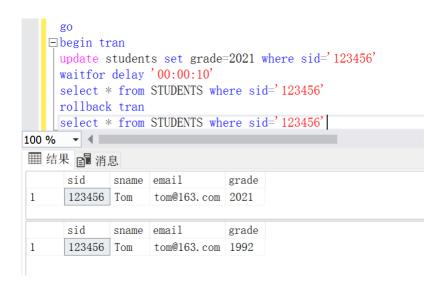
# 19335262 张航悦 实验 14

### 练习一

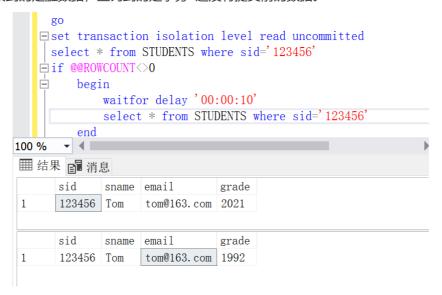
设置"未提交读"隔离级别(READ UNCOMMITTED),在students表上演示读"脏"数据。

- 步骤
  - 。 新建事务1, 在其中更新grade, 延时20秒后, 事务回滚至初始状态, 如代码1所示。
  - 。 在事务1执行过程中,执行查询2。查询2为查询grade,延时20秒后,再次查询grade。
- 代码
  - o 代码1



#### ○ 代码2

第一次读到的是脏数据,查询到的是事务1还没有提交前的数据。



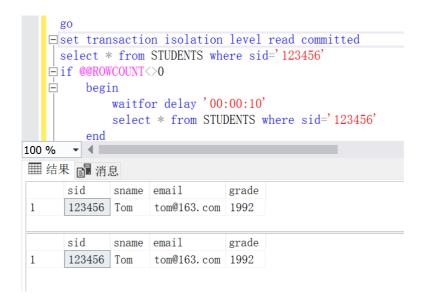
### 练习二

设置"提交读"隔离级别(READ COMMITTED),在students表上演示避免读"脏"数据。

• 步骤

将查询2的隔离级别改为"READ COMMITTED"

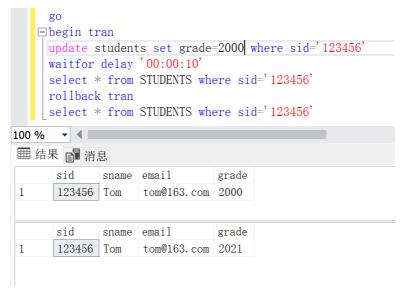
• 代码3



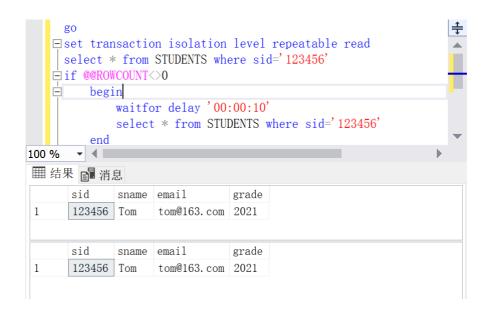
## 练习三

设置"可重复读"隔离级别(REPEATABLE READ),在students表上演示避免读"脏"数据、不可重复读,但不能避免幻象读。

- 演示避免读脏数据
  - o 代码4: 在事务中更新grade

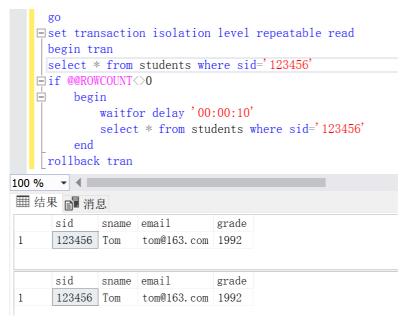


。 代码5: 查询到的结果是事务提交后的结果



#### • 演示避免不可重复读

。 代码6:在事务中设置隔离级别为"隔离级别",在事务中查询grade,延时10秒后,再次相同查询。

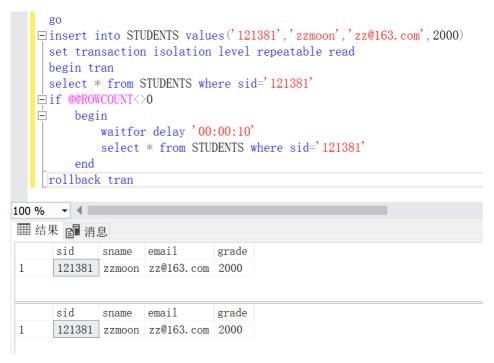


。 代码7: 新建查询,在执行事务过程中,执行代码更新grade

```
set transaction isolation level repeatable read update STUDENTS set grade=2021 where sid='123456'
```

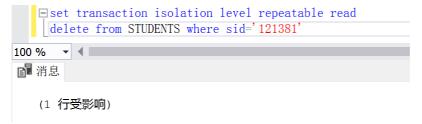
#### • 演示不可避免幻读

。 代码8:插入一条新数据,执行查询students表,延迟10秒后再次相同查询。



。 代码9: 在另一查询中删除刚刚新插入的记录

在上图中可以看到该记录已被删除,但还能查询到,这就是幻视。



## 练习四

设置 "可串行化"隔离级别(SERIALIZABLE),在students表上演示防止其他用户在事务提交之前更新数据。

步骤
在执行代码10两次查询的过程中,执行代码11插入数据。

- 代码
  - 。 代码10

o 代码11

```
□ set transaction isolation level serializable insert into STUDENTS values ('121381', 'zzmoon', 'zz@163.com', 2000)
```