



## 中山大学数据科学与计算机学院

### 移动信息工程专业-数据库系统

### 本科生实验报告

(2017-2018 学年秋季学期)

课程名称: 数据库系统实验

教学班级	15M1	专业(方向)	移动互联网
学号	15352408	姓名	张镓伟

## 一、实验目的

1.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、实验内容

事务并发不一致问题:

- 读“脏”数据: 一个事务读取另一个事务尚未提交的数据引起。
- 不可重复读: 事务 T1 读取数据 a 后, 事务 T2 对数据 a 进行更新, 事务 T1 再次读取, 无法读取前一次的结果。
- 幻象读: 事务 T1 两次查询过程中, 事务 T2 对数据进行插入或删除, 导致事务 T1 两次查询的记录数不一致。

事务隔离级别:

- READ UNCOMMITTED(未提交读, 读脏)
- READ COMMITTED(已提交读, 不读脏, 但允许不重复读, SQL 默认级别)
- REPEATABLE READ(可重复读, 禁止读脏和不重复读, 但允许幻象读)
- SERIALIZABLE(可串行化, 最高级别, 事务不能并发, 只能串行)

## 三、实验过程及结果

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

**Step1:** 新建事务 1, 实现在事务 1 中更新 grade, 延时 20 秒后, 事务回滚至初始状态, 如代码 1 所示。

**Step2:** 在事务 1 执行过程中, 执行查询 2, 查询 2 位查询事务 1 修改的那个学生的 grade, 延时 20 秒, 再次查询该 grade, 比较两次查询的结果。如代码 2 所示。

代码 1:



```
SQLQuery2.sql - (\\...\\UTE\\freedom (55))* SQLQuery1.sql - (\\...\\UTE\\freedom (52))*
use school
go
begin tran
update STUDENTS set grade=1990 where sid='800001216'
waitfor delay '00:00:20' -- 延时20秒
select * from STUDENTS where sid='800001216'
rollback tran
select * from STUDENTS where sid='800001216'
```

sid	sname	email	grade
800001216	gfixgs	hhce4@qhldj.gov	1990

事务执行过程中, grade为1990

sid	sname	email	grade
800001216	gfixgs	hhce4@qhldj.gov	1992

事务回滚后, grade为1992

## 代码 2:

```
use School
go
set transaction isolation level read uncommitted -- 未提交读
--模拟实现脏读
select * from students where sid='800001216'
if @@ROWCOUNT<>0
begin
    waitfor delay '00:00:20'
    -- PRINT 模拟实现不可重复读
    select * from STUDENTS where sid='800001216'
end
```

sid	sname	email	grade
800001216	gfixgs	hhce4@qhldj.gov	1990

读脏数据, 查询的是事务1没提交前的数据

sid	sname	email	grade
800001216	gfixgs	hhce4@qhldj.gov	1992

延时20秒后再次查询, 与第一次查询结果不一样, 发生了不可重复读

如图, 代码 2 第一次查询得到的是脏数据, 后面再次查询就不一样了, 原因是事务 1 更新数据过程与查询 2 的执行过程没有隔离开来。

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

将代码 2 的隔离级别设置成“READ COMMITTED”, 重复执行 (1) 中过程。

```
use School
go
set transaction isolation level read committed -- 事务隔离级别为“提交读”
--模拟实现脏读
select * from students where sid='800001216'
if @@ROWCOUNT<>0
begin
    waitfor delay '00:00:20'
    -- PRINT 模拟实现不可重复读
    select * from STUDENTS where sid='800001216'
end
```

sid	sname	email	grade
800001216	gfixgs	hhce4@qhldj.gov	1992

sid	sname	email	grade
800001216	gfixgs	hhce4@qhldj.gov	1992

由图可以看出, 两次查询结果是一样的, 都是事务 1 提交后的值, 避免了读脏数据。



(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

**Step1:** 新建代码 3, 在代码 3 中先插入一条新纪录, 在查询该记录, 延时 20 秒, 再次查询。

**Step2:** 在执行代码 3 的过程中, 执行代码 4, 代码 4 为删除代码 3 插入的记录。

代码 3:

```
use School
insert into STUDENTS values('800000000','abc','aa@aa.com',1990)
set transaction isolation level repeatable read 可重复读
begin tran
select * from students where sid='800000000'
if @@ROWCOUNT<>0
begin
waitfor delay '00:00:10'
select * from STUDENTS where sid='800000000'
end
rollback tran
```

sid	sname	email	grade
800000000	abc	aa@aa.com	1990

sid	sname	email	grade
800000000	abc	aa@aa.com	1990

代码 4:

```
set transaction isolation level repeatable read
delete from STUDENTS where sid='800000000'
```

可以发现, 代码 3 的两次查询结果相同, 但事实上该记录已经被删除, 出现了“幻想读”现象。

(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。

在代码 5 中执行两次查询, 间隔 10 秒, 在执行代码 5 的过程中, 执行代码 6, 代码 6 为插入一条新数据, 代码 5 的两次查询都是查询这条新数据。

代码 5:

```
SQLQuery2.sql - (\\...UTE\\freedom (55))* SQLQuery1.sql - (\\...UTE\\freedom (52))*
use School
set transaction isolation level serializable 可串行化
begin tran
select * from students where sid='800000000'
waitfor delay '00:00:10'
select * from STUDENTS where sid='800000000'
rollback tran
```

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

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

代码 6:

```
set transaction isolation level serializable
insert into STUDENTS values('800000000','abc','aa@aa.com',1990)
```

可以看到, 事务执行顺序是完全串行的, 事务 5 在执行中防止了代码 6 向其插入数



据，所以查询为空。

## 四、 实验感想

这次实验，我们学习解决了事务并发不一致的问题，其实归根到底就是跟事务的过程的并发执行顺序有关，由此也产生了解决这个问题的 4 个级别的事物隔离级别（如实验内容中所述）。通过这次实验，我也了解到了如何灵活运用这 4 个隔离级别来解决不同的事物并发不一致问题。