实验二　数据库表/视图的创建与维护实验

班级：2015211305　　姓名：顾溢杰　　学号：2015211257

实验目的

1. 通过进行数据库表的建立操作，熟悉并掌握 SQL SERVER 数据库表的建立方法，理解关系数据库表的结构，巩固 SQL 标准中关于数据库表的建立语句；
2. 通过进行数据库表数据的增加、删除和插入等维护操作，熟悉并掌握 SQL SERVER 数据库数据的操作方法，巩固 SQL 中关于数据维护的语句；
3. 通过对 SQL SERVER 中建立、维护视图的实验，熟悉 SQL SERVER 中对视图的操作方法和途径，理解和掌握视图的概念。

实验内容

1. 熟悉课程实验背景知识；
2. 根据物理模式使用 SQL Server 创建相应的表；
3. 将教师提供的数据导入表；
4. 对每个表都尝试增加、删除或者修改一条记录，看看是否能够成功，原因何在？
5. 在 SQL SERVER 中建立、维护视图。

实验环境

两台装有 Windows 10 操作系统和 Microsoft SQL Server 2008 R2 的笔记本电脑，一台充当服务器，一台充当客服端。

实验步骤、结果及分析

创建数据库和表

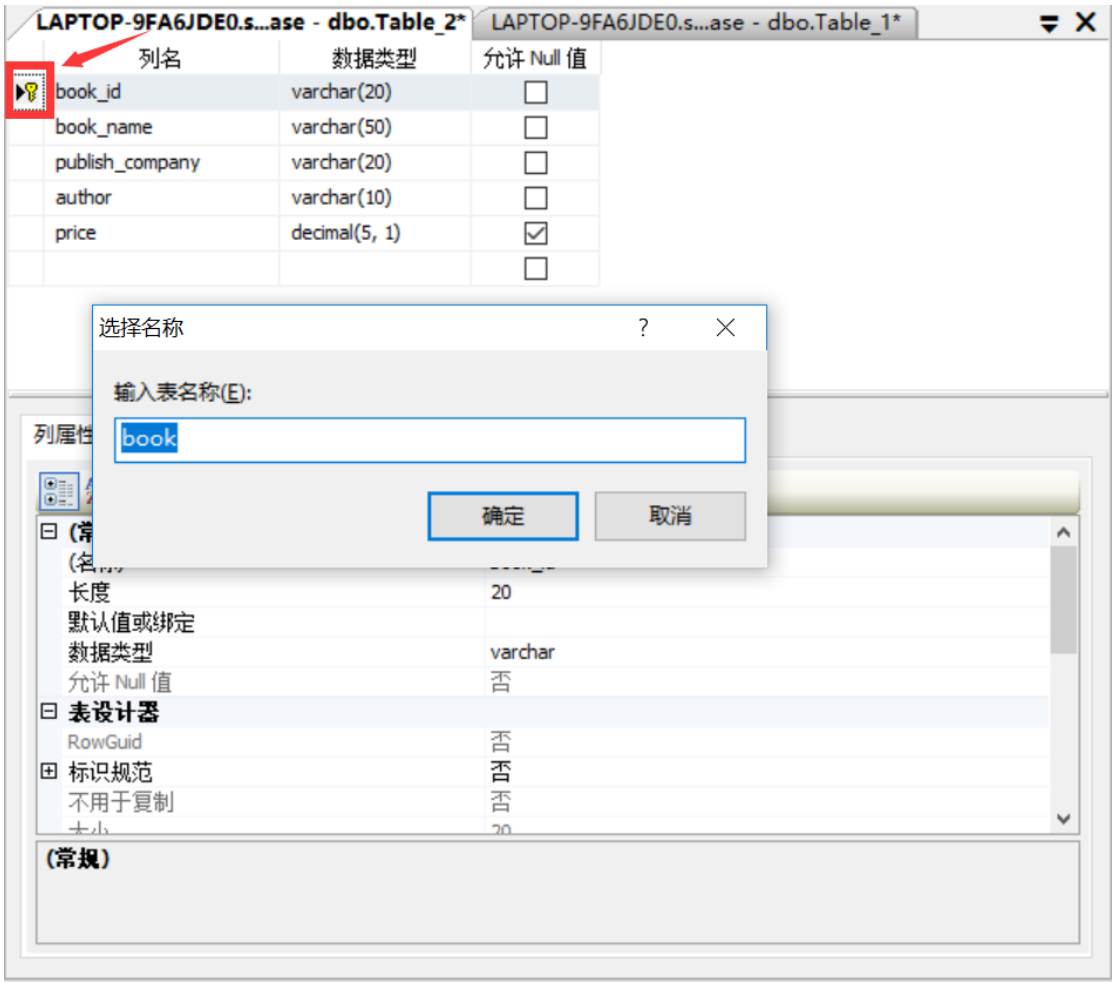
实验一中已创建 student_class_database 数据库。接下来创建各表。

book 表

属性名	类型	是否可以为 NULL
[主键]book_id	varchar(20)	否
book_name	varchar(50)	否
publish_company	varchar(50)	否
author	varchar(10)	否

price	decimal(5,1)	是

该表利用图形化界面进行创建。我们在对象资源管理器的 student_class_database 数据库中的表条目上点击右键，选择新建表，弹出新建表的界面，然后填入各列信息。右键点击图中箭头所指取余，选择设置为主键。然后 crtl+s 保存，输入表名，再点击确定，即可创建成功。图中箭头所指处的图标显示 book_id 已经被设置为了主键。



course 表

属性名	类型	是否可以为 NULL
[主键]course_id	varchar(20)	否
course_name	varchar(50)	否
book_id	varchar(20)	是
total_perior	smallint	否
week_perior	smallint	否
credit	tinyint	是

除了要将 course_id 设置为主键，还需将 book_id 设置为该表相对于 book 表的外键。
该表利用 SQL 脚本创建，代码如下：

```
create table course(
    course_id varchar(20) not null,
    course_name varchar(50) not null,
```

```
    book_id varchar(20),
    total_perior smallint not null,
    week_perior smallint not null,
    credit tinyint,
    primary key(course_id)
);
go
```

class 表

属性名	类型	是否可以为 NULL
[主键]class_id	varchar(10)	否
monitor	varchar(10)	是
classroom	varchar(20)	否
department_id	varchar(10)	否

SQL 脚本如下：

```
create table class(
    class_id varchar(10) not null,
    monitor varchar(10),
    classroom varchar(20) not null,
    department_id varchar(10) not null,
    primary key(class_id)
);
go
```

class_course 表

属性名	类型	是否可以为 NULL
[主键]class_id	varchar(10)	否
[主键]course_id	varchar(20)	否

特别需要注意的是，该表中的主键由 class_id 和 course_id 共同构成。
SQL 脚本如下：

```
create table class_course(
    class_id varchar(10) not null,
    course_id varchar(20) not null,
    primary key(class_id,course_id)
);
go
```

department 表

属性名	类型	是否可以为 NULL
[主键]department_id	varchar(10)	否
department_name	varchar(50)	否

department_header	varchar(10)	是
teacher_num	smallint	是

SQL 脚本如下：

```sql
create table department(
    department_id varchar(10) not null,
    department_name varchar(50) not null,
    department_header varchar(10),
    teacher_num smallint,
    primary key(department_id)
);
go
```

student 表

属性名	类型	是否可以为 NULL
[主键]student_id	varchar(10)	否
student_name	varchar(10)	否
sex	varchar(2)	否
birth	date	是
class_id	varchar(10)	否
entrance_date	date	否
home_addr	varchar(200)	是

SQL 脚本如下：

```sql
create table student(
    student_id varchar(10) not null,
    student_name varchar(10) not null,
    sex varchar(2) not null,
    birth date,
    class_id varchar(10) not null,
    entrance_date date not null,
    home_addr varchar(200),
    primary key(student_id)
);
go
```

student_course 表

属性名	类型	是否可以为 NULL
[主键]course_id	varchar(20)	否
[主键]student_id	varchar(10)	否
grade	tinyint	是
credit	tinyint	否
semester	tinyint	否

school_year	varchar(10)	否

SQL 脚本如下：

```
create table student_course(
    course_id varchar(20) not null,
    student_id varchar(10) not null,
    grade tinyint,
    credit tinyint not null,
    semester tinyint not null,
    school_year varchar(10) not null,
    primary key(course_id,student_id)
);
go
```

teacher 表

属性名	类型	是否可以为 NULL
[主键]teacher_id	varchar(10)	否
teacher_name	varchar(10)	否
sex	varchar(2)	否
birth	date	是
department_id	varchar(10)	否
profession	varchar(10)	否
telephone	decimal(15,0)	否
home_addr	varchar(200)	是
postalcode	varchar(10)	是

SQL 脚本如下：

```
create table teacher(
    teacher_id varchar(10) not null,
    teacher_name varchar(10) not null,
    sex varchar(2) not null,
    birth date,
    department_id varchar(10) not null,
    profession varchar(10) not null,
    telephone decimal(15,0) not null,
    home_addr varchar(200),
    postalcode varchar(10),
    primary key(teacher_id)
);
go
```

teacher_course_class 表

属性名	类型	是否可以为 NULL

[主键]teacher_id	varchar(10)	否
[主键]course_id	varchar(20)	否
[主键]class_id	varchar(10)	否
semester	tinyint	否
school_year	varchar(10)	否
course_schedule	decimal(10,0)	否
course_classroom	varchar(10)	否
book_id	varchar(20)	是

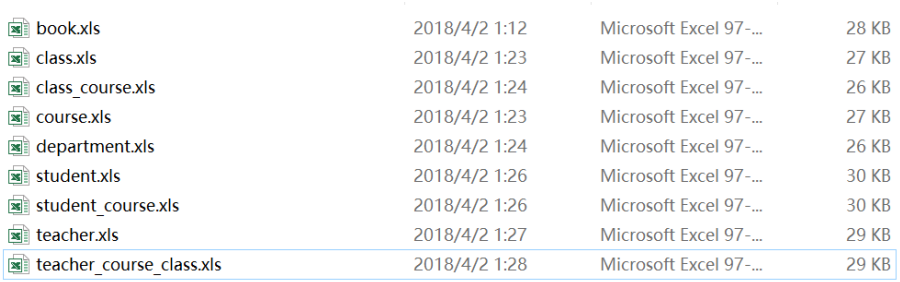
SQL 脚本如下：

```sql
create table teacher_course_class(
    teacher_id varchar(10) not null,
    course_id varchar(20) not null,
    class_id varchar(10) not null,
    semester tinyint not null,
    school_year varchar(10) not null,
    course_schedule decimal(10,0) not null,
    course_classroom varchar(10) not null,
    book_id varchar(20),
    primary key(teacher_id,course_id,class_id)
);
go
```

导入表数据

SQL 提供的导入数据的方法很多。可以通过 SQL 语句一条记录、一条记录的插入；也可以通过图形化表格界面键入；还可以直接从 EXCEL 导入一个表（在数据库中新建一个表或者添加到数据库中已有的表中）；也可以通过 SQL 语句，从 EXCEL 中将数据插入到已有的表中。我们将采用第四种方法。

老师提供的表中，所有的表混合在了一个 sheet 中，因此我们先将其拆分成 9 个 sheet，或者 9 个 EXCEL 文件。



book.xls	2018/4/2 1:12	Microsoft Excel 97-...	28 KB
class.xls	2018/4/2 1:23	Microsoft Excel 97-...	27 KB
class_course.xls	2018/4/2 1:24	Microsoft Excel 97-...	26 KB
course.xls	2018/4/2 1:23	Microsoft Excel 97-...	27 KB
department.xls	2018/4/2 1:24	Microsoft Excel 97-...	26 KB
student.xls	2018/4/2 1:26	Microsoft Excel 97-...	30 KB
student_course.xls	2018/4/2 1:26	Microsoft Excel 97-...	30 KB
teacher.xls	2018/4/2 1:27	Microsoft Excel 97-...	29 KB
teacher_course_class.xls	2018/4/2 1:28	Microsoft Excel 97-...	29 KB

例如 book.xls 内部数据如图所示

book_id	book_name	publish_company	author	price
dep01_s001_01	电磁波工程	电子工业出版社	顾华	21
dep04_b001_01	计算机基础	清华大学出版社	洪涛	16
dep04_b001_02	计算机应用	电子工业出版社	李群	19.8
dep04_p001_01	C语言程序设计	清华大学出版社	钟军	18.8
dep04_s001_01	SQL Server数据库开发技术	北方交通出版社	成虎	21.5
dep04_s002_01	JAVA语言程序设计	东南大学出版社	王平	22.5
dep04_s003_01	单片机原理	东南大学出版社	肖红	16.8
dep04_s004_01	软件开发技术	南京大学出版社	刘雨	15
dep04_s005_01	网页设计	地质出版社	张凯芝	12
dep04_s006_01	计算机网络	机械工业出版社	张旸	23.5
dep04_s007_01	计算机导论	北京邮电大学出版社	周艳	28
dep01_s002_01	计算机网络技术与实践	人民邮电出版社	潘晓	38

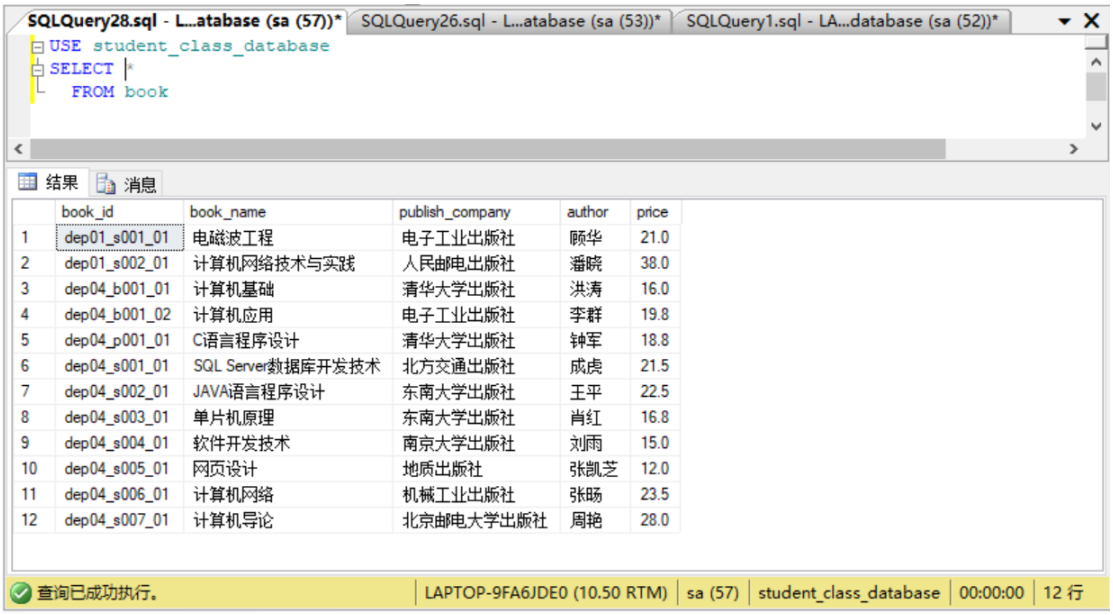
启用 Ad Hoc Distributed Queries

```
exec sp_configure 'show advanced options',1
reconfigure
exec sp_configure 'Ad Hoc Distributed Queries',1
reconfigure
```

从 book.xls 导入数据到 book 表中

```
INSERT INTO book
SELECT *
FROM OPENROWSET('MICROSOFT.JET.OLEDB.4.0', 'EXCEL 5.0;HDR=YES;DATABASE=C:\Users\16417\Desktop\notation\database\courses\实验\实验二\数据\book.xls', sheet1$)
```

编写 SQL 脚本查询 book 表中数据，如下图，表中数据无误，导入成功。



	book_id	book_name	publish_company	author	price
1	dep01_s001_01	电磁波工程	电子工业出版社	顾华	21.0
2	dep01_s002_01	计算机网络技术与实践	人民邮电出版社	潘晓	38.0
3	dep04_b001_01	计算机基础	清华大学出版社	洪涛	16.0
4	dep04_b001_02	计算机应用	电子工业出版社	李群	19.8
5	dep04_p001_01	C语言程序设计	清华大学出版社	钟军	18.8
6	dep04_s001_01	SQL Server数据库开发技术	北方交通出版社	成虎	21.5
7	dep04_s002_01	JAVA语言程序设计	东南大学出版社	王平	22.5
8	dep04_s003_01	单片机原理	东南大学出版社	肖红	16.8
9	dep04_s004_01	软件开发技术	南京大学出版社	刘雨	15.0
10	dep04_s005_01	网页设计	地质出版社	张凯芝	12.0
11	dep04_s006_01	计算机网络	机械工业出版社	张旸	23.5
12	dep04_s007_01	计算机导论	北京邮电大学出版社	周艳	28.0

其余表格用相同方法导入，不再赘述。

数据库表数据的增加、删除、修改

本步骤中，前四个表的操作在本机上进行；后五个表的操作通过客户端远程连接到本机进行。首先介绍远程连接的相关配置。

远程连接配置

1. 在 sql server 配置管理器中启动 sql server browser 服务。如果无法启动，到 windows 10 的服务管理中启用（为了方便，选择自动启动）。
2. 在 windows 防火墙设置中开放本机的 1433 端口。在入站规则中添加 1433 端口即可。
3. 在 windows 防火墙例外中添加 sqlservr.exe 程序（在 SQL Server 的安装目录中）
4. 重启 sql server 中的相关服务或者重启服务器主机。
5. 保证服务器主机网络通畅后，在客户端主机上远程连接。输入服务器主机的 ip 地址，选择 SQL Server 身份验证，输入用户名和密码即可登陆成功。

接下来是在本机进行的前四个表的操作。

book 表

采用图形化界面操作进行表内记录的增加、删除、修改。

1. 删除

在对象资源管理器中选中 book 表，右键单击选择编辑前 200 行。

	book_id	book_name	publish_company	author	price
	dep01_s001_01	电磁波工程	电子工业出版社	顾华	21.0
	dep01_s002_01	计算机网络技...	人民邮电出版社	潘晓	38.0
	dep04_b001_01	计算机基础	清华大学出版社	洪涛	16.0
▶	dep04_b001_02	计算机应用	电子工业出版社	李群	19.8
	dep04_p001_01	C语言程序设计	清华大学出版社	钟军	18.8
	dep04_s001_01	SQL Server数据...	北方交通出版社	成虎	21.5
	dep04_s002_01	JAVA语言程序...	东南大学出版社	王平	22.5
	dep04_s003_01	单片机原理	东南大学出版社	肖红	16.8
	dep04_s004_01	软件开发技术	南京大学出版社	刘雨	15.0
	dep04_s005_01	网页设计	地质出版社	张凯芝	12.0
	dep04_s006_01	计算机网络	机械工业出版社	张旸	23.5
	dep04_s007_01	计算机导论	北京邮电大学...	周艳	28.0
*	NULL	NULL	NULL	NULL	NULL

在上图所示的编辑界面中，点击最左边的空白区域即可选中某一行，右键单击，选择删除，即可删除该行的记录。也可以选中多行。

	book_id	book_name	publish_company	author	price
	dep01_s001_01	电磁波工程	电子工业出版社	顾华	21.0
	dep01_s002_01	计算机网络技...	人民邮电出版社	潘晓	38.0
	dep04_b001_01	计算机基础	清华大学出版社	洪涛	16.0
	dep04_p001_01	C语言程序设计	清华大学出版社	钟军	18.8
	dep04_s001_01	SQL Server数据...	北方交通出版社	成虎	21.5
	dep04_s002_01	JAVA语言程序...	东南大学出版社	王平	22.5
	dep04_s003_01	单片机原理	东南大学出版社	肖红	16.8
	dep04_s004_01	软件开发技术	南京大学出版社	刘雨	15.0
	dep04_s005_01	网页设计	地质出版社	张凯芝	12.0
	dep04_s006_01	计算机网络	机械工业出版社	张旸	23.5
	dep04_s007_01	计算机导论	北京邮电大学...	周艳	28.0
▸*	NULL	NULL	NULL	NULL	NULL

从这图可见，《计算机应用》这本书的记录已经成功被我们删除。

2. 增加

接下来我们在下面显示为 NULL 的行中将该记录添加回去。

	book_id	book_name	publish_company	author	price
	dep01_s001_01	电磁波工程	电子工业出版社	顾华	21.0
	dep01_s002_01	计算机网络技...	人民邮电出版社	潘晓	38.0
	dep04_b001_01	计算机基础	清华大学出版社	洪涛	16.0
	dep04_p001_01	C语言程序设计	清华大学出版社	钟军	18.8
	dep04_s001_01	SQL Server数据...	北方交通出版社	成虎	21.5
	dep04_s002_01	JAVA语言程序...	东南大学出版社	王平	22.5
	dep04_s003_01	单片机原理	东南大学出版社	肖红	16.8
	dep04_s004_01	软件开发技术	南京大学出版社	刘雨	15.0
	dep04_s005_01	网页设计	地质出版社	张凯芝	12.0
	dep04_s006_01	计算机网络	机械工业出版社	张旸	23.5
	dep04_s007_01	计算机导论	北京邮电大学...	周艳	28.0
▸	dep04_b001_02	计算机应用	电子工业出版社	李群	19.8
*	NULL	NULL	NULL	NULL	NULL

从上图可见，该记录被我们成功添加了回去。

3. 修改

我们也可以修改某一条记录，例如，将电磁波工程这本书的价格改为 25.5

	book_id	book_name	publish_company	author	price
	dep01_s001_01	电磁波工程	电子工业出版社	顾华	25.5
	dep01_s002_01	计算机网络技...	人民邮电出版社	潘晓	38.0
	dep04_b001_01	计算机基础	清华大学出版社	洪涛	16.0
	dep04_p001_01	C语言程序设计	清华大学出版社	钟军	18.8
	dep04_s001_01	SQL Server数据...	北方交通出版社	成虎	21.5
	dep04_s002_01	JAVA语言程序...	东南大学出版社	王平	22.5
	dep04_s003_01	单片机原理	东南大学出版社	肖红	16.8
	dep04_s004_01	软件开发技术	南京大学出版社	刘雨	15.0
	dep04_s005_01	网页设计	地质出版社	张凯芝	12.0
	dep04_s006_01	计算机网络	机械工业出版社	张旸	23.5
	dep04_s007_01	计算机导论	北京邮电大学...	周艳	28.0
	dep04_b001_02	计算机应用	电子工业出版社	李群	19.8
*	NULL	NULL	NULL	NULL	NULL

1 / 12 单元格已修改。

course 表

采用 SQL 脚本。

1. 增加

我们向 course 表中增加一条记录

course_id	course_name	book_id	total_perior	week_perior	credit
dep04_j001	计算机系统结构	dep04_j001_01	68	4	6

SQL 脚本如下：

```
USE student_class_database
INSERT INTO course VALUES('dep04_j001','计算机系统结构','dep04_j001_02',68,4,6);
SELECT *
FROM course;
```

结果如下：

	course_id	course_name	book_id	total_perior	week_perior	credit
1	dep01_s001	电磁波工程	dep01_s001_01	80	6	6
2	dep01_s002	网络技术与实践	dep01_s002_01	68	4	4
3	dep04_b001	计算机基础	dep04_b001_01	68	4	4
4	dep04_j001	计算机系统结构	dep04_j001_02	68	4	6
5	dep04_p001	C语言程序设计	dep04_p001_01	68	4	4
6	dep04_s001	SQL Server数据库开发技术	dep04_s001_01	80	5	5
7	dep04_s002	JAVA程序设计与开发	dep04_s002_01	68	4	4
8	dep04_s003	单片机原理	dep04_s003_01	68	4	4
9	dep04_s004	软件开发技术	dep04_s004_01	51	3	3
10	dep04_s005	网页设计	dep04_s005_01	80	5	5
11	dep04_s006	计算机网络	dep04_s006_01	60	3	3
12	dep04_s007	计算机导论	dep04_s007_01	80	4	6

2. 删除

我们再将计算机系统结构这本书删除，以其 course_id 作为筛选条件。

SQL 脚本如下：

```
USE student_class_database
DELETE FROM course WHERE course_id='dep04_j001';
SELECT *
FROM course;
```

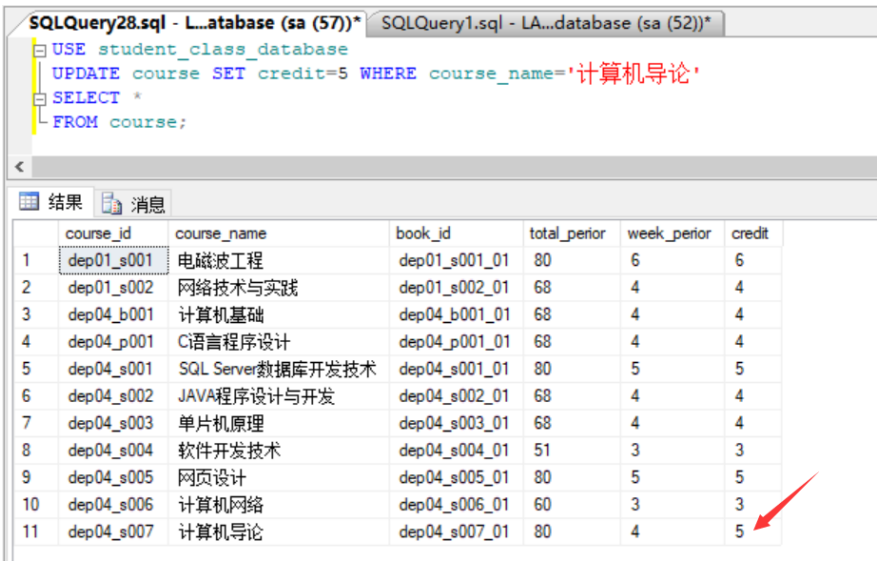
3. 修改

我们修改计算机导论的 credit 属性取值为 5（原先为 6），采用 course_name 作为筛选条件。

SQL 脚本如下：

```
USE student_class_database
UPDATE course SET credit=5 WHERE course_name='计算机导论'
SELECT *
FROM course;
```

结果如下：



	course_id	course_name	book_id	total_perior	week_perior	credit
1	dep01_s001	电磁波工程	dep01_s001_01	80	6	6
2	dep01_s002	网络技术与实践	dep01_s002_01	68	4	4
3	dep04_b001	计算机基础	dep04_b001_01	68	4	4
4	dep04_p001	C语言程序设计	dep04_p001_01	68	4	4
5	dep04_s001	SQL Server数据库开发技术	dep04_s001_01	80	5	5
6	dep04_s002	JAVA程序设计与开发	dep04_s002_01	68	4	4
7	dep04_s003	单片机原理	dep04_s003_01	68	4	4
8	dep04_s004	软件开发技术	dep04_s004_01	51	3	3
9	dep04_s005	网页设计	dep04_s005_01	80	5	5
10	dep04_s006	计算机网络	dep04_s006_01	60	3	3
11	dep04_s007	计算机导论	dep04_s007_01	80	4	5

class 表

1. 增加

我们向 class 表中增加一条记录

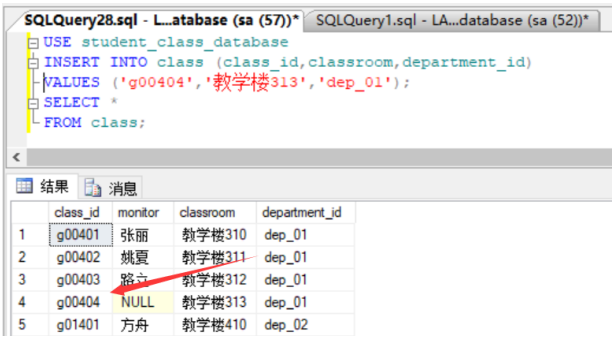
class_id	monitor	classroom	department_id
g00404	NULL	教学楼 313	dep_01

之前我们在新建表的时候，设定了 monitor 属性取值可以为 NULL。（可能新建了班级，但是还未选举出班长）

SQL 脚本如下：

```sql
USE student_class_database
INSERT INTO class (class_id,classroom,department_id)
VALUES ('g00404','教学楼 313','dep_01');
SELECT *
FROM class;
```

结果如下：



2. 删除

删除 department_id 为 dep_01 的所有班级记录

SQL 脚本如下：

```sql
USE student_class_database
DELETE FROM class WHERE department_id='dep_01';
SELECT *
FROM class;
```

结果如下：

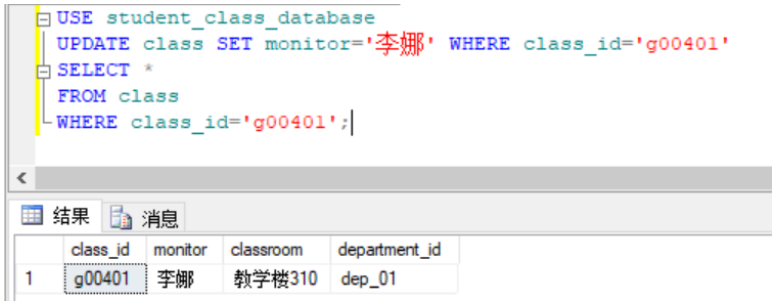
3. 修改

我们将 class_id 为 g00401 的班级的 monitor 修改为'李娜'，class_id 作为筛选条件

SQL 脚本如下：

```
USE student_class_database
UPDATE class SET monitor='李娜' WHERE class_id='g00401'
SELECT *
FROM class
WHERE class_id='g00401';
```

结果如下：



class_course 表

采用 SQL 脚本。

增加 class_id 为 g99401 的课程记录，要求所有课程都和班级 g99402 一样；然后删除 class_id 为 g99402 的所有记录；最后将 class_id 为 g99403 的记录的 class_id 改为 g99402。

SQL 脚本如下：

```
USE student_class_database
INSERT INTO class_course(class_id,course_id)(
        SELECT 'g99401',T.course_id
        FROM class_course AS T
        WHERE T.class_id='g99402'
);

DELETE FROM class_course
WHERE class_id='g99402';

UPDATE class_course
SET class_id='g99402'
WHERE class_id='g99403'
```

结果如下：

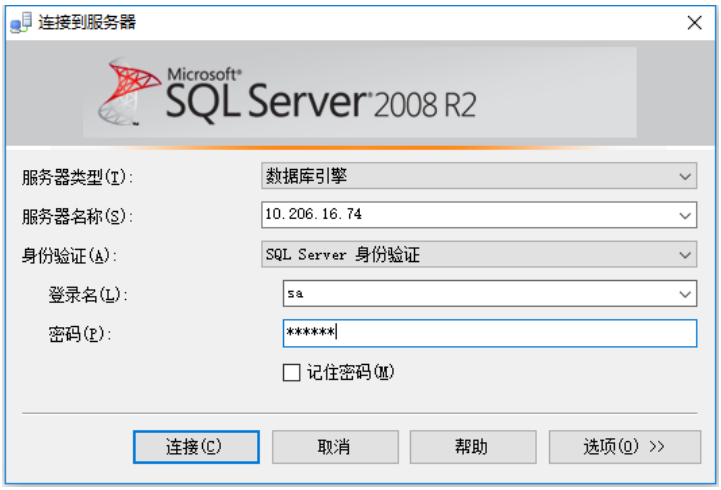
结果 消息

	class_id	course_id
1	g99401	dep04_s001
2	g99401	dep04_s002
3	g99401	dep04_s003
4	g99401	dep04_s004
5	g99401	dep04_s005
6	g99402	dep04_s001
7	g99402	dep04_s002
8	g99402	dep04_s003
9	g99402	dep04_s004
10	g99402	dep04_s005

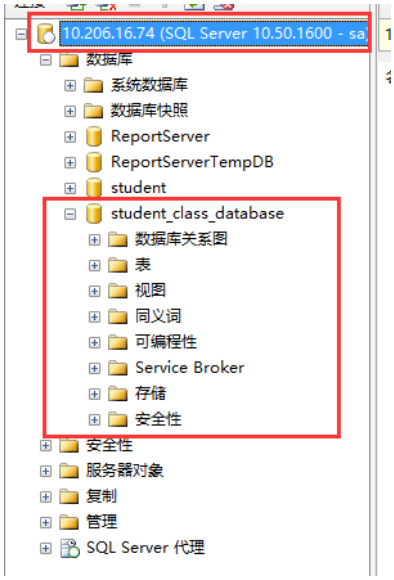
接下来是在客服端主机的远程操作。

department 表

首先在客户端主机上远程连接到服务器主机。



如下图，成功连接后，在客户端主机的 Management studio 的对象资源管理器中，可以看到我们在服务器主机上创建的数据库 student_course_database。



接下来我们便可以远程操作该数据库了。

修改 department_header 为'王敬远'的记录的 teacher_num 为 100；

删除 department_name 为通信与信息工程的记录；

新增如下记录：

department_id	department_name	department_header	teacher_num
dep_02	通信与信息工程	王杰	99

SQL 脚本如下：

```
use student_class_database

UPDATE department SET teacher_num=100
WHERE department_header='王敬远';

DELETE FROM department
WHERE department_name='通信与信息工程'

INSERT INTO department
VALUES('dep_02','通信与信息工程','王杰',99);

select *
from department;
```

结果如下：

结果 消息

	department_id	department_name	department_header	teacher_num
1	dep_01	无线电	王敬远	100
2	dep_02	通信与信息工程	王杰	99
3	dep_03	电子工程	董一平	72
4	dep_04	计算机科学	纪云	85

student 表

采用 SQL 脚本。

新增一条记录如下：

student_id	student_name	sex	birth	class_id	entrance_date	home_addr
g9940208	张伟	男	1980-09-22	g990402	1999-09-01	上海市南京路 206 号

然后将 student_name 为李红[大]的记录删除，将 student_name 为李红[小]的记录的 student_name 改为李红。

SQL 脚本如下：

```
use student_class_database

INSERT INTO student
VALUES('g9940208','张伟','男','1980-09-22',
'g99402','1999-09-01','上海市南京路 206 号');

DELETE FROM student WHERE student_name='李红[大]';

UPDATE student SET student_name='李红'
WHERE student_name='李红[小]';

select *
from student;
```

结果如下：

结果 消息

	student_id	student_name	sex	birth	class_id	entrance_date	home_addr
1	g9940201	张虹	男	1981-09-12	g99402	1999-09-01	南京市鼓楼区古平岗3号
2	g9940202	林红	女	1981-06-16	g99402	1999-09-01	扬州市五亭街41号
3	g9940203	曹雨	男	1980-12-11	g99402	1999-09-01	南京玄武区林荫大道4号
4	g9940204	芮华	男	1982-09-09	g99402	1999-09-01	江苏仪征市青年路6号
5	g9940206	林海国	男	1981-04-05	g99402	1999-09-01	北京市中关村太平北路1号
6	g9940207	李红	女	1981-05-23	g99402	1999-09-01	南京虎踞北路43号
7	g9940208	张伟	男	1980-09-22	g99402	1999-09-01	上海市南京路206号
8	g9940210	赵青	男	1980-01-09	g99402	1999-09-01	南京中山北路10号
9	g9940211	赵凯	男	1982-01-11	g99402	1999-09-01	南京中山北路10号
10	g9940301	左群声	男	1982-09-08	g99403	1999-09-01	NULL
11	g9940302	许小华	女	1981-10-15	g99403	1999-09-01	上海市龙盘路97号
12	g9940303	张雅丽	女	1980-01-21	g99403	1999-09-01	上海市北京东路191号
13	g9940304	林红	女	1980-09-19	g99403	1999-09-01	南京中山路9号

student_course 表

新增张伟（student_id=g9940208）的一条选课记录如下：

course_id	student_id	grade	credit	semester	school_year
dep04_b001	g9940208	NULL	4	0	1999/2000

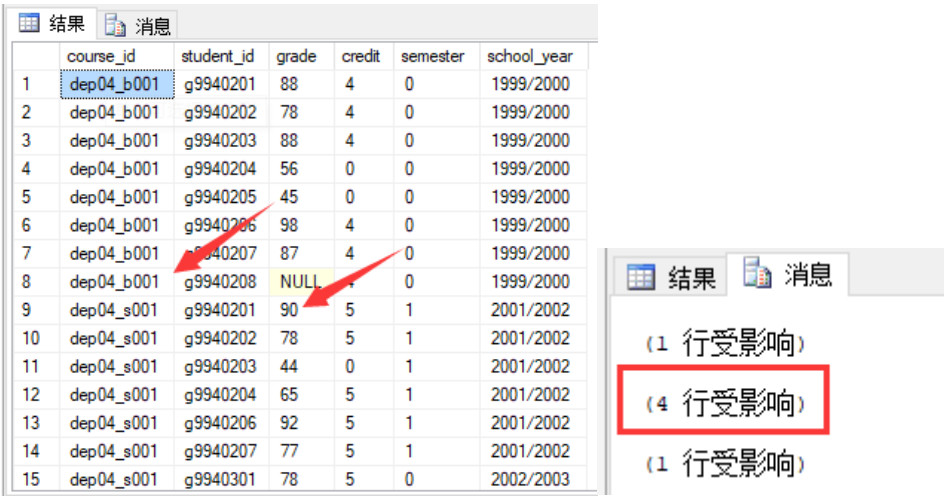
删除 student_id 为 g9940205 的学期 1 的选课记录；

修改 student_id 为 g9940201 的 course_id 为 dep04_s001 的课程成绩为 90。

SQL 脚本如下：

```
use student_class_database

INSERT INTO student_course
VALUES('dep04_b001','g9940208',NULL,4,0,'1999/2000');

DELETE FROM student_course
WHERE student_id='g9940205' and semester = 1;

UPDATE student_course SET grade=90
WHERE student_id='g9940201' and course_id='dep04_s001';

select *
from student_course;
```

结果如下：



	course_id	student_id	grade	credit	semester	school_year
1	dep04_b001	g9940201	88	4	0	1999/2000
2	dep04_b001	g9940202	78	4	0	1999/2000
3	dep04_b001	g9940203	88	4	0	1999/2000
4	dep04_b001	g9940204	56	0	0	1999/2000
5	dep04_b001	g9940205	45	0	0	1999/2000
6	dep04_b001	g9940206	98	4	0	1999/2000
7	dep04_b001	g9940207	87	4	0	1999/2000
8	dep04_b001	g9940208	NULL	4	0	1999/2000
9	dep04_s001	g9940201	90	5	1	2001/2002
10	dep04_s001	g9940202	78	5	1	2001/2002
11	dep04_s001	g9940203	44	0	1	2001/2002
12	dep04_s001	g9940204	65	5	1	2001/2002
13	dep04_s001	g9940206	92	5	1	2001/2002
14	dep04_s001	g9940207	77	5	1	2001/2002
15	dep04_s001	g9940301	78	5	0	2002/2003

说明 student_id 为 g9940205 的学期 1 的选课记录有 4 条，均被删除。

teacher 表

修改 teacher_id 为 dep04_005 的 profession 为副教授；删除 teacher_id 为'dep04_004'的记录；新增一条记录如下：

teacher_id	teacher_name	sex	birth	department_id
dep01_002	严为	男	1978-09-01	dep_01
profession	telephone	home_addr	postalcode	
助教	7654987	南京鼓楼区长虹路 3 号	210002	

SQL 脚本如下：

```
use student_class_database

UPDATE teacher SET profession='副教授'
```

```
WHERE teacher_id='dep04_005';

DELETE FROM teacher
WHERE teacher_id='dep04_004';

INSERT INTO teacher
VALUES('dep01_002','严为','男','1978/09/01',
'dep_01','助教','7654987','南京鼓楼区长虹路 3 号','210002');

select *
from teacher
```

结果如下：

结果 消息

	teacher_id	teacher_name	sex	birth	department_id	profession	telephone	home_addr
1	dep01_001	王敬远	男	1956-09-09	dep_01	副教授	6211544	南京先贤路
2	dep01_002	严为	男	1978-09-01	dep_01	助教	7654987	南京鼓楼区
3	dep02_001	康辉	男	1949-08-06	dep_02	教授	13951606754	南京鼓楼区
4	dep03_001	董一平	男	1968-01-01	dep_03	副教授	3728543	南京长江路
5	dep03_002	潘惠	女	1966-04-01	dep_01	讲师	5887654	南京长江路
6	dep04_001	纪云	男	1965-09-09	dep_04	副教授	6245321	南京长江路
7	dep04_002	章红	女	1970-07-04	dep_04	教授	13027816022	南京北京西
8	dep04_003	李伟	男	1954-06-06	dep_04	教授	3728543	南京先贤路
9	dep04_005	乔红	女	1969-05-31	dep_04	副教授	8802888	南京太州青

teacher_course_class 表

将教学楼 211 的记录的上课教室改为教学楼 203；删除 school_year 为 1999/2000 的所有记录；新增一条记录如下：

teacher_id	course_id	class_id	semester
dep04_002	dep04_j001	g99402	1
school_year	course_schedule	course_classroom	book_id
1999/2000	110320234	教学楼 303	dep04_j001_01

SQL 脚本如下：

```
use student_class_database

UPDATE teacher_course_class
SET course_classroom='教学楼 203'
WHERE course_classroom='教学楼 211';

DELETE FROM teacher_course_class
WHERE school_year='1999/2000';

```

```
INSERT INTO teacher_course_class VALUES('dep04_002','dep04_j001','g99402',1,'1999/2000','110320234','教学楼 303','dep04_j001_01');

select *
from teacher_course_class
```

结果如下：

	teacher_id	course_id	class_id	semester	school_year	course_schedule	course_classroom	boo
1	dep01_001	dep01_s002	g99102	1	2001/2002	120320520	教学楼202	dep
2	dep04_001	dep04_s001	g99402	1	2002/2003	110320512	教学楼601	dep
3	dep04_001	dep04_s005	g99402	0	2001/2002	130310511	教学楼602	dep
4	dep04_002	dep04_j001	g99402	1	1999/2000	110320234	教学楼303	dep
5	dep04_002	dep04_s002	g99402	1	2002/2003	210420321	教学楼601	dep
6	dep04_003	dep04_s004	g99402	1	2001/2002	120410132	教学楼203	dep
7	dep04_004	dep04_b001	g99403	0	2001/2002	310520789	教学楼603	dep
8	dep04_004	dep04_s001	g99403	1	2001/2002	210512456	教学楼212	dep
9	dep04_004	dep04_s004	g99402	1	2001/2002	220521897	教学楼203	dep

可见有上课教室为教学楼 211 的记录有两条，均被改为了教学楼 203；school_year 为 1999/2000 的记录有三条，均被删除；最后还剩一条新添加进去的记录的 school_year 为 1999/2000。

建立、维护视图

在进行视图的实验之前，为了消除上述实验的影响，我们先将数据库恢复到初始状态。

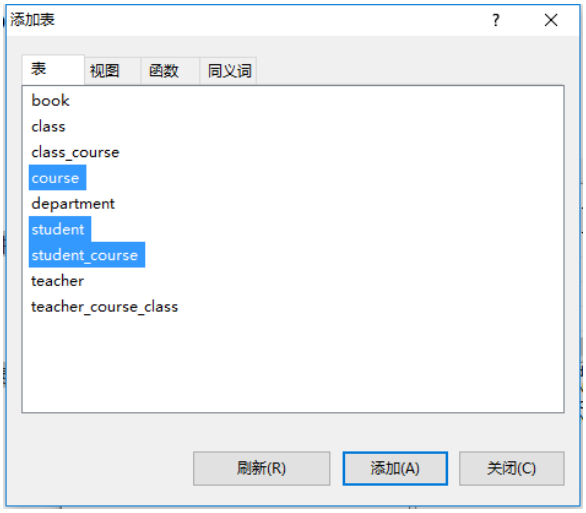
什么是视图？

在 SQL 中，视图是基于 SQL 语句的结果集的可视化的表。
视图包含行和列，就像一个真实的表。视图中的字段就是来自一个或多个数据库中的真实的表中的字段。我们可以向视图添加 SQL 函数、WHERE 以及 JOIN 语句，我们也可以提交数据，就像这些来自于某个单一的表。
注释：数据库的设计和结构不会受到视图中的函数、where 或 join 语句的影响。

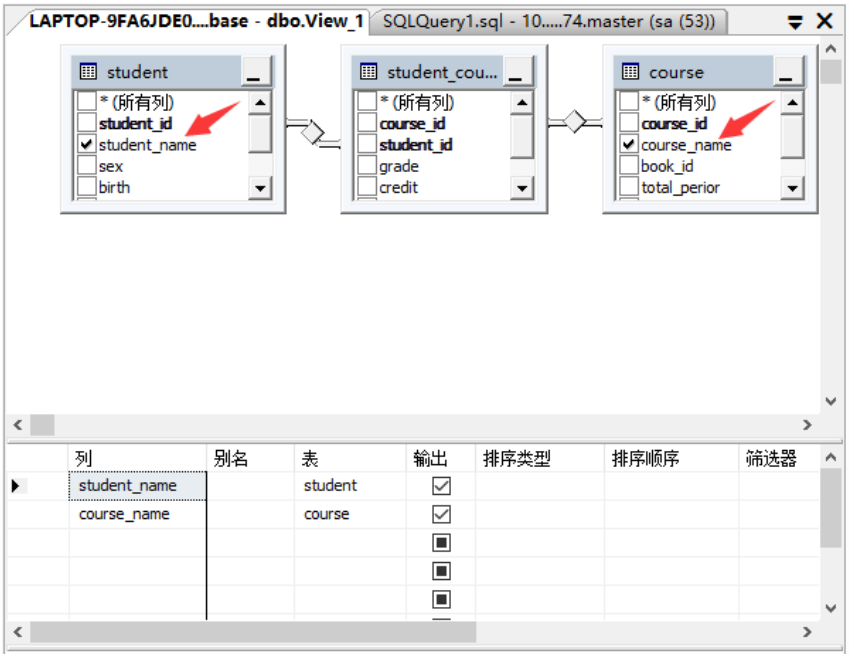
利用图形化界面创建视图

- 在 “对象资源管理器”中，展开要创建新视图的数据库。
- 右键单击“视图”文件夹，然后单击“新建视图...”。
- 在 “添加表” 对话框中，从以下选项卡之一选择要在新视图中包含的元素：“表”、“视图”、“函数”和“同义词”。
- 单击 “添加”，再单击 “关闭”。
- 在 “关系图窗格”中，选择要在新视图中包含的列或其他元素。
- 在 “条件窗格”中，选择列的其他排序或筛选条件。
- 在“文件”菜单上，单击“保存***视图名称”。
- 在 “选择名称” 对话框中，输入新视图的名称并单击 “确定”。

我们想要构造一个有学生姓名和选课名称的视图。
首先选择 student、student_course、course 表：

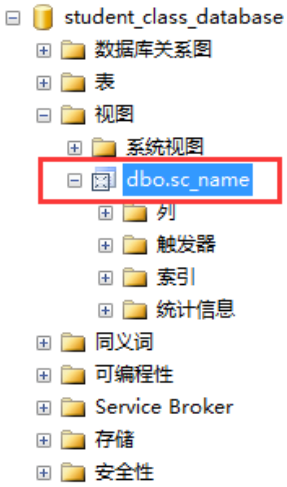


然后在视图的设计界面选择 student 表的 student_name 属性和 course 表的 course_name 属性。



点击保存，取名为 sc_name。

可以看到对象资源管理器中多出了 dbo.sc_name 这个视图。

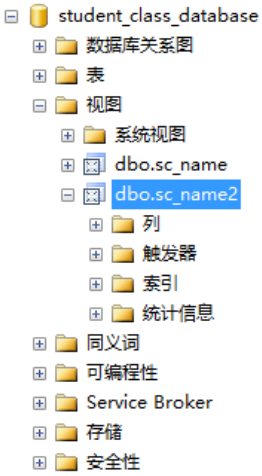


利用 SQL 脚本创建视图

SQL 脚本如下：

```
USE student_class_database;
GO
CREATE VIEW sc_name2
AS
SELECT s.student_name, c.course_name
FROM student as s,student_course as sc,course as c
WHERE s.student_id=sc.student_id and sc.course_id=c.course_id;
GO
```

结果如下：



可以看到又多出了一个 db.sc_name2 的视图，其本质上与 sc_name 视图的功能一样。

查询视图

我们输出之前创建的视图的所有列，SQL 脚本如下：

```
USE student_class_database;
SELECT *
FROM sc_name2
```

结果如下：

结果 消息

	student_name	course_name
1	张虹	计算机基础
2	林红	计算机基础
3	曹雨	计算机基础
4	芮华	计算机基础
5	李红[大]	计算机基础
6	林海国	计算机基础
7	李红[小]	计算机基础
8	张虹	SQL Server数据库开发技术
9	林红	SQL Server数据库开发技术
10	曹雨	SQL Server数据库开发技术
11	芮华	SQL Server数据库开发技术
12	李红[大]	SQL Server数据库开发技术
13	林海国	SQL Server数据库开发技术
14	李红[小]	SQL Server数据库开发技术
15	左群吉	SQL Server数据库开发技术

查询已成功执行。

结果 消息

	student_name	course_name
31	张虹	软件开发技术
32	林红	软件开发技术
33	曹雨	软件开发技术
34	芮华	软件开发技术
35	李红[大]	软件开发技术
36	林海国	软件开发技术
37	李红[小]	软件开发技术
38	张虹	网页设计
39	林红	网页设计
40	曹雨	网页设计
41	芮华	网页设计
42	李红[大]	网页设计
43	林海国	计算机网络
44	李红[小]	计算机导论

查询已成功执行。

共 44 条记录，与 student_course 表中的记录相符。

向视图中增加数据

我们试图向该视图中增加一条(‘李娜’,’计算机系统结构’)的记录。结果如下：

消息

消息 4405，级别 16，状态 1，第 2 行
视图或函数 'sc_name2' 不可更新，因为修改会影响多个基表。

可见无法在这种涉及多个基表的视图中添加记录。
我们另外再创建一个较简单的视图 teacher_simple，该试图是 teacher 表的简化版本，只包含那些可以不为 NULL 的属性。
SQL 脚本如下：

```
USE student_class_database;
GO
CREATE VIEW teacher_simple
AS
SELECT teacher_id,teacher_name,sex,department_id,profession,telephone
FROM teacher
```

我们在视图中新增一条记录如下：

teacher_id	teacher_name	sex	department_id	profession	telephone
dep01_002	高晓松	男	dep_01	教授	123456789

SQL 脚本如下：

```
USE student_class_database;
INSERT INTO teacher_simple VALUES('dep01_002','高晓松','男','dep_01','教授
','123456789')
```

```
SELECT *
FROM teacher_simple
```

结果如下：

结果 消息

	teacher_id	teacher_name	sex	department_id	profession
1	dep01_001	王敬远	男	dep_01	副教授
2	dep01_002	高晓松	男	dep_01	教授
3	dep02_001	康辉	男	dep_02	教授
4	dep03_001	董一平	男	dep_03	副教授
5	dep03_002	潘惠	女	dep_01	讲师
6	dep04_001	纪云	男	dep_04	副教授
7	dep04_002	章红	女	dep_04	教授
8	dep04_003	李伟	男	dep_04	教授
9	dep04_004	严为	男	dep_04	助教
10	dep04_005	乔红	女	dep_04	讲师

我们再去 teacher 表中查询检验
SQL 脚本如下：

```
USE student_class_database;
SELECT *
FROM teacher
WHERE teacher_name='高晓松'
```

结果如下：

结果 消息

	teacher_id	teacher_name	sex	birth	department_id	profession	telephone	home_addr	postalcode
1	dep01_002	高晓松	男	NULL	dep_01	教授	123456789	NULL	NULL

可见对视图中记录的修改，也会影响到基表。

修改视图

我们将之前创建的 sc_name2 视图修改为姓李的学生的姓名和选课名字。
SQL 脚本如下：

```
ALTER VIEW sc_name2
AS
SELECT s.student_name, c.course_name
FROM student as s,student_course as sc,course as c
WHERE s.student_id=sc.student_id and sc.course_id=c.course_id and s.student_name like '李%';
GO
SELECT *
FROM sc_name2
```

结果如下：

结果 消息

	student_name	course_name
1	李红[大]	计算机基础
2	李红[小]	计算机基础
3	李红[大]	SQL Server数据库开发技术
4	李红[小]	SQL Server数据库开发技术
5	李红[大]	JAVA程序设计与开发
6	李红[小]	JAVA程序设计与开发
7	李红[大]	单片机原理
8	李红[小]	单片机原理
9	李红[大]	软件开发技术
10	李红[小]	软件开发技术
11	李红[大]	网页设计
12	李红[小]	计算机导论

实验总结

通过本实验，我加深理解了数据库表的结构，掌握了建立数据库表的方法，熟悉了 SQL 中关于数据库表的建立语句；掌握了从外部将数据导入到 SQL Server 数据库中的若干种方法；掌握了远程连接、操作数据库的方法；通过对每个数据表的增加、删除和插入记录的维护操作，熟悉并掌握了 SQL Server 数据库数据的操作方法；最后通过建立、维护视图，初步理解和掌握了视图的概念。

当然也碰到了不少问题。例如在导入数据时，发现 excel 中多余列虽然是空的，但是 SQL Server 仍认为这其中有数据，这就导致了导入时属性个数不匹配，因此需要将处理 excel 文件。

对于一些较长的数字型字符串，例如电话号码，在导入时，SQL 会将其认为是浮点数，然后自动转化为科学计数法，如果表中对应属性的类型设置为了 varchar，则显示的数据将是科学计数法形式的字符串，为了解决这个问题，该属性的类型不再设置为 varchar，而改为 decimal(n,0)，即一个总位数为 n，小数点后位数为 0 的浮点数，如果必要的话，可以在此基础上再转化为 varchar。

在实现远程连接数据库时，也碰到了端口未打开等问题，为此查了一些资料，进行了相关设置，才最终连接成功。

总之，通过本次实验，掌握更多 SQL Server 的使用方法，为之后的学习奠定了基础。