**《数据库系统实验》**

**实验报告**

|  |  |
| --- | --- |
| 题目 | 实验12 |
| 姓名 | 宋渝杰 |
| 学号 | 18340146 |
| 班级 | 计科六班 |

**一、实验环境：**

win10+MySQL 8.0

**二、实验内容与完成情况：**

**第12章表12-7/8/9/10：**

先创建数据库和表、输入数据，代码如下：

create database jxgl;

use jxgl;

# 建表

create table student

(sno varchar(7),

sname varchar(5),

sage numeric(2,0),

ssex varchar(1),

sdept varchar(5),

primary key(sno));

create table course

(cno varchar(4),

cname varchar(6),

cpno varchar(4),

ccredit numeric(1,0),

primary key(cno));

create table sc

(sno varchar(7),

cno varchar(4),

grade float(3),

primary key(sno,cno),

foreign key(sno) references student(sno),

foreign key(cno) references course(cno));

# 加入课本数据

insert into student values('2005001','钱横',18,'男','Cs');

insert into student values('2005002','王林',19,'女','Cs');

insert into student values('2005003','李民',20,'男','Is');

insert into student values('2005004','赵欣然',16,'女','Ma');

insert into course values('1','数据库系统','5',4);

insert into course values('2','数学分析',null,2);

insert into course values('3','信息系统导论','1',3);

insert into course values('4','操作系统原理','6',3);

insert into course values('5','数据结构','7',4);

insert into course values('6','数据处理基础',null,4);

insert into course values('7','C语言','6',3);

insert into sc values('2005001','1',87);

insert into sc values('2005001','2',67);

insert into sc values('2005001','3',90);

insert into sc values('2005002','2',95);

insert into sc values('2005003','3',88);

**表12-7：**

**Session1：**

use jxgl;

set @@transaction\_isolation='read-uncommitted';

set autocommit=0;

start transaction;

select \* from sc where sno = '2005001' and cno = '1';

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

commit;

**Session2：**

use jxgl;

set @@transaction\_isolation='read-uncommitted';

set autocommit=0;

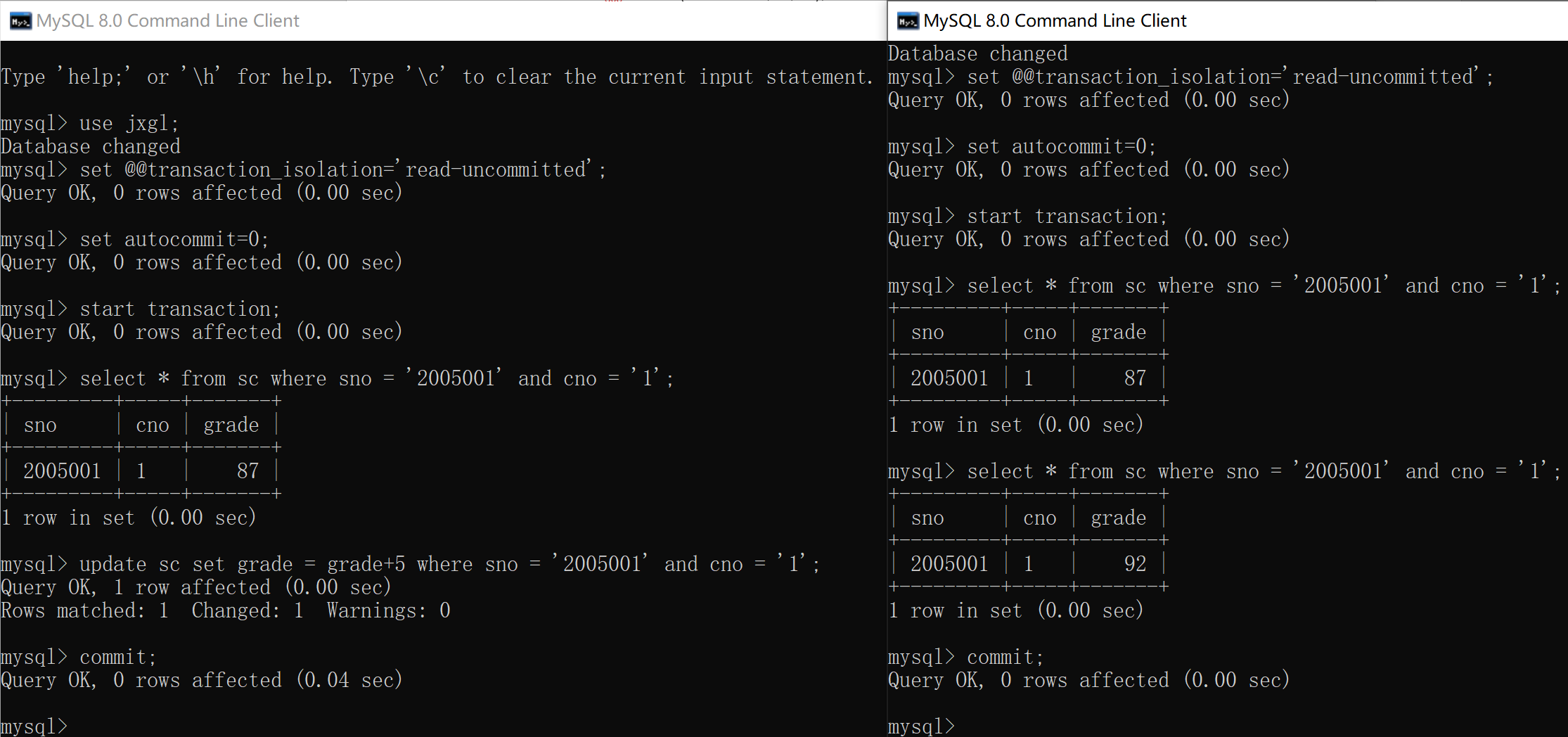
start transaction;

select \* from sc where sno = '2005001' and cno = '1';

select \* from sc where sno = '2005001' and cno = '1';

commit;

**结果：**



左边为事务1的结果，右边为事务2的结果。

可以看出，出现了不可重复读的情况，事务2两次读取的数值不一样了

**表12-8：**

**Session1：**

use jxgl;

set @@transaction\_isolation='read-uncommitted';

set autocommit=0;

start transaction;

select \* from sc where sno = '2005001' and cno = '1' for update;

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

commit;

**Session2：**

use jxgl;

set @@transaction\_isolation='read-uncommitted';

set autocommit=0;

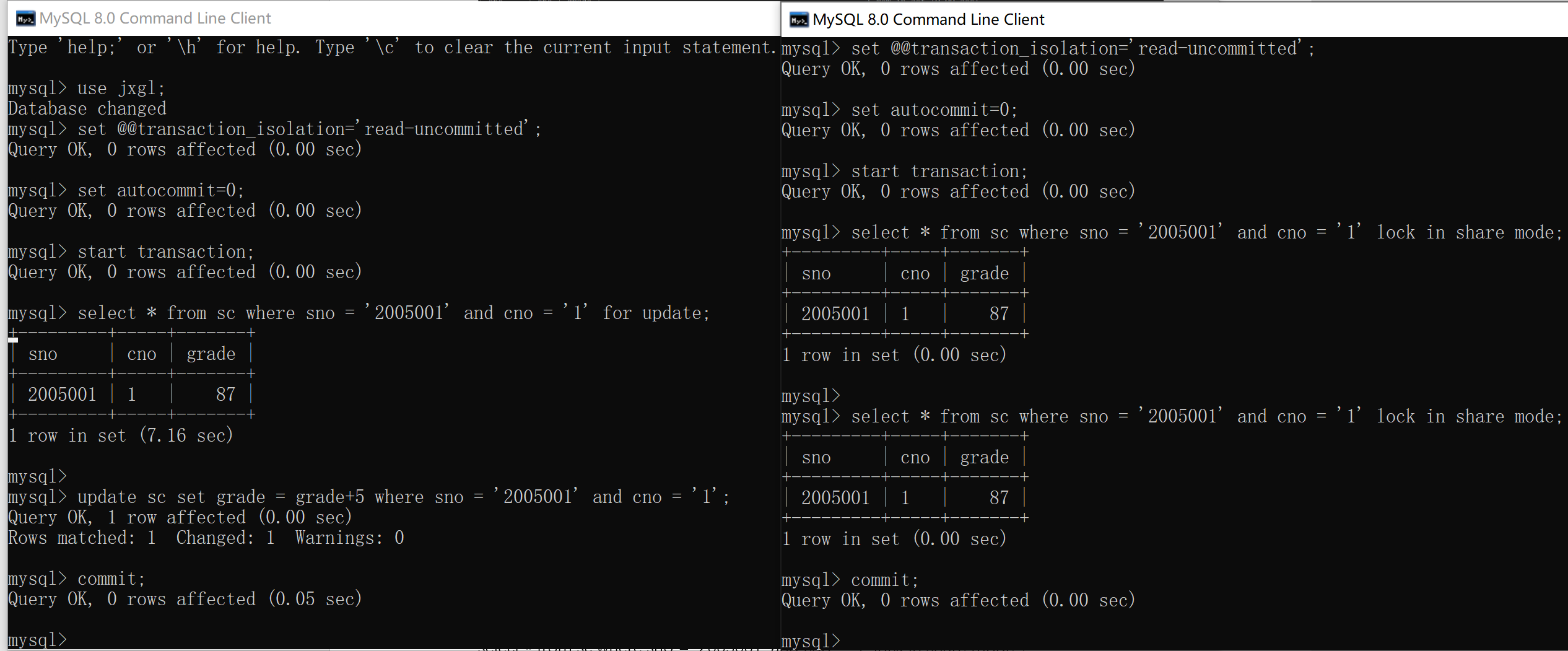
start transaction;

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

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

commit;

**结果：**



左边为事务1的结果，右边为事务2的结果。

可以看出，使用了共享锁之后，事务1的更新操作会等待事务2的第二次查询解锁之后再进行，因此事务1的更新操作会被固定限制在事务2第二次查询之后，确保了事务2两次查询结果的相同。

**表12-9：**

**Session1：**

use jxgl;

set @@transaction\_isolation='read-uncommitted';

set autocommit=0;

start transaction;

select \* from sc where grade > 90;

insert into sc values('2005003', '1', 97);

commit;

**Session2：**

use jxgl;

set @@transaction\_isolation='read-uncommitted';

set autocommit=0;

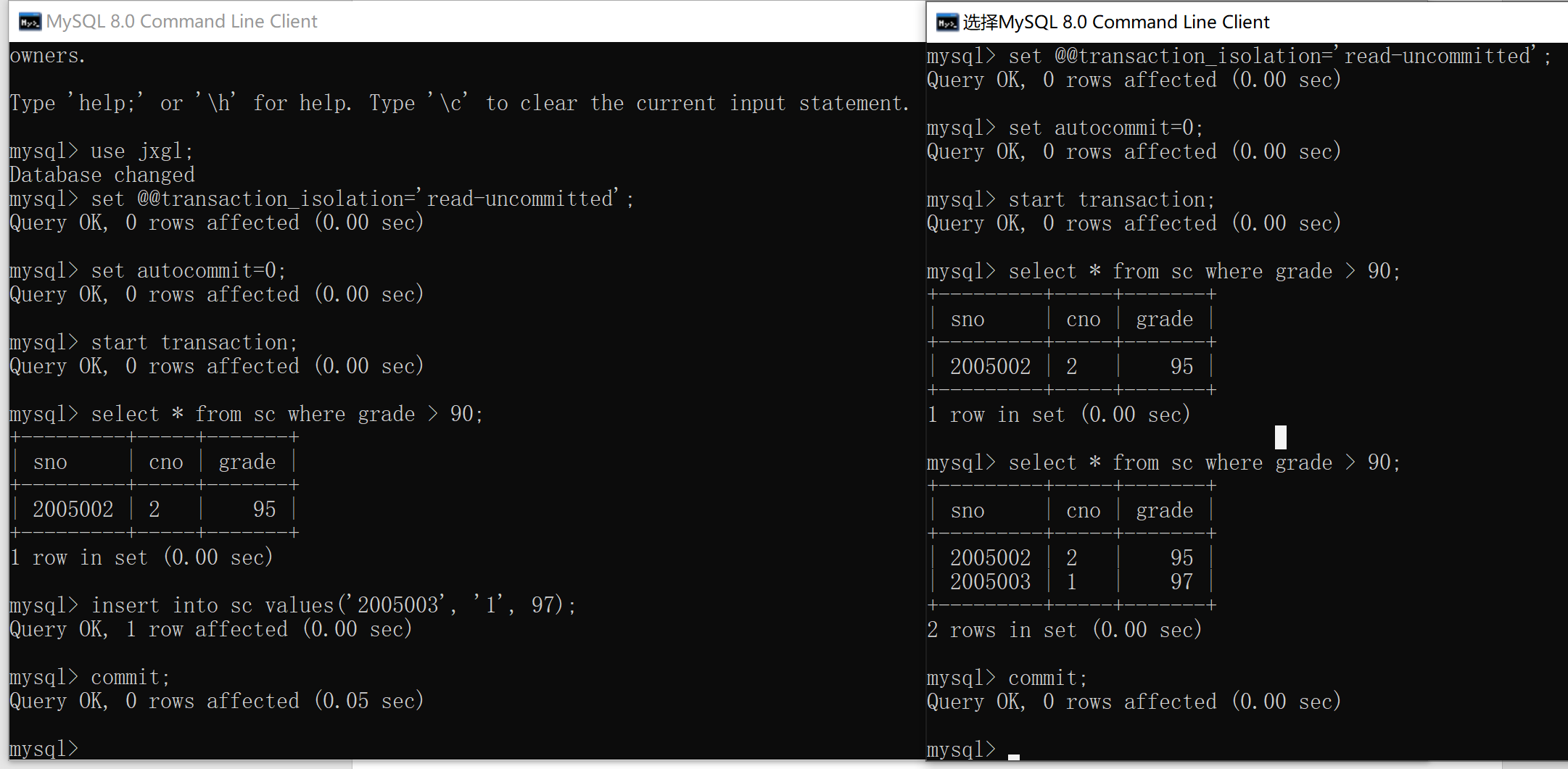
start transaction;

select \* from sc where grade > 90;

select \* from sc where grade > 90;

commit;

**结果：**



左边为事务1的结果，右边为事务2的结果。

事务2在第一次select访问数据之后，事务1插入一行数据，所以事务2第二次select访问数据的结果和第一次不一样，即产生了幻影现象。

**表12-10：**

**Session1：**

use jxgl;

set @@transaction\_isolation='read-uncommitted';

set autocommit=0;

start transaction;

select \* from sc where grade > 90 for update;

insert into sc values('2005003', '1', 97);

commit;

**Session2：**

use jxgl;

set @@transaction\_isolation='read-uncommitted';

set autocommit=0;

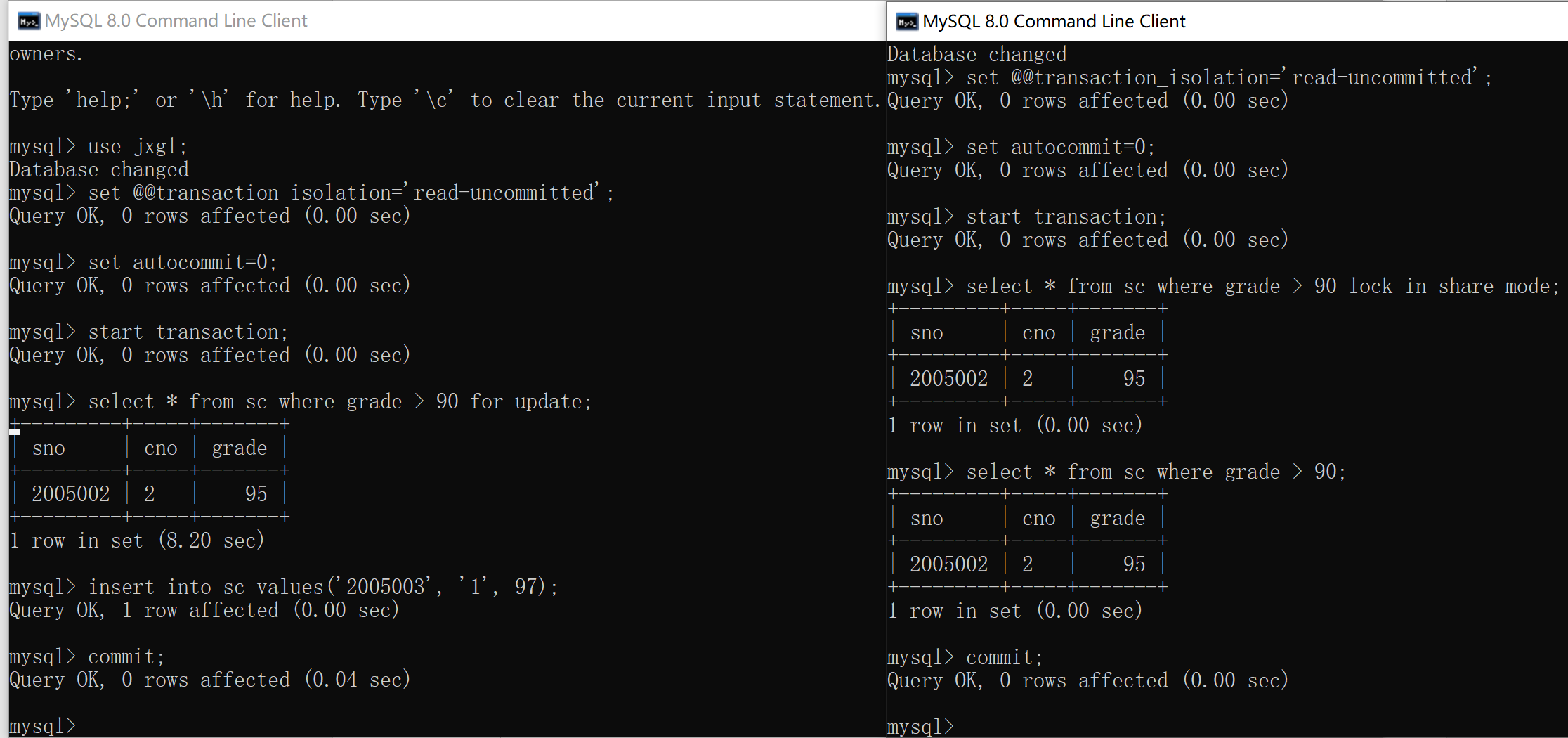
start transaction;

select \* from sc where grade > 90 lock in share mode;

select \* from sc where grade > 90;

commit;

**结果：**



左边为事务1的结果，右边为事务2的结果。

可以看出，使用了共享锁之后，事务1的插入操作会等待事务2的第二次查询解锁之后再进行，因此事务1的插入操作会被固定限制在事务2第二次查询之后，确保了事务2两次查询结果的相同。

**三、实验心得：**

本次实验内容不多，而且内容就是复现实验课本的内容，因此也基本没有遇到困难，只需要了解一下复现的实验原理即可~