一. 实验目的

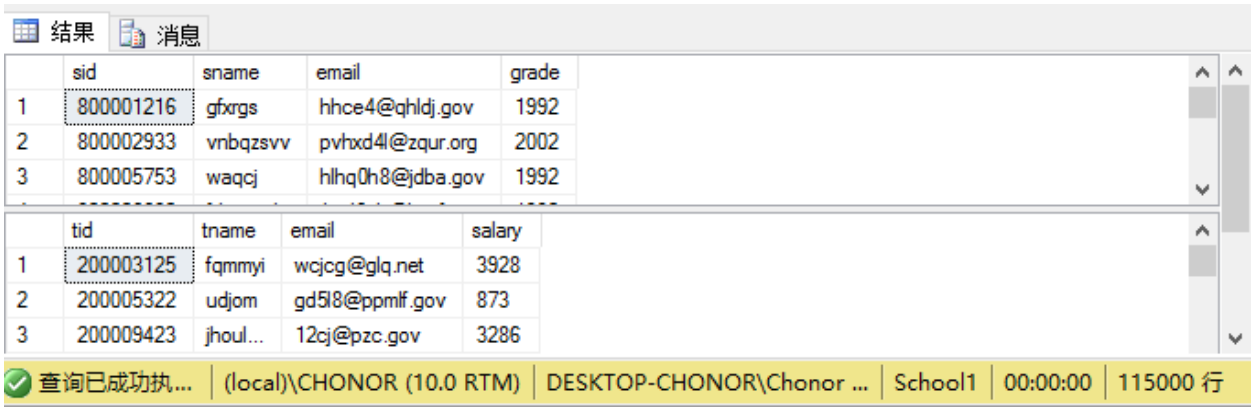
- 熟悉SQL Server的事务控制语言，能够熟练使用事务控制语言来编写事务处理程序。

二. 实验结果

1. **编写一个嵌套事务。外层修改students表某记录，内层在teachers表插入一条记录。演示内层插入操作失败后，外层修改操作回滚**

   我们先查看一下原来的数据

```
SELECT * FROM STUDENTS ORDER BY SID
SELECT * FROM TEACHERS ORDER BY TID
```



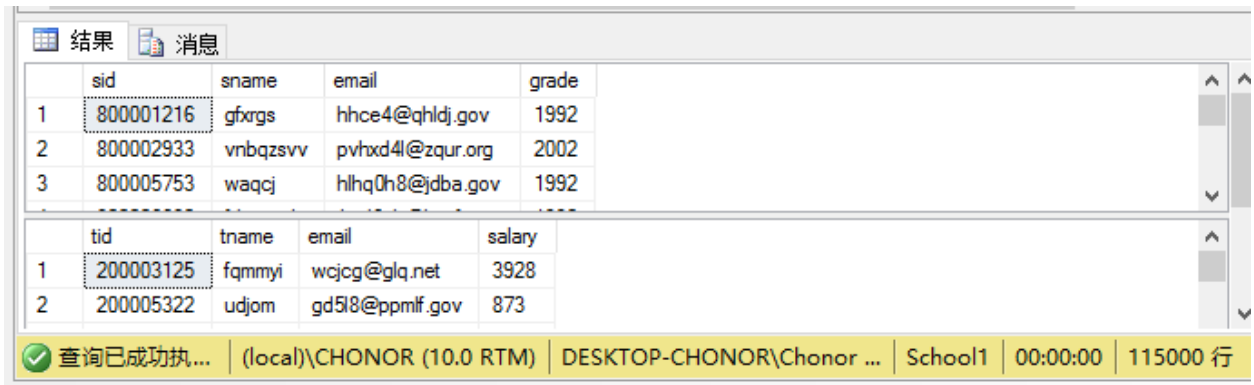
之后我们更新第一条学生数据，然后插入个第一条教师数据相同tid的数据，此时因为tid不能重复所以会插入失败。

```
SET XACT_ABORT ON
BEGIN TRAN up_stu
UPDATE  STUDENTS
SET grade='1992'
WHERE sid='800001216'
    BEGIN TRAN insert_tea
        INSERT INTO TEACHERS
        VALUES('200003125','test','xxx@xxx','9999')
    COMMIT TRAN insert_tea
COMMIT TRAN up_stu
```

结果如下：

此时我们再次查询

```
SELECT * FROM STUDENTS ORDER BY SID
SELECT * FROM TEACHERS ORDER BY TID
```

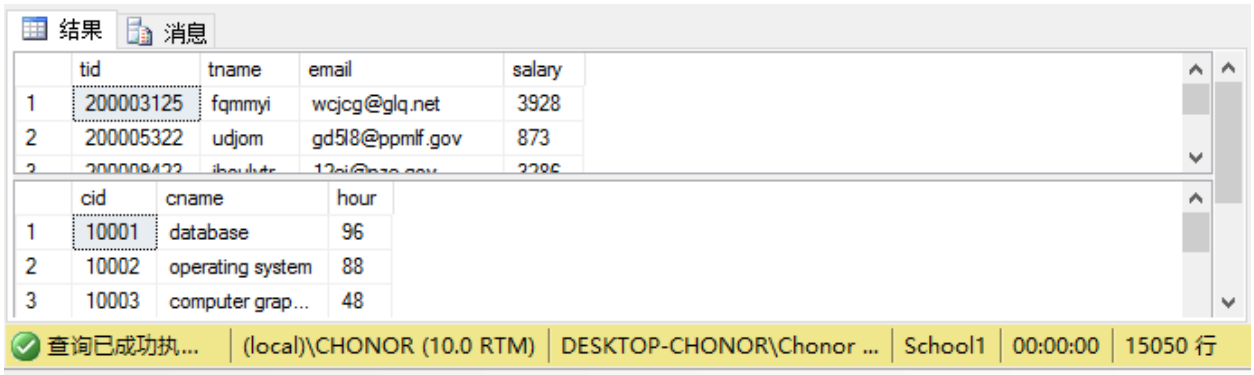


发现外层事务在成功执行之后因为内层事务的执行失败而回滚，符合要求。

2. **编写一个带有保存点的事务。更新teachers表中数据后，设置事务保存点，然后在表courses中插入数据，如果courses插入数据失败，则回滚到事务保存点。演示courses插入失败，但teachers表更新成功的操作**

```
SELECT * FROM TEACHERS ORDER BY TID
SELECT * FROM COURSES ORDER BY CID
```

查询结果如下：



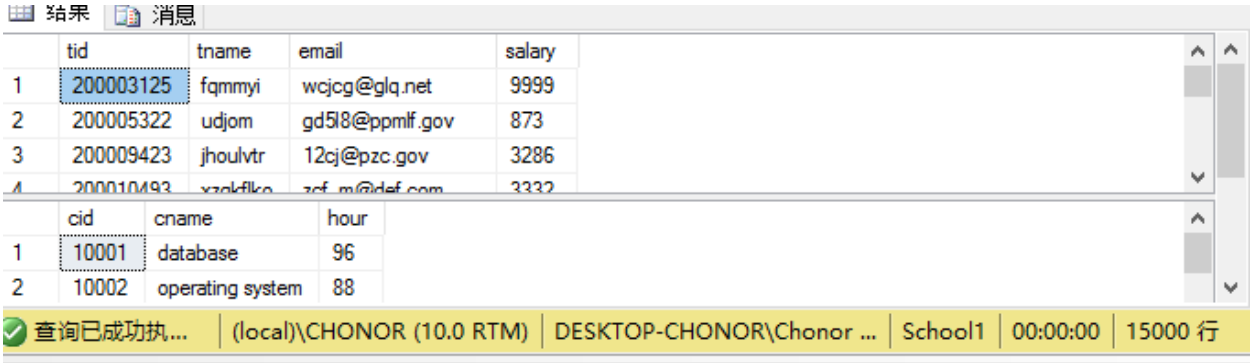
我们修改tid为200003125的教师工资为9999，然后插入一条cid为10001的课程

```
SET XACT_ABORT OFF
BEGIN TRAN up_tea
```

```
    UPDATE  TEACHERS
    SET salary='9999'
    WHERE tid='200003125'
    SAVE TRAN up_tea_done
    INSERT INTO COURSES
    VALUES ('10001','test','999')
    IF @@ERROR!=0
    BEGIN
        ROLLBACK TRAN up_tea_done
        PRINT'插入课程表失败'
        RETURN
    END
COMMIT TRAN up_tea
```

此时再次查询

```
SELECT * FROM TEACHERS ORDER BY TID
SELECT * FROM COURSES ORDER BY CID
```



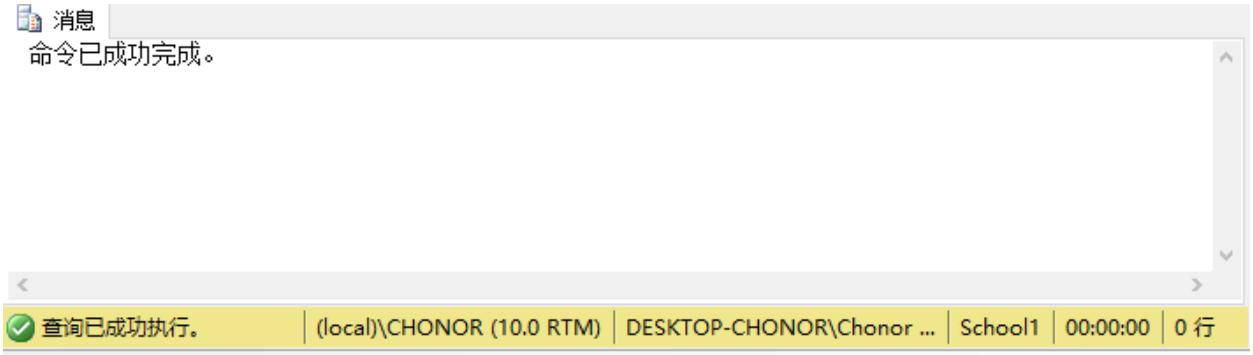
此时教师工资更新成功没有回滚，而课程表没有变化。

3. **编写一个包含事务的存储过程，用于更新courses表的课时。如果更新记录的cid不存在，则输出“课程信息不存在”，其他错误输出“修改课时失败”，如果执行成功，则输出“课时修改成功”。调用该存储过程，演示更新成功与更新失败的操作。**

```
CREATE PROCEDURE UPDATECOURSEINFO
    @cid_ char(10),
    @hour_ int,
    @returnStr varchar(100) out
AS
BEGIN TRAN
```

```
    IF NOT EXISTS(SELECT CID FROM COURSES WHERE CID=@cid_)
        BEGIN
            SELECT @returnStr='课程信息不存在'
            GOTO ONERROR
        END
    UPDATE COURSES
    SET hour=@hour_
    WHERE cid=@cid_
    IF @@ERROR<>0
        BEGIN
            SELECT @returnStr='修改课时失败'
            GOTO ONERROR
        END
    SELECT @returnStr='修改课时成功'
    PRINT @returnStr
    COMMIT TRAN
ONERROR:
    PRINT @returnStr
    ROLLBACK TRAN
GO
```

结果如下：



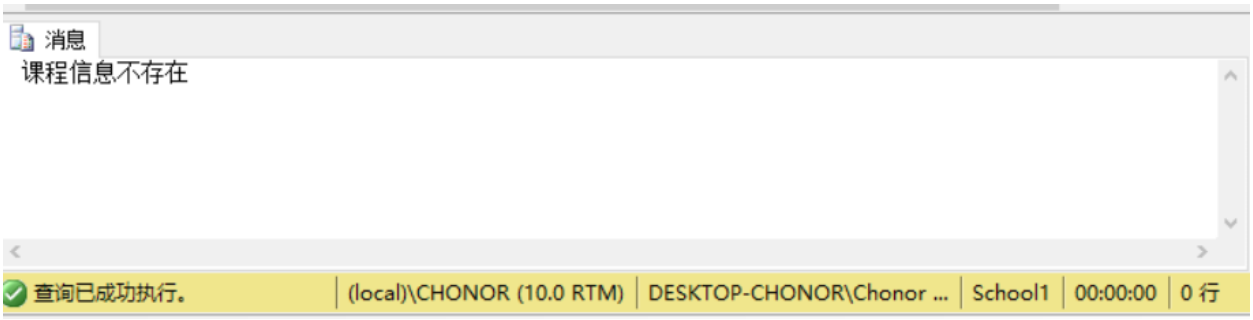
演示更新成功

```
declare @cid_ char(10)
declare @hour_ int
declare @returnStr varchar(100)
exec UPDATECOURSEINFO '10001',100,@returnStr out
SELECT * FROM COURSES ORDER BY CID
```

	cid	cname	hour
1	10001	database	100
2	10002	operating system	88
3	10003	computer graphics	48
4	10004	java	48
5	10005	c++	60

更新失败的操作，此时修改一个不存在的课程

```
declare @cid_ char(10)
declare @hour_ int
declare @returnStr varchar(100)
exec UPDATECOURSEINFO '11001',100,@returnStr out
```

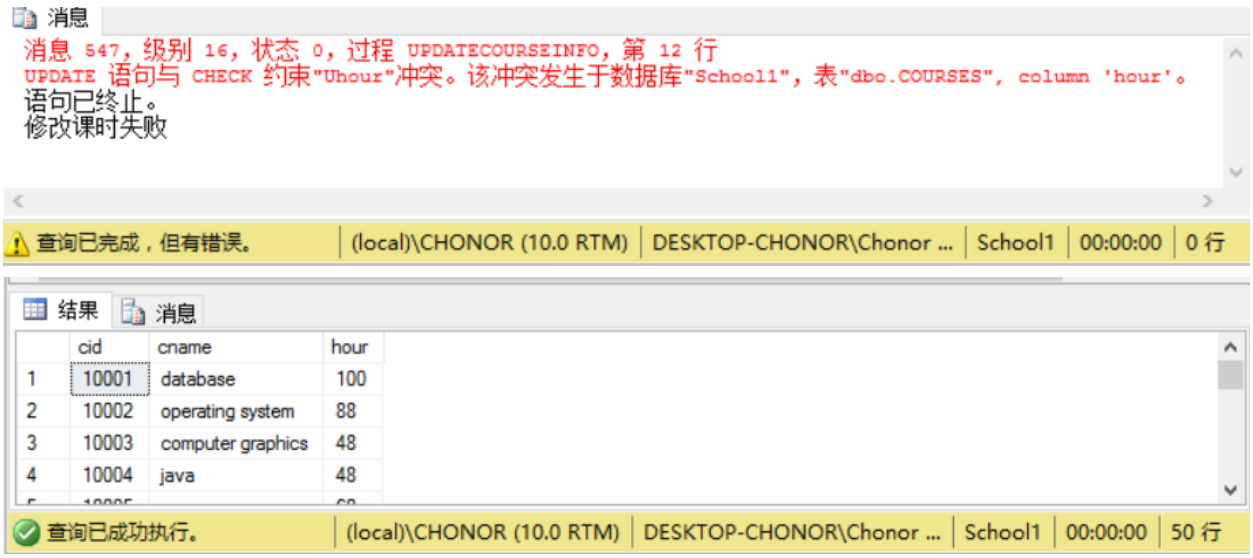


演示修改课时失败的操作2，先加入一个约束要求课时大于0,之后进行更新hour=0

```
ALTER TABLE COURSES ADD CONSTRAINT Uhour CHECK (hour>0)

declare @cid_ char(10)
declare @hour_ int
declare @returnStr varchar(100)
exec UPDATECOURSEINFO '10001',0,@returnStr out
```

查询结果如下



此时更新失败

三. 实验感想

这次实验是在学习事务的使用方式，之前实验中也有事务的内容，这次实验多增加了事务的嵌套，回滚，保存点和事务的存储过程，第三个任务中想要演示修改失败，结果发现原数据表中并没有对课时进行约束，也就是怎么修改都能成功，只要id存在，所以多加了一个约束进行修改失败的演示。