

# 《数据库系统原理》实验指导书

课程编码：6063121145

课程类型：课内实训课

学时：24

学分：3（理论+实验）

适用专业：计算机科学与技术

开课学期：5

编写执笔人：赵丽婷

审定负责人：刘红梅

必开实验个数：6

## 一、实验要求及学时分配

本课程是计算机科学与技术专业的专业必修课，重点介绍数据库设计原理、关系数据库理论及关系数据库管理系统的使用。通过学习理论并结合上机操作，掌握数据库设计方法及数据库实施方法和过程，为开发和使用数据库打下理论和实践基础。

本课程含教学规定的 24 学时上机实验课，在机房进行课堂内容的补充学习与讨论、上机测试及完成规定的实验练习，使学生能更好地融会贯通课堂教学的内容，进行实践能力的训练。上机环节要求实验 6 个，各实验按照实验指导书的要求学生独立或分组完成，并提交实验报告。

序号	实验项目名称	类 型		学 时
		验证性	综合设计性	
一	数据库的创建与管理	√		4
二	表的创建与管理	√		4
三	数据查询		√	4
四	数据库的视图定义及维护		√	4
五	数据安全性控制		√	4
六	数据完整性控制		√	4

## 二、考核方式

实验成绩=实验报告成绩。共 6 份实验报告，每份 100 分，6 份实验报告的均分为实验成绩。

## 三、配套的实验教材或指导书

《数据库系统概论》，王珊，萨师焯主编，高等教育出版社

## 四、实验项目

## 实验一 数据库的创建与管理

学时：4

1. 实验类型：验证性
2. 实验类别：专业实验
3. 每组人数：5-6 人
4. 实验要求：必修
5. 实验目的
  - (1) 巩固数据库的基础知识；
  - (2) 掌握创建数据库的两种方法；
  - (3) 掌握如何查看、修改数据库属性的方法；
  - (4) 掌握数据库的缩小、更名、与删除的方法。
6. 实验内容

### (1) 创建数据库

#### 1) 使用企业管理器直接创建数据库

打开企业管理器。

右键单击数据库并选择【新建数据库】。

会弹出窗体如下图所示，在常规标签中输入数据库名称。

然后单击【数据文件】标签，可以输入文件名、文件位置、文件大小以及文件组。在这个标签中还提供了让数据文件在数据需要更多空间时自动扩展的选项和数据文件大小的最大值。

单击【事务日志】标签，在此键入日志文件的名字、位置、大小。在这个标签中还提供了让日志文件在日志需要更多空间时自动扩展选项和日志文件大小的最大值。

#### 2) 使用 Transact-SQL 语句创建数据库

当我们无法用企业管理器创建生成一个大的数据库，或者用户想用 Transact-SQL 语句写一段程序自动的计划的生成几个数据库时，可以用 Transact-SQL 语句向服务器请求创建数据库。使用 Transact-SQL 语句创建数据库的语法如下：

```
CREATE DATABASE database_name
```

```
[ ON
```

[PRIMARY]

( [NAME = logical\_file\_name,]

FILENAME = 'os\_file\_name'

[,SIZE = size]

[,MAXSIZE = {max\_size|UNLIMITED}]

[,FILEGROWTH = growth\_increment)][,...n]

<filegroup>::=

FILEGROUP filegroup\_name <filespec> [...n]

LOG ON

(NAME = logical\_file\_name,

FILENAME = 'os\_file\_name'

SIZE = size,

MAXSIZE = {max\_size | UNLIMITED},

FILEGROWTH = growth\_increment)]

例 1：创建一个 lfm 数据库，主要文件组包含主要数据文件文件 lfm1\_dat 和次要数据文件 lfm2\_dat。有 2 个次要文件组：lfmGroup1 包含 2 个次要数据文件 lfm1Fi1\_dat 和 lfmFi2\_dat, lfmGroup2 包含 lfm2Fi1\_dat 和 lfm2Fi2\_dat 两个次要数据文件。日志的逻辑文件名为 lfm\_log，此日志文件存储在 c:\mssql\data\lfmlog.ldf 中。

CREATE DATABASE lfm

ON PRIMARY

(NAME = lfm1\_dat,

FILENAME = 'c:\lfm1\_dat.mdf',

SIZE = 10,

MAXSIZE = 50,

FILEGROWTH = 15%),

```

(NAME = lfm2_dat,

FILENAME = 'c:\lfm2_dat.ndf',

SIZE = 10,

MAXSIZE = 50,

FILEGROWTH = 15%),

FILEFROUP lfmGroup1

( NAME = lfm1Fi1_dat,

FILENAME = 'c:\lfm1Fildt.ndf',

SIZE = 10,

MAXSIZE = 50,

FILEGROWTH = 5) ,

( NAME = lfm1Fi2_dat,

FILENAME = 'c:\lfm2Fildt.ndf',

SIZE = 10,

MAXSIZE = 50,

FILEGROWTH = 5),

FILEGROUP lfmGroup2

(NAME = lfm2Fil_dat,

FILENAME = 'c:\SG2Fi1dt.ndf',

SIZE = 10,

MAXSIZE = 50,

FILEGROWTH = 5),

(NAME = lfm2Fi2_dat,

FILENAME = 'c:\SG2Fi1dt.ndf',

SIZE = 10,

```

MAXSIZE = 50,  
FILEGROWTH = 5)

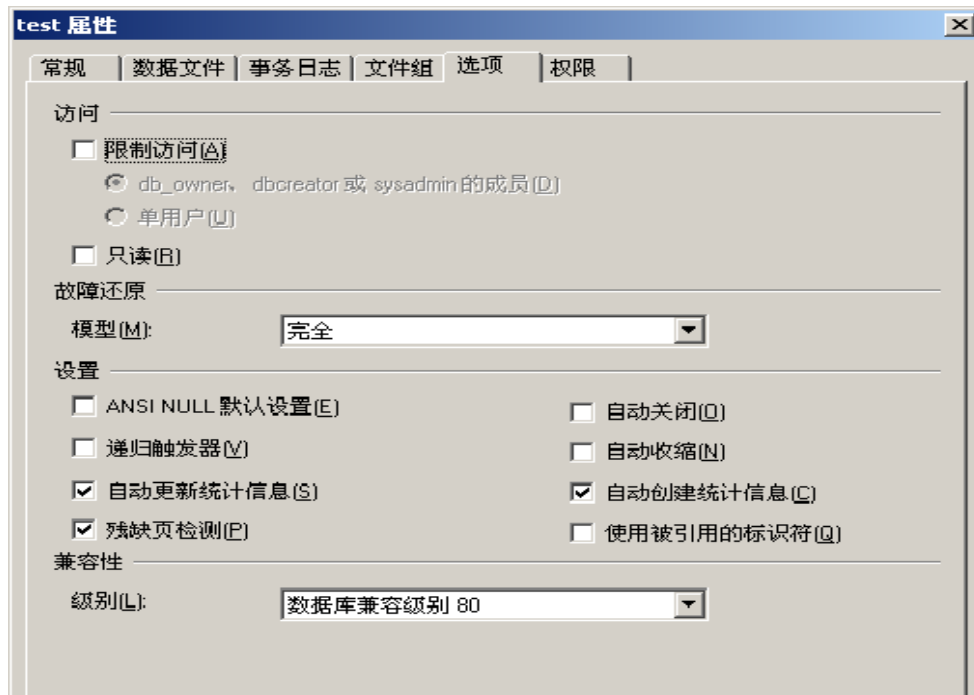
LOG ON

(NAME = 'lfm\_log',  
FILENAME = 'c:\lfmlog.ldf',  
SIZE = 5MB,  
MAXSIZE = 25MB,  
FILEGROWTH = 5MB)

(2) 查看数据库、修改数据库的属性

1) SQLSERVER2008 有许多选项可以改变数据库的行为。因此，在使用数据库之前，可以考虑其中一些选项的设置。

进入企业管理器，用鼠标右键单击数据库 test，再单击“属性”，然后选择“选项”标签即可进入如下图所示 test 数据库的选项窗体。



访问控制：用于控制哪些用户能访问该数据库

只读：这个选项使数据库只能读取不能写入

模型：指定数据库的恢复模型类型

ANSI NULL 默认设置：指定将数据库列默认定义为 NULL 还是 NOT NULL。

递归触发器：启用触发器的递归触发。

自动更新统计信息：指定在优化期间自动生成查询优化所需的过时统计信息。

残缺页检测：指定可以检测不完整页。

自动关闭：指定数据库资源被释放以及所有用户退出之后关闭数据库。

自动收缩：指定数据库文件可以周期性自动收缩。

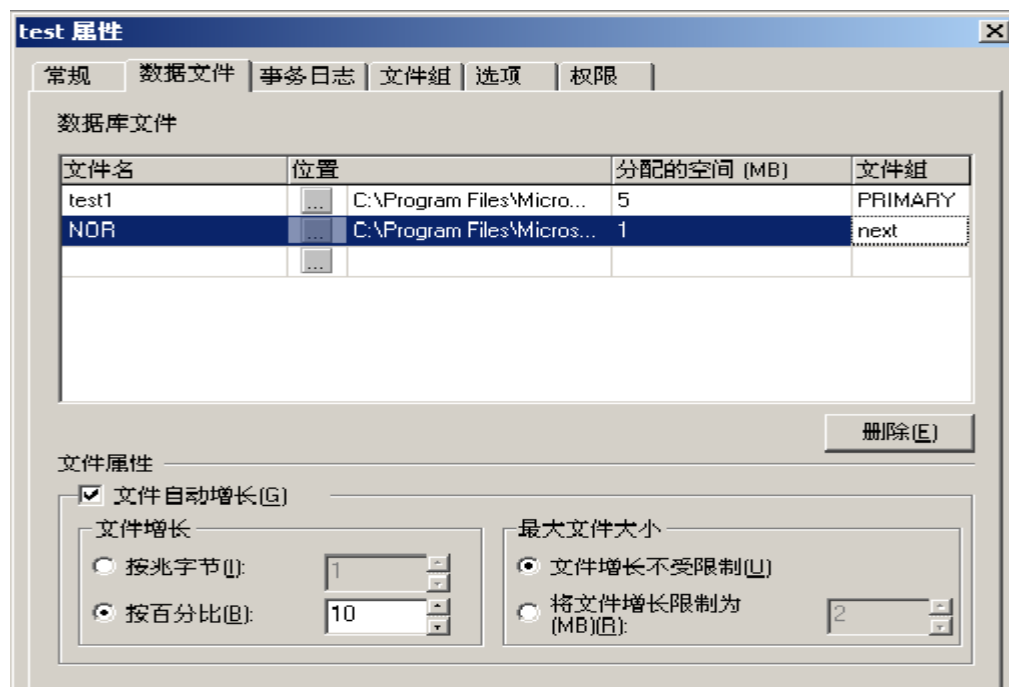
自动创建统计信息：指定在优化期间自动生成优化查询所需的任何缺少的统计信息。

使用引用的标识符：指定 SQLSERVER 将强制执行关于引号的 ANSI 规则。

级别：指定数据库兼容性级别。

## 2) 整数据库大小

点击“数据文件”或“事物日志”标签，可以得到如下图所示 test 数据库的窗体



文件自动增长：选中表示数据库自动，这时可以选择按兆增长还是按百分比增长。

文件增长不受限制：意味着数据库没有最大限制，可以一直充满磁盘

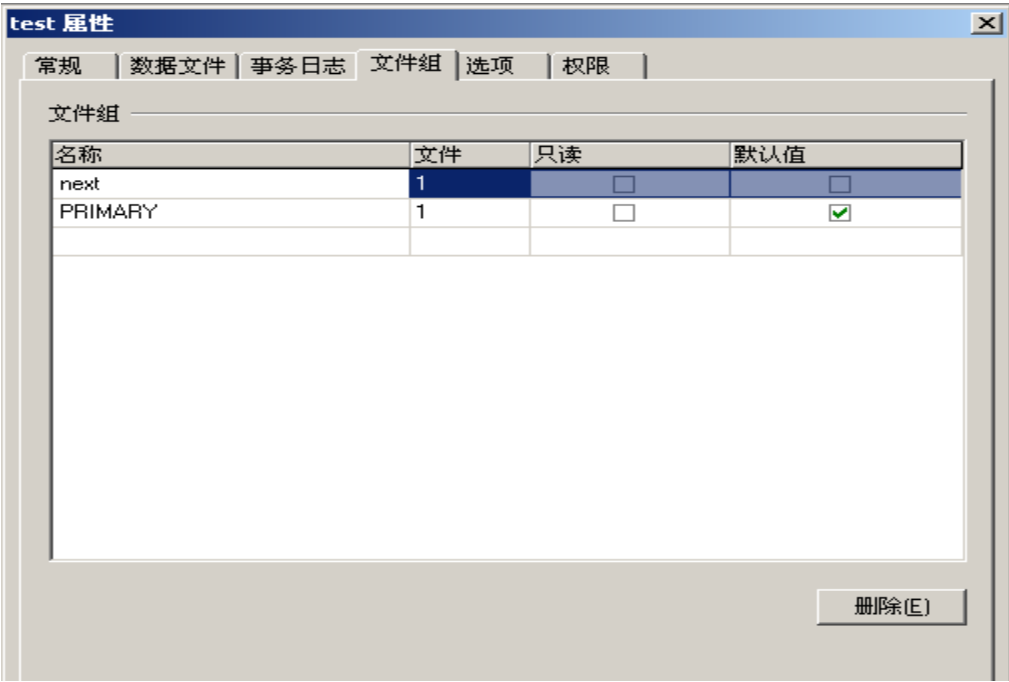
将文件增长限制为，为数据库选择一个最大限制。

分配空间：用户可以直接增加数据库文件大小

扩大数据库的另外一个方法是增加辅助数据文件。

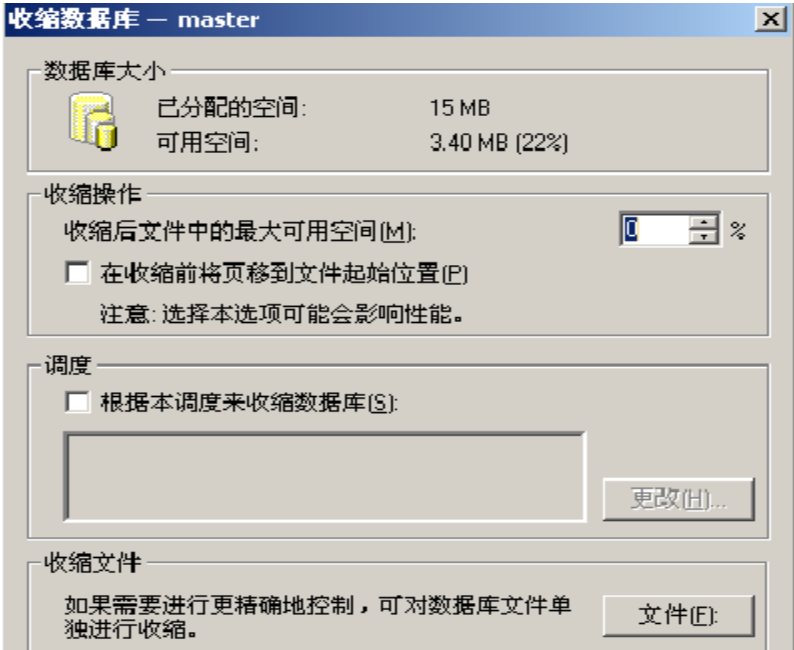
3) 加文件组

生成一辅助文件后，可将其逻辑组织到文件组中，帮助管理磁盘空间分配。增加文件组的方法是：在文件组列中输入一个新的文件组名称，如“next”，单击“确定”，然后右击数据库，选择“属性”，选择“文件组”标签会看到增加的文件组。如图示：



4) 缩小数据文件

右击数据库，单击“所有任务”，选择收缩数据库，会弹出下图所示窗体，利用



该窗体可以输入数据库大小，并且可以定制任务，让 SQL Server 自动定期收缩数据库大小。

### (3) 利用 T-SQL 语句查看、修改数据库属性

语法为：

[@dbname =] 'database' 设置指定选项的数据库名称

[@optname =] 'option\_name' 设置的选项名称

[@optname =] 'option\_name' 可以示 true、false、on 或 off

#### 1) 修改日志文件的最大值

将 test1\_log 日志文件的最大值更改为 150M，SQL 脚本程序如下：

```
USE test
```

```
GO
```

```
ALTER DATABASE test
```

```
MODIFY FILE(NAME=test1_log,MAXSIZE=150MB)
```

```
GO
```

#### 2) 修改日志文件的初始值

将 test1\_log 日志文件的初始值由 5M 更改为 20M，SQL 脚本程序如下：

```
USE test
```

```
GO
```

```
ALTER DATABASE test
```

```
MODIFY FILE(NAME=test1_log,SIZE=20MB)
```

```
GO
```

#### 3) 查看数据库信息



执行系统存储过程 sp\_helpdb,可以查看 test 数据库信息，并验证数据库是否已经修改成功。

Sp\_helpdb test

#### 4) 更改数据库

更改数据库 test 时为 test 添加一个辅助文件，文件名为 testfz，存储在 c:\ testfz.ndf 中，初始大小为 3MB，最大值为 50MB，增长量为 3MB。

USE test

ALTER DATABASE test

ADD FILE

(NAME = testfz,

FILENAME='c:\testfz.ndf',

SIZE=3MB,

MAXSIZE=50,

FILEGROWTH=3)

向数据库 test 中添加一个含有 2 个文件的文件组

USE test

GO

ALTER DATABASE test

ADD FILEGROUP testfg1

GO

ALTER DATABASE test

ADD FILE

(NAME=testfz1,

FILENAME='C:\testfz1.ndf',

SIZE =5MB,

```
MAXSIZE=50MB,  
  
FILEGROWTH =5),  
  
(NAME=testfz2,  
  
FILENAME='C:\testfz2.ndf',  
  
SIZE =5MB,  
  
MAXSIZE=50MB,  
  
FILEGROWTH =5)  
  
TO FILEGROUP testfg1
```

向 test 数据库中添加 2 个 5MB 的日志文件

```
USE test
```

```
GO
```

```
ALTER DATABASE test
```

```
ADD LOG FILE
```

```
(NAME =testlog2,  
  
FILENAME='c:\testlog2.ldf',  
  
SIZE =5MB,  
  
MAXSIZE = 10MB,  
  
FILEGROWTH =5),
```

```
(NAME =testlog3,  
  
FILENAME='c:\testlog3.ldf',  
  
SIZE =5MB,  
  
MAXSIZE = 10MB,  
  
FILEGROWTH =5)
```

删除数据库 test 中的一个文件 testfz.ndf

```
USE test
```

GO

ALTER DATABASE test

REMOVE FILE testfz

删除 test 数据库

USE test

DROP DATABASE test

## 练习题

### 练习 1：创建数据库

用企业管理器创建数据库

- 1) 创建数据库 Test:
- 2) 右击数据库，单击【新建数据库】
- 3) 输入数据库名称 Test
- 4) 选择【数据文件】标签，增加一个文件 test1，设置初始大小为 5MB
- 5) 选择【事务日志】标签，增加一个日志文件 test1\_log，初始大小为 5MB
- 6) 单击【确定】按钮，开始创建数据库

### 练习 2：用 Transact-SQL 语句创建数据库

打开查询分析器，创建数据库 Test1。数据库的参数如下表所示。

参数	参数值
----	-----

数据库名称	Test1
-------	-------

数据库逻辑文件名	Test1_dat
----------	-----------

操作系统数据文件名	C:\mssql\data\Test1_dat.mdf
-----------	-----------------------------

数据文件的初始大小	10MB
-----------	------

数据文件最大大小	50MB
----------	------

数据文件增长量	原来的 20%
---------	---------

日志逻辑文件名 Test\_log

操作系统日志文件名 C:\mssql\data\Test1\_log.ldf

日志文件初始大小 5MB

日志文件的最大大小 25MB

日志文件增长增量 5MB

#### 1) 设置数据库选项:

使用企业管理器修改数据库选项。右击数据库，单击“属性”，选择“选项”标签，弹出如图所示窗体，选中“只读”复选框。这样数据库就变为只读数据库。然后打开查询分析器使用 T-SQL 语句更改数据库选项。

```
EXEC SP_DBOPTION 'test', 'read only', 'TRUE'
```

#### 2) 压缩数据库:

使用企业管理器压缩数据库。右击数据库，选择“所有任务”，单击“收缩数据库”，弹出图示窗体。键入收缩量。使用 T-SQL 语句压缩数据库。打开查询分析器键入

```
DBCC SHRINKDATABASE (test, 2) 将 test 数据库压缩成为 2M。
```

#### 3) 更改数据库:

打开查询分析器更改数据库 test，参数如下:

参数 参数值

数据库名 Test

增加的文件组名 TestFG1

增加的文件 1 的逻辑名 Testdat3

文件 1 在磁盘中的目录 C:\tdat3.ndf

文件 1 的初始大小 5MB

文件 1 的最大值 50MB

文件 1 的增长量 5MB

增加的文件 2 逻辑名 Testdat4

文件 2 磁盘中的目录 C:\tdat3.ndf

文件 2 初始大小 5MB

文件 2 最大值 50MB

文件 2 增长量 5MB

新增日志逻辑文件名 Testlog2

日志文件在磁盘中的目录 D:\testlog.ldf

日志文件初始值 5MB

日志文件最大值 100MB

日志文件增长量 5MB

## 实验二 表的创建与管理

学时：4

### 1. 实验类型：验证性

2. 实验类别：专业实验

3. 每组人数：5-6 人

4. 实验要求：必修

5. 实验目的

- (1) 巩固数据库的基础知识；
- (2) 掌握创建与删除表的两种方法；
- (3) 掌握修改表结构的方法。

6. 实验内容

### (1) 创建表

#### 1) 利用企业管理器创建表

打开企业管理器。

选中创建好的数据库，单击数据库中的表对象，然后右击窗口右侧选择新建表，弹出如图所示窗体。

在这个窗体中，列名列就示表的字段名，可以在这个窗体中为字段选择数据类型和长度以及是否可以为空值。

在此处可以为表选择文件组，在表中单击右键，选择属性，如果已经为数据库创建了一个次要组，可以在属性窗体的表文件组中为此表选择文件组。应该把一些竞争使用的表放于不同的文件组中，并且让文件组属于不同磁盘，这样可以在表竞争读写时提高并发性能。

创建完成后，单击磁盘图标，并为表取一个名字，注意，应该为表取一个有意义的名字。

#### 2) 用 Transact\_SQL 语句创建表

语法为：

CREATE TABLE

[ database\_name.[owner].] table\_name

( {column\_name data\_type [DEFAULT 'default\_value']

[CONSTRAINT constraint\_name]

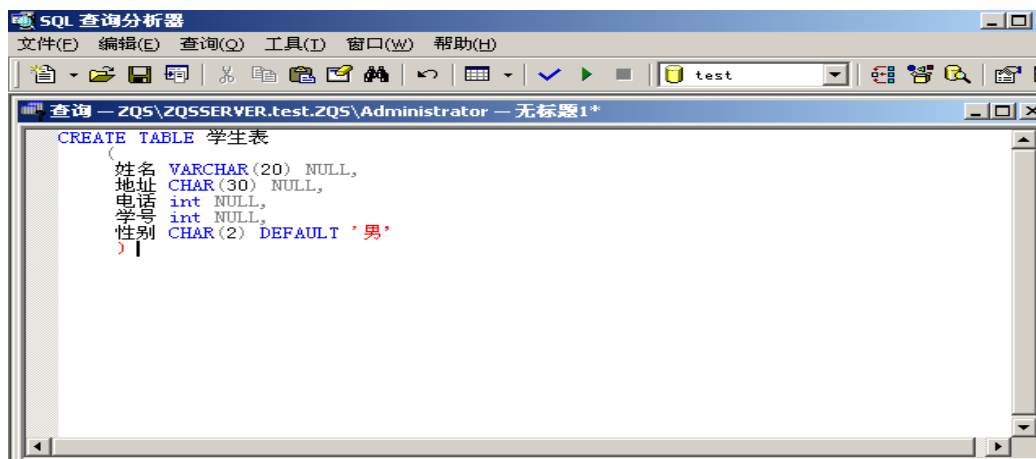
}, [...n] [IDENTITY [(seed, increment )]]

)

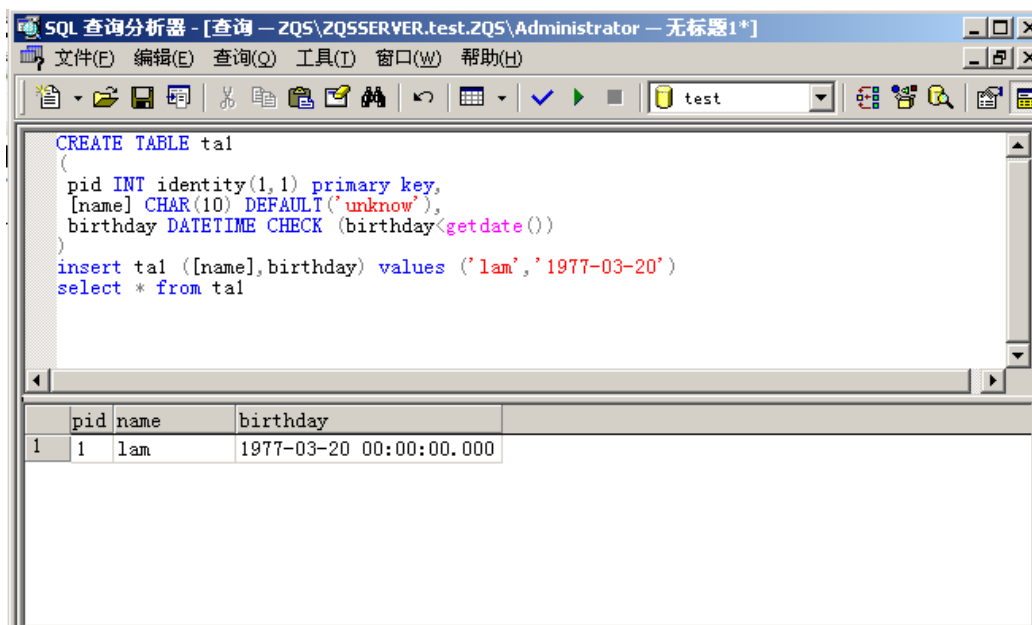
[ON { filegroup | DEFAULT }]

[TEXTIMAGE\_ON {filegroup | DEFAULT }]

例 1：生成一个表名为学生表的表



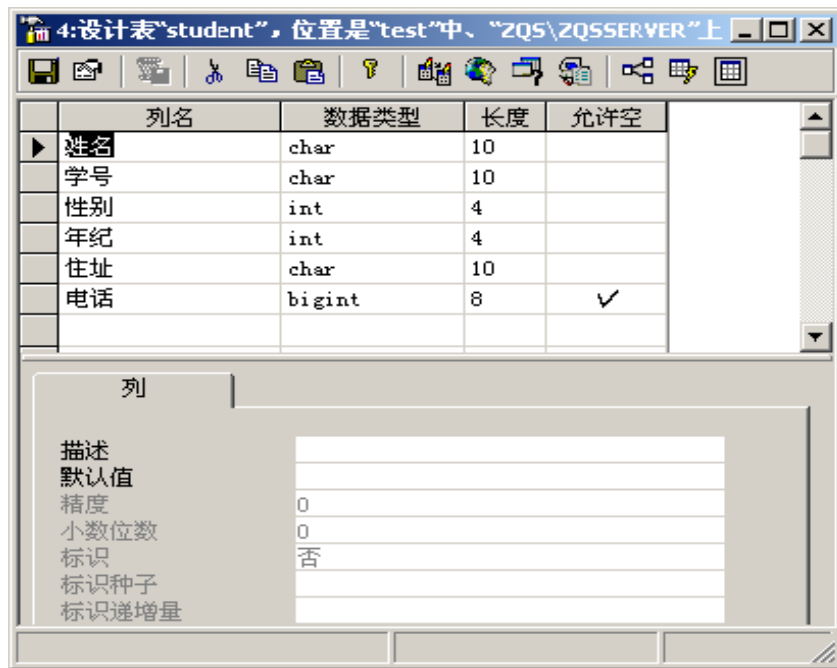
例 2: 创建一张名为 Ta1 的表，此表中有 3 列，第一列 Pid 定义为主键，并且自动增长。第二列[Name]默认值为 Unknown，第三列定义一个约束，即此列中的日期不能大于输入当天的日期，插入一条记录并进行查询，显示如下图。



## (2) 表的修改

### 1) 用企业管理器修改表

右键点击所要修改的表 student，选择“设计表”，会弹出如图所示的窗体，在这个窗体中可以更改数据表的字段。在空白处点右键，选择“属性”，弹出另一窗体，在此可以更改约束，也可以增加约束。



## 2) 利用 T-SQL 语句修改表

向表中添加新的字段：在 student 表中添加一个“籍贯”字段，数据类型为字符型。

```
ALTER TABLE student add 籍贯 char
```

删除表中的旧列：将 student 表中的“电话”字段删除。

```
ALTER TABLE student DROP column 籍贯
```

更改表以添加具有约束的列：

给 student 表的增加“年级”字段并加上 CHECK 约束，让其不可以大于 100。

```
ALTER TABLE student
```

```
ADD 年级 int
```

```
Constraint chname check(年级<100)
```

```
Exec sp_help chname
```

给 student 表中添加“出生日期”字段，并且这个日期不能在录入当天的日期之后。

```
Alter table student
```

```
Add 出生日期 DATETIME NULL
```

```
CONSTRAINT ch_birthday CHECK (出生日期<getdate())
```

添加具有默认值的可为空的列：

在 student 表中加入“进校日期”字段，并且这一字段的默认值为录入当天的日期。

```
Alter table student
```

```
Add 进校日期 smalldatetime NULL
```

```
Constraint adddateflt
```



Default getdate())

### (3) 表的删除

删除表对象，我们可以在企业管理器中选择要删除的表直接删除，也可以通过 T-SQL 语句 DROP 删除表的定义及表中的所有数据、索引、出发器、约束和权限规范。

删除 table2 表。

DROP TABLE table2

### 练习题

练习 1：在 Test 数据库中创建教工表

打开查询分析器，用 sa 登录

在数据库列表中单击 test 数据库

写出并执行语句创建教工表。条件如下：

列名	数据类型	允许为空	约束	IDENTITY 属性
教工号	Int	否	主键	
姓名	Varchar(20)	是	无	
家庭地址	Char(30)	是	无	
电话	Int	是	无	
性别	Char(2)	否	默认为‘男’	
课程名称	Varchar(10)	否	必须在 (‘数学’、 ‘语文’、 ‘化学’、 ‘英语’、 ‘物理’) 中	

根据条件在查询分析器中输入以下语句：

CREATE TABLE 教工表

(

教工号 INT PRIMARY KEY IDENTITY (1, 1) ,

姓名 VARCHAR (20) NULL,

家庭住址 CHAR (30) NULL,

电话 INT NULL,

性别 CHAR (2) DEFAULT ‘男’ ,  
 课程名 VARCHAR (10) CONSTRAINT chaname1  
 CHECK (课程名 IN ( ‘数学’ , ‘语文’ , ‘化学’ , ‘英语’ , ‘物  
 理’ ) )

)

删除此表：在查询分析器中输入语句

**DROP TABLE 教工表**

自己试着用企业管理器重新创建此表。

**练习 2：表的管理**

(1) 用企业管理器创建教工表 **teacher**，条件如下

列名	数据类型	允许为 空？	约束	IDENT ITY 属性
教工号	Int	否	主键	无
姓名	Varchar(20)	是	无	无
家庭住址	Char(30)	是	无	无
电话	Int	是	无	无
性别	Char(2)	否	默认为 ‘男’	无
系别	Varchar(16)	否	必须是 ( ‘计算 机软件’、‘计算机 体系结构’、‘计算 机网络’ ) 之一	无

(2) 在查询分析器中用命令：**DROP TABLE teacher** 删除该表。

(3) 根据条件在查询分析器中输入以下语句：

**CREATE TABLE teacher**

(

教工号 INT PRIMARY KEY IDENTITY(1,1),

姓名 VARCHAR(18) NULL,

家庭住址 CHAR(30) NULL,

电话 INT NULL,

职称 CHAR(14) DEFAULT '讲师',

所在专业 VARCHAR(16) CONSTRAINT chname1

CHECK(所在专业 IN ('计算机体系结构','计算机网络', ‘计算机软件’ ))

)

### 实验三 数据查询

学时：4

1. 实验类型：综合设计性

2. 实验类别：专业实验

3. 每组人数：5-6 人

4. 实验要求：必修

5. 实验目的

(1) 了解查询的概念和方法；

(2) 掌握查询分析器的使用方法；

(3) 掌握 SELECT 语句在单表查询中的应用；

(4) 掌握复杂查询的使用方法；

(5) 掌握多表连接的方法；

(6) 掌握 SELECT 语句在多表查询中的应用。

6. 实验内容

(1) 表的创建

1) 写出并执行语句创建教工表，条件如下：

列名	数据类型	允许为空	约束	IDENTITY 属性
教工号	Int	否	主键	无
姓名	Varchar(20)	是	无	无
家庭地址	Char(30)	是	无	无
性别	Char(2)	是	默认为“男”	无
电话	Int	是	无	无
工资	Int	是	无	无
系别	Varchar(16)	是	必须是（计算机软件， 计算机体系结构，计 算机网络）之一	无
职称	Varchar(10)	是	无	无
课程名称	Varchar(10)	是	必须在（语文，数学， 英语，化学，物理） 中	无

```
use flm
```

```
go
```

```

create table teacher
(
  teacherno int primary key,
  tname varchar (20),
  troom char(30),
  tsex char (2),
  tphone int,
  tsalary int,
  tdept varchar (16),
  tprof varchar(10),
  tcourse varchar(10),
)

```

(2) 插入数据 (不少于 5 条记录)

```

use flm
go
insert into teacher
values(2110,'伍三','学院路 1 号','男',15926456,5000,'计算机软件','教授','语文')
insert into teacher
values(2111,'张勇','学院路 2 号','男',16526488,5050,'计算机网络','副教授','数学')
insert into teacher
values(2112,'李立','学院路 1 号','女',16525488,5000,'计算机体系结构','教授','英语')
insert into teacher
values(2120,'王旭','学院路 3 号','男',13426488,5000,'计算机网络','副教授','化学')
insert into teacher
values(2140,'张寿','学院路 2 号','男',15526688,6000,'计算机软件','教授','物理')

```

(2) 单表查询

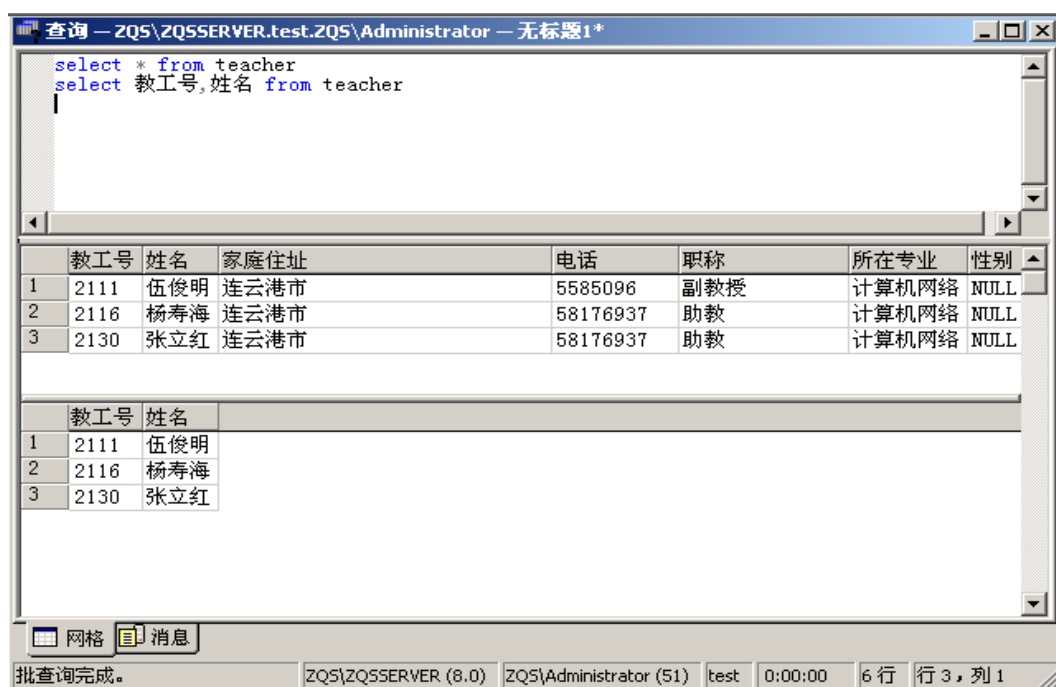
1) 基本查询

打开查询分析器，从 teacher 表中分别检索出教师的所有信息，以及教工号、姓名和职称。语句及查询结果如下：

```

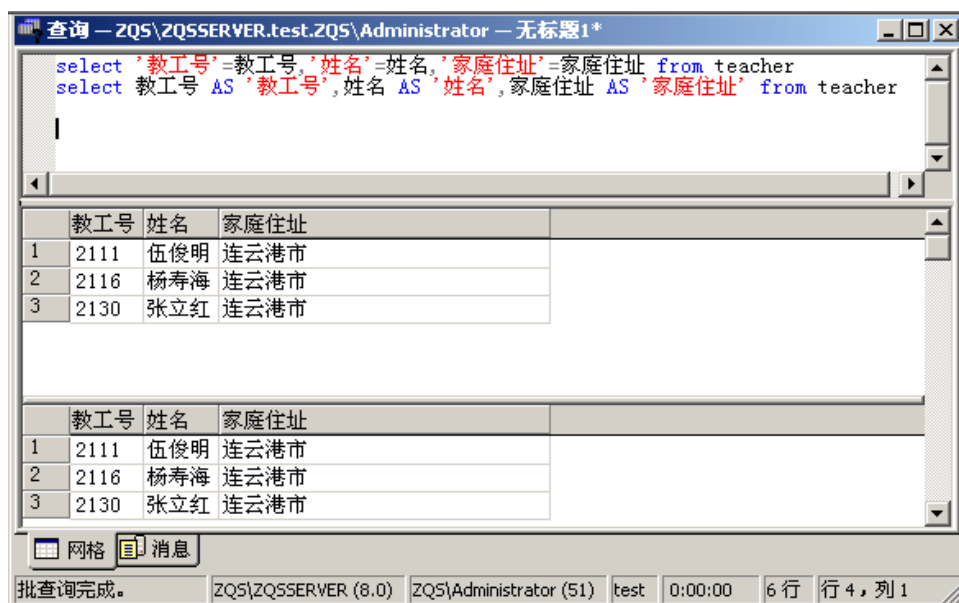
select * from teacher
select 教工号,姓名 from teacher

```



## 2) 查询时改变列标题的显示

从 teacher 表中分别检索出教师教工号、姓名、家庭住址信息并分别加上‘教师姓名’、‘教工号’、‘家庭住址’等标题信息，语句及查询结果如下：



## 3) 基于 WHERE 语句进行数据查询

a 基于比较条件。从 teacher 表中查询出教工号小于 2130 的教师资料。语句如下：

select \* from teacher where 教工号 < 2130

b 基于 BETWEEN 语句。从 teacher 表中查询出教工号介于 2100 和 2130 之间的教师资料。语句如下：

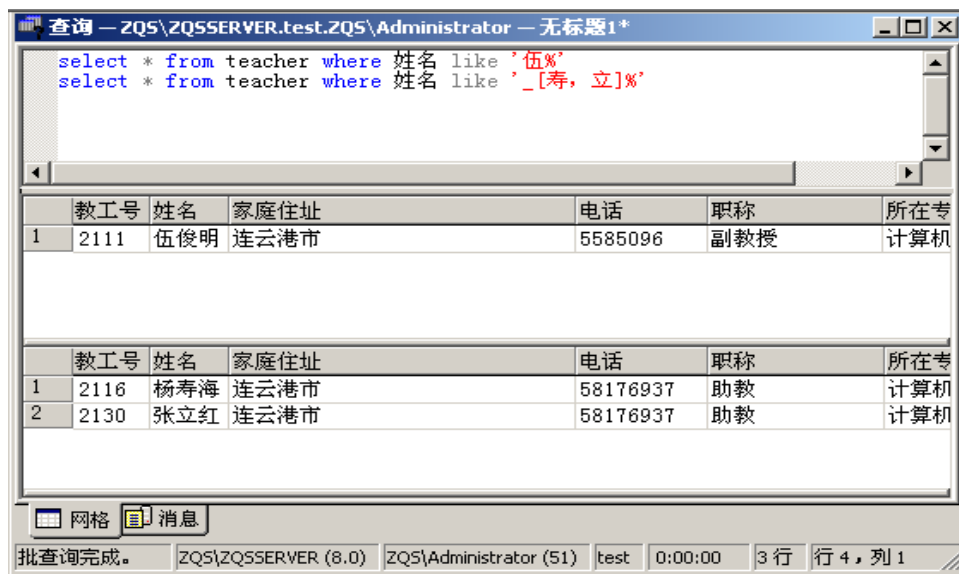
select \* from teacher where 教工号 between 2100 and 2130

c 基于 IN 子句的数据查询。从 teacher 表中查询出职称为“教授”或“副教授”的教师的教工号、教师姓名、职称及家庭住址。语句如下：

```
select 教工号,姓名,家庭住址,职称 from teacher
where 职称 in ('教授','副教授')
```

d 基于 LIKE 子句的查询。从 teacher 表中分别检索出姓伍的教师的资料，或者姓名的第 2 个字是寿或立的教师的资料。语句如下：

```
select * from teacher where 姓名 like '伍%'
select * from teacher where 姓名 like '_[寿,立]%'
```



查询 — ZQS\ZQSSERVER.test.ZQS\Administrator — 无标题1\*

```
select * from teacher where 姓名 like '伍%'
select * from teacher where 姓名 like '_[寿,立]%'
```

	教工号	姓名	家庭住址	电话	职称	所在专
1	2111	伍俊明	连云港市	5585096	副教授	计算机

	教工号	姓名	家庭住址	电话	职称	所在专
1	2116	杨寿海	连云港市	58176937	助教	计算机
2	2130	张立红	连云港市	58176937	助教	计算机

网格 消息

批查询完成。 ZQS\ZQSSERVER (8.0) ZQS\Administrator (51) test 0:00:00 3行 行 4, 列 1

4)使用 TOP 关键字查询。分别从 teacher 中检索出前 2 条及前面 67%的教师的信  
息。语句及查询结果如下：

```
select top 2 * from teacher
select top 67 percent * from teacher
```

查询 — ZQS\ZQSSERVER.test.ZQS\Administrator — 无标题1\*

```
select top 2 * from teacher
select top 70 percent * from teacher
```

	教工号	姓名	家庭住址	电话	职称	所在专
1	2111	伍俊明	连云港市	5585096	副教授	计算机
2	2116	杨寿海	连云港市	58176937	助教	计算机

	教工号	姓名	家庭住址	电话	职称	所在专
1	2111	伍俊明	连云港市	5585096	副教授	计算机
2	2116	杨寿海	连云港市	58176937	助教	计算机
3	2130	张立红	连云港市	58176937	助教	计算机

批查询完成。 ZQS\ZQSSERVER (8.0) ZQS\Administrator (51) test 0:00:00 5行 行 2, 列 15

5)使用 DISTINCT 关键字查询。从 teacher 表中检索出教师的职称并且要求显示的职称不重复。语句及查询结果如下：

select distinct 职称 from teacher

查询 — ZQS\ZQSSERVER.test.ZQS\Administrator — 无标题1\*

```
select distinct 职称 from teacher
```

	职称
1	副教授
2	助教

批查询完 ZQS\ZQSSERVER (8.0) ZQS\Administrator (51) test 0:00:00 2行 行 3, 列 1

6)用计算列查询。将 teacher 表中各教师的姓名、教工号及工资按 95%发放的信息，第 2 条语句将工资按 95%发放后列名该为‘预发工资’。语句如下：

select 教工号,姓名,工资\*0.95 from teacher

select 教工号,姓名,工资\*0.95 AS 预发工资 from teacher

查询 — ZQ5\ZQSSERVER.test.ZQ5\Administrator — 无标题1\*

```

select 教工号,姓名,工资*0.95 from teacher
select 教工号,姓名,工资*0.95 AS 预法工资 from teacher

```

	教工号	姓名	(无列名)
1	2111	伍俊明	3800.00
2	2116	杨寿海	2736.00
3	2130	张立红	2451.00

	教工号	姓名	预法工资
1	2111	伍俊明	3800.00
2	2116	杨寿海	2736.00
3	2130	张立红	2451.00

网格 消息

批查询完 ZQ5\ZQSSERVER (8.0) ZQ5\Administrator (51) test 0:00:00 6行 行 3, 列 1

### 7) 使用 ORDER BY 语句对查询的结果进行排序

使用 ORDER BY 语句可以对查询的结果进行排序, ASC、DESC 分别是升序和降序排列的关键字,系统默认的是升序排列。从 teacher 表中查询工资大于 2800 的教师的教工号、姓名,并按升序排列,语句及查询结果如下:

```

select 教工号,姓名 from teacher
where 工资>2800 order by 工资 ASC

```

查询 — ZQ5\ZQSSERVER.test.ZQ5\Administrator — 无标题1\*

```

select 教工号,姓名 from teacher
where 工资>2800 order by 工资 ASC

```

	教工号	姓名
1	2116	杨寿海
2	2111	伍俊明

网格 消息

批查询完 ZQ5\ZQSSERVER (8.0) ZQ5\Administrator (51) test 0:00:00 2行 行 3, 列 1

### (3) 多表查询

数据库的各个表中存放着不同的数据,用户经常需要用多个表中的数据来组合提炼出所需要的信息,如果一个查询需要对多个表进行操作,就称为联表查询,联表查询的结果集或结果表称为表之间的连接。联表查询实际上是通过各表之间共同列的关联来查询数据的,它是关系数据库查询最基本的特征。

按照下表所示,分别在数据库 test 中构造 student、course 和 student\_course 三张表,并写入记录。



Student 表

列名称	类型	宽度	允许为空	缺省值	主键
学号	Char	8	否		是
学生姓名	Nvarchar	8	否		
性别	Bit		否		
出生年月	Smalldatetime		否		
班级号	Char	6	否		
入学时间	Smalldatetime		否		
家庭住址	Nvarchar	40	是		

Course 表:

列名称	类型	宽度	允许为空	缺省值	主键
课程号	char	10	否		是
课程名称	Nvarchar	20	否		
书标识	Char	13	否		
课程总学时	Tinyint		是		
周学时	Tinyint		是		
课程学分	Tinyint		是		

Student\_course 表

列名称	类型	宽度	允许为空	缺省值	主键
课程号	Char	10	否		是
学号	Char	8	否		是
成绩	Tinyint		否		
学分	Tinyint		否		
学期	Bit		是		
学年	Char	9	是		

SQL 2008 兼容 2 种连接形式:用于 FROM 子句的 ANSI 连接语法和用于 WHERE 子句的 SQL SERVER 连接语法形式。

### 1) 进行多表查询

从 student、course 和 student\_course 三张表中检索学生的学号、姓名、学习课程号、学习课程名及课程成绩。语句如下:

```
select student. 学号, student_course. 学生姓名, student_course. 课程号, course. 课程名, student_course. 成绩 from student, student_course, course
where student. 学号=student_course. 学号
and course. 课程号=student_course. 课程号
```

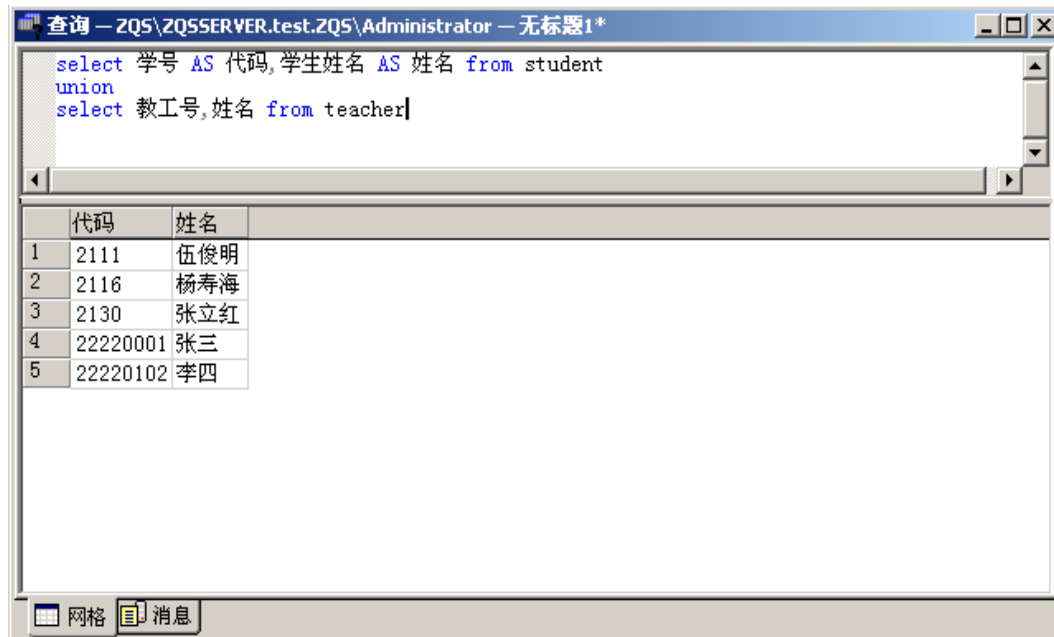
### 2) 使用 UNION 子句进行查询

使用 UNION 子句可以将一个或者多个表的某些数据类型相同的列显示在同一列上。如从 teacher 表中列出教工号、姓名并从 student 表中列出学号及学生姓名。语句及查询结果如下:

```
select 学号 AS 代码,学生姓名 AS 姓名 from student
```

union

select 教工号,姓名 from teacher



### 3) 用 GROUP 子句进行查询

如果要在数据检索时对表中数据按照一定条件进行分组汇总或求平均值,就要在 SELECT 语句中与 GROUP BY 子句一起使用集合函数。使用 GROUP BY 子句进行数据检索可得到数据分类的汇总统计、平均值或其他统计信息。

a 使用不带 HAVING 的 GROUP BY 子句。使用不带 HAVING 的 GROUP BY 子句汇总出 student\_course 表中的学生的学号及总成绩, 语句如下:

```
select '学号' = student. 学号, '总成绩' = SUM(成绩) from student_course
group by 学号
```

b 使用带 HAVING 的 GROUP BY 子句。使用带 HAVING 的 GROUP BY 子句汇总出 student\_course 表中总分大于 450 分的学生的学号及总成绩, 语句如下:

```
select '学号' = 学号, '总成绩' = SUM(成绩) FROM student_course
GROUP BY 学号 HAVING SUM(成绩) > 450
```

### 4) 使用 COMPUTE 和 COMPUTE BY 子句进行查询

使用 COMPUTE 和 COMPUTE BY 既能浏览数据又能看到统计的结果。

a 用 COMPUTE 子句汇总出 student\_course 表中每个学生的学号及总成绩, 语句如下:

```
select '学号' = 学号, '成绩' = 成绩 from student_course
ORDER BY 学号 COMPUTE SUM(成绩)
```

b 用 COMPUTE BY 子句按学号汇总出 student\_course 表中每个学生的学号及总成绩, 语句如下:

```
select '学号' = 学号, '成绩' = 成绩 from student_course
```

ORDER BY 学号

COMPUTE SUM (成绩) BY 学号

观察执行 COMPUTE 和 COMPUTE BY 子句的结果有什么不同?

(3)进行嵌套查询

1)使用 IN 或 NOT IN 关键字

使用 IN 关键字查询出 j10011 班所有男生的学号、课程号及相应的成绩，语句如下：

```
select student_course.学号, student_course.课程号, student_course.成绩
FROM student_course
WHERE 学号 IN
    (SELECT 学号 FROM student
```

```
    WHERE 班级= 'j10011' AND 性别=1)
```

2)使用 EXISTS 或 NOT EXISTS 关键字。使用 EXISTS 关键字查询出 'j10011' 班的学生的学号、课程号及相应的成绩，语句如下：

```
SELECT student_course.学号, student_course.课程号, student_course.成绩
FROM student_course
WHERE EXIST
    (SELECT * FROM student
    WHERE student_course.学号=student.学号 AND student.班级= 'j10011')
```

## 练习题

练习 1: 从 PUBS (SQL Server 样本数据库) 中的 authors, titleauthor, titles 表中选择出当年图书销量大于 10000 册的图书作者名称和图书名称。

练习 2: 统计各类图书的平均价格。

练习 3: 列出 business 类的图书名称和价格，结果按价格降序、书名升序排列。

练习 4: 列出所有作者为 Oakland 籍的图书。

练习 5: 查询所在州没有出版社的那些作者及其所在州名。

## 实验四 数据库的视图定义及维护

学时：4

1. 实验类型：综合设计性
2. 实验类别：专业实验
3. 每组人数：5-6 人
4. 实验要求：必修
5. 实验目的

- (1) 使学生掌握视图的定义与维护操作
- (2) 加深对视图在关系数据库中的作用的理解决

### 6. 实验内容

#### (1) 创建视图

写出创建满足下述要求的视图的 SQL 语句，并执行所写代码。

1) 包含学生的学号、姓名、所在系、年龄，课程号、课程名、课程学分的视图。

```
create view v1 as
select Student.Sno,Sname,Sdept,Sage,SC.Cno,Cname,Credit from
Student,SC,Course where Student.Sno=SC.Sno and Course.Cno=SC.Cno
```

2) 学生的学号、姓名、选修的课程名和考试成绩的视图。

```
create view v2 as
select Student.Sno,Sname,Cname,Grade from Student,SC,Course
where Student.Sno=SC.Sno and SC.Cno=Course.Cno
```

3) 统计每个学生的选课门数，要求列出学生学号和选课门数的视图。

```
create view v3 as
select Sno,COUNT(Cno) CourseNum
from SC group by Sno
```

4) 统计每个学生的修课总学分，要求列出学生学号和总学分的视图（说明：考试成绩大于等于 60 才可获得此门课程的学分）。

```
create view v4 as
select Sno,SUM(Credit) SumCredit
from SC,Course
where SC.Cno=Course.Cno and Grade>=60 group by Sno
```

#### (2) 查询和更新视图

利用第 1 题建立的视图，完成如下查询。

1) 查询考试成绩大于等于 90 分的学生的姓名、课程名和成绩。

```
select Sname,Cname,Grade
```

from v2

where Grade>90

2) 查询选课门数超过 3 门的学生的学号和选课门数。

select Sno,CourseNum

from v3

where CourseNum>3

3) 查询计算机系选课门数超过 3 门的学生的姓名和选课门数。

select Sname,CourseNum

from v3,Student

where v3.Sno=Student.Sno and CourseNum>3 and Sdept='计算机系'

4) 查询修课总学分超过 10 分的学生的学号、姓名、所在系和修课总学分。

select v4.Sno, Sname, Sdept, SumCredit

from v4, Student

where v4.Sno=Student.Sno and SumCredit>10

5) 查询年龄大于等于 20 岁的学生中, 修课总学分超过 10 分的学生的姓名、年龄、所在系和修课总学分。

select Sname, Sage, Sdept, SumCredit from v4, Student where  
v4.Sno=Student.Sno and SumCredit>10 and Sage>=20

(3) 使用 WITH CHECK OPTION, 在第 1 题建立的视图 (1) 的基础上, 再创建一个“计算机系”同学的学号、姓名、所在系、年龄, 课程号、课程名、课程学分的视图。

create view v5 as

select \* from v1 where Sdept='计算机系'

with check option;

增加一条记录

(0841101, 张新, 计算机系, 20, C001, 高等数学, 4)

修改这条记录为

(0841101, 张新, 物理系, 20, C001, 高等数学, 4)

以上两个操作是否能够实现? 为什么? 可以实现的, 写出 SQL 语句。

不能实现, 因为修改会影响多个基表。

## 练习题

1 建立信息系学生的视图

2 建立信息系学生的视图, 并要求进行修改和插入操作时仍须保证该视图只有信息系的学生。

- 3 建立信息系选修了' C1' 号课程的学生视图
- 4 建立信息系选修了' C1' 号课程且成绩在 90 分以上的学生视图
- 5 定义一个反映学生出生年份的视图

## 实验五 数据安全性控制

学时：4

1. 实验类型：综合设计性
2. 实验类别：专业实验
3. 每组人数：5-6 人
4. 实验要求：必修
5. 实验目的

(1) 通过实验使学生加深对数据安全性的理解

(2) 掌握 SQL Server 中有关用户，角色及操作权限的管理方法

### 6. 实验内容

在 SQL Server 企业管理器中，设置 SQL Server 的安全认证模式。

通过 SQL Server 企业管理器，实现对 SQL Server 的用户和角色管理。

分别通过 SQL Server 企业管理器和 SQL 的数据控制功能，设置和管理数据操作权限。

实验步骤：

●在 SQL Server 企业管理器中为所属的 SQL 服务器设置为 SQL Server 和 Windows NT 混合安全认证模式。其步骤如下：

在企业管理器窗口中展开服务器组，用鼠标右击需要设置的 SQL 服务器，在弹出的菜单中选择“属性”项，则出现 SQL Server 属性对话框，如图 1 所示。

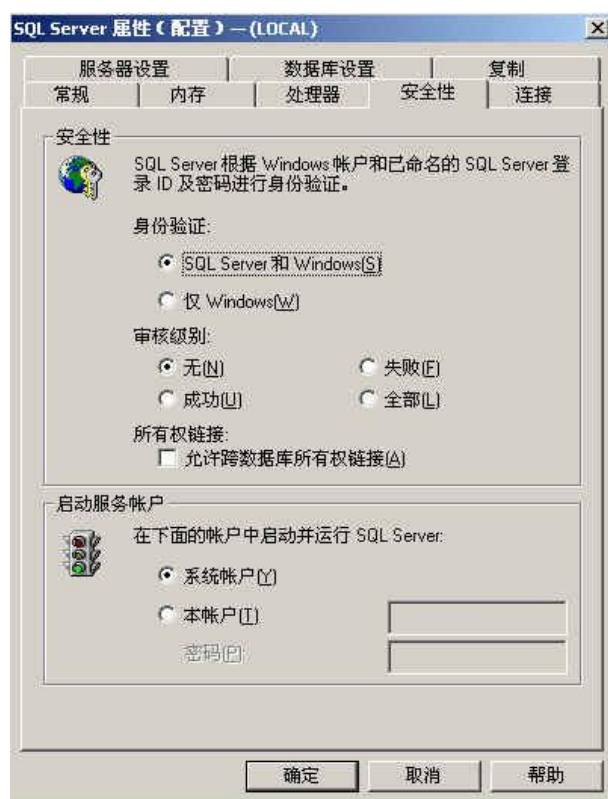


图 1 SQL Server 属性对话框

1) 在 SQL Server 属性对话框中选择“安全性”选项卡。

2) 身份验证选择“SQL Server 和 Windows”单选项。

●在 SQL Server 企业管理器中为自己建立一个服务器用户、数据库用户和数据库角色。

在企业管理器窗口中展开服务器组，展开服务器，用鼠标单击“安全性”文件夹右侧的‘+’，用鼠标右击“登录”，在弹出的菜单中选择“新建登录”项，则出现新建登录对话框，如图 2 所示。



图 2 新建登录对话框

1) 在新建登录对话框中有常规，服务器角色和数据库访问三个选项卡。

在常规选项卡中，输入用户名（本例为王平），选择 SQL Server 安全验证，输入用户口令。

在服务器角色选项卡中，需要确定用户所属的服务器角色，在本例中采用缺省值即可。

在数据库访问选项卡中，需要指定此登录可以访问的数据库（本例选中学生选课数据库）和“学生选课”的数据库角色（在本例中采用缺省值即可）。

单击“确定”按钮，既完成了创建登录用户的工作。



- 关闭 SQL Server 企业管理器，打开 SQL 查询分析器。选择 SQL Server 安全验证，用户名为王平，输入用户口令，连接到 SQL Server。在“查询”窗口中可以输入 SQL 语句。如“SELECT \* FROM 学生”。运行后，得到消息“拒绝了对对象‘学生’（数据库‘学生选课’，所有者‘dbo’）的 SELECT 权限。”。可见用户王平没有对学生表的 SELECT 权限。
- 要想将学生选课数据库的操作权限赋予数据库用户王平，有两种方法。

方法一：通过企业管理器

- 1) 在企业管理器窗口中展开服务器组，展开服务器，用鼠标单击“数据库”文件夹右侧的‘+’，用鼠标单击学生选课数据库文件夹右侧的‘+’，用鼠标右击“用户”。在屏幕右侧的“用户”窗口中选择“王平”项，用鼠标右击，在弹出的菜单中选择“属性”项，则出现数据库用户属性对话框，如图 3 所示。



图 3 数据库用户属性对话框

- 2) 单击图 3 中的权限按钮，则出现数据库用户权限对话框，如图 4 所示对话框的下面是有关数据库用户和角色所对应的权限表。这些权限均以复选框的形式表示。复选框有三种状态：“√”为授权；“×”为

废除权；空为撤权。在表中可以对用户或角色的各种对象操作权（SELECT，INSERT，UPDATE，DELETE，EXEC 和 DRI）进行授予或撤消。



图 4 数据库用户权限对话框

3) 在图 4 中找到学生表，授予 SELECT 权限，即让学生表与 SELECT 列交叉的复选框为“√”即可。

方法二：通过 SQL 的数据控制功能

对用户王平授权，必须是数据库对象拥有者以上用户授予。我们可以以系统管理员或 sa 用户登录进入查询分析器。在查询分析器中输入授权语句“GRANT SELECT ON 学生 TO 王平；”，然后执行即可。

## 实验六 数据完整性控制

学时：4

1. 实验类型：验证性
2. 实验类别：专业实验
3. 每组人数：5-6 人
4. 实验要求：必修
5. 实验目的
  - (1) 通过实验使学生加深对数据完整性的理解
  - (2) 学会创建和使用触发器。
6. 实验内容

相关知识：

SQL Server 实现数据完整性的具体方法有 4 种：约束、缺省、规则和触发器。其中约束和缺省在实验一中已接触过。本实验重点学会创建和使用触发器。

触发器是实施复杂完整性的特殊类型的存储类型。触发器不需要专门语句调用，当对它所保护数据进行修改时自动激活，以防止对数据进行不正确，未授权或不一致的修改。

创建触发器的语法为：

```
CREATE TRIGGER < 触发器> ON <表名>
[WITH ENCRYPTION]
FOR {[DELETE][, ][INSERT][, ][UPDATE]}
[WITH APPEND]
[NOT FOR REPLICATION]
AS <SQL 语句组>
```

其中：

- 1) WITH ENCRYPTION 为加密选项。
- 2) DELETE 选项为创建 DELETE 触发器。DELETE 触发器的作用是当对表执行 DELETE 操作时触发器被激活，并从指定表中删除元组，同时将删除的元组放入一个特殊的逻辑表（delete 表）中。触发器的动作可以检查 delete 表中的数据，以确定下一步该如何处理。
- 3) INSERT 选项为创建 INSERT 触发器。INSERT 触发器在对指定表中执行插入数据操作时激活，激活后将插入表中的数据拷贝并送入一个特殊的逻辑表（inserted 表）中，触发器会根据 INSERT 表中的值决定如何处理。
- 4) UPDATE 选项为创建 UPDATE 触发器。UPDATE 触发器仅在对指定表中进行更新数据操作时激活。UPDATE 触发器激活后把将要被更新的原数据移入 delete 表中再将要被更新后的新数据的备份送入 insert 表中，UPDATE

触发器对 delete 和 inserted 表进行检查，并决定如何处理。

- 5) NOT FOR REPLICATION 选项说明当一个复制过程在修改一个触发器表时，与该表相关联的触发器不能被执行。

一个触发器只适用于一个表，每个表最多只能有三个触发器，它们分别是 INSERT、UPDATE 和 DELETE 触发器。触发器仅在实施数据完整性和处理业务规则时使用。

#### 设计内容及步骤:

- 1、创建触发器，要求在学生表中删除一个学生时，同时从选课表中将其所有选课信息删除，并进行测试。

```
CREATE TRIGGER student_delete on student
FOR DELETE
AS
DELETE FROM SC
WHERE SC.SNO IN(SELECT SNO FROM DELETED)
```

说明：该触发器是删除记录事件产生时，系统自动运行的。请同学们在 student 中选中几个学生，查看其在 SC 表有几条记录。然后，在 student 中，删除这个学生，再查看其在 SC 中是否还有记录。

- 2、创建触发器限定一个学生最多只能选择 4 门课，并进行测试。

```
CREATE TRIGGER student_INSERT ON SC
FOR INSERT
AS
DECLARE @CL INT
SELECT @CL=COUNT(*) FROM SCORE WHERE SNO = (SELECT SNO
FROM INSERTED )
IF (@CL>=5)
BEGIN
ROLLBACK
PRINT'超过门课，不能再选'
END
```

说明：该触发器是插入记录事件产生时，系统自动运行的。请同学们使用下面的 SQL 语句，测试该触发器：

```
INSERT SC('201215121','1',60)
```

- 3、利用触发器限定修改后的分数只能比原来高，并进行测试。

```
CREATE TRIGGER student_update ON SC
```

```

FOR UPDATE
AS
IF((SELECT COUNT(*) FROM INSERTED WHERE
GRADE > (SELECT GRADE FROM DELETED))=0)
BEGIN
ROLLBACK
PRINT ' 修改后的分数比原来的低，不允许'
END

```

说明：该触发器是修改记录事件产生时，系统自动运行的。执行以下 SQL 语句，测试该触发器：

```

(1) SELECT * FROM SC
WHERE SNO = '201215121' AND CNO='1'
(2) UPDATE SC SET GRADE=60 WHERE SNO=' 201215121' AND
CNO=' 1'

```

4、创建一个按名字模糊查询学生基本信息的存储过程。

```

CREATE PROCEDURE Pro_student (@sname varchar(20))
AS
(
SELECT SNO AS '学号', SNAME AS '姓名', SSEX AS '性别',
SAGE AS '年龄', SDEPT AS '系部'
WHERE SNAME LIKE @sname
)

```

说明：存储过程必须先定义，然后在客户端进行调用

在查询编辑器中，执行方法为：EXEC 存储过程名 参数

本题中调用存储过程查询名字中含有的学生信息方法为，在查询编辑器中，执行如下语句：

```
EXEC PROCEDURE '%丽%'
```

5、创建一个存储过程，可以根据指定的学生学号，通过参数返回该学生的姓名和所选课程的平均分。

```

CREATE PROCEDURE student_nameaverage(
@c1 varchar(8),@c2 varchar(10) OUTPUT,@c3 float OUTPUT)
AS
SELECT @c2=SNAME,@c3=AVG(GRADE)
FROM student, sc

```

Where student.SNO=sc.SNO AND student.SNO=@c1

Group by student.sno,student.sname

说明：存储过程可以通过实参返回数据。请同学们使用下面的 SQL 语句调用该存储过程，查询学号为 ‘20002059’ 的学生的姓名和平均分：

```
DECLARE @S_NAME CHAR(10)
```

```
DECLARE @S_AVG FLOAT
```

```
EXEC student_nameaverage '201215121', @S_NAME OUTPUT, @S_AVG
```

```
OUTPUT
```

```
PRINT @S_NAME
```

```
PRINT @S_AVG
```