《数据库系统实验》

实验报告

题目	（实验 12）
姓名	TRY
学号	
班级	计科 X 班

一. 实验环境：

操作系统：windows

图形界面：mysql3.7.31, mysql workbench

二. 实验内容与完成情况：

本次实验在以前建立的教学管理系统（jxgl）的基础上完成。

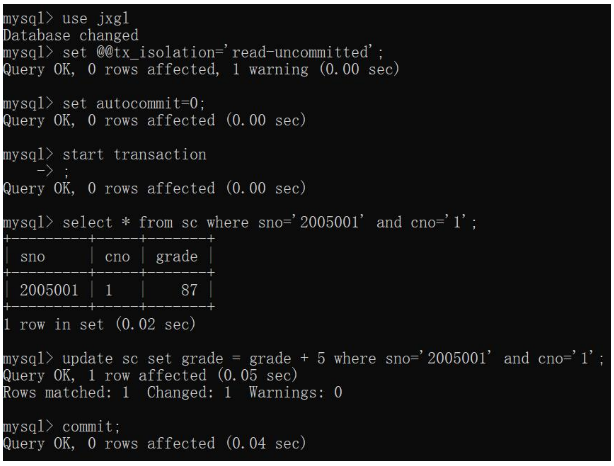
2.1 表 12-7 InnoDB 存储引擎不可重复读

代码：（按执行顺序显示）

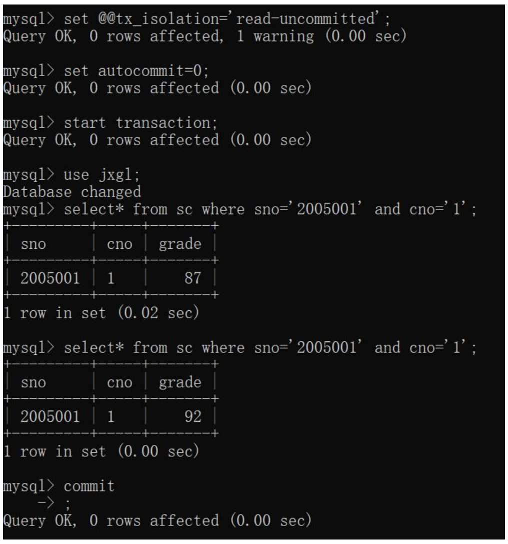
```
session_1：
    use jxgl
    set @@tx_isolation='read-uncommitted';
    set autocommit=0;
    start transaction;
    select * from sc where sno='2005001' and cno='1';

session_2：
    use jxgl
    set @@tx_isolation='read-uncommitted';
    set autocommit=0;
    start transaction;
    select * from sc where sno='2005001' and cno='1';

session_1：
    update sc set grade = grade + 5 where sno='2005001' and cno='1';
    commit;

session_2:
    select* from sc where sno='2005001' and cno='1';
    commit;
```

运行结果：

session_1 & session_2：



```
mysql> use jxgl
Database changed
mysql> set @@tx_isolation='read-uncommitted';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction
    -> ;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from sc where sno='2005001' and cno='1';
+---------+-----+-------+
| sno     | cno | grade |
+---------+-----+-------+
| 2005001 | 1   |    87 |
+---------+-----+-------+
1 row in set (0.02 sec)

mysql> update sc set grade = grade + 5 where sno='2005001' and cno='1';
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> commit;
Query OK, 0 rows affected (0.04 sec)
```



```
mysql> set @@tx_isolation='read-uncommitted';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> use jxgl;
Database changed
mysql> select* from sc where sno='2005001' and cno='1';
+---------+-----+-------+
| sno     | cno | grade |
+---------+-----+-------+
| 2005001 | 1   |    87 |
+---------+-----+-------+
1 row in set (0.02 sec)

mysql> select* from sc where sno='2005001' and cno='1';
+---------+-----+-------+
| sno     | cno | grade |
+---------+-----+-------+
| 2005001 | 1   |    92 |
+---------+-----+-------+
1 row in set (0.00 sec)

mysql> commit
    -> ;
Query OK, 0 rows affected (0.00 sec)
```

分析：由于这两个事务是并发执行的，且两个事务在访问数据、修改数据的时候都没有加共享锁和排它锁，因此事务二 session_2 的第二次访问 select 操作会读到事务一 session_1 已经提交的数据，破坏了数据库的一致性要求，两次读出来的数据是不一样的。

2.2 表 12-8 InnoDB 存储引擎避免不可重复读

代码：（按执行顺序显示）

session_2：

```
    use jxgl
    set @@tx_isolation='read-uncommitted';
    set autocommit=0;
    start transaction;
    select * from sc where sno='2005001' and cno='1' lock in share mode;
```

session_1：

```
    use jxgl
    set @@tx_isolation='read-uncommitted';
    set autocommit=0;
    start transaction;
    select * from sc where sno='2005001' and cno='1' for update;
```

session_2:

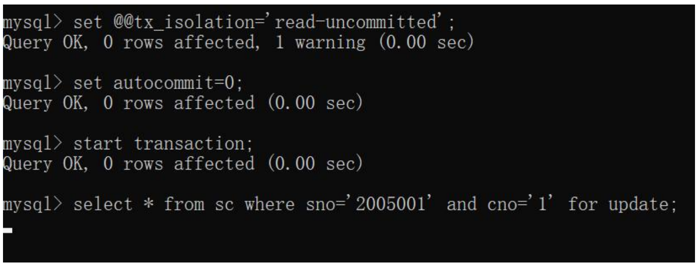
```
    select* from sc where sno='2005001' and cno='1' lock in share mode;
    commit;
```

session_1：

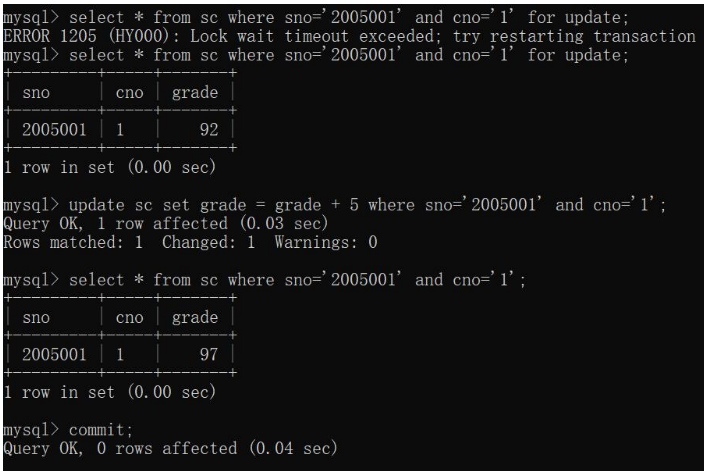
```
    update sc set grade = grade + 5 where sno='2005001' and cno='1';
    select * from sc where sno='2005001' and cno='1';
    commit;
```

运行结果：

session_1：

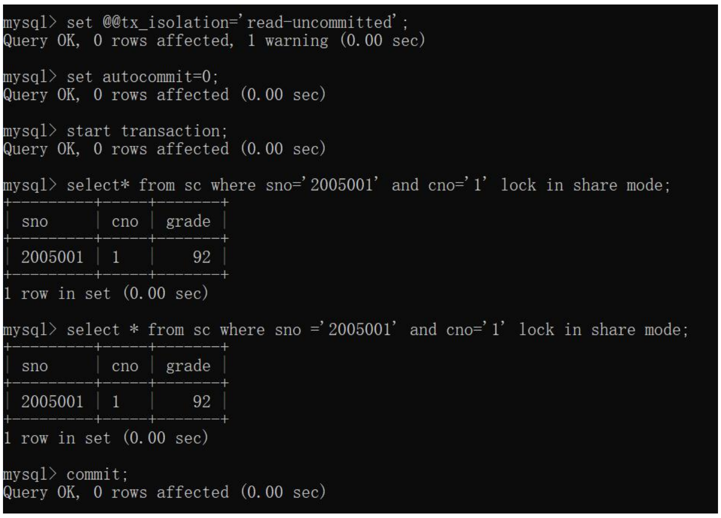


```
mysql> set @@tx_isolation='read-uncommitted';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from sc where sno='2005001' and cno='1' for update;
```



```
mysql> select * from sc where sno='2005001' and cno='1' for update;
ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting transaction
mysql> select * from sc where sno='2005001' and cno='1' for update;
+---------+-----+-------+
| sno     | cno | grade |
+---------+-----+-------+
| 2005001 | 1   |    92 |
+---------+-----+-------+
1 row in set (0.00 sec)

mysql> update sc set grade = grade + 5 where sno='2005001' and cno='1';
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from sc where sno='2005001' and cno='1';
+---------+-----+-------+
| sno     | cno | grade |
+---------+-----+-------+
| 2005001 | 1   |    97 |
+---------+-----+-------+
1 row in set (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.04 sec)
```

session_2：



```
mysql> set @@tx_isolation='read-uncommitted';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> select* from sc where sno='2005001' and cno='1' lock in share mode;
+---------+-----+-------+
| sno     | cno | grade |
+---------+-----+-------+
| 2005001 | 1   |    92 |
+---------+-----+-------+
1 row in set (0.00 sec)

mysql> select * from sc where sno ='2005001' and cno='1' lock in share mode;
+---------+-----+-------+
| sno     | cno | grade |
+---------+-----+-------+
| 2005001 | 1   |    92 |
+---------+-----+-------+
1 row in set (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)
```

分析：由于此时事务一和事务二分别加上了共享锁和排它锁，事务一在事务二获得共享锁的过程中一直在等待排它锁的授予，因此不可以修改数据库，所以事务二两次读出来的数据都是相同的，避免了不可重复读情况的出现。

2.3 表 12-9 InnoDB 存储引擎幻影

代码：按执行顺序：

session_2:

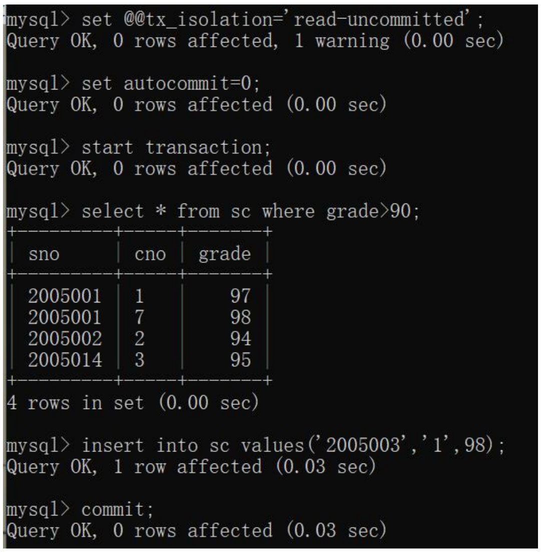
set @@tx_isolation='read-uncommitted';

set autocommit=0;

```
        start transaction;
        select* from sc where grade>90;
    session_1：
        set @@tx_isolation='read-uncommitted';
        set autocommit=0;
        start transaction;
        select* from sc where grade>90;
        insert into sc values('2005003','1',98);
        commit;
    session_2：
        select* from sc where grade>90;
        commit;
```

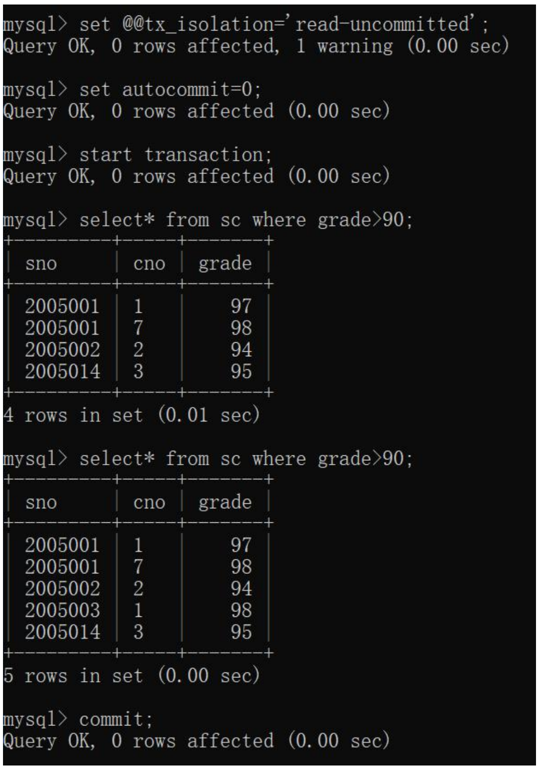
运行结果：

session_1：



```
mysql> set @@tx_isolation='read-uncommitted';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from sc where grade>90;
+---------+-----+-------+
| sno     | cno | grade |
+---------+-----+-------+
| 2005001 | 1   |    97 |
| 2005001 | 7   |    98 |
| 2005002 | 2   |    94 |
| 2005014 | 3   |    95 |
+---------+-----+-------+
4 rows in set (0.00 sec)

mysql> insert into sc values('2005003','1',98);
Query OK, 1 row affected (0.03 sec)

mysql> commit;
Query OK, 0 rows affected (0.03 sec)
```

session_2：



```
mysql> set @@tx_isolation='read-uncommitted';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> select* from sc where grade>90;
+---------+-----+-------+
| sno     | cno | grade |
+---------+-----+-------+
| 2005001 | 1   |    97 |
| 2005001 | 7   |    98 |
| 2005002 | 2   |    94 |
| 2005014 | 3   |    95 |
+---------+-----+-------+
4 rows in set (0.01 sec)

mysql> select* from sc where grade>90;
+---------+-----+-------+
| sno     | cno | grade |
+---------+-----+-------+
| 2005001 | 1   |    97 |
| 2005001 | 7   |    98 |
| 2005002 | 2   |    94 |
| 2005003 | 1   |    98 |
| 2005014 | 3   |    95 |
+---------+-----+-------+
5 rows in set (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)
```

分析：由于 session_1 和 session_2 是并发的，且两者没有加共享锁和排他锁，因此事务二的第二次 select 操作会发现由 session_1 新插入的数据（第一次 select 有 4 条数据，第二次 select 有 5 条数据），出现“幻影问题”。

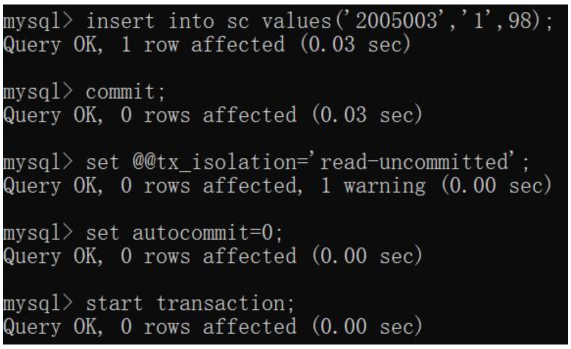
2.4 InnoDB 存储引擎解决幻影

代码：按执行顺序：

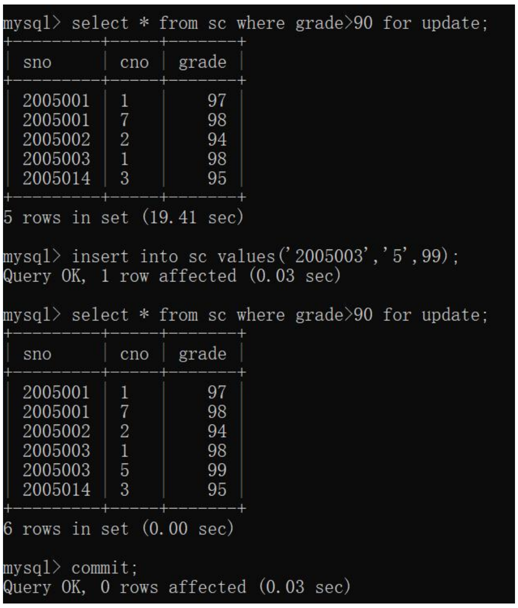
```
session_2:
    set @@tx_isolation='read-uncommitted';
    set autocommit=0;
    start transaction;
    select* from sc where grade>90 lock in share mode;
session_1：
    set @@tx_isolation='read-uncommitted';
    set autocommit=0;
    start transaction;
    select* from sc where grade>90 for update;
session_2：
    select* from sc where grade>90 lock in share mode;
    commit;
session_1:
    insert into sc values('2005003','5',98);
    commit;
```

运行结果：

session_1：

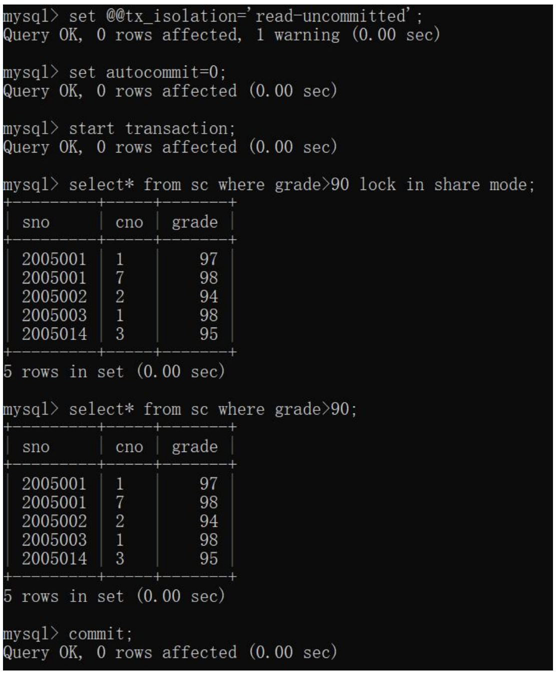


```
mysql> insert into sc values('2005003','1',98);
Query OK, 1 row affected (0.03 sec)

mysql> commit;
Query OK, 0 rows affected (0.03 sec)

mysql> set @@tx_isolation='read-uncommitted';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)
```



```
mysql> select * from sc where grade>90 for update;
+---------+-----+-------+
| sno     | cno | grade |
+---------+-----+-------+
| 2005001 | 1   |    97 |
| 2005001 | 7   |    98 |
| 2005002 | 2   |    94 |
| 2005003 | 1   |    98 |
| 2005014 | 3   |    95 |
+---------+-----+-------+
5 rows in set (19.41 sec)

mysql> insert into sc values('2005003','5',99);
Query OK, 1 row affected (0.03 sec)

mysql> select * from sc where grade>90 for update;
+---------+-----+-------+
| sno     | cno | grade |
+---------+-----+-------+
| 2005001 | 1   |    97 |
| 2005001 | 7   |    98 |
| 2005002 | 2   |    94 |
| 2005003 | 1   |    98 |
| 2005003 | 5   |    99 |
| 2005014 | 3   |    95 |
+---------+-----+-------+
6 rows in set (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.03 sec)
```

session_2：



```
mysql> set @@tx_isolation='read-uncommitted';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> select* from sc where grade>90 lock in share mode;
+---------+-----+-------+
| sno     | cno | grade |
+---------+-----+-------+
| 2005001 | 1   |    97 |
| 2005001 | 7   |    98 |
| 2005002 | 2   |    94 |
| 2005003 | 1   |    98 |
| 2005014 | 3   |    95 |
+---------+-----+-------+
5 rows in set (0.00 sec)

mysql> select* from sc where grade>90;
+---------+-----+-------+
| sno     | cno | grade |
+---------+-----+-------+
| 2005001 | 1   |    97 |
| 2005001 | 7   |    98 |
| 2005002 | 2   |    94 |
| 2005003 | 1   |    98 |
| 2005014 | 3   |    95 |
+---------+-----+-------+
5 rows in set (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)
```

分析：由于此时 session_2 加了共享锁，session_1 加了排他锁，因此 session_1 要一直等待 session_2 共享锁的释放，才能得到排他锁，而 session_2 的两次 select 操作不会读到 session_1 加入的数据，因此两次的结果相同，避免了幻影现象的出现。