上机实验报告

课程名：_____ 指导教师：_____

姓名		班级		实验得分	
		学号			
实验时间		实验地点			
实验主题	事务与并发控制				
实验目的	（1）掌握事务机制，学会创建事务。 （2）理解事务并发操作所可能导致的数据不一致性问题：丢失修改、读脏数据、不可重复读以及幻读现象。 （3）理解锁类型、锁机制、锁的相容性，理解锁协议，学会采用锁实现不同级别的事务隔离，相应解决各种数据不一致的问题。				

实验内容	假设在 Study 数据库中，除了学生表 S、课程表 C，选课表 SC 以外，还有一个银行卡表 BANK，银行卡间的转账流水表 TranFlowIn、TranFlowOut。三表的定义如下：
	（1）BankCard(BID，Sno，Money)//银行卡号，学号，银行卡金额
	（2）TranFlowIn(TFINID，InID，FromID，TranMoney，Time)//转账流水号 ID，转入银行卡号，来源银行卡号，入账时间。TFINID 为 32 位流水号，自动生成。
	（3）TranFlowOut(TFOUTID，OutID，ToID，TranMoney，Time)//转账流水号 ID，转出银行卡号，转入银行卡号，出账时间。TFID 为 32 位流水号，自动生成。
	（1）编写一个事务处理(begin tran)实现如下的操作：某学号为“2017300647”的学生要从银行卡“B20200506001”中转账 20.21 元到学号“2020302206”的银行卡“B20210506004”中。每一笔转账，均分别在 TranFlowIn 和 TranFlowOut 中各记录 1 个转账流水（一个位出账流水，一个位入账流水）。若中间出现故障则进行 rollback。
	提示：①SQL 代码已经完成了出账。
	②此时事务发生故障，导致事务尚未结束，结果不能提交，回滚。
	③SQL 代码执行过程中，一步一步执行，观察变量值（可在事务中设置延时等待）。				

	（2）针对本实验的数据库和表，在并发条件下，构造出现三种数据不一致的情况： ①丢失修改、②读脏数据、③不可重复读。提示： ①构建两个事务，两个事务并发执行。②SQL 代码执行过程中，一步一步执行，观察变量值（可在事务中设置延时等待）。 （3）针对本实验的数据库表中的某一数据，进行读锁和写锁的相容性验证。 （4）利用锁机制、实施不同的封锁协议，分别解决上述丢失修改、读脏数据和不可重复读的数据不一致问题。 提示：①构建两个事务 T1、T2，并发执行，并且操作相同的数据。 ②T1 先操作本数据库表中的某一数据 D，加锁。 ③接着 T2 事务操作同一数据 D，需要等待 T1 解锁。 ④接着 T1 事务解锁数据 D。 ⑤然后 T2 才可以操作数据 D。 ⑥SQL 代码执行过程中，一步一步执行，观察变量值（可在事务中设置延时等待）。 （5）构造一个出现死锁的情形。 提示：①构建两个事务 T1、T2，并发执行，并且同时操作本数据库表中两个数据 D1、D2。 ②T1 先操作本数据库表中某一数据 D1，加锁。 ③同时 T2 事务操作同一数据 D2，加锁。 ④接着 T1 要操作本数据库表中的数据某一 D2，但 T2 尚未解锁 D2，因此 T1 等待。 ⑤此时 T2 要操作数据 D1，但 T1 尚未解锁数据 D1，因此 T2 等待。 ⑥这样事务 T1 与 T2 就形成了死锁。
实验结果 / 实验结论	建表和函数默认值

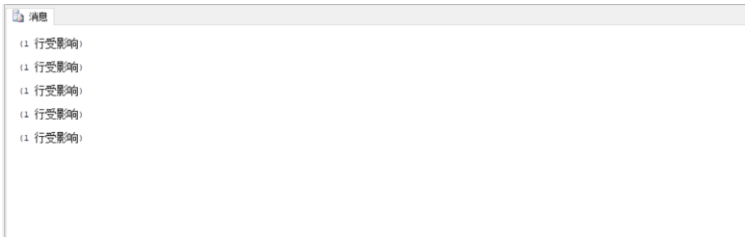
```
create Function pad()
returns char(32)
as
begin
declare @i char(32);
declare @count int;
SELECT @i=datediff(ss,'1970-01-01',GETDATE());
select @count=(30-len(@i));
return (select 'OD'+REPLICATE('0',@count)+@i);
end


create Function pad_1()
returns char(32)
as
begin
declare @i char(32);
declare @count int;
SELECT @i=datediff(ss,'1970-01-01',GETDATE());
select @count=(30-len(@i));
return (select 'ID'+REPLICATE('0',@count)+@i);
end

create table BANK(
     BID char(32),
     Sno char(32),
     MoneyNum money
);

create table TranFlowIn(
     InID char(32),
     FromID char(32),
     TranMoney money,
     Getime TIME,
     TFINID char(32) DEFAULT DBO.pad()

);
create table TranFlowOut(
     InID char(32),
     FromID char(32),
     TranMoney money,
     Getime TIME,
     TFOUTID char(32) DEFAULT DBO.pad_1(),
);
```

：某学号为“2017300647”的学生要从银行卡“B20200506001”中转账 20.21 元到学号“2020302206”的银行卡“B20210506004”中。每一笔转账，均分别在 TranFlowIn 和 TranFlowOut 中各记录 1 个转账流水（一个位出账流水，一个位入账流水）。若中间出现故障则进行 rollback。

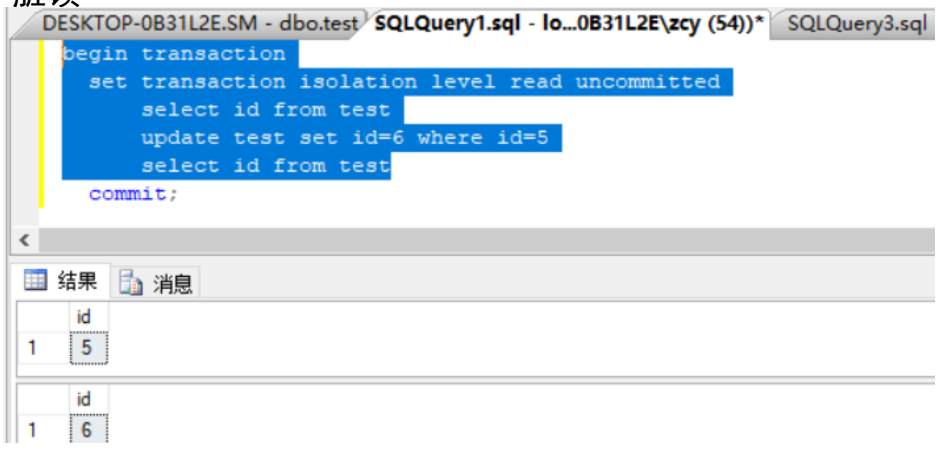
```
begin tran TranStart
insert into TranFlowIn(InID,FromID,TranMoney,Getime) values('B20210506004','B20200506001',20.21,GETDATE());
if @@ERROR<>0
    rollback tran TranStart
else
begin
    insert into TranFlowOut(InID,FromID,TranMoney,Getime) values('B20210506004','B20200506001',20.21,GETDATE());
    if @@ERROR<>0
    begin
        rollback tran TranStart
    end
    else
    begin
        update BANK set MoneyNUm = MoneyNUm-20.21 where sno='2017300647'
        if @@ERROR<>0
        begin
            rollback tran TranStart
        end
        else
        begin
            update BANK set MoneyNUm = MoneyNUm-20.21 where sno='2017300647'
            if @@ERROR<>0
            begin
                rollback tran TranStart
            end
            else
            begin
                update BANK set MoneyNUm = MoneyNUm+20.21 where sno='2020302206'
                if @@ERROR<>0
                begin
                    rollback tran TranStart
                end
                else
```

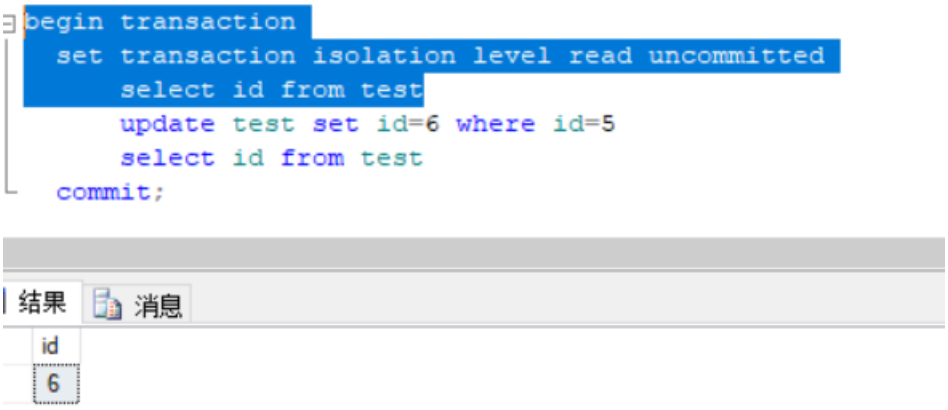
```
            begin
                commit tran TranStart
            end
        end
    end
end
```

若是结果正确，则插入，失败则会回滚



脏读





不可重复读

```
begin transaction
  set transaction isolation level read uncommitted
      select id from test
      update test set id=6 where id=5
      select id from test
  commit;
```

DESKTOP-0B31L2E.SM - dbo.test | SQLQuery1.sql - lo...0B31L2E\zcy (54))* | SQLQuery3.sq

```
begin transaction
  set transaction isolation level read uncommitted
      select id from test
      select id from test
  commit;
```

结果 消息

	id
1	5

	id
1	6

幻读

```
  begin transaction
set transaction isolation level read uncommitted
    select id from test
    insert into test values(1)
    select id from test
commit;
```

结果 消息

id
6

```
  begin transaction
set transaction isolation level read uncommitted
    select id from test
    insert into test values(1)
    select id from test
commit;
```

结果 消息

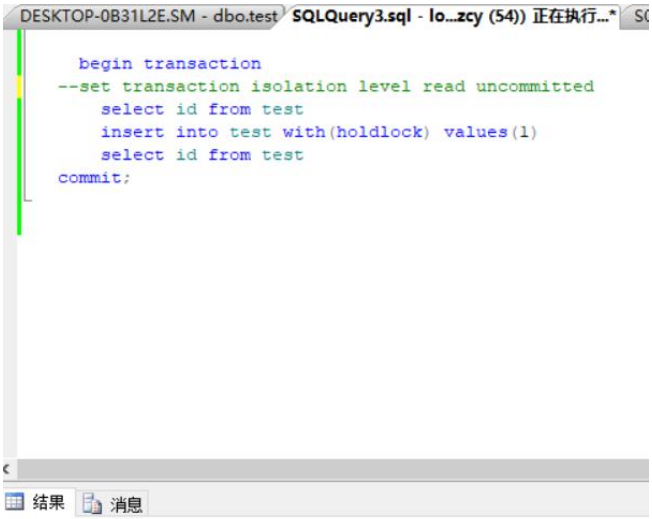
	id
1	6

```
    insert into test values(1)
    select id from test
commit;
```

结果 消息

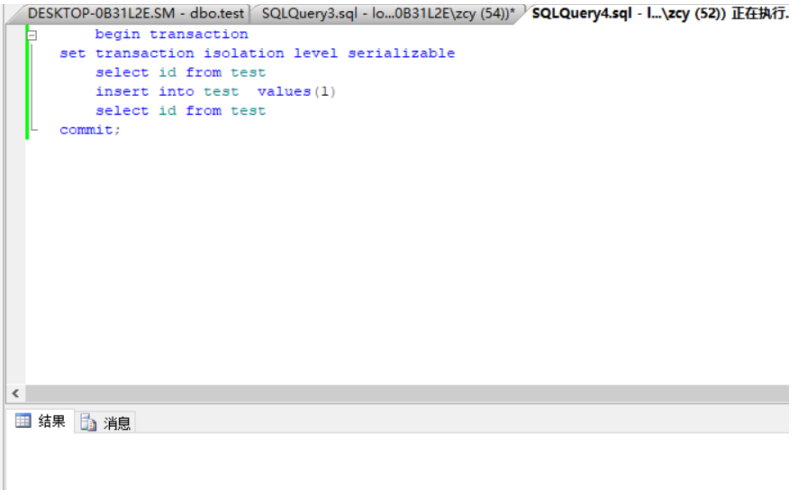
id
6
1
1

加锁避免幻读，脏读等现象

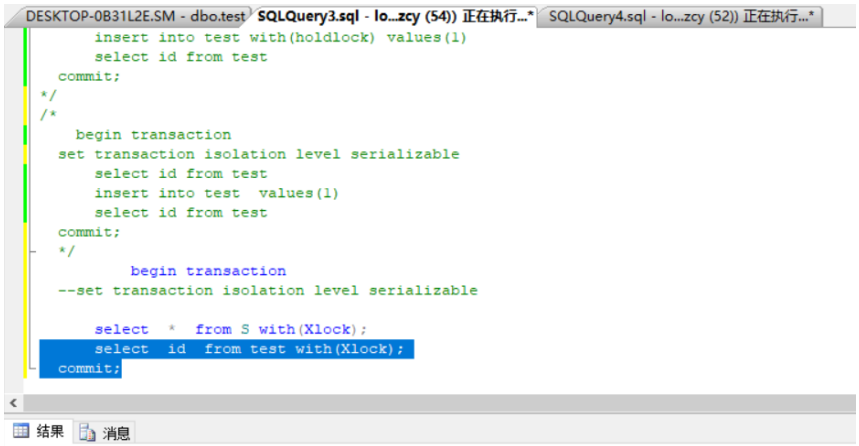


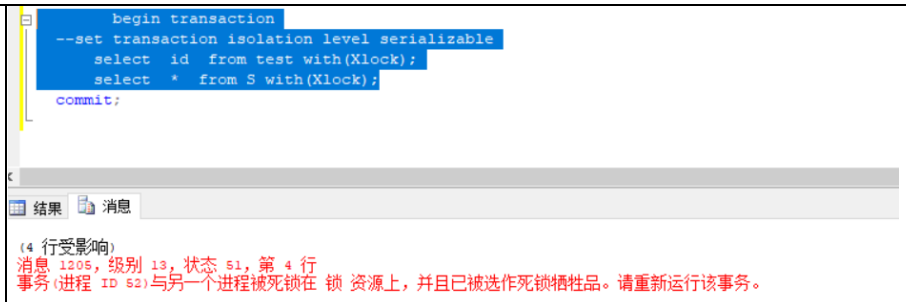
会卡在正在执行，直到最后 commit

加隔离避免此类情况



构造死锁



	
实验心得	通过本次实验，我对于 mssql 的强大功能有了进一步的认知，学会了然后处理并发下的数据完整性，懂得了加锁和设置隔离级别的方法，又懂得了该如何避免死锁。