

# 数据库系统实验十三

18340013 陈琮昊

## 一、实验目的：

理解事务并发中不一致的问题，以及通过设置隔离级别解决不一致问题。

## 二、实验题目：

以下练习均在school数据库上进行。

- 1.设置“未提交读”隔离级别( `READ UNCOMMITTED` ), 在students表上演示读“脏”数据。
- 2.设置“提交读”隔离级别( `READ COMMITTED` ), 在students表上演示避免读“脏”数据。
- 3.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。
- 4.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。

## 三、实验过程与结果：

注, 对于 STUDENTS 表内sid='800001216'的数据原来如下 (后面用来做对比) :

2	800001216	gfxrgs	hhce4@qhldj.gov	1992
---	-----------	--------	-----------------	------

1. SQL 代码如下:

```
begin tran
  update STUDENTS set grade=1993 where sid='800001216'
  waitfor delay '00:00:20'
  select * from STUDENTS where sid='800001216'
rollback tran
  select * from STUDENTS where sid='800001216'
```

```
set transaction isolation level read uncommitted
select * from STUDENTS where sid='800001216'
if @@ROWCOUNT<>0
  begin
    waitfor delay '00:00:20'
    select * from STUDENTS where sid='800001216'
  end
```

按 PPT 所给执行方式执行后得到的结果如下：

	sid	sname	email	grade
1	800001216	gfixrgs	hhce4@qhldj.gov	1993

	sid	sname	email	grade
1	800001216	gfixrgs	hhce4@qhldj.gov	1992

可以看到读“脏”数据，出现了不一致。

2. SQL 代码如下：

```
begin tran
    update STUDENTS set grade=1993 where sid='800001216'
    waitfor delay '00:00:20'
    select * from STUDENTS where sid='800001216'
rollback tran
    select * from STUDENTS where sid='800001216'
```

```
set transaction isolation level read committed
select * from STUDENTS where sid='800001216'
if @@ROWCOUNT<>0
    begin
        waitfor delay '00:00:20'
        select * from STUDENTS where sid='800001216'
    end
end
```

结果如下：

	sid	sname	email	grade
1	800001216	gfixrgs	hhce4@qhldj.gov	1992

  

	sid	sname	email	grade
1	800001216	gfixrgs	hhce4@qhldj.gov	1992

可以看到并没有读到脏数据。

按照 PPT 所给演示，这里加一步提交读隔离级别，避免脏读，允许不可重复读的示例：

```

set transaction isolation level read committed
begin tran
select * from STUDENTS where sid='800001216'
if @@ROWCOUNT<>0
    begin
        waitfor delay '00:00:20'
        select * from STUDENTS where sid='800001216'
    end
end
rollback tran

```

```

set transaction isolation level read committed
update STUDENTS set grade=1999 where sid='800001216'

```

结果如下:

	sid	sname	email	grade
1	800001216	gfxrgs	hhce4@qhldj.gov	1993

  

	sid	sname	email	grade
1	800001216	gfxrgs	hhce4@qhldj.gov	1999

可以看到结果不一致，因为提交读允许其他事务在两次读取间隔内修改。

3. SQL 代码如下:

```

set transaction isolation level repeatable read
begin tran
select * from STUDENTS where sid='800001216'
if @@ROWCOUNT<>0
    begin
        waitfor delay '00:00:20'
        select * from STUDENTS where sid='800001216'
    end
end
rollback tran

```

```

set transaction isolation level repeatable read
update STUDENTS set grade=1995 where sid='800001216'

```

结果如下:

	sid	sname	email	grade
1	800001216	gfxrgs	hhce4@qhldj.gov	1999

	sid	sname	email	grade
1	800001216	gfxrgs	hhce4@qhldj.gov	1999

可以看到保证了读一致，接下来演示幻象读：

SQL 代码如下：

```
insert into STUDENTS values('300000000','fgf','fgdfg@126.com',1980)
set transaction isolation level repeatable read
begin tran
select * from STUDENTS where sid='300000000'
if @@ROWCOUNT<>0
begin
waitfor delay '00:00:10'
select * from STUDENTS where sid='300000000'
end
rollback tran
```

```
set transaction isolation level repeatable read
delete from STUDENTS where sid='300000000'
```

结果如下：

	sid	sname	email	grade
1	300000000	fgf	fgdfg@126.com	1980

	sid	sname	email	grade
1	300000000	fgf	fgdfg@126.com	1980

可以看到sid='300000000'的数据其实已经被删除，但还是显示了出来，出现了幻象读。

4. SQL 代码如下：

```
set transaction isolation level serializable
begin tran
select * from STUDENTS where sid='300000000'
waitfor delay '00:00:10'
select * from STUDENTS where sid='300000000'
rollback tran
```

```
set transaction isolation level serializable
insert into STUDENTS values('300000000','fgf','fgdfg@126.com',1980)
```

结果如下：

sid	sname	email	grade
-----	-------	-------	-------

sid	sname	email	grade
-----	-------	-------	-------

可以看到执行顺序是串行的，因为第一个事务在执行时并没有让插入数据的那个事务进来，因此查询为空。

## 四、实验体会：

本次实验比较简单，把 PPT 所给示例看懂，注意一些细节再做起来就很容易；不过学到了很多。通过这次实验，我了解到了一些关于事务并发出现不一致性的问题，通过设置事务隔离级别来解决这种不一致性，让我对于事务一致性这一部分内容有了更深刻的理解。