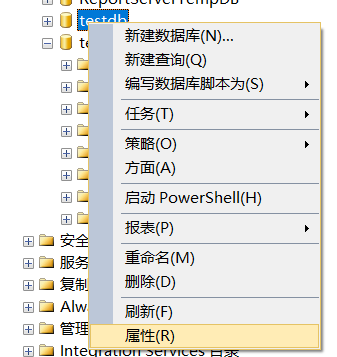
点击分析按钮，代码无误以后，点击执行按钮，完成命令。



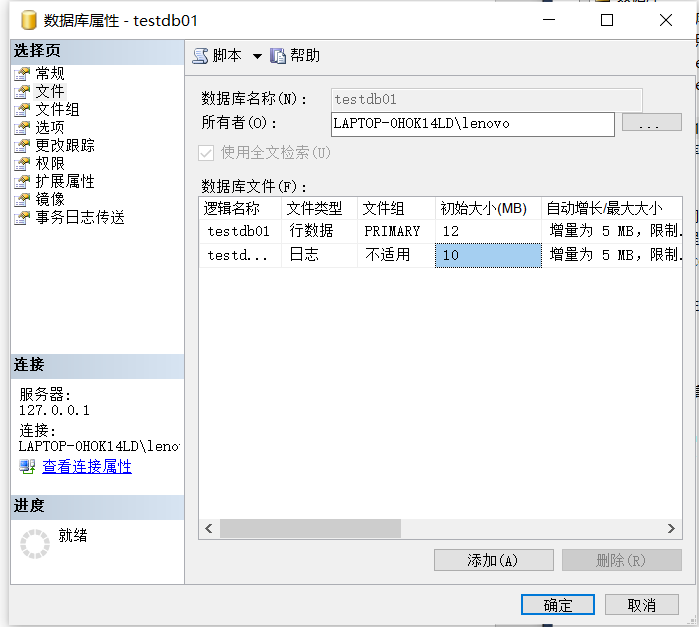
**3.3.3数据库属性的更改**

**3.3.3.1图形界面方法进行数据库属性的修改**

选中要更改属性的数据库，右键选择属性按钮。



在弹出的对话框中就可以进行数据库的属性的修改了。



值得注意的是，修改的时候，存储路径不能够进行修改的，但是数据库的名字依然进行修改。

**3.3.3.2脚本方法进行数据库属性的修改**

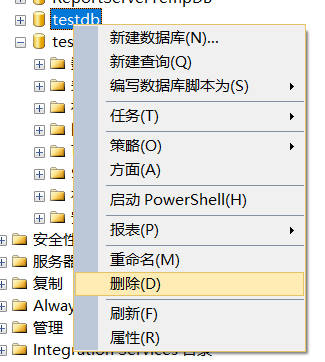
一下是一个脚本的样例，可以进行数据库大小，开销上限，名称，增长指数等参数进行修改。

|  |
| --- |
| ALTER DATABASE TESTDB01  MODIFY NAME = TESTDB01;  ALTER DATABASE TESTDB  MODIFY FILE( NAME=TESTDT,  SIZE=20MB,  MAXSIZE=80MB,  FILEGROWTH=5MB );  EXEC SP\_HELPDB TESTDB; |

**3.3.4数据库的删除**

**3.3.4.1界面方法进行数据库的删除操作**

右键数据库，选择删除按钮。



**3.3.4.2脚本删除操作命令**

相关代码如下：

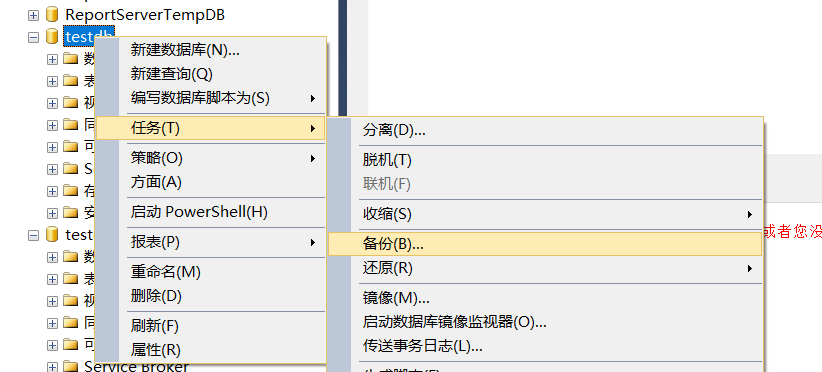
|  |
| --- |
| DROP DATABASE Sales, NewSales; |

**3.3.5数据库的备份与还原**

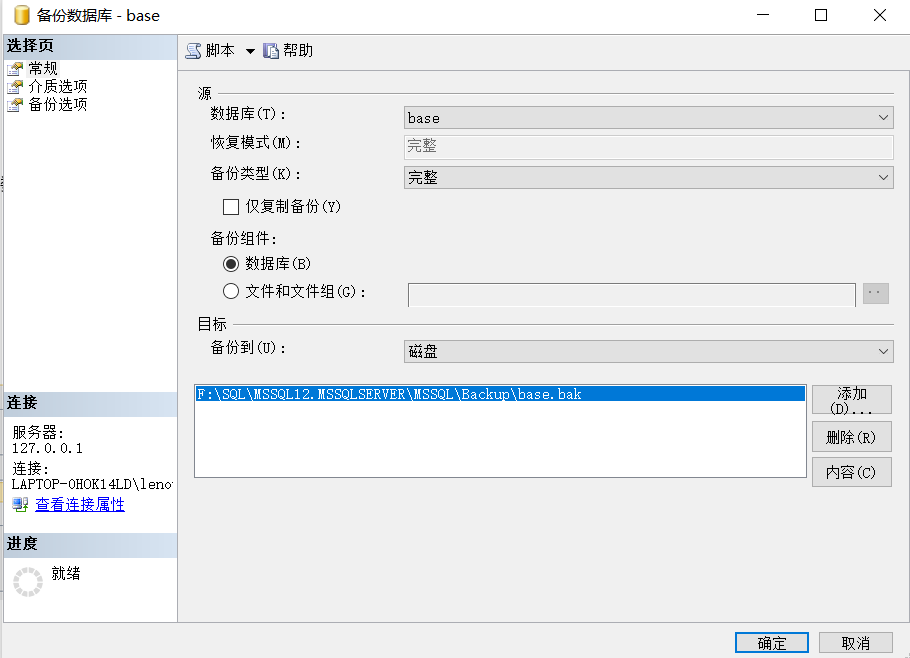
通过还原一组备份然后恢复数据库的策略，能够最大限度地降低灾难性数据丢失的风险。

还原方式主要分为差异备份和完整备份两种。

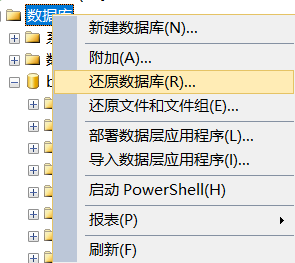
选择需要进行备份的数据库，右键选择“任务”->”备份”



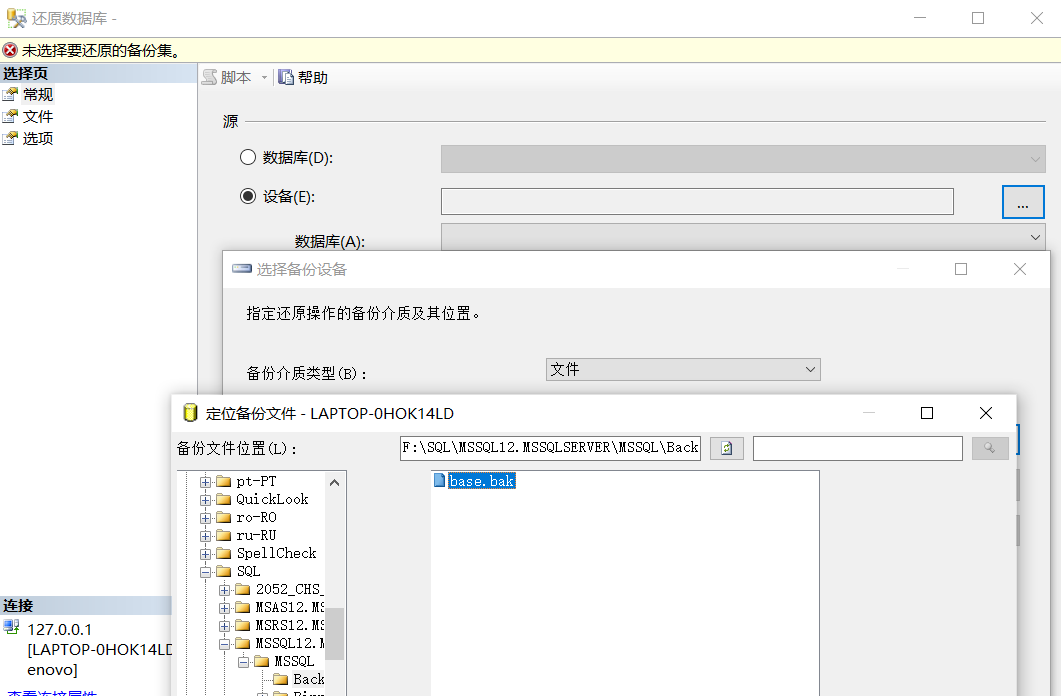
在弹出的对话框中，配置备份的属性，包括选择备份的类型、备份的位置等等。

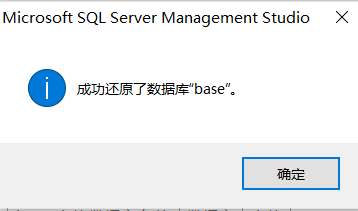


根据备份的文件信息可以进行数据库的还原。具体操作如下：选择相应的数据库，右键选择还原数据库。



在弹出的窗口中找到目标备份文件，即可进行数据库的还原。

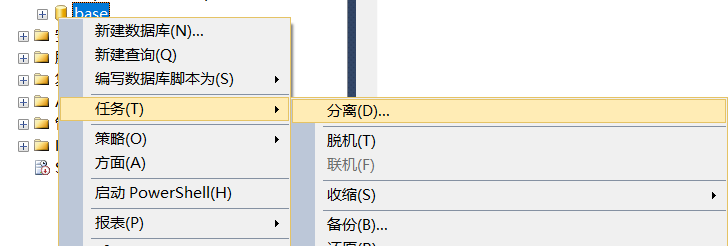




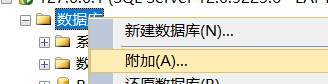
**3.3.6数据库分离和附加**

数据库的分离是指将整个数据库文件进行拷贝，附加是将数据库文件完整的导入目标的服务器。与数据库的备份与还原相比，由于不产生中间的备份文件，所以移动过程中要耗费更多的存储空间。

分离的方法是，右键所选的数据库，点击“任务”->“分离”。



附加的方法是，右键数据库，点击附加，然后倒入目标的数据库文件路径即可。



**3.4 SQL Server下的数据类型**

**3.4.1数字类型**

|  |  |
| --- | --- |
| 数据类型 | 输出 |
| bigint | 8 byte |
| int | 4 byte |
| smallint | 2 byte |
| tinyint | 1 byte |
| float | 取决于n的值 |

**3.4.2时间类型**

|  |  |
| --- | --- |
| 数据类型 | 输出 |
| time | 12:35:29.1234567 |
| date | 2021-3-15 |
| smalldatetime | 2021-3-15 12:35:00 |
| datetime | 2021-3-15 12:35:29.123 |
| datetime2 | 2021-3-15 12:35:29.1234567 |

**3.4.3字符串的类型**

char[n]

varchar[(n|max)]

支持Unicode编码，所以每个字符占用两个字节:

nchar[n]

nvarchar[n|max]

**3.5表的创建和操作**

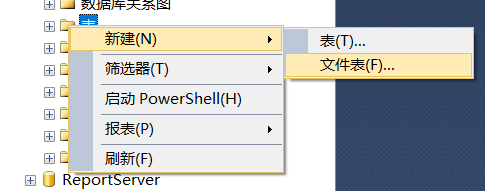
**3.5.1新建表**

若要创建表，必须提供表的名称和该表中每个列的名称和数据类型，指出每个列中是否允许空值。

此外，大多数表中都有一个主键，由表的一列或者多列组成，主键始终是唯一的。

**3.5.1.1图形界面进行创建**

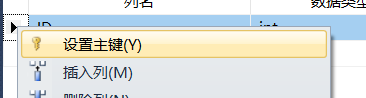
右键数据库栏下的“表”，点击新建“表”



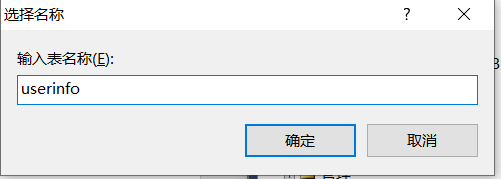
在弹出的窗口下，可以对各个字段进行属性的初始化。



点击一个字段，右键可以将其设为该表的主键。



保存时，可以对表进行命名。



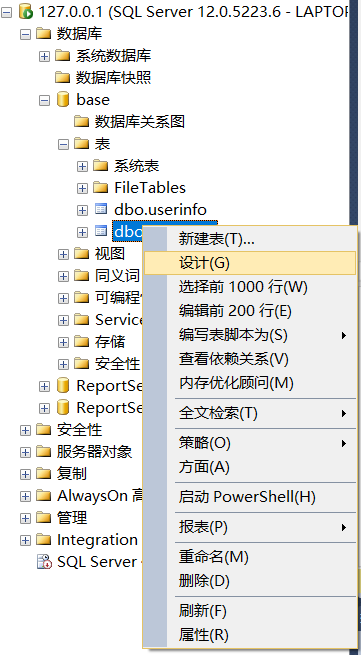
**3.5.1.2使用SQL语言进行表的创建**

|  |
| --- |
| create table userinfo2  (ID int primary key not null,  name varchar(20) not null  ); |

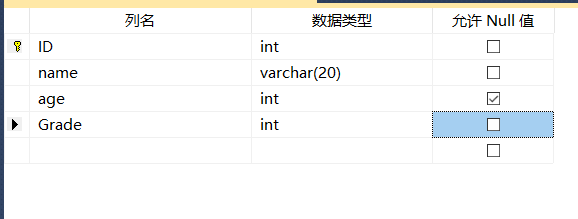
**3.5.2修改、删除表结构**

**3.5.2.1操作界面进行表的修改**

右键相应的表，点击“设计”。



在弹出的对话框中，可以对表的各个字段的属性进行调整。



**3.5.2.2 SQL语句进行**

使用SQL语句进行类型长度的修改。比如将varchar(10)改成varchar(100)。

|  |
| --- |
| alter table userinfo  alter column name varchar(100); |

使用SQL语句进行字段类型的修改。

|  |
| --- |
| alter table userinfo  alter column age float; |

添加是否可以为空。

|  |
| --- |
| alter table userinfo  alter column age float not null; |

添加主键。

|  |
| --- |
| alter table userinfo  add constraint KID primary key(ID); |

新建字段。

|  |
| --- |
| alter table userinfo  add grade varchar(10); |

first：第一个位置；

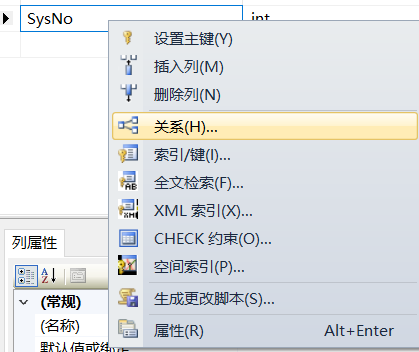
after：在哪个字段之后；默认在最后一个字段的后面。

|  |
| --- |
| alter table userinfo  add newkey FIRST; |

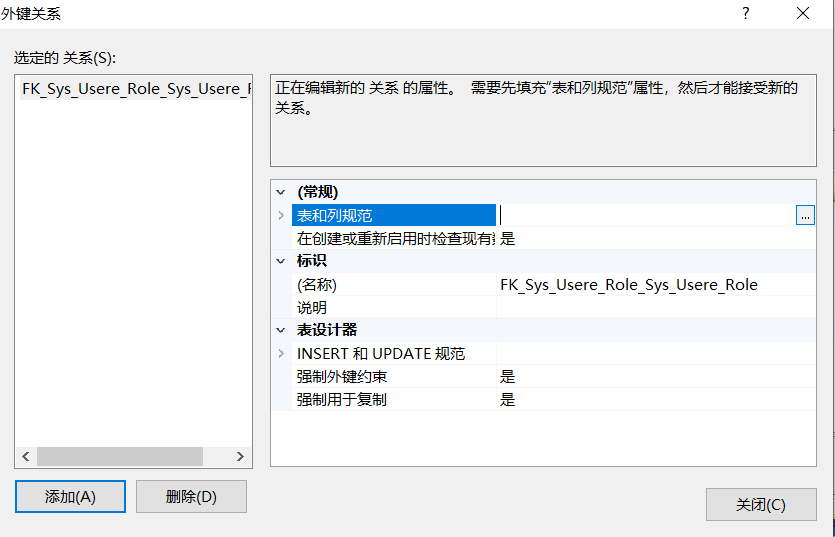
**3.5.3定义主键外键**

**3.5.3.1使用界面方法定义主键和外键**

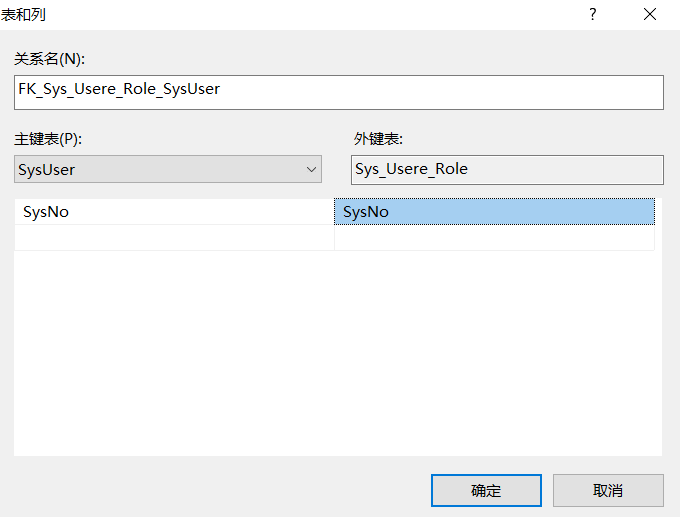
选择关系。



在弹出的对话框中添加关系。



选择主键表和外键表，并指定各自的主键和外键。这样两个表之间就构建了联系。



**3.5.3.2 SQL语句方法定义主键和外键**

|  |
| --- |
| alter table sys\_usere\_role  add constraint FK\_user\_role foreign key(RoleSysNo) references sys\_role(sysno); |

**3.6一些常用的SQL语句**

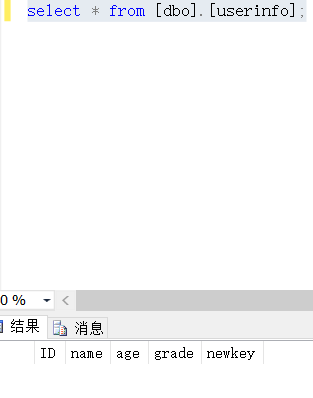
**3.6.1新增表记录**

新增表记录的时候，我们可以插入单行数据，也可以插入多行数据。

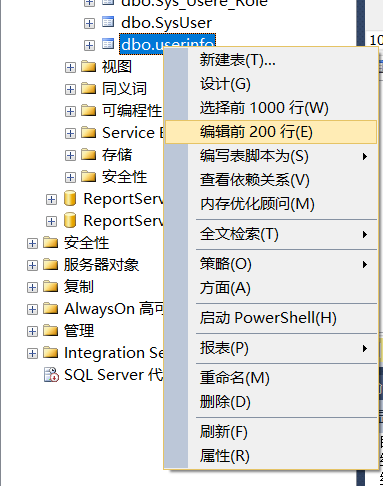
在讲述如何新增表记录之前，先了解一下如何查询表，这是一个基础的SQL语句。我们输入以下代码查询表的内容：

|  |
| --- |
| select \* from [dbo].[userinfo]; |

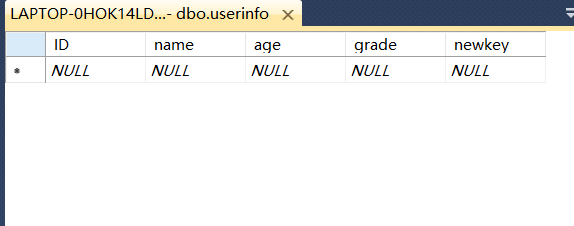
可以发现，当前是一个空表。



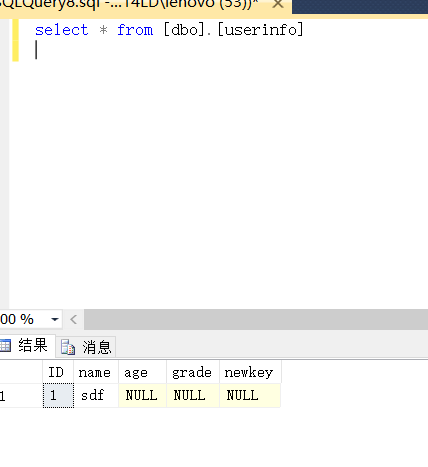
或者还可以从图形界面进行相关的查找。鼠标右键相应的表，选择编辑前200行。



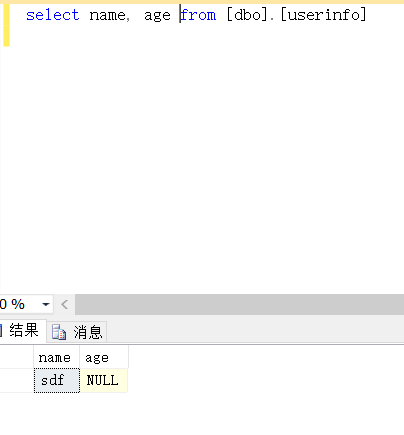
可以观察到，当前的表也显示为空。



|  |
| --- |
| select \* from [dbo].[userinfo] |

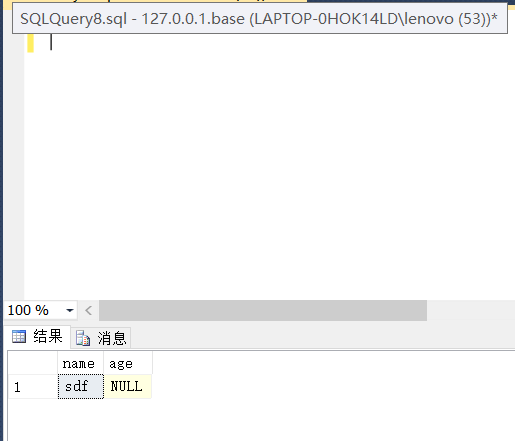


|  |
| --- |
| select name, age from [dbo].[userinfo] |



distinct/top的用法

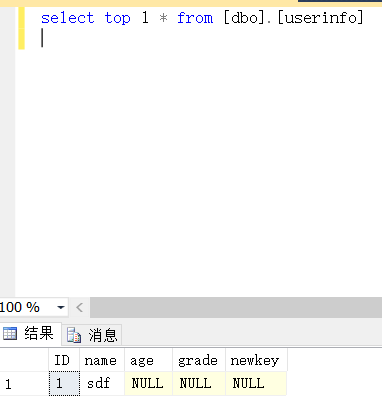
|  |
| --- |
| select distinct name, age from [dbo].[userinfo] |



|  |
| --- |
| select top 0 name, age from [dbo].[userinfo] |



|  |
| --- |
| select top 1 \* from [dbo].[userinfo] |



可以用SQL语句进行新增表记录:

|  |
| --- |
| insert into userinfo(age, grade, ID)  values(15, '一年级', 1031004); |

注意，要一一对应。

插入多行记录的方法类似，同样需要一一对齐。

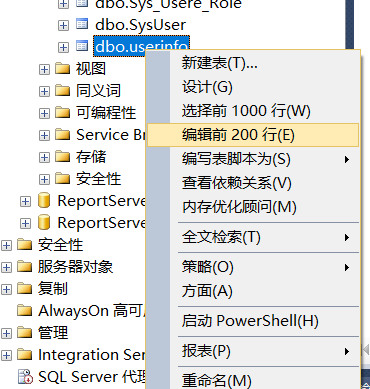
|  |
| --- |
| insert into userinfo(age, grade, ID)  values(15, '一年级', 1031004),  (18, '四年级', 1031005),  (10, '一年级', 1031006),  (13, '六年级', 1031007),  (90, '二年级', 1031048); |

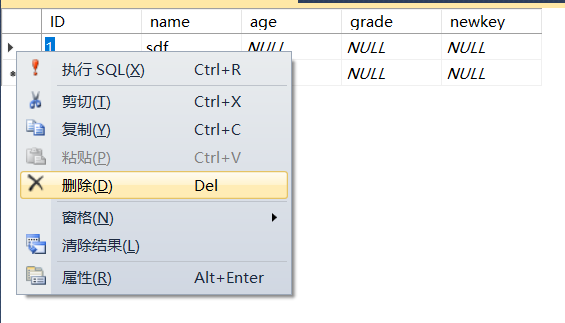
插入其他表的大数据集：

|  |
| --- |
| insert into userinfo(age, grade, ID)  select age, grade, email from [dbo][bigdata]; |

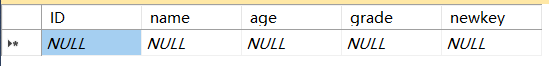
**3.6.2 SQL Server 删除表记录**

选择相应的表格，右键选择“编辑前200行”，即可在弹出的窗口中进行表记录的删除操作。





删除完毕：

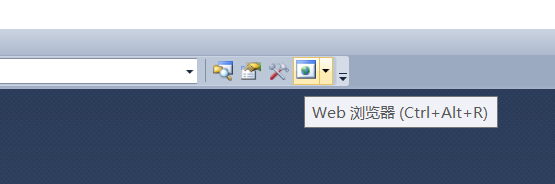


同样地，也可以使用SQL语句进行表记录的删除操作。

|  |
| --- |
| delete from [dbo].[userinfo]  where name = 'zhangsan'; |

**3.7熟悉在线帮助系统的使用**

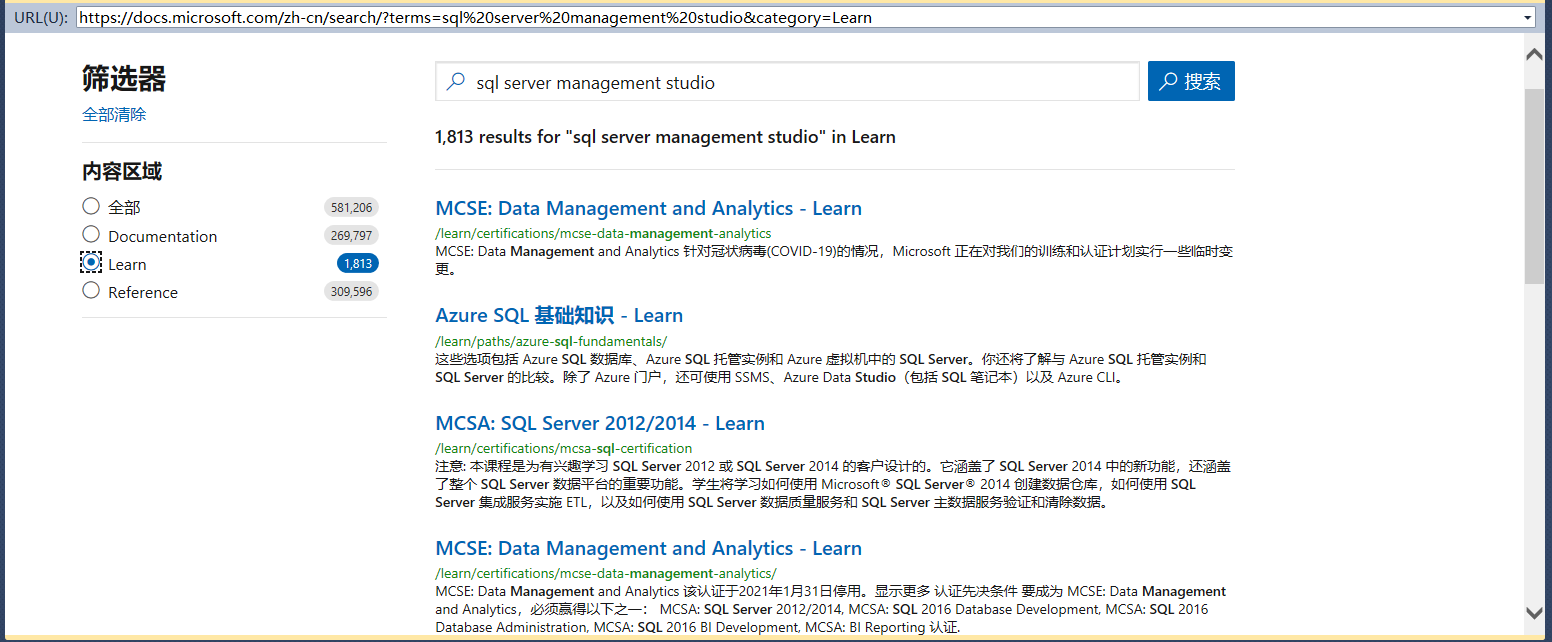
在SQL Server Management Studio窗口中单击“Web浏览器”，将弹出网络界面。



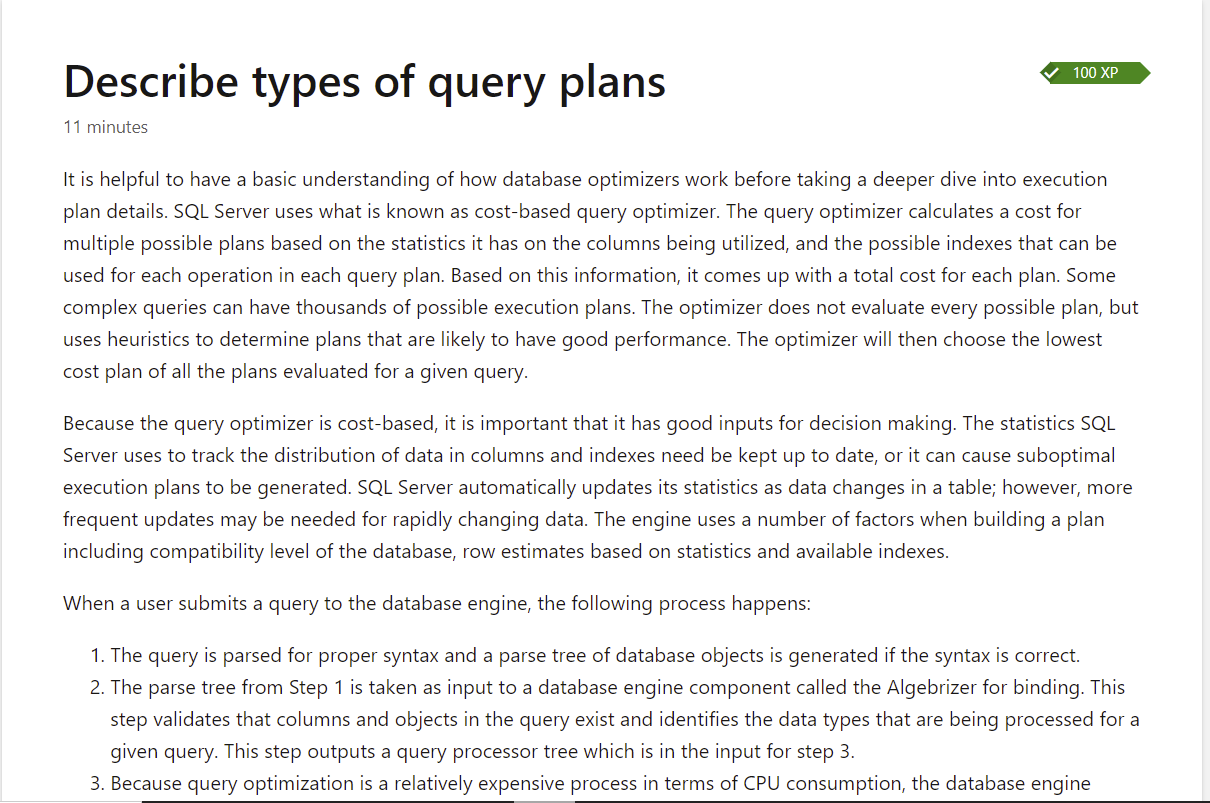
输入MSDN网站的网址，进入微软的在线帮助系统。



可以在搜索栏内搜索相应的问题，在线获取帮助。



可以在线获取很多的帮助文档以及学习的教程。



我学到了一些关于SQL语句中查询字段的原理和哈希值有关，并且基于此进行查找的优化。

