**实验2 SQL数据定义和操作**

**课程名称：数据库系统**

**专业：计算机科学与技术**

**学号：3190105359**

**姓名：段皞一**

**一 实验目的**

1. 掌握关系数据库语言SQL的使用
2. 使所有的SQL作业都能上机通过

**二 实验平台**

1. 数据库管理系统：SQL Server

**三 实验内容和要求**

1. 视图操作：通过视图的数据查询和数据修改

**3.1建立数据库**

打开Microsoft SQL Server Management Studio, 连接自己的服务器，建立一个基于教材的详细的大学模式数据库。



在完成数据库University的建立后，在数据库的表选项中添加相应的一系列表。也可以使用SQL语言直接在新建查询中进行表的新建。

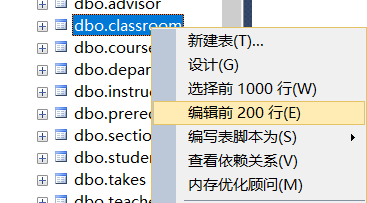
本次实验，采用SQL语句和图形界面结合的方式，进行表的创建。

**3.1.1创建表classroom**

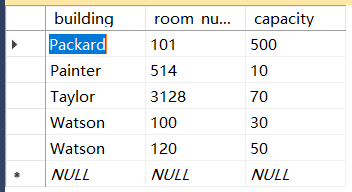
创建表“classroom”.相关代码如下所示：

|  |
| --- |
| create table classroom  (  building varchar(15),  room\_number varchar(7),  capacity numeric(4,0),  primary key (building, room\_number)  ); |

表建成以后，右键dbo.classroom单击选择“编辑前200行”选项，可以使用图形界面进行字段的添加。



完成添加字段。效果图如下：

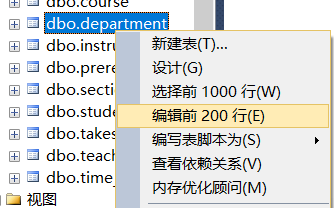


**3.1.2创建表department**

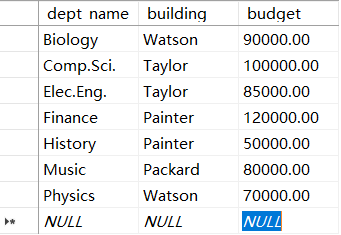
创建表”department”,相关SQL语句如下所示：

|  |
| --- |
| create table department  (  dept\_name varchar(20),  building varchar(20),  budget numeric(12,2) check (budget > 0),  primary key (dept\_name)  ); |

创建表成功后，可以在表的栏目下查找到”dbo.department”记录。右键以后，在弹出的窗口栏中选择“编辑前200行”选项，即可进行该表相应字段的添加。



该表字段添加完成以后的示意图如下所示：

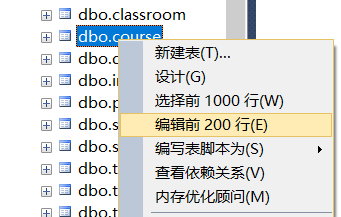


**3.1.3创建表course**

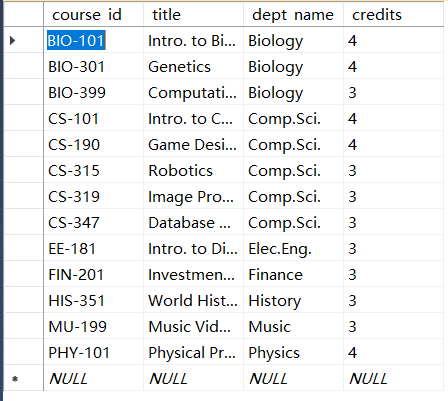
该表创建的相关代码如下所示：

|  |
| --- |
| create table course  (  course\_id varchar(8),  title varchar(50),  dept\_name varchar(20),  credits numeric(2, 0) check (credits > 0),  primary key (course\_id),  foreign key (dept\_name) references department  on delete set null  ); |

创建成功以后，可以在表窗口栏下查到dbo.course.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



添加成功后的效果图如下：

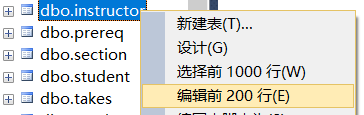


**3.1.4创建表instructor**

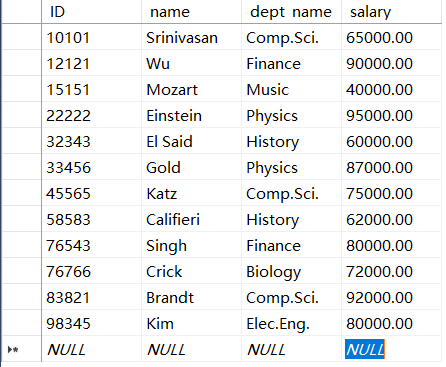
该表的创建代码如下所示：

|  |
| --- |
| create table instructor  (  ID varchar(5),  name varchar(20) not null,  dept\_name varchar(20),  salary numeric (8,2) check (salary > 29000),  primary key(ID),  foreign key (dept\_name) references department  on delete set null  ); |

创建成功以后，可以在表窗口栏下查到dbo.instructor.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



字段添加完成以后的效果图如下：

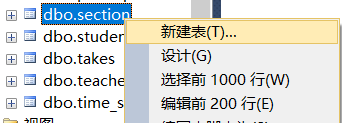


**3.1.5创建表section**

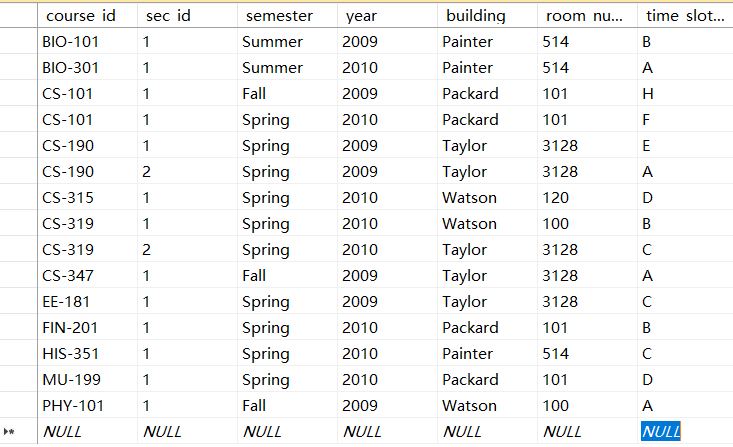
该表的创建的代码如下：

|  |
| --- |
| create table section  (  course\_id varchar(8),  sec\_id varchar(8),  semester varchar(6) check (semester in  ('Fall', 'Winter', 'Spring', 'Summer')),  year numeric(4, 0) check (year > 1701 and year < 2100),  building varchar(15),  room\_number varchar(7),  time\_slot\_id varchar(4),  primary key (course\_id, sec\_id, semester, year),  foreign key (course\_id) references course  on delete cascade,  foreign key (building, room\_number) references classroom  on delete set null  ); |

创建成功以后，可以在表窗口栏下查到dbo.section.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



添加字段以后的示意图如下：

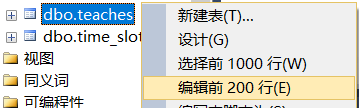


**3.1.6创建表teaches**

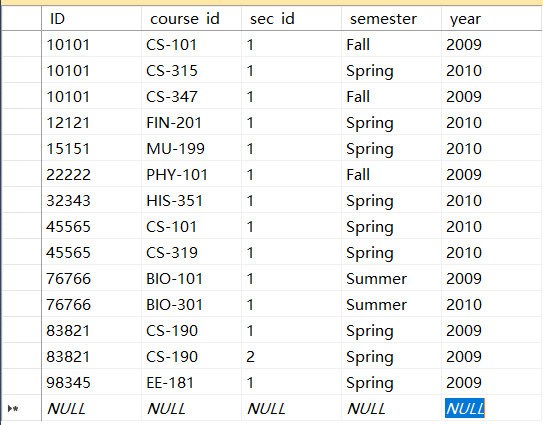
创建表的相关代码如下：

|  |
| --- |
| create table teaches  (  ID varchar(5),  course\_id varchar(8),  sec\_id varchar(8),  semester varchar(6),  year numeric(4, 0),  primary key (ID, course\_id, sec\_id, semester, year),  foreign key (course\_id, sec\_id, semester, year) references section  on delete cascade,  foreign key (ID) references instructor  on delete cascade  ); |

创建成功以后，可以在表窗口栏下查到dbo.teaches.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



添加字段完成以后的示意图如下：

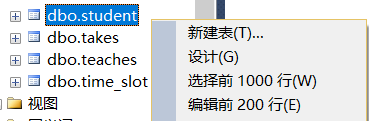


**3.1.7创建表student**

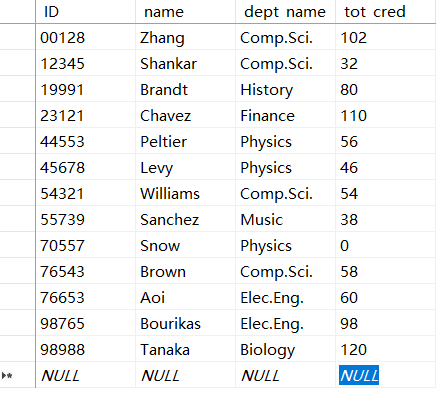
创建相关表的相应代码如下所示：

|  |
| --- |
| create table student  (  ID varchar(5),  name varchar(20) not null,  dept\_name varchar(20),  tot\_cred numeric(3, 0) check(tot\_cred >= 0),  primary key (ID),  foreign key (dept\_name) references department  on delete set null  ); |

创建成功以后，可以在表窗口栏下查到dbo.student.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



编辑字段成功后，可以看到相应的示意图如下：

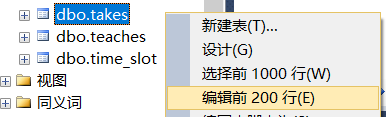


**3.1.8创建表takes**

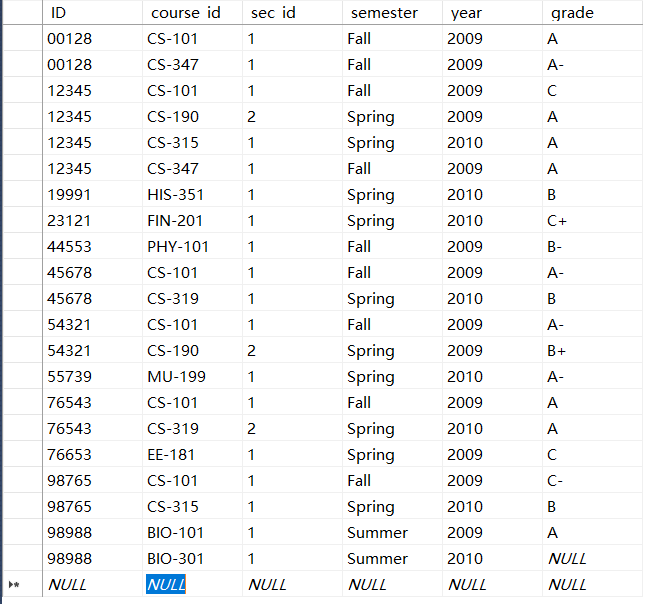
该表创建的相关代码如下所示：

|  |
| --- |
| create table takes  (  ID varchar(5),  course\_id varchar(8),  sec\_id varchar(8),  semester varchar(6),  year numeric(4, 0),  grade varchar(2),  primary key (ID, course\_id, sec\_id, semester, year),  foreign key (course\_id, sec\_id, semester, year) references section  on delete cascade,  foreign key(ID) references student  on delete cascade  ); |

创建成功以后，可以在表窗口栏下查到dbo.takes.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



完成字段的添加以后，相关示意图如下：

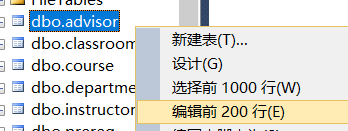


**3.1.9创建表advisor**

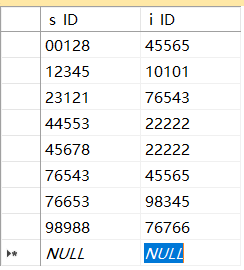
该表创建的相关代码如下：

|  |
| --- |
| create table advisor  (  s\_ID varchar(5),  i\_ID varchar(5),  primary key (s\_ID),  foreign key (i\_ID) references instructor(ID)  on delete set null,  foreign key (s\_ID) references student(ID)  on delete cascade  ); |

创建成功以后，可以在表窗口栏下查到dbo.advisor.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



完成字段的添加以后，相关示意图如下：

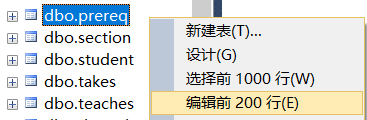


**3.1.10创建表prereq**

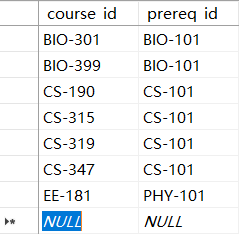
创建表的相关代码如下：

|  |
| --- |
| create table prereq  (  course\_id varchar(8),  prereq\_id varchar(8),  primary key (course\_id, prereq\_id),  foreign key (course\_id) references course  on delete cascade,  foreign key (prereq\_id) references course  ); |

创建成功以后，可以在表窗口栏下查到dbo.prereq.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



完成字段的添加以后，可以看到示意图如下：

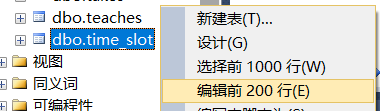


**3.1.11创建表time\_slot**

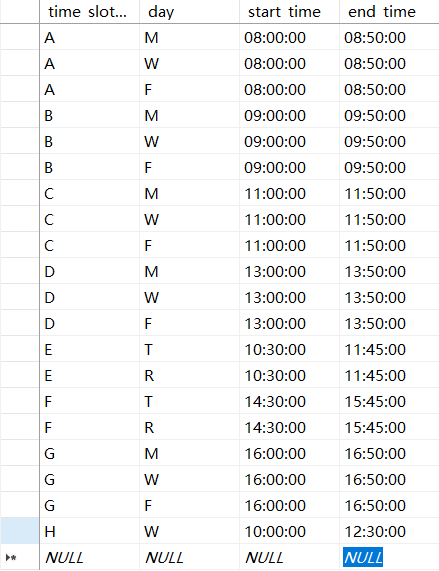
该表创建的相关代码如下：

|  |
| --- |
| create table time\_slot  (  time\_slot\_id varchar(4),  day varchar(1) check (day in ('M', 'T', 'W', 'R', 'F', 'S', 'U')),  start\_time time,  end\_time time,  primary key(time\_slot\_id, day, start\_time)  ); |

创建成功以后，可以在表窗口栏下查到dbo.time\_slot.右键以后将弹出窗口栏，选择“编辑前200行”选项，即可进行相应字段的添加。



完成该表字段的添加以后，可以看到相关的示意图如下：



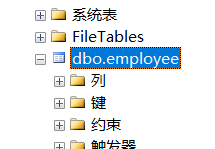
**3.2数据定义**

**3.2.1表的建立、删除和修改**

建立一个表，SQL语句如下所示：

|  |
| --- |
| create table employee  (  person\_name char(10) not null,  street char(10),  city char(10)  ); |

可以看到索引下多了一个dbo.employee.可见表创建成功。



修改一个表。相关的SQL语句的代码如下：

|  |
| --- |
| ALTER TABLE employee  add number\_id char(10) |

可以看到employee表下多了一个字段名number\_id.修改表成功。



实验中其他修改表的例子如下：

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); |

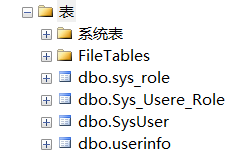
first：第一个位置；

after：在哪个字段之后；默认在最后一个字段的后面。

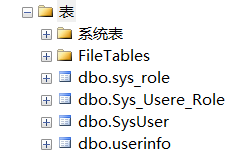
|  |
| --- |
| alter table userinfo  add newkey FIRST; |

删除一个表，相关的SQL语句如下所示：

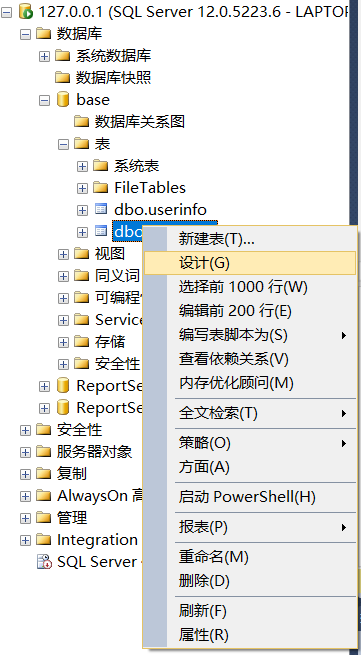
|  |
| --- |
| DROP TABLE employee |



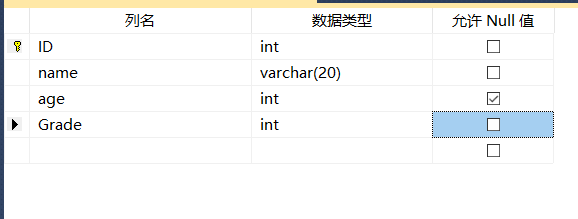
可以看到，索引下dbo.employee消失了。删除表成功。



通过视图方式修改表结构。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。



**3.2.2索引的建立、删除和修改**

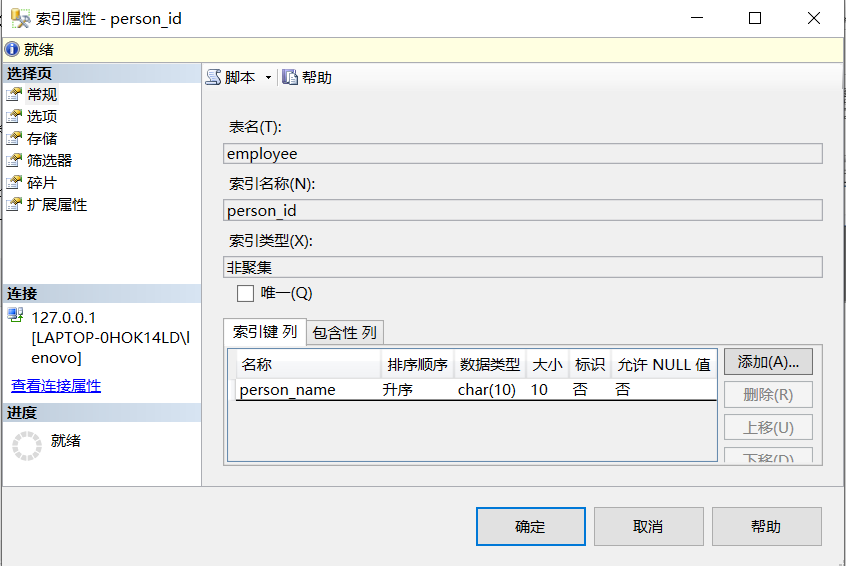
建立索引：

|  |
| --- |
| create index person\_id  on employee(person\_name) |

可以在搜索栏找到新建的索引。



之后还可以对索引的属性进行更改。



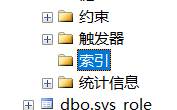
重建索引：

|  |
| --- |
| alter index person\_id on employee  rebuild |

删除索引：

|  |
| --- |
| drop index employee.person\_id |

可以看见，已经不存在索引person\_id.删除索引成功。



**3.2.3视图的建立、删除和修改**

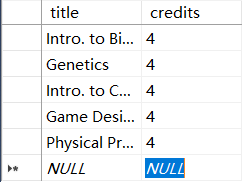
视图的建立，相关代码如下所示：

|  |
| --- |
| CREATE VIEW member  AS  SELECT title, credits  FROM course  WHERE credits = 4 |

可以看见视图栏下出现了dbo.member的视图，新建试图成功。



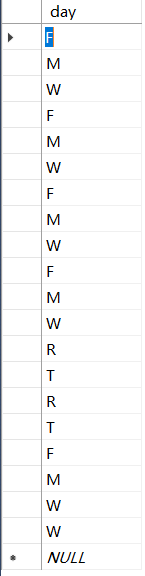
可以观察新建的视图。



视图的修改，相关代码如下：

|  |
| --- |
| ALTER view dbo.member  as  select day  from time\_slot  where start\_time is not null |

可以观察到member发生了改变，视图的修改成功。



**3.3数据更新**

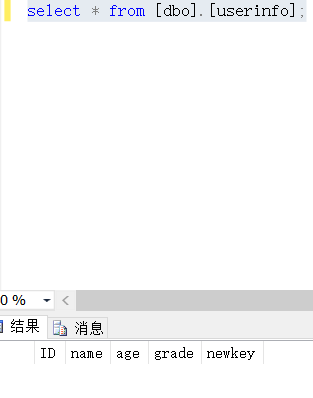
**3.3.1用insert命令插入表命令**

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。

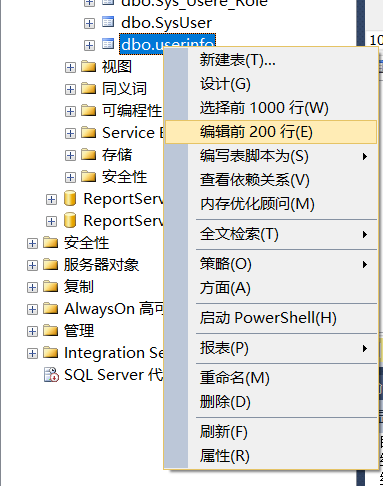
在讲述如何新增表记录之前，先了解一下如何查询表，这是一个基础的SQL语句。我们输入以下代码查询表的内容：

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

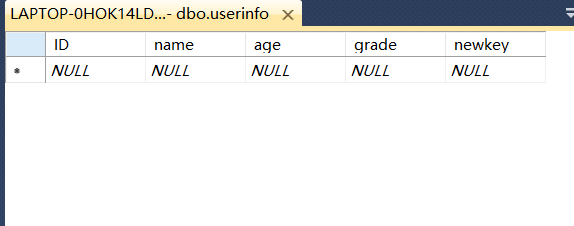
可以发现，当前是一个空表。



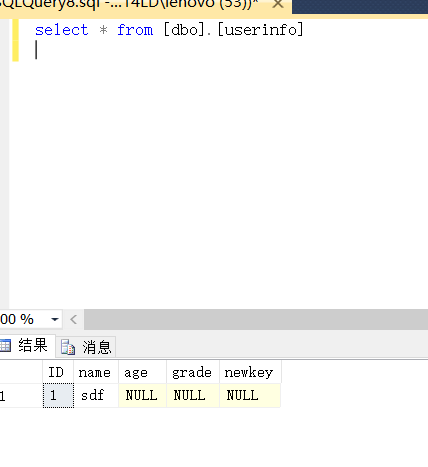
或者还可以从图形界面进行相关的查找。鼠标右键相应的表，选择编辑前200行。



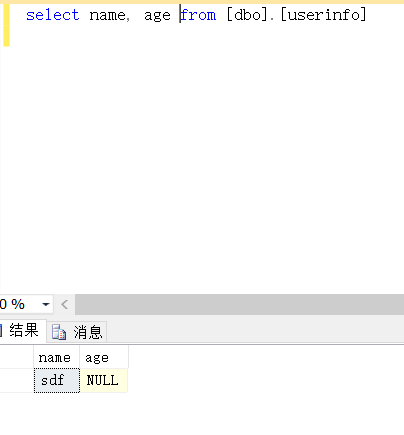
可以观察到，当前的表也显示为空。



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

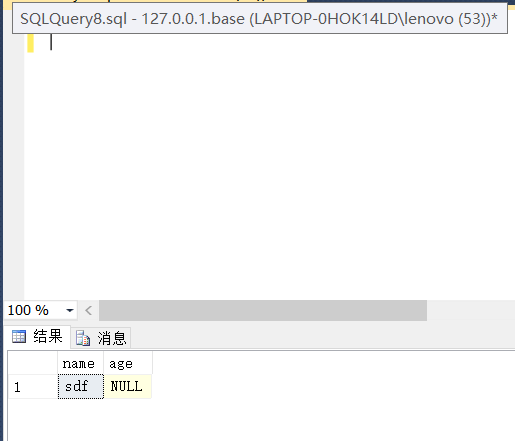


|  |
| --- |
| select name, age from [dbo].[userinfo] |



distinct/top的用法：

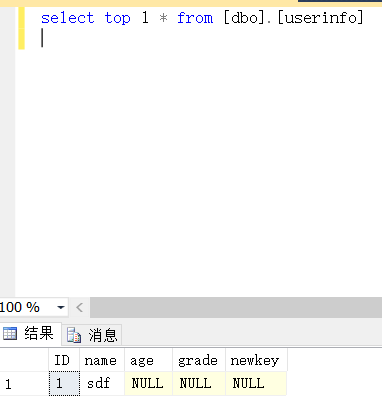
|  |
| --- |
| select distinct name, age from [dbo].[userinfo] |



|  |
| --- |
| 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，相关代码如下:

|  |
| --- |
| 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。

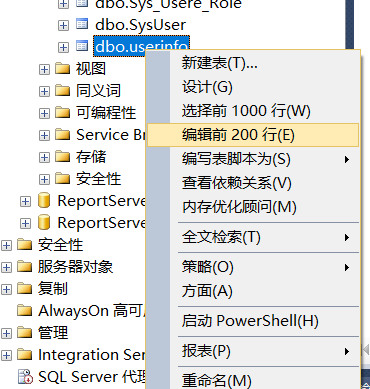
|  |
| --- |
| insert into userinfo(age, grade, ID)  values(15, '一年级', 1031004),  (18, '四年级', 1031005),  (10, '一年级', 1031006),  (13, '六年级', 1031007),  (90, '二年级', 1031048); |

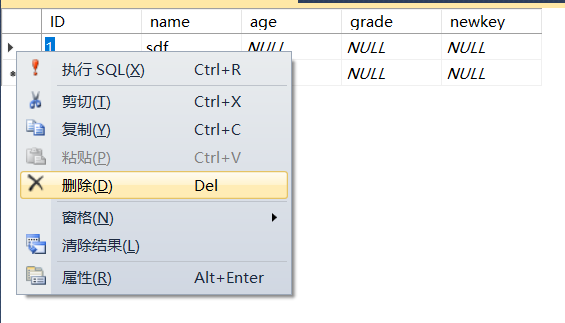
插入其他表的大数据集：

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

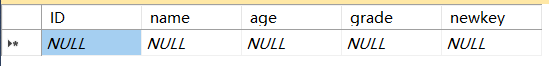
**3.3.2用delete命令删除表命令**

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'; |

**3.3.3用updata命令修改表数据**

|  |
| --- |
| update department  set building = 'Watson'  where dept\_name = 'Physics' |

观察到physics的系大楼改到了’Watsond’.



**3.4数据查询**

**3.4.1单表查询**

单表查询的代码案例：

|  |
| --- |
| //选出ID=-1的1字段  select \* from [dbo].[userinfo]  where ID = -1;  //选出RoleName以“管理员”作为结尾的字段  select \* from [dbo].[userinfo]  where RoleName like '%管理员'  //选出RoleName以“仓库”作为开头的字段  select \* from [dbo].[userinfo]  where RoleName like '仓库%'  //选出RoleName包含“中心”的字段  select \* from [dbo].[userinfo]  where RoleName like '%中心%'  //选出name是三个字节的字段  select \* from [dbo].[userinfo]  where name like '\_\_\_' |

Between用法。限制条件表达式，指定表达式范围值。

|  |
| --- |
| select \* from [dbo].[userinfo]  where name between 100 and 200;  select \* from [dbo].[userinfo]  where name between '9DDBL006' and '9GDBL006';  select \* from [dbo].[userinfo]  where time between '2015-07-29 12:00:00.001' and '2019-5-18 23:59:59.999'; |

In语句的用法实例如下：

|  |
| --- |
| select \* from [dbo].[userinfo]  where StudantNo in (1503, 1501, 1505); |

取反的用法如下：

|  |
| --- |
| select \* from [dbo].[userinfo]  where StudantNo not in (1503, 1501, 1505);  select \* from [dbo].[userinfo]  where StudantName not in ('张三', '李四', '王五');  select \* from [dbo].[Students]  where StudantName in (select studentname from [dbo].[Student\_Lesson]); |

exists语法，exits语句返回布尔值True 或者 False。

|  |
| --- |
| select a.StudentNo, a.StudentName, a.Age from [dbo].[Students] a  where exists (select ID from [dbo].[Student\_Lesson] b  where a.StudentNo = b.StudentNo); |

返回记录排序，示例代码如下：

|  |
| --- |
| select \* from [dbo].[Students]  order by parentid;  select \* from [dbo].[Students]  order by parentid desc;  select \* from [dbo].[Students]  order by parentid, ordernum; |

表关联查询，交叉关联示例代码：

|  |
| --- |
| select \* from [dbo].[Students] a  inner join [dbo].[Classes] b  on a.ClassesNo = b.ClassNo; |

左关联：左表的全部数据和右表满足关联的数据。

|  |
| --- |
| select \* from [dbo].[Students] a  left join [dbo].[Classes] b  on a.ClassesNo = b.ClassNo; |

聚合函数的用法：

|  |
| --- |
| select avg(Score) from [dbo].[Score]  select sum(Score) from [dbo].[Score]  select min(Score) as SumScore from [dbo].[Score]  select max(Score) as SumScore from [dbo].[Score] |

取别名：

|  |
| --- |
| select sum(Score) as SumScore from [dbo].[Score] |

LEN()函数，返回指定字符串表达式的字符数：

|  |
| --- |
| select \*, LEN() as len\_grade from [dbo].[Classes] |

DATALENGTH()返回表达式的字节数：

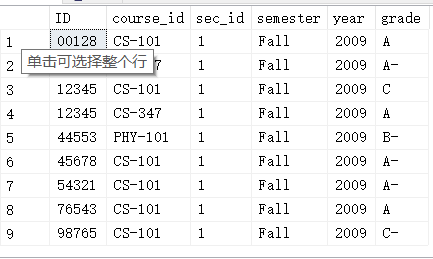
|  |
| --- |
| select \*, DATALENGTH() as len\_grade from [dbo].[Classes] |

随机数的产生：

|  |
| --- |
| select rand();  select floor(rand()\*10);  //  select rand();  select ceiling(rand()\*1000); |

具体的案例，查询秋季开课的字段。

|  |
| --- |
| select \*  from takes  where semester = 'Fall' |

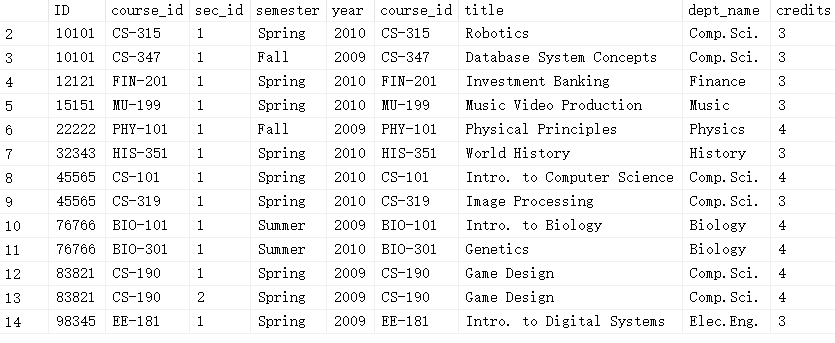


**3.4.2多表查询**

多表查询的示例如下：

|  |
| --- |
| select \*  from teaches, course  where teaches.course\_id = course.course\_id |

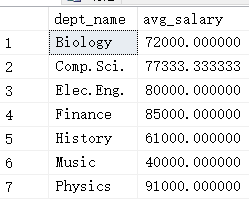
示意图如下：



**3.4.3嵌套子表查询**

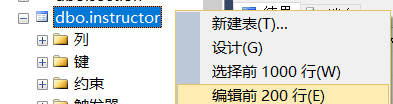
|  |
| --- |
| select dept\_name, avg\_salary  from( select dept\_name, avg(salary) as avg\_salary  from instructor  group by dept\_name)  where avg\_salary > 30000 |

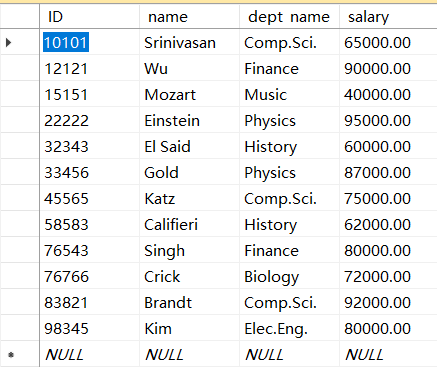
执行该嵌套查询语句后的示意图如下所示：



**3.5视图操作**

通过视图的数据查询和数据修改方法是，右键相应的表，选择“编辑前200行”选项，即可进行查询和修改操作。





**3.6其他操作**

时间的获取：

|  |
| --- |
| //格式1  select getdate()  //格式2  select getutcdate() |

效果如下：





Convert函数：把日期转换成新的数据类型。

|  |
| --- |
| //格式1  select convert(varchar(10), getdate(), 110);  //格式2  select convert(varchar(10), getdate(), 111);  //格式3  select convert(varchar(20), getdate(), 120); |





