

第二部分 综合设计实验

实验一 课程成绩管理数据库

一、实验目的

1. 理解数据库系统的三级模式结构与两级映像机制；
2. 体会数据库的数据独立性特征；
3. 掌握 ER 模型的使用；
4. 掌握将 ER 模型转换为关系模型的方法；
5. 了解数据库设计的完整过程；
6. 掌握数据库的建立和重构方法；
7. 掌握数据添加、数据删除和数据更新的方法；
8. 掌握各种数据查询方法；
9. 了解数据库的安全性控制机制。

二、实验内容`

1. 课程成绩管理的业务要求
 - 1) 与课程有关的信息有：课程编号、课程名称、学分；
 - 2) 与学生有关的信息有：学号、姓名、班级；
 - 3) 与教师有关的信息有：工作证编号、姓名、所在学院；
 - 4) 与班级有关的信息有：班级名称；
 - 5) 一名学生属于一个班级，一个班级有多名学生；
 - 6) 一个班级可以开设多门课程，一门课程可能为多个班级开设；
 - 7) 一名学生可以选修为其所在班级开设的多门课程，为一个班级开设的一门课程可以有多名学生选修；
 - 8) 一名教师可以主讲为多个班级开设的多门课程，一门课程

可以有多名教师为多个班级主讲，为一个班级开设的一门课程只能由一名教师主讲；

- 9) 一名学生选修的一门课程只能有一个成绩，这个成绩只能由主讲该生所在班级该课程的教师评定；
- 10) 对各种数据的格式要求如表 9.1 所示；
- 11) 开设 3 门课程，具体信息如 表 9.2 所示；
- 12) 共有 6 名学生，具体信息如表 9.3 所示；
- 13) 共有 5 名教师，具体信息如表 9.4 所示；
- 14) 共有 2 个班级，具体信息如表 9.5 所示；
- 15) 教务员负责建立和维护课程、班级、教师和学生的档案，为各班开设课程，为各个班的各门课程分配主讲教师，按班级打印各门课程每个学生的成绩并计算平均成绩。各班课程开设及主讲教师分配情况如表 9.6 所示；

表 9.1 各种数据格式要求

数据	类型	长度	其他说明
课程编号	字符	6	课程的唯一标识
课程名称	字符	20	无
学分	正整数	2	无
学号	字符	8	学生的唯一标识
姓名	字符	20	无
班级名称	字符	16	班级的唯一标识
工作证编号	字符	6	教师的唯一标识
所在学院	字符	20	无
成绩	非负整数	3	不大于 100

表 9.2 课程数据表

课程编号	课程名称	学分
050304	计算机科学导引	4
120509	建筑力学	6
910201	世界科技史	3

表 9.3 学生数据表

学号	姓名	班级
40533123	诸葛飞扬	信息 0519
40533099	雷铭	信息 0519
40533103	贺舞	信息 0519
40525009	卫小虎	建筑 0501
40525011	邵剑青	建筑 0501
40525024	欧阳雨晨	建筑 0501

表 9.4 教师数据表

工作证编号	姓名	所在学院
032011	向华泽	信息科学与技术学院
032036	梅鹤	信息科学与技术学院
011221	张天啸	文法学院
011224	秦放	文法学院
021121	萧雁鸣	建筑科学与工程学院

表 9.5 班级数据表

班级名称
信息 0519
建筑 0501

表 9.6 各班课程开设及主讲教师分配情况

班级	课程	主讲教师
信息 0519	计算机科学导引	向华泽
信息 0519	世界科技史	张天啸
建筑 0501	计算机科学导引	梅鹤
建筑 0501	建筑力学	萧雁鸣
建筑 0501	世界科技史	秦放

- 16) 学生可以选课和查看成绩；
- 17) 教师负责给出成绩、更正成绩和按班级打印各门课程每个学生的成绩并计算平均成绩；
- 18) 学生选课情况如表 9.7 所示；

19) 学生各门课程成绩如表 9.8 所示;

表 9.7 学生选课情况

学生	课程	班级
诸葛飞扬	计算机科学导引	信息 0519
诸葛飞扬	世界科技史	信息 0519
雷铭	计算机科学导引	信息 0519
雷铭	世界科技史	信息 0519
贺舞	计算机科学导引	信息 0519
贺舞	世界科技史	信息 0519
卫小虎	计算机科学导引	建筑 0501
卫小虎	建筑力学	建筑 0501
卫小虎	世界科技史	建筑 0501
邵剑青	建筑力学	建筑 0501
邵剑青	世界科技史	建筑 0501
欧阳雨晨	计算机科学导引	建筑 0501
欧阳雨晨	建筑力学	建筑 0501

表 9.8 学生各门课程成绩

学生	课程	成绩
诸葛飞扬	计算机科学导引	85
诸葛飞扬	世界科技史	77
雷铭	计算机科学导引	69
雷铭	世界科技史	88
贺舞	计算机科学导引	75
贺舞	世界科技史	89
卫小虎	计算机科学导引	84
卫小虎	建筑力学	90
卫小虎	世界科技史	79
邵剑青	建筑力学	80
邵剑青	世界科技史	72
欧阳雨晨	计算机科学导引	81
欧阳雨晨	建筑力学	83

2. 实验任务

- 1) 根据业务要求建立数据库
- 2) 根据业务要求建立该数据库的用户
- 3) 根据业务要求以不同的用户身份使用该数据库

三、实验步骤

1. 使用 ER 模型进行概念设计。
2. 将得到的 ER 模型转换成关系模型。
3. 定义数据库模式(全局模式)。
4. 定义数据库外模式(子模式)。
5. 定义数据库用户：
 - 1) 教务员；
 - 2) 学生；
 - 3) 教师。
6. 以不同用户的身份使用数据库：
 - 1) 以教务员的身份输入课程、学生、教师的数据，并为课程分配教师；
 - 2) 以学生的身份选课；
 - 3) 以教师的身份给出成绩；
 - 4) 以学生的身份查看成绩；
 - 5) 以教师的身份更改成绩；
 - 6) 以教务员的身份更改成绩；
 - 7) 以教师的身份打印成绩单；
 - 8) 以教务员的身份打印成绩单。
7. 重新分配用户权限，收回教师的更改成绩权限，分配给教务员更改成绩权限。
8. 以不同用户的身份使用数据库：
 - 1) 以教师的身份更改成绩；
 - 2) 以教务员的身份更改成绩；
 - 3) 以教师的身份打印成绩单；
 - 4) 以教务员的身份打印成绩单。
9. 修改数据库模式，在教师信息中增加“职称”一项(字符类型，长度为 6)，并且删除“所在学院”一项。
10. 以不同的身份使用数据库：

- 1) 以学生的身份查看成绩；
- 2) 以教师的身份打印成绩单；
- 3) 以教务员的身份增加一名教师。

四、思考题

1. 数据库系统的三级模式结构与两级映像机制在 SQL server 2000 中是如何实现的？
2. 本实验中如何发挥数据库的数据独立性优点？
3. 在数据库设计过程中 ER 模型起到什么作用？
4. 本实验中数据库的安全性控制机制如何发挥作用？

五、实验报告内容及要求

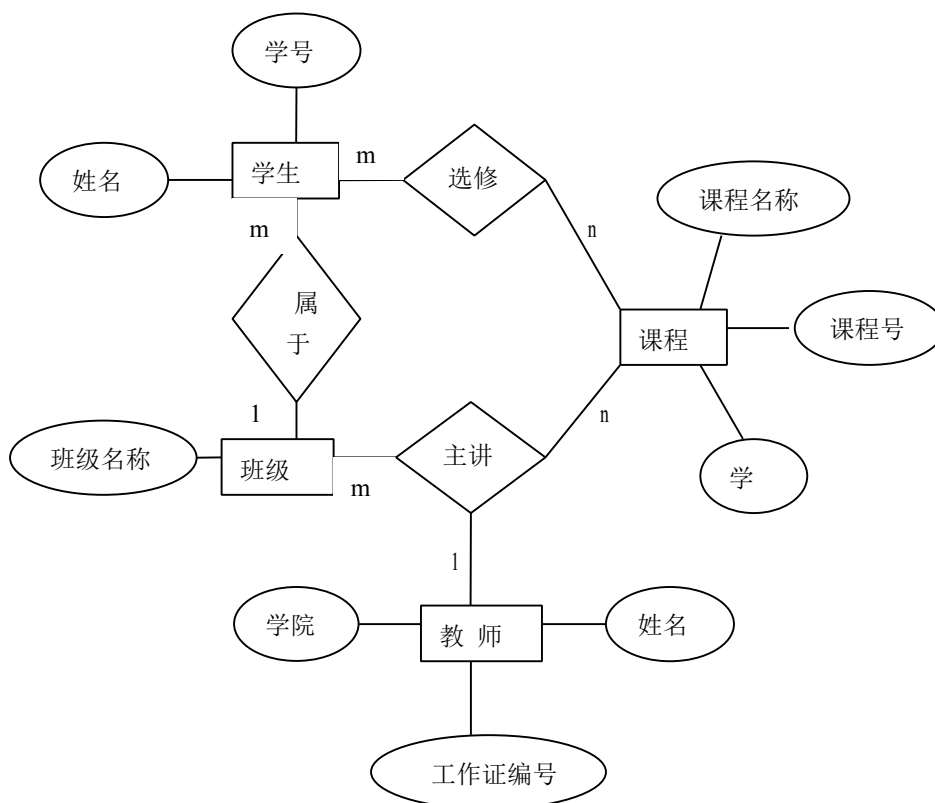
1. 本实验的数据库设计及建立的完整过程及各环节的产品；
2. 本实验所建立的数据库的三级模式结构与两级映像机制及实现；
3. 本实验所建立的数据库的数据独立性分析；
4. 本实验所建立的数据库的安全性控制机制；
5. 实验结论。

附录一：实验报告格式

1. 实验报告要求用 A4 纸张；
2. 一级标题使用黑体 4 号字，二级标题使用黑体小 4 号字，三级标题使用加黑宋体小 4 号字，正文使用宋体小 4 号字，不得使用斜体字，页面行距取 1.5 倍；
3. 所有的图表均要注明图号或表号。

附录二：实验参考文档

1. 课程成绩管理数据库系统的 ER 模型如附图 1 所示。



附图 1 课程成绩管理数据库系统的 ER 模型

2. 关系模型

实体“学生”的关系模式：学生（学号，姓名，班级），该关系模式包含联系“属于”所对应的关系模式。

实体“课程”的关系模式：课程（课程号，课程名称，学分）。

实体“教师”的关系模式：教师（工作证编号，姓名，学院）。

联系“选修”的关系模式：选修（学号，课程号，分数）。

联系“主讲”的关系模式：主讲（工作证编号，课程号，班级）。

文档中各模式的英文表示分别为：

Student(Sno,Sname,Ssex,Class)；


```
Course(Cno,Cname,Cridet) ;
Teacher(Tno,Tname,Tdept) ;
Enrollment(Sno,Cno,Grade) ;
TeachCourse(Tno,Cno,Class).
```

3. 数据库模式

1) 创建 TeachManagement 的脚本文件

```
USE master
GO
CREATE DATABASE TeachManagement
ON
( NAME = TeachManagement_dat,
  FILENAME = 'd:\program files\microsoft sql server\mssql\data\
TeachManagement_dat.mdf',
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5 )
LOG ON
( NAME = 'TeachManagement_log',
  FILENAME = 'd:\program files\microsoft sql server\mssql\data\
TeachManagement_log.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB )
GO
```

2) 创建 TeachManagement 中各个表的脚本文件

```
USE TeachManagement
if exists (select * from dbo.sysobjects where id = object_id
(N'[dbo].[FK_Enrollment_Course]') and OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [dbo].[Enrollment] DROP CONSTRAINT
FK_Enrollment_Course
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id
(N'[dbo].[FK_Enrollment_Student]') and OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [dbo].[Enrollment] DROP CONSTRAINT
```

FK_Enrollment_Student
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Course]')
and OBJECTPROPERTY(id, N'IsUSERTable') = 1)
drop table [dbo].[Course]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Enrollment]') and OBJECTPROPERTY(id,
N'IsUSERTable') = 1)
drop table [dbo].[Enrollment]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Student]') and OBJECTPROPERTY(id, N'IsUSERTable')
= 1)
drop table [dbo].[Student]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[TeachCourse]') and OBJECTPROPERTY(id,
N'IsUSERTable') = 1)
drop table [dbo].[TeachCourse]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Teacher]') and OBJECTPROPERTY(id, N'IsUSERTable')
= 1)
drop table [dbo].[Teacher]
GO

CREATE TABLE [dbo].[Course] (
[Cno] [char] (6) COLLATE Chinese_PRC_CI_AS NOT NULL PRIMARY
KEY ,
[Cname] [char] (20) COLLATE Chinese_PRC_CI_AS NULL ,
[Credit] [int] NULL

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Enrollment] (
[Sno] [char] (8) COLLATE Chinese_PRC_CI_AS NOT NULL,
[Cno] [char] (6) COLLATE Chinese_PRC_CI_AS NOT NULL ,
[Grade] [int] NULL,
PRIMARY KEY ([Sno], [Cno])
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Student] (
[Sno] [char] (8) COLLATE Chinese_PRC_CI_AS NOT NULL PRIMARY
KEY,
[Sname] [char] (20) COLLATE Chinese_PRC_CI_AS NULL ,
[Class] [char] (16) COLLATE Chinese_PRC_CI_AS NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[TeachCourse] (
[Class] [char] (16) COLLATE Chinese_PRC_CI_AS NOT NULL,
[Cno] [char] (6) COLLATE Chinese_PRC_CI_AS NOT NULL ,
[Tno] [char] (6) COLLATE Chinese_PRC_CI_AS NOT NULL,
PRIMARY KEY ([Class], [Cno])
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Teacher] (
[Tno] [char] (6) COLLATE Chinese_PRC_CI_AS NOT NULL PRIMARY
KEY ,
[Tname] [char] (20) COLLATE Chinese_PRC_CI_AS NULL ,
[Tdept] [char] (20) COLLATE Chinese_PRC_CI_AS NULL
) ON [PRIMARY]
GO
```

4. 数据库外模式

创建 **Grade** 视图，供学生和教师查询。
成绩（班级，姓号，课程名称，学分，成绩）

Grade(Class,Sname,Cname,Credit,Grade)

创建 Grade 视图的脚本文件如下：

```
CREATE VIEW dbo.Grade
AS
SELECT  dbo.Student.Class,  dbo.Student.Sname,  dbo.Course.Cname,
        dbo.Course.Credit,
        dbo.Enrollment.Grade
FROM  dbo.Enrollment INNER JOIN
        dbo.Course ON dbo.Enrollment.Cno = dbo.Course.Cno INNER JOIN
        dbo.Student ON dbo.Enrollment.Sno = dbo.Student.Sno
```

5. 定义数据库用户：

1) 教务员 (Dean)

定义数据库用户方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

为该用户分配权限的脚本文件如下：

```
USE TeachManagement
GO
GRANT ALL PRIVILEGES ON Student, Course, Teacher, TeachCourse TO
Dean
GO
```

2) 学生 (Student)

定义数据库用户方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

为该用户分配权限的脚本文件如下：

```
USE TeachManagement
GO
GRANT SELECT, INSERT(Sno,Cno), UPDATE(Sno,Cno) ON Enrollment
TO Student
GO
```

3) 教师 (Teacher)

定义数据库用户方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

为该用户分配权限的脚本文件如下：

```
USE TeachManagement
GO
GRANT SELECT, UPDATE(Grade) ON Enrollment TO Teacher
GO
```

6. 以不同用户的身份使用数据库:

- 1) 以教务员的身份输入课程、学生、教师的数据, 并为课程分配教师;

在查询分析器中以教务员身份登录, 进行下面操作。以不同身份登录的方法见第一部分第5章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Course]')
and OBJECTPROPERTY(id, N'IsUSerTable') = 1)
```

```
delete from [dbo].[Course]
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Student]') and OBJECTPROPERTY(id, N'IsUSerTable')
= 1)
```

```
delete from [dbo].[Student]
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Teacher]') and OBJECTPROPERTY(id, N'IsUSerTable')
= 1)
```

```
delete from [dbo].[Teacher]
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[TeachCourse]') and OBJECTPROPERTY(id,
N'IsUSerTable') = 1)
```

```
delete from [dbo].[TeachCourse]
```

```
GO
```

```
INSERT INTO Course (Cno,Cname,Credit) VALUES ('050304','计算机科学
导引','4')
```

```
INSERT INTO Course (Cno,Cname,Credit) VALUES ('120509','建筑力学
','6')
```

```
INSERT INTO Course (Cno,Cname,Credit) VALUES ('910201','世界科技史
','3')
```

```
GO
```

```
INSERT INTO Student(Sno,Sname,Class)VALUES('40533123','诸葛飞扬','
信息 0519')
```

```
INSERT INTO Student(Sno,Sname,Class)VALUES('40533099','雷铭','信息
0519')
```

```
INSERT INTO Student(Sno,Sname,Class)VALUES('40533103','贺舞','信息
```

```

0519')
INSERT INTO Student(Sno,Sname,Class)VALUES('40525009','卫小虎','建筑 0501')
INSERT INTO Student(Sno,Sname,Class)VALUES('40525011','邵剑青','建筑 0501')
INSERT INTO Student(Sno,Sname,Class)VALUES('40525024','欧阳雨晨','建筑 0501')
GO
INSERT INTO Teacher(Tno,Tname,Tdept)VALUES('032011','向华泽','信息科学与技术学院')
INSERT INTO Teacher(Tno,Tname,Tdept)VALUES('032036','梅鹤','信息科学与技术学院')
INSERT INTO Teacher(Tno,Tname,Tdept)VALUES('011221','张天啸','文法学院')
INSERT INTO Teacher(Tno,Tname,Tdept)VALUES('011224','秦放','文法学院')
INSERT INTO Teacher(Tno,Tname,Tdept)VALUES('021121','萧雁鸣','建筑科学与工程学院')
GO
INSERT INTO TeachCourse(Class,Cno,Tno)VALUES('建筑 0501','050304','032036')
INSERT INTO TeachCourse(Class,Cno,Tno)VALUES('建筑 0501','120509','021121')
INSERT INTO TeachCourse(Class,Cno,Tno)VALUES('建筑 0501','910201','011224')
INSERT INTO TeachCourse(Class,Cno,Tno)VALUES('信息 0519','050304','032011')
INSERT INTO TeachCourse(Class,Cno,Tno)VALUES('信息 0519','910201','011221')
GO

```

2) 以学生的身份选课；

在查询分析器中以学生身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```

USE TeachManagement
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Enrollment]') and OBJECTPROPERTY(id,
N'IsUSERTable') = 1)

```

```
delete from [dbo].[Enrollment]
```

```
GO
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40533123','050304')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40533123','910201')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40533099','050304')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40533099','910201')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40533103','050304')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40533103','910201')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40525009','050304')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40525009','120509')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40525009','910201')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40525011','120509')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40525011','910201')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40525024','050304')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40525024','120509')
```

```
GO
```

3) 以教师的身份给出成绩；

在查询分析器中以教师身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
GO
```

```
UPDATE Enrollment SET Grade='85' WHERE Sno='40533123' AND  
Cno='050304'
```

```
UPDATE Enrollment SET Grade='77' WHERE Sno='40533123' AND  
Cno='910201'
```

```
UPDATE Enrollment SET Grade='69' WHERE Sno='40533099' AND  
Cno='050304'
```

```
UPDATE Enrollment SET Grade='88' WHERE Sno='40533099' AND  
Cno='910201'
```

```
UPDATE Enrollment SET Grade='75' WHERE Sno='40533103' AND  
Cno='050304'
```

```
UPDATE Enrollment SET Grade='89' WHERE Sno='40533103' AND  
Cno='910201'
```

```
UPDATE Enrollment SET Grade='84' WHERE Sno='40525009' AND  
Cno='050304'
```

```
UPDATE Enrollment SET Grade='90' WHERE Sno='40525009' AND  
Cno='120509'
```

```
UPDATE Enrollment SET Grade='79' WHERE Sno='40525009' AND  
Cno='910201'
```

```
UPDATE Enrollment SET Grade='80' WHERE Sno='40525011' AND  
Cno='120509'
```

```
UPDATE Enrollment SET Grade='72' WHERE Sno='40525011' AND  
Cno='910201'
```

```
UPDATE Enrollment SET Grade='81' WHERE Sno='40525024' AND  
Cno='050304'
```

```
UPDATE Enrollment SET Grade='83' WHERE Sno='40525024' AND  
Cno='120509'
```

```
GO
```

4) 以学生的身份查看成绩；

在查询分析器中以学生身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
SELECT * FROM Grade
```

```
GO
```

5) 以教师的身份更改成绩；

在查询分析器中以教师身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
UPDATE Enrollment SET Grade='90' WHERE Sno='40533123' AND  
Cno='050304'
```

```
GO
```

6) 以教务员的身份更改成绩；

在查询分析器中以教务员身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
UPDATE Enrollment SET Grade='95' WHERE Sno='40533123' AND  
Cno='050304'
```

```
GO
```

7) 以教师的身份打印成绩单；

在查询分析器中以教师身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

这里使用了游标，可以依次打印各个班级的成绩，并有一些相应的格式上的设置。当然也可以不使用游标，分别对各班进行打印。

在“查询分析器”下，将查询结果的显示模式设置为“文本显

示结果”（设置方法见第一部分第4章），运行如下脚本：

```
USE TeachManagement
GO
SET NOCOUNT ON
DECLARE @Class char(16), @Average float
PRINT '----- 各班成绩打印 -----'
PRINT "
DECLARE print_cursor CURSOR FOR
SELECT DISTINCT Class FROM Student
ORDER BY Class
OPEN print_cursor
FETCH NEXT FROM print_cursor
INTO @Class
WHILE @@FETCH_STATUS = 0
BEGIN
    PRINT '-----'+@Class+'班成绩-----'
    -- Get average
    SELECT @Average=AVG(Grade) FROM Enrollment, Student WHERE
(Enrollment.Sno = Student.Sno) AND (Class=@Class)
    --Print grade and average
    SELECT * FROM Grade WHERE Class=@Class
    PRINT @Class+'班平均成绩:'+STR(@Average)
PRINT "
    -- Get the next Class
    FETCH NEXT FROM print_cursor
    INTO @Class
END
CLOSE print_cursor
DEALLOCATE print_cursor
GO
```

8) 以教务员的身份打印成绩单。

在查询分析器中以教务员身份登录，进行下面操作。以不同身份登录的方法见第一部分第5章 SQL Server 2000 用户与权限管理。

在“查询分析器”下，将查询结果的显示模式设置为“文本显示结果”（设置方法见第一部分第4章），运行第7)小题中的成绩打印脚本。

7. 重新分配用户权限，收回教师的更改成绩权限，分配给教务员更改成绩

权限。

在查询分析器中以管理员(sa)身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
REVOKE UPDATE(Grade) ON TABLE Grade FROM Teacher
```

```
GRANT UPDATE(Grade) ON Grade TO Dean
```

8. 以不同用户的身份使用数据库：

1) 以教师的身份更改成绩；

在查询分析器中以教师身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
UPDATE Grade SET Grade='90' WHERE Sno='40533123' AND  
Cno='050304'
```

```
GO
```

2) 以教务员的身份更改成绩；

在查询分析器中以教务员身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
UPDATE Grade SET Grade='95' WHERE Sno='40533123' AND  
Cno='050304'
```

```
GO
```

3) 以教师的身份打印成绩单；

在查询分析器中以教师身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

在“查询分析器”下，将查询结果的显示模式设置为“文本显示结果”（设置方法见第一部分第 4 章），运行 6 题第 7) 小题中的成绩打印脚本。

4) 以教务员的身份打印成绩单。

在查询分析器中以教务员身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

在“查询分析器”下，将查询结果的显示模式设置为“文本显示结果”（设置方法见第一部分第 4 章），运行 6 题第 7) 小题中的成绩打印脚本。

9. 修改数据库模式，在教师信息中增加“职称”一项（字符类型，长度为 6），并且删除“所在学院”一项。

在查询分析器中以管理员(sa)身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

在教师信息中增加“职称”一项的 SQL 语句如下：

```
USE TeachManagement
```

```
ALTER TABLE Teacher ADD Position CHAR(6)
```

```
GO
```

在 SQL 中，由于不能删除表的某一属性列，采用下面的替代方案，即先将所需列复制至备份表中，再删除原表。

```
if exists (select * from dbo.sysobjects where id =  
object_id(N'[dbo].[TeacherTemp]') and OBJECTPROPERTY(id,  
N'IsUSERTable') = 1)
```

```
drop table [dbo].[TeacherTemp]
```

```
GO
```

```
CREATE TABLE [dbo].[TeacherTemp] (  
[Tno] [char] (6) COLLATE Chinese_PRC_CI_AS NOT NULL ,  
[Tname] [char] (20) COLLATE Chinese_PRC_CI_AS NULL ,  
[Position] [char] (6) COLLATE Chinese_PRC_CI_AS NULL  
) ON [PRIMARY]
```

```
GO
```

```
INSERT INTO TeacherTemp(Tno,Tname) SELECT Tno,Tname FROM  
Teacher
```

```
GO
```

```
if exists (select * from dbo.sysobjects where id =  
object_id(N'[dbo].[Teacher]') and OBJECTPROPERTY(id, N'IsUSERTable')  
= 1)
```

```
drop table [dbo].[Teacher]
```

```
GO
```

```
CREATE TABLE [dbo].[Teacher] (  
[Tno] [char] (6) COLLATE Chinese_PRC_CI_AS NOT NULL ,  
[Tname] [char] (20) COLLATE Chinese_PRC_CI_AS NULL ,  
[Position] [char] (6) COLLATE Chinese_PRC_CI_AS NULL  
) ON [PRIMARY]
```

```
GO
```

```
INSERT INTO Teacher SELECT Tno,Tname FROM TeacherTemp
```

```
GO
```

```
if exists (select * from dbo.sysobjects where id =  
object_id(N'[dbo].[TeacherTemp]') and OBJECTPROPERTY(id,  
N'IsUSERTable') = 1)
```

```
drop table [dbo].[TeacherTemp]
```

GO

10. 以不同的身份使用数据库:

1) 以学生的身份查看成绩;

在查询分析器中以学生身份登录, 进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
SELECT * FROM Grade
```

GO

2) 以教师的身份打印成绩单;

在查询分析器中以教师身份登录, 进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

在“查询分析器”下, 将查询结果的显示模式设置为“文本显示结果”(设置方法见第一部分第 4 章), 运行 6 题第 7) 小题中的成绩打印脚本。

3) 以教务员的身份增加一名教师。

在查询分析器中以教务员身份登录, 进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
INSERT INTO Teacher(Tno,Tname,Tdept)VALUES('032005','李新','信息科学与技术学院')
```

GO