

中山大学数据科学与计算机学院本科生实验报告

(2020学年秋季学期)

课程名称：数据库系统 任课教师：刘玉葆

年级&班级	18级 计科一班	专业(方向)	计科 (大数据与人工智能方向)
学号	18340014	姓名	陈嘉宁
电话	18475934419	Email	734311072@qq.com
开始日期	2020/12/25	完成日期	2020/12/25

一、实验题目

事务并发的隔离级别

二、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。

三、实验内容

1. 事务并发不一致问题：
 - 读“脏”数据：一个事务T2读取另一个事务T1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。
2. 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)

四、实验过程

1. 设置“未提交读”隔离级别 (READ UNCOMMITTED)，在students表上演示读“脏”数据：
首先在一个查询框中输入事务T1的代码，在其中进行更新和回滚：

```

1  -- S1
2  begin tran T1
3      update STUDENTS set sname='cbkyyds' where sid='60000001'
4      waitfor delay '00:00:20'
5      select * from STUDENTS where sid='60000001'
6  rollback tran T1
7  select * from STUDENTS where sid='60000001'

```

接着在另一个查询框中进行与事务T1中相同的查询，隔离级别设置为read uncommitted：

```

1  -- S2
2  set transaction isolation level read uncommitted
3  select * from STUDENTS where sid='60000001'
4  if @@ROWCOUNT <> 0 begin
5      waitfor delay '00:00:20'
6      select * from STUDENTS where sid='60000001'
7  end

```

依次运行上述查询，则在S1中会首先将相应的sname更新为'cbkyyds'，之后S1和S2均读取到该更新；在S1进行rollback后，S1和S2又会读取到回滚后的数据。结果如下：

The screenshot displays two SQL query windows. The top window, titled 'SQLQuery2.sql - (\\...KLS675B\\dell (56))', contains a transaction T1 that updates the student name to 'cbkyyds', waits for 20 seconds, and then rolls back. The bottom window, titled 'SQLQuery1.sql - (\\...KLS675B\\dell (53))', contains a transaction that sets the isolation level to read uncommitted, selects the student record, waits for 20 seconds, and then selects the record again. Both windows show the results of their queries.

sid	sname	email	grade
1	60000001	cbkyyds	00020212121
1	60000001	waqcj	00020212121

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

在这一步中，仅需要将S2的隔离级别修改为read committed即可。运行结果如下：

The screenshot displays a SQL query window titled 'SQLQuery2.sql - (\\...KLS675B\\dell (56))' with the isolation level set to read committed. The query selects the student record, waits for 20 seconds, and then selects the record again. The results show the student name as 'waqcj'.

sid	sname	email	grade
1	60000001	waqcj	00020212121
1	60000001	waqcj	00020212121

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:

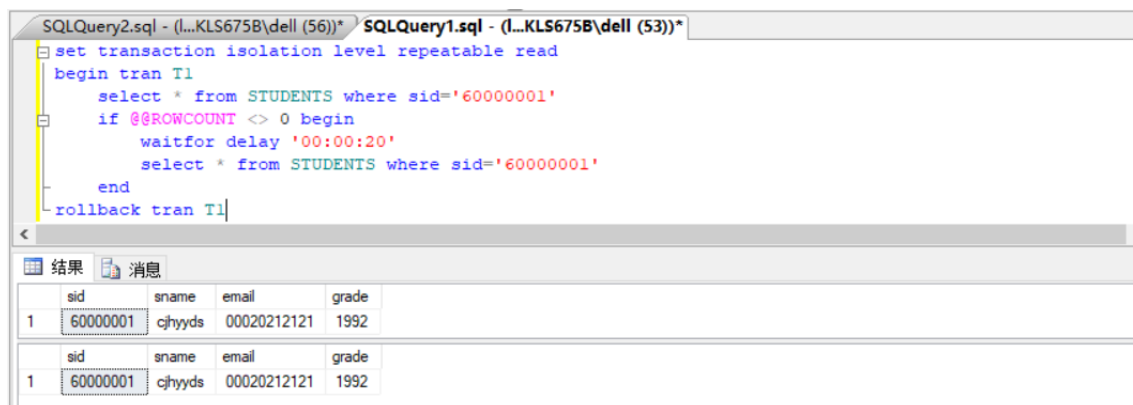
首先在一个查询框中输入事务T1的代码, 在其中进行数据的读取, 隔离级别设置为repeatable read:

```
1  -- S1
2  set transaction isolation level repeatable read
3  begin tran T1
4      select * from STUDENTS where sid='60000001'
5      if @@ROWCOUNT <> 0 begin
6          waitfor delay '00:00:20'
7          select * from STUDENTS where sid='60000001'
8      end
9  rollback tran T1
```

接着在另一个查询框中进行数据表项的删除:

```
1  -- S2
2  set transaction isolation level read committed
3  begin tran T2
4  delete from STUDENTS where sid='60000001'
5  rollback tran T2
```

在S1中的运行结果为:



sid	sname
60000001	cjhyds

可以发现, 读取到了在S2中被删除的数据, 即出现了“幻象读”。

将S2的查询语句修改为:

```
1  -- S2
2  set transaction isolation level read committed
3  begin tran T2
4  update STUDENTS set sname='lxdyds' where sid='60000001'
5  rollback tran T2
```

S1的执行结果变为:

SQLQuery2.sql - (\\...KLS675B\\dell (56))*SQLQuery1.sql - (\\...KLS675B\\dell (53))*

```
set transaction isolation level repeatable read
begin tran T1
    select * from STUDENTS where sid='60000001'
    if @@ROWCOUNT <> 0 begin
        waitfor delay '00:00:20'
        select * from STUDENTS where sid='60000001'
    end
rollback tran T1
```

<

结果消息

	sid	sname	email	grade
1	60000001	cjhyds	00020212121	1992

	sid	sname	email	grade
1	60000001	cjhyds	00020212121	1992

说明repeatable read情况下可以避免不可重复读。

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:

首先在一个查询框中输入事务T1的代码, 在其中进行数据的读取, 隔离级别设置为serializable:

```

1  -- S1
2  set transaction isolation level serializable
3  begin tran T1
4      select * from STUDENTS where sid='60000001'
5      if @@ROWCOUNT = 0 begin
6          waitfor delay '00:00:20'
7          select * from STUDENTS where sid='60000001'
8      end
9  rollback tran T1

```

接着在另一个查询框中进行数据表项的插入:

```

1  -- S2
2  set transaction isolation level read committed
3  begin tran T2
4  insert into STUDENTS values('60000002','czxyds','czx@163.com',2000)
5  rollback tran T2

```

S1的执行结果为:

SQLQuery2.sql - (\\...KLS675B\\dell (56))*

SQLQuery1.sql - (\\...KLS675B\\dell (53))*

```
set transaction isolation level SERIALIZABLE
begin tran T1
    select * from STUDENTS where sid='60000002'
    if @@ROWCOUNT = 0 begin
        waitfor delay '00:00:20'
        select * from STUDENTS where sid='60000002'
    end
rollback tran T1
```

结果

消息

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

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

五、实验感想

本次实验较为简单且工作量较小，跟随PPT的指导可以较快完成。同时本实验有效地强化了课堂上学习的关于事务隔离级别的知识。