

# 苏州大学实验报告

院、系	计科院	年级专业	软件工程	姓名	高歌	学号	2030416018
课程名称	数据库课程实践					同组实验者	/
指导教师	瞿剑锋	实验日期	2022-04-13	成绩			

实 验 名 称    SQL 语言（试验十一-试验十二）触发器、恢复技术

## 试验十一 触发器

目的：了解触发器的机制及编程设计、使用。

一、建立学生表的触发器 `usp_addstudent`，当增加学生时，SX 系的学生不能超过 30 岁。

1 写出触发器

```
1. CREATE TRIGGER usp_addstudent
2. ON Student
3. FOR INSERT
4. AS
5. BEGIN
6.     DECLARE @dept CHAR(10);
7.     DECLARE @age INT;
8.     SELECT @dept = Sdept, @age = Sage
9.     FROM inserted;
10.    IF @dept = 'SX' AND @age > 30
11.        BEGIN
12.            RAISERROR('SX 系的学生不能超过 30 岁', 16, 1);
13.            ROLLBACK TRANSACTION;
14.        END
15. END
```

2 执行下列语句块：

```
begin tran
    insert into student (sno, sname, ssex, sage, sdept)
        values ( '0701' , ' 刘欢' , ' 男' , 26, ' SX' )
    if @@error=0
        commit
    else
        rollback
end
观察该学生是否加入到 student
```

```
[09:34:53]      Started executing query at Line 21  
                (1 行受到影响)  
                Total execution time: 00:00:00.002
```

成功添加

\*注：上面这段代码存在语法错误。在 Microsoft SQL Server 中，BEGIN TRANSACTION 不以“END”结尾，而是以 COMMIT TRANSACTION 结尾。因此上面这段代码应该改为：

```
1. BEGIN TRANSACTION  
2. INSERT INTO Student  
3.   (Sno, Sname, Ssex, Sage, Sdept)  
4. VALUES  
5.   ('0701', '刘欢', '男', 26, 'SX');  
6. IF @@ERROR <> 0 ROLLBACK TRANSACTION;  
7. COMMIT TRANSACTION
```

下面题目中的代码修改同理，不再过多赘述。

3 执行下列语句块：

```
begin tran  
    insert into student (sno,sname,ssex,sage,sdept) values ('0702','赵欢','男',31,'SX')  
    if @@error=0  
        commit  
    else  
        rollback  
end
```

观察该学生是否加入到 student

```
[09:41:10]      Started executing query at Line 32  
                Msg 50000, Level 16, State 1, Procedure usp_addstudent, Line 12  
                SX系的学生不能超过30岁  
                Msg 3609, Level 16, State 1, Line 2  
                事务在触发器中结束。批处理已中止。  
                Total execution time: 00:00:00
```

事务被触发器中止，添加失败

## 二、实现下列触发器

1 不能删除年龄大于 25 岁的学生记录。

```
1. CREATE TRIGGER usp_delstudent  
2. ON Student  
3. FOR DELETE  
4. AS  
5. BEGIN  
6.   DECLARE @age INT;  
7.   SELECT @age = Sage
```

```

8. FROM deleted;
9. IF @age > 25
10. BEGIN
11. RAISERROR('不能删除年龄大于 25 岁的学生记录', 16, 1);
12. ROLLBACK TRANSACTION;
13. END
14. END

```

- 2 建立触发器 usp\_delcourse，使课程表中 1001, 1002, 1003 三门课不会被删除。  
注意如何调试。

```

1. CREATE TRIGGER usp_delcourse
2. ON Course
3. FOR DELETE
4. AS
5. BEGIN
6. DECLARE @cno CHAR(4);
7. SELECT @cno = Cno
8. FROM deleted;
9. IF @cno IN ('1001', '1002', '1003')
10. BEGIN
11. DECLARE @tmp CHAR(16);
12. SET @tmp = '不能删除课程' + @cno;
13. RAISERROR(@tmp, 16, 1);
14. ROLLBACK TRANSACTION;
15. END
16. END

```

- 3 对学生表建立一触发器，使更改后的年龄只能比原值大

```

1. CREATE TRIGGER usp_updatestudent
2. ON Student
3. FOR UPDATE
4. AS
5. BEGIN
6. DECLARE @age INT;
7. SELECT @age = Sage
8. FROM inserted;
9. IF @age <= (SELECT Sage FROM deleted)
10. BEGIN
11. RAISERROR('年龄必须大于原值', 16, 1);
12. ROLLBACK TRANSACTION;
13. END
14. END

```

- 4 对 sc 表建立触发器，使‘JSJ’系的学生不可选择‘1004’号课程

```

1. CREATE TRIGGER usp_addsc
2. ON SC

```

```

3. FOR INSERT
4. AS
5. BEGIN
6. DECLARE @sno CHAR(6);
7. DECLARE @cno CHAR(4);
8. SELECT @sno = Sno, @cno = Cno
9. FROM inserted;
10. IF @sno = 'JSJ' AND @cno = '1004'
11. BEGIN
12. RAISERROR('JSJ 系的学生不可选择 1004 号课程', 16, 1);
13. ROLLBACK TRANSACTION;
14. END
15. END

```

- 5 对表 course 建触发器，实现级联删除的功能，但某课选修人数大于 3 则不能删除。  
(先删除 sc 表对 course 的外码)

首先删除外码：

```

1. CREATE PROCEDURE del_fk_of_table
2. (@table_name VARCHAR(255))
3. AS
4. DECLARE fk_cursor CURSOR
5. FOR
6. SELECT name
7. FROM sys.foreign_key_columns AS f
8. JOIN sys.objects AS o
9. ON f.constraint_object_id = o.object_id
10. WHERE f.parent_object_id = object_id(@table_name);
11. OPEN fk_cursor;
12. DECLARE @fk VARCHAR(255);
13. FETCH NEXT FROM fk_cursor INTO @fk;
14. WHILE @@FETCH_STATUS = 0
15. BEGIN
16. EXEC ('ALTER TABLE ' + @table_name + ' DROP CONSTRAINT ' + @fk);
17. FETCH NEXT FROM fk_cursor INTO @fk;
18. END
19. CLOSE fk_cursor;
20. DEALLOCATE fk_cursor;
21. GO;
22.
23. del_fk_of_table 'SC'

```

创建触发器：

```

1. CREATE TRIGGER usp_delcourse2
2. ON Course
3. FOR DELETE
4. AS

```

```

5. BEGIN
6. DECLARE @cno CHAR(4);
7. SELECT @cno = Cno
8. FROM deleted;
9. IF (SELECT COUNT(*) FROM SC WHERE Cno = @cno) > 3
10. BEGIN
11. RAISERROR('课程选修人数大于 3，不能删除', 16, 1);
12. ROLLBACK TRANSACTION;
13. END
14. ELSE
15. DELETE FROM SC
16. WHERE Cno = @cno;
17. END

```

### \*三、建立一个触发器，使对 SC 表成绩的修改自动记录修改日志。

日志文件表(tablog)记录如下：

用户名	学号	课程号	原成绩	修改后成绩	更改日期
-----	----	-----	-----	-------	------

首先建立 TABLOG 日志文件表：

```

1. CREATE TABLE TABLOG (
2.   UserName VARCHAR(128) NOT NULL,
3.   Sno CHAR(6) NOT NULL,
4.   Cno CHAR(4) NOT NULL,
5.   OldScore INT NOT NULL,
6.   NewScore INT NOT NULL,
7.   ChangeDate DATETIME NOT NULL
8. )

```

定义触发器：

```

1. CREATE TRIGGER usp_updatesc
2. ON SC
3. FOR UPDATE
4. AS
5. BEGIN
6. DECLARE @sno CHAR(6);
7. DECLARE @cno CHAR(4);
8. DECLARE @oldscore INT;
9. DECLARE @newscore INT;
10. SELECT @oldscore = Grade
11. FROM deleted;
12. SELECT @sno = Sno, @cno = Cno, @newscore = Grade
13. FROM inserted;
14. IF @oldscore <> @newscore
15. BEGIN
16. INSERT INTO TABLOG
17. (UserName, Sno, Cno, OldScore, NewScore, ChangeDate)

```

```

18.      VALUES
19.      (ORIGINAL_LOGIN(), @sno, @cno, @oldscore, @newscore, GETDATE());
20.  END
21. END

```

运行下面这段测试代码：

```

1. UPDATE SC
2. SET Grade = 68
3. WHERE Sno = '0002'
4. AND Cno = '1003';

```

#### MESSAGES

```

[18:14:16]      Started executing query at Line 194
                  (1 行受到影响)
                  (1 行受到影响)
                  Total execution time: 00:00:00.010

```

```

1. SELECT *
2. FROM TABLOG;

```

#### RESULTS

	UserName	Sno	Cno	OldScore	NewScore	ChangeDate
1	sa	0002	1003	67	68	2022-04-22 18:14:16.427

四、在 School 数据库中建立一个试验用的发票表 bill，然后为发票 bill 建立触发器 utr\_money，实现当输入单价和数量后，自动填写金额，即发票金额不输入，由单价、数量相乘后自动填写到金额中。

```

Create table bill(
    billID char(8), --发票编号
    date datetime, --开票日期
    product char(10), --产品编号
    price int , --单价
    qty int , --数量
    charge int , --金额
    primary key (billid) )

```

```

1. CREATE TRIGGER utr_money
2. ON Bill
3. FOR INSERT
4. AS
5. BEGIN
6.     DECLARE @id CHAR(8);
7.     DECLARE @price INT;
8.     DECLARE @qty INT;

```

```
9.  DECLARE @charge INT;
10.  SELECT @id = billID, @price = price, @qty = qty
11.  FROM inserted;
12.  SET @charge = @price * @qty;
13.  UPDATE Bill
14.  SET charge = @charge
15.  WHERE billID = @id;
16. END
```

思考：触发器中 inserted, deleted 表的作用？在触发器中如没有用到此两个表中的任何一个，你认为触发器还有意义吗？

答:inserted 表为插入的记录,对 UPDATE 和 INSERT 语句有效;deleted 表为删除的记录,对 UPDATE 和 DELETE 语句有效。即使某一个触发器没有用到两个表中的任意一个,也可以具有意义,例如创建某个触发器,当某个表更新时记录日志信息(仅记录当前登录名和时间),虽然没有使用到这两个表中的任意一个,但仍然有效。或者也可以设置一个总是取消事务的触发器,用于暂时“冻结”某个表,待之后再删除该触发器。

## 试验十二 恢复技术

- 目的：
- 1 掌握数据库的备份及恢复的方法。
  - 2 了解备份方案的设定

### 一、完全备份的建立与恢复

#### 1 建立完全备份

```
1. USE school;
2. BACKUP DATABASE school
3. TO DISK='C:\schooldata.bak';
```

#### 2 查看备份文件中的信息

```
1. RESTORE FILELISTONLY FROM DISK='C:\schooldata.bak';
2. RESTORE HEADERONLY FROM DISK='C:\schooldata.bak';
```

结果 消息													
	LogicalName	PhysicalName	Type	FileGroupName	Size	MaxSize	FileId	CreateLSN	DropLSN	UniqueId	ReadOnlyLSN	ReadWriteLSN	BackupSizeIn
1	school	C:\DATA\school.mdf	D	PRIMARY	153092096	35184372080640	1	0	0	C4A8C552-9036-4E1C-9C2E-CD0D3071B27C	0	0	3997696
2	school_log	C:\DATA\school_log.ldf	L	NULL	679477248	2199023255552	2	0	0	34871390-4DA7-41A3-8849-45C92B601AFF	0	0	0

	BackupName	BackupDescription	BackupType	ExpirationDate	Compressed	Position	DeviceType	UserName	ServerName	DatabaseName	DatabaseVersion	DatabaseCreationDate	BackupSize	F
1	NULL	NULL	1	NULL	0	1	2	sa	Y7000\SQLEXPRESS	school	904	2022-03-15 10:30:28.000	4253696	5

#### 3 恢复完全备份

##### 1) 先删除数据库 School

```
1. USE Master;
2. DROP DATABASE school;
```

##### 2) 然后恢复.

```
1. RESTORE DATABASE school
2. FROM DISK='C:\schooldata.bak';
```

##### 3) 查看 school 的 Student 中的数据

```
1. SELECT *
2. FROM Student;
```



RESULTS					
	Sno	Sname	Ssex	Sage	Sdept
1	0001	周志林	男	20	SX
2	0002	李文庆	男	23	JSJ
3	0003	陈小明	男	20	SX
4	0004	杨秀红	女	21	JSJ
5	0009	钱明明	男	20	SX
6	0078	王振	男	21	JSJ
7	0081	刘亭	女	22	SX
8	0091	贺秋雪	女	20	SX
9	0092	赵三	男	22	SX
10	8001	张华	男	23	SX
11	8002	赵颖	女	21	SX
12	8003	钱凯	男	22	JSJ
13	8004	王华	男	21	SX
14	8005	张英	女	21	JSJ
15	8006	赵章	女	22	JSJ
16	8007	钱利	男	23	JSJ
17	8008	王铁	男	21	JSJ

## 二、建立差异备份

### 1 建立备份

1) 制作数据文件备份 schoolDiff.bak

```
1. BACKUP DATABASE school
2. TO DISK='C:\schoolDiff.bak';
```

2) 把学号 7001, 姓名: 王海, 性别: 男, 年龄为 23 的学生加入 Student

```
1. INSERT INTO Student
2. (Sno, Sname, Ssex, Sage)
3. VALUES
4. (7001, '王海', '男', 23);
```

3) 制作 school 的差异备份, 存入 schoolDiff.bak

```
1. BACKUP DATABASE school
2. TO DISK='C:\schoolDiff.bak'
3. WITH DIFFERENTIAL;
```

4) 把学号 7002, 姓名: 赵燕, 性别: 女, 年龄为 22 的学生加入 student

```
1. INSERT INTO Student
2. (Sno, Sname, Ssex, Sage)
3. VALUES
4. (7002, '赵燕', '女', 22);
```

5) 制作 school 的差异备份, 存入 schoolDiff.bak

```
1. BACKUP DATABASE school
2. TO DISK='C:\schoolDiff.bak'
3. WITH DIFFERENTIAL;
```

## 2 查看备份文件 schoolDiff.bak 中的信息

1. RESTORE FILELISTONLY FROM DISK='C:\schoolDiff.bak';
2. RESTORE HEADERONLY FROM DISK='C:\schoolDiff.bak';

结果 消息

	LogicalName	PhysicalName	Type	FileGroupName	Size	MaxSize	FileId	CreateLSN	DropLSN	UniqueId	ReadOnlyLSN	ReadWriteLSN	BackupSizeIn
1	school	C:\DATA\school.mdf	D	PRIMARY	153092096	35184372080640	1	0	0	C4A8C552-9036-4E1C-9C2E-CD0D3071B27C	0	0	3997696
2	school_log	C:\DATA\school_log.ldf	L	NULL	679477248	2199023255552	2	0	0	34871390-4DA7-41A3-8849-45C92B601AFF	0	0	0

	BackupName	BackupDescription	BackupType	ExpirationDate	Compressed	Position	DeviceType	UserName	ServerName	DatabaseName	DatabaseVersion	DatabaseCreationDate	BackupSize	F
1	NULL	NULL	1	NULL	0	1	2	sa	Y7000\SQLEXPRESS	school	904	2022-04-22 21:07:41.000	4253696	3
2	NULL	NULL	5	NULL	0	2	2	sa	Y7000\SQLEXPRESS	school	904	2022-04-22 21:07:41.000	579584	3
3	NULL	NULL	5	NULL	0	3	2	sa	Y7000\SQLEXPRESS	school	904	2022-04-22 21:07:41.000	579584	3

## 3 删除 school 数据库

1. USE Master;
2. DROP DATABASE school;

## 4 恢复数据库 school 到第 2 步状态

1. RESTORE DATABASE school
2. FROM DISK='C:\schoolDiff.bak'
3. WITH file=1, NORECOVERY;
4. RESTORE DATABASE school
5. FROM DISK='C:\schoolDiff.bak'
6. WITH file=2;

观察 Student 数据

1. SELECT \*
2. FROM Student;

	Sno	Sname	Ssex	Sage	Sdept
1	0001	周志林	男	20	SX
2	0002	李文庆	男	23	JSJ
3	0003	陈小明	男	20	SX
4	0004	杨秀红	女	21	JSJ
5	0009	钱明明	男	20	SX
6	0078	王振	男	21	JSJ
7	0081	刘亨	女	22	SX
8	0091	贺秋雪	女	20	SX
9	0092	赵三	男	22	SX
10	7001	王海	男	23	JSJ
11	8001	张华	男	23	SX
12	8002	赵颖	女	21	SX
13	8003	钱凯	男	22	JSJ
14	8004	王华	男	21	SX
15	8005	张英	女	21	JSJ
16	8006	赵章	女	22	JSJ
17	8007	钱利	男	23	JSJ
18	8008	王铁	男	21	JSJ

## 5 恢复数据库 school 到最新状态

```
1. RESTORE DATABASE school
2. FROM DISK='C:\schoolDiff.bak'
3. WITH file=1, NORECOVERY;
4. RESTORE DATABASE school
5. FROM DISK='C:\schoolDiff.bak'
6. WITH file=3;
```

观察 Student 数据

```
1. SELECT *
2. FROM Student;
```

结果		消息			
	Sno	Sname	Ssex	Sage	Sdept
1	0001	周志林	男	20	SX
2	0002	李文庆	男	23	JSJ
3	0003	陈小明	男	20	SX
4	0004	杨秀红	女	21	JSJ
5	0009	钱明明	男	20	SX
6	0078	王振	男	21	JSJ
7	0081	刘亨	女	22	SX
8	0091	贺秋雪	女	20	SX
9	0092	赵三	男	22	SX
10	7001	王海	男	23	JSJ
11	7002	赵燕	女	22	JSJ
12	8001	张华	男	23	SX
13	8002	赵颖	女	21	SX
14	8003	钱凯	男	22	JSJ
15	8004	王华	男	21	SX
16	8005	张英	女	21	JSJ
17	8006	赵章	女	22	JSJ
18	8007	钱利	男	23	JSJ
19	8008	王铁	男	21	JSJ

**思考:** 如果仅执行下述恢复语句,能查看 student 的数据吗?

```
1. RESTORE DATABASE school
2. FROM DISK='C:\schoolDiff.bak'
3. WITH file=1, NORECOVERY;
```

指定 NORECOVERY 恢复后数据库不能直接使用:

```
1. SELECT *
2. FROM Student;
```

消息

消息 927, 级别 14, 状态 2, 第 1 行  
无法打开数据库 'school'。该数据库正在进行还原。

完成时间: 2022-04-22T21:33:09.5494082+08:00



5) 制作日志备份存入 schoollog.bak

1. BACKUP LOG school
2. TO DISK='C:\schoollog.bak';

3 观察 schoollog 中的信息

LogicalName	PhysicalName	Type	FileGroupName	Size	MaxSize	FileId	CreateLSN	DropLSN	UniqueId	ReadOnlyLSN	ReadWriteLSN	BackupSizeIn
1 school	C:\DATA\school.mdf	D	PRIMARY	153092096	35184372080640	1	0	0	C4A8C552-9036-4E1C-9C2E-CD0D3071B27C	0	0	0
2 school_log	C:\DATA\school_log.ldf	L	NULL	679477248	2199023255552	2	0	0	34871390-4DA7-41A3-8849-45C92B601AFF	0	0	0

BackupName	BackupDescription	BackupType	ExpirationDate	Compressed	Position	DeviceType	UserName	ServerName	DatabaseName	DatabaseVersion	DatabaseCreationDate	BackupSize
1 NULL	NULL	2	NULL	0	1	2	sa	Y7000\SQLEXPRESS	school	904	2022-04-22 21:36:09.000	116736

4 删除 school 数据库

1. USE Master;
2. DROP DATABASE school;

5 利用 schooldata1.bak 及 schoollog.bak 恢复数据库 school 到最新状态

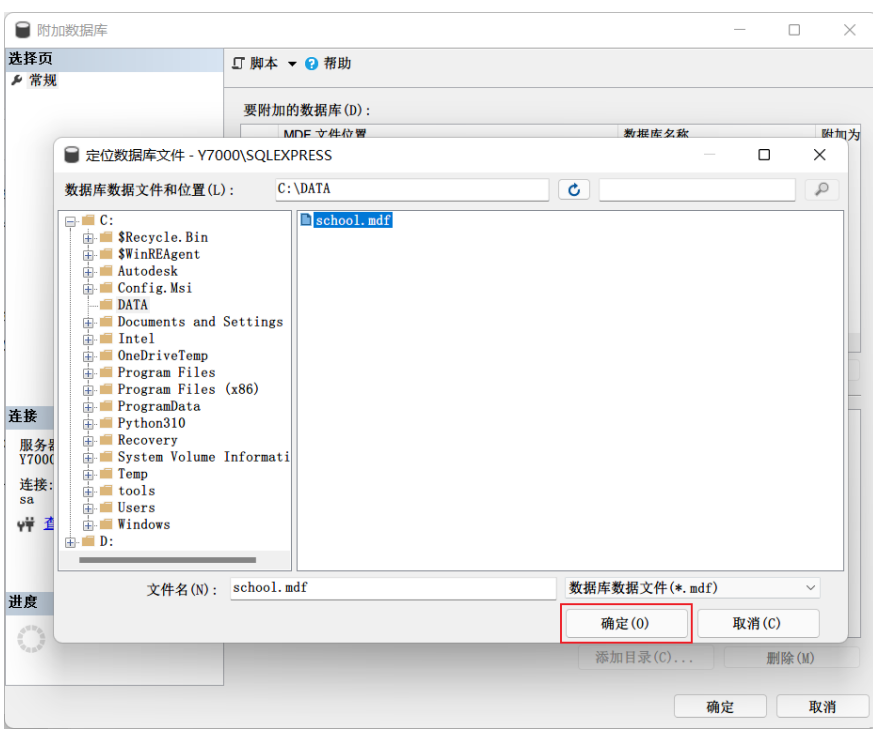
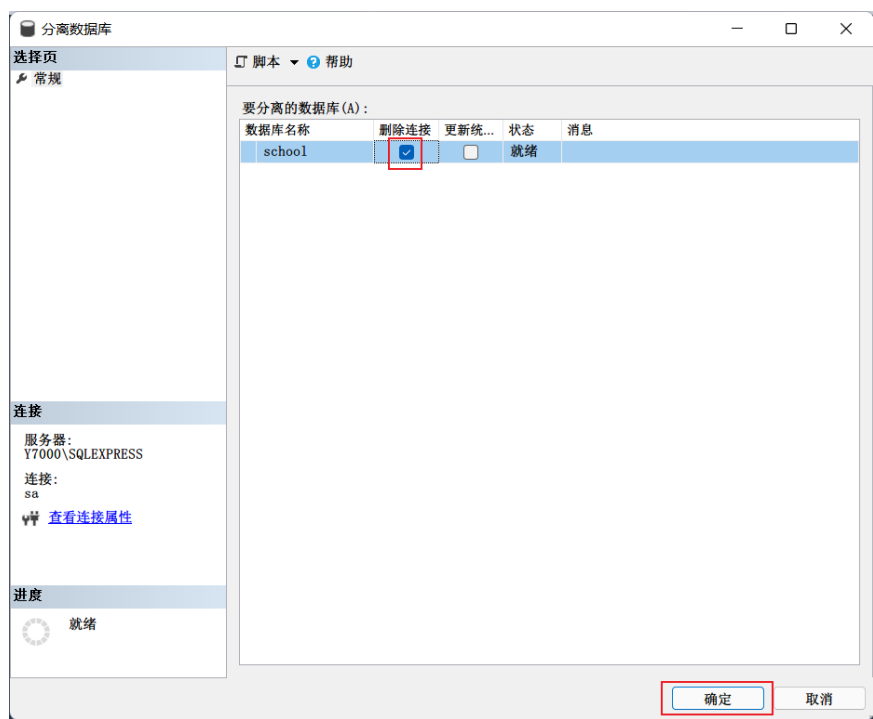
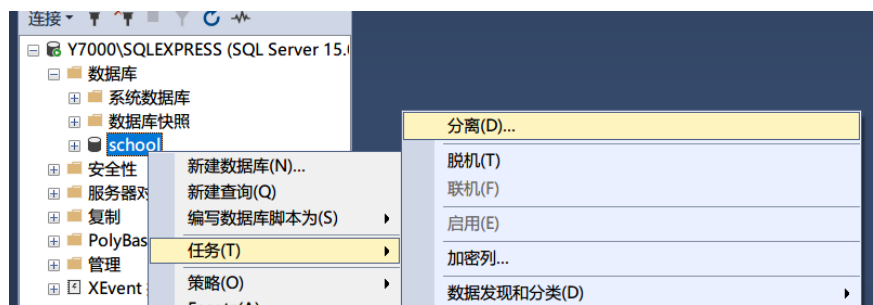
1. RESTORE DATABASE school
2. FROM DISK='C:\schooldata1.bak'
3. WITH NORECOVERY;
4. RESTORE LOG school
5. FROM DISK='C:\schoollog.bak';

观察 Student 数据

1. SELECT \*
2. FROM Student;

	Sno	Sname	Ssex	Sage	Sdept
1	0001	周志林	男	20	SX
2	0002	李文庆	男	23	JSJ
3	0003	陈小明	男	20	SX
4	0004	杨秀红	女	21	JSJ
5	0009	钱明明	男	20	SX
6	0078	王振	男	21	JSJ
7	0081	刘亭	女	22	SX
8	0091	贺秋雪	女	20	SX
9	0092	赵三	男	22	SX
10	7001	王海	男	23	JSJ
11	7002	赵燕	女	22	JSJ
12	7003	王江	男	23	JSJ
13	8001	张华	男	23	SX
14	8002	赵颖	女	21	SX
15	8003	钱凯	男	22	JSJ
16	8004	王华	男	21	SX
17	8005	张英	女	21	JSJ
18	8006	赵章	女	22	JSJ
19	8007	钱利	男	23	JSJ
20	8008	王铁	男	21	JSJ

四、使用企业管理器练习数据库的分离及附加



## 五、如何清除日志文件。

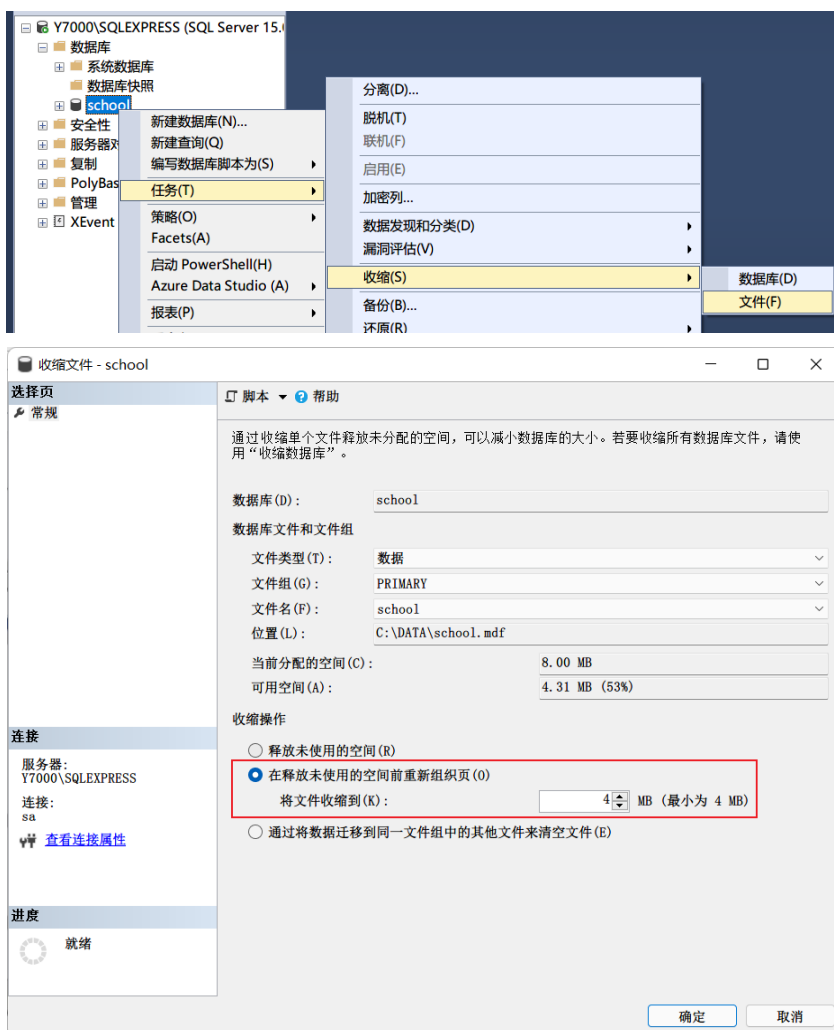


(上述步骤需要将数据库恢复模式设为“简单”)

或者运行下面的 T-SQL 语句：

1. **ALTER DATABASE** school **SET** RECOVERY SIMPLE **WITH** NO\_WAIT;
2. **ALTER DATABASE** school **SET** RECOVERY SIMPLE;
3. USE school;
4. DBCC SHRINKFILE (N'school\_Log', 11, TRUNCATEONLY);
5. USE Master;
6. **ALTER DATABASE** school **SET** RECOVERY FULL **WITH** NO\_WAIT;
7. **ALTER DATABASE** school **SET** RECOVERY FULL;

## 六、使用企业管理器练习数据库的压缩



## 七、把 school 备份到其他计算机上。

将备份后的文件复制到其他计算机即可。

1. BACKUP DATABASE school
2. TO DISK='C:\schooldata.bak';

然后运行恢复语句：

1. RESTORE DATABASE school
2. FROM DISK='C:\schooldata.bak';

[加上如果没有 norecovery 发生的情况]

恢复时加上 NORECOVERY 关键字的目的是防止在恢复过程中数据库被修改, 导致因为不一致无法进行后续恢复。差异备份与日志备份恢复时都要求除最后一条恢复语句外前面数条恢复语句都必须加上 NORECOVERY 关键字, 否则将无法执行命令。

[企业管理器恢复如何实现 norecovery ]

