数据库实验lab12

曹广杰

15352015 数据科学与计算机

授课教师：刘玉葆

Content

数据库实验lab12
Content
实验目的
句法汇总
实验内容
未提交读模式
提交读模式
可重复读隔离模式
幻象读
可串行化隔离模式
附录

实验目的

1. 理解事务并发中不一致的问题
2. 通过设置隔离级别解决不一致问题

句法汇总

1. 为事务设置模式
2. 未提交模式
3. 提交模式
4. 可重复读模式与幻象读
5. 可串行化模式

实验内容

未提交读模式

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

题意分析：

所谓脏数据就是，在一个事务中进行修改但是尚未提交的数据。

为了演示读取到脏数据，需要编写读取数据的事务，这里笔者选择自己插入的数据进行修改（`values('1000', tom, NULL, NULL)`）：

```
set transaction isolation level read uncommitted
    select *
    from students
    where sid = '1000'

    if @@rowcount <> 0
        begin
            waitfor delay '00:00:10'
            select *
            from students
            where sid = '1000'
        end
```

此处设置每隔10s读取一次数据，一共读取两次，这样我们可以在其等待的时间段中使用另一个事务修改数据库，并且该事务不能提交，应该回滚：

```
set transaction isolation level read uncommitted
begin tran
    update students
    set grade = '68'
    where sid = '1000'
rollback tran
```

此时查询到的结果是两次不同的数据，第一次查询的时候查询的结果是`NULL`，第二次查询的结果是`68`。因为第一次查询的时候，整个事务是允许其他的事务插入进来修改数据库信息的，所以可以查询到脏数据。

提交读模式

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

题意分析：

提交读模式中允许其他事务的插入，但是只有当插入的事务是真正提交到数据库之后才可以修改数据库的信息，对于回滚的事务，是不能够影响到我们查询的数据的。

所以先编写读取的事务：

```
set transaction isolation level read committed
    select *
    from students
    where sid = '1000'

    if @@rowcount <> 0
        begin
            waitfor delay '00:00:10'
            select *
            from students
            where sid = '1000'
        end
```

运行的时候也是先运行该事务，待到该事务在等待的10s之中，运行另一个修改数据库的事务，修改数据库的事务需要是正常运行并且可以提交的。

```
set transaction isolation level read committed
begin tran
    update students
    set grade = '68'
    where sid = '1000'
commit tran
```

此时两次可以查询到不同的数据，每一次的数据都是数据库中真实的数据。表示在这个事务运行的时候是允许其他事务插入，共同运行的。

可重复读隔离模式

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。

题意分析：

可重复读的模式并不允许其他事务的写入，所以两次运行查询的结果是相同的。

读取事务的编写：

```
set transaction isolation level read repeatable read
begin tran
    select *
    from students
    where sid = '1000'

    if @@rowcount <> 0
        begin
            waitfor delay '00:00:10'
            select *
            from students
            where sid = '1000'
        end
commit tran
```

先运行该事务，在此事务运行等待的时候，其他事务——无论是使用insert、update还是delete操作都不能插入运行，不允许打断，所以两次查询到的信息都是一样的。

但是，该模式下可能会出现幻象读取的情况。

幻象读

幻象读是指在事务A运行的时候，不允许另一个事务B写数据，但是允许B读数据。此时二者得到的数据都是一样的，但是A对数据后续进行修改之后，B之前读取的数据就已经失效了。B的第一次就是幻象读。

综上，可重复读取模式下的事务是不允许其他事务对其正在使用的数据进行写入操作的，但是允许其他事务读取。

可串行化隔离模式

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。

依然是需要编写两个事务，其中一个事务用于读：

```
set transaction isolation level serializable
begin tran
    select *
    from students
    where sid = '1001'

    if @@rowcount <> 0
        begin
            waitfor delay '00:00:10'
            select *
            from students
            where sid = '1001'
        end
commit tran
```

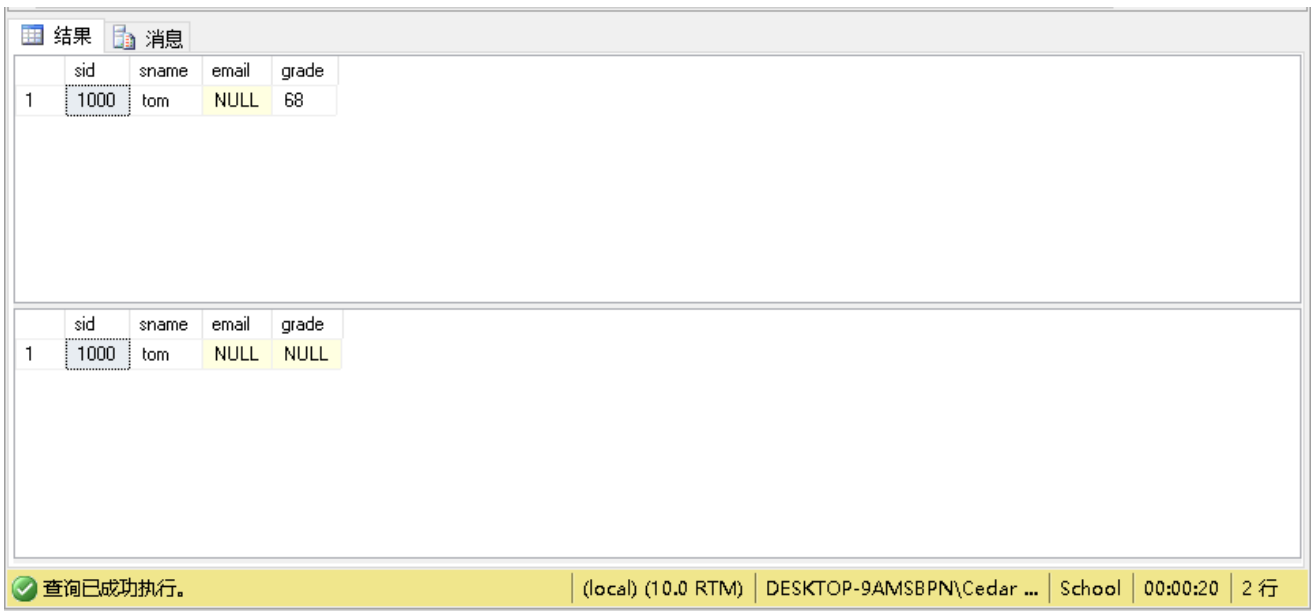
第一次读取与第二次读取中等待10s，在等待过程中运行另一个写的事务：

```
set transaction isolation level serializable
insert into students
values('1001', 'tommy', NULL, NULL);
```

这里笔者在写的事务中选择的操作是插入——`insert`其实什么操作都没有意义，因为在串行化中，对于事务A中使用到的数据信息，都使用的是X锁，互斥。其他的事务既不能写也不能读。

附录

实验结果展示：



未提交读模式

结果 消息

	sid	sname	email	grade
1	1000	tom	NULL	NULL

	sid	sname	email	grade
1	1000	tom	NULL	NULL

查询已成功执行。 (local) (10.0 RTM) DESKTOP-9AMSBPN\Cedar ... School 00:00:18 2 行

提交读模式

结果 消息

	sid	sname	email	grade
1	1000	tom	NULL	NULL

	sid	sname	email	grade
1	1000	tom	NULL	NULL

查询已成功执行。 (local) (10.0 RTM) DESKTOP-9AMSBPN\Cedar ... School 00:00:10 1 行

可重复读取模式