

《数据库系统实验》

实验报告

题目	实验 12
姓名	郝裕玮
学号	18329015
班级	计科 1 班

一、实验环境

MySQL 命令行

二、实验内容与完成情况

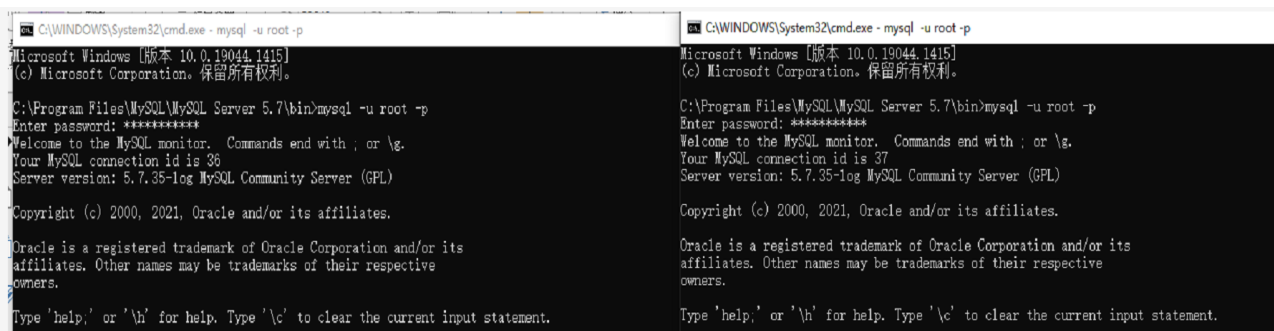
12.1 针对表 12-5

表 12-5 InnoDB 存储引擎脏读

session_1	session_2
<pre>mysql>set @@tx_isolation='read- uncommitted'; Query OK, 0 rows affected (0.00 sec) mysql>set autocommit=0; Query OK, 0 rows affected (0.00 sec) mysql>start transaction; mysql>select * from sc where sno='2005001' and cno='1' ; +-----+-----+-----+ sno cno grade +-----+-----+-----+ 2005001 1 87 +-----+-----+-----+</pre>	<pre>mysql>set @@tx_isolation='read- uncommitted'; Query OK, 0 rows affected (0.00 sec) mysql>set autocommit=0; Query OK, 0 rows affected (0.00 sec) mysql>start transaction; mysql>select * from sc where sno='2005001' and cno='1'; +-----+-----+-----+ sno cno grade +-----+-----+-----+ 2005001 1 87 +-----+-----+-----+</pre>
<pre>mysql>update sc set grade=grade+ 5 where sno='2005001' and cno='1'; mysql>select * from sc where sno='2005001' and cno='1'; +-----+-----+-----+ sno cno grade +-----+-----+-----+ 2005001 1 92 +-----+-----+-----+</pre>	
<pre>mysql>rollback;</pre>	<pre>mysql>select * from sc where sno='2005001' and cno='1'; +-----+-----+-----+ sno cno grade +-----+-----+-----+ 2005001 1 87 +-----+-----+-----+</pre>
	<pre>mysql>commit;</pre>

首先我们在该目录 C:\Program Files\MySQL\MySQL Server 5.7\bin 下打开两个 cmd 窗口。并连接到 SQL。

连接到 SQL 的语句为：mysql -u root -p，执行后输入密码即可。



两个 cmd 窗口连接到 SQL 后，第一条语句均为：

use jxgl

执行该语句以调用 jxgl 数据库。

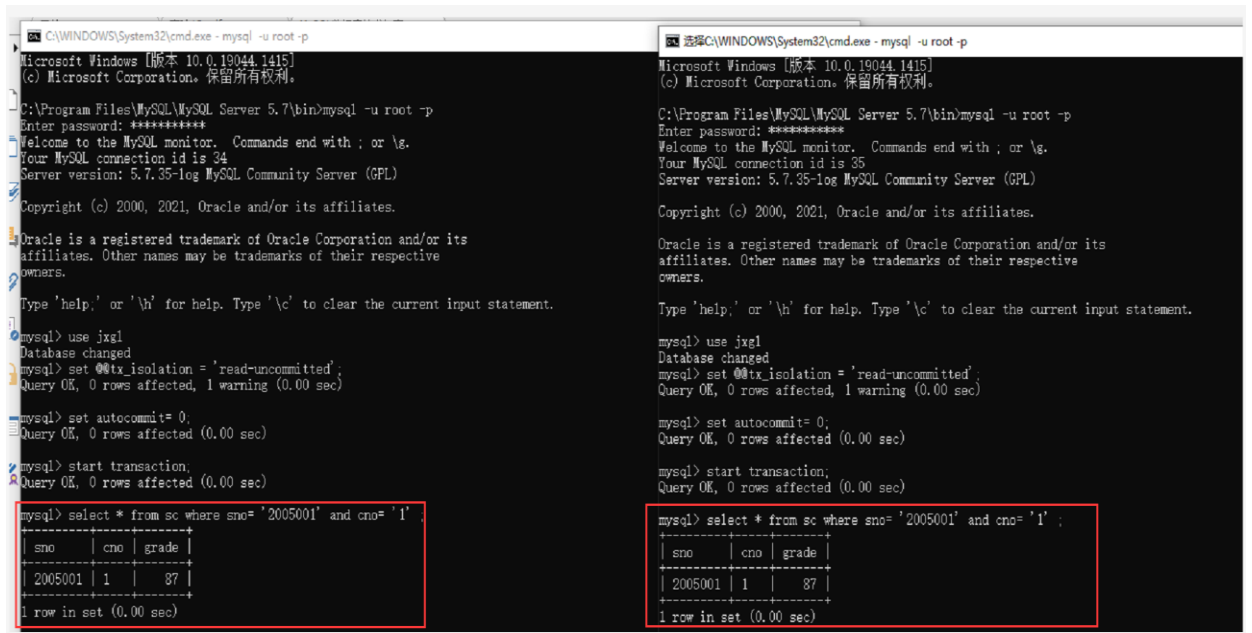
执行后根据上图顺序输入语句即可。

结果如下：

(1) 两边均执行到这步时（结果图见下页）：

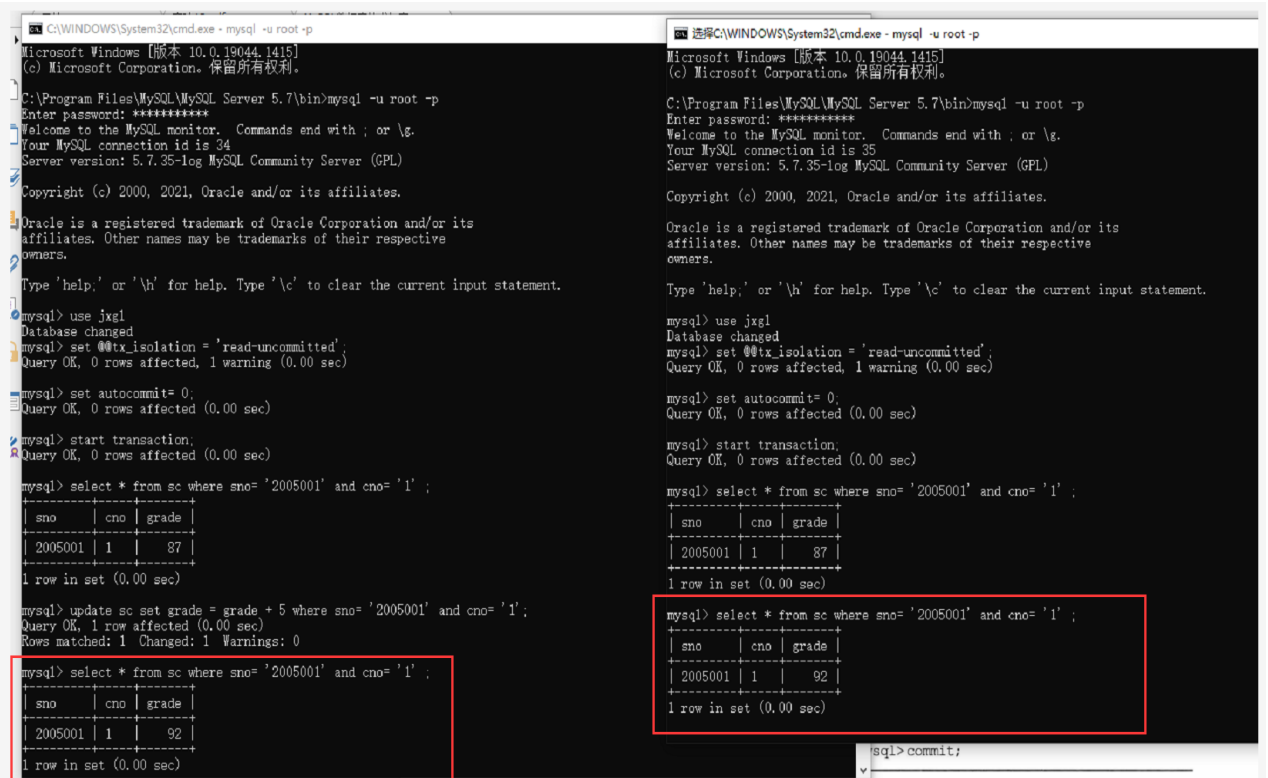
表 12-5 InnoDB 存储引擎脏读

session_1	session_2
mysql>set @@tx_isolation='read-uncommitted'; Query OK, 0 rows affected (0.00 sec)	mysql>set @@tx_isolation='read-uncommitted'; Query OK, 0 rows affected (0.00 sec)
mysql>set autocommit=0; Query OK, 0 rows affected (0.00 sec)	mysql>set autocommit=0; Query OK, 0 rows affected (0.00 sec)
mysql>start transaction;	mysql>start transaction;
mysql>select * from sc where sno='2005001' and cno='1';	mysql>select * from sc where sno='2005001' and cno='1';
+-----+-----+-----+ sno cno grade +-----+-----+-----+ 2005001 1 87 +-----+-----+-----+	+-----+-----+-----+ sno cno grade +-----+-----+-----+ 2005001 1 87 +-----+-----+-----+



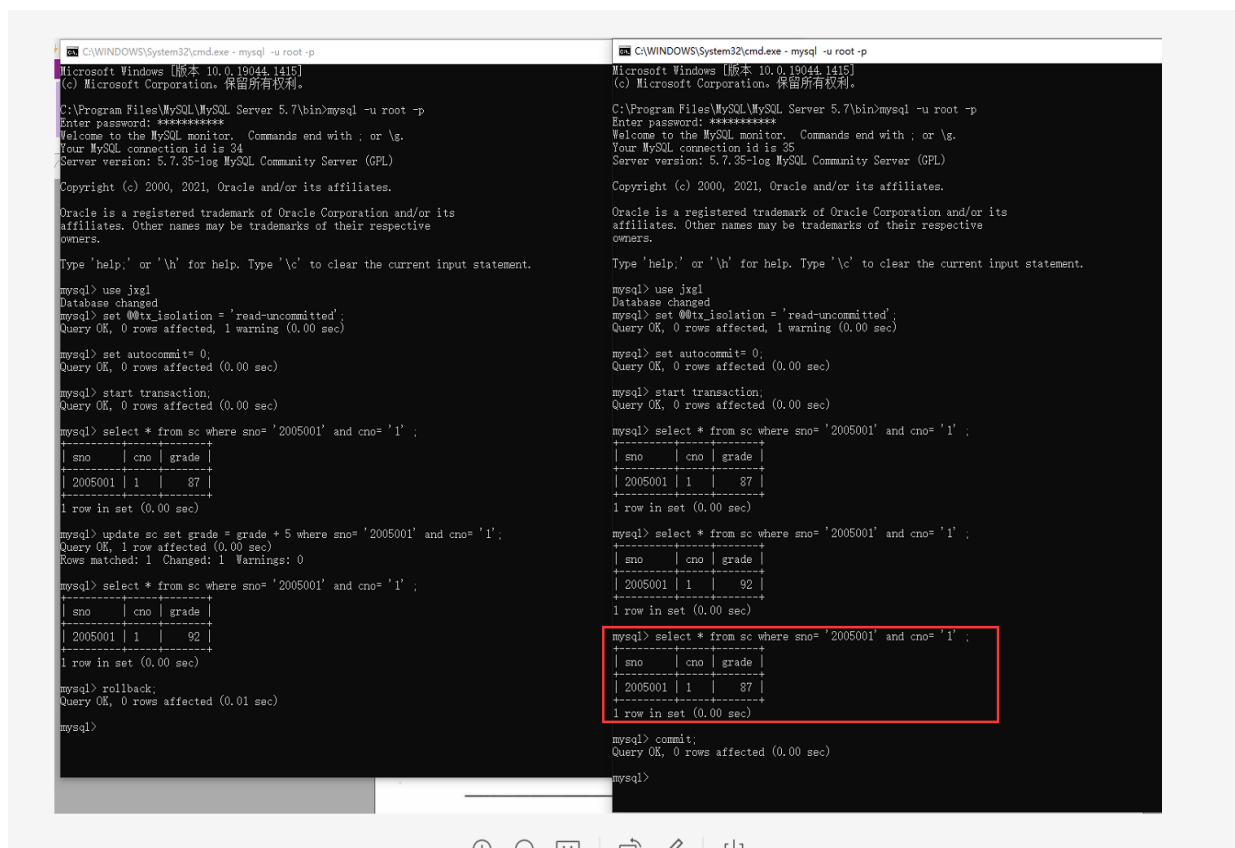
(2) session_1 更新完 grade 后:

mysql> update sc set grade=grade+5 where sno='2005001' and cno='1';	
mysql> select * from sc where sno='2005001' and cno='1';	mysql> select * from sc where sno='2005001' and cno='1';
+-----+-----+-----+	+-----+-----+-----+
sno cno grade	sno cno grade
+-----+-----+-----+	+-----+-----+-----+
2005001 1 92	2005001 1 92
+-----+-----+-----+	+-----+-----+-----+



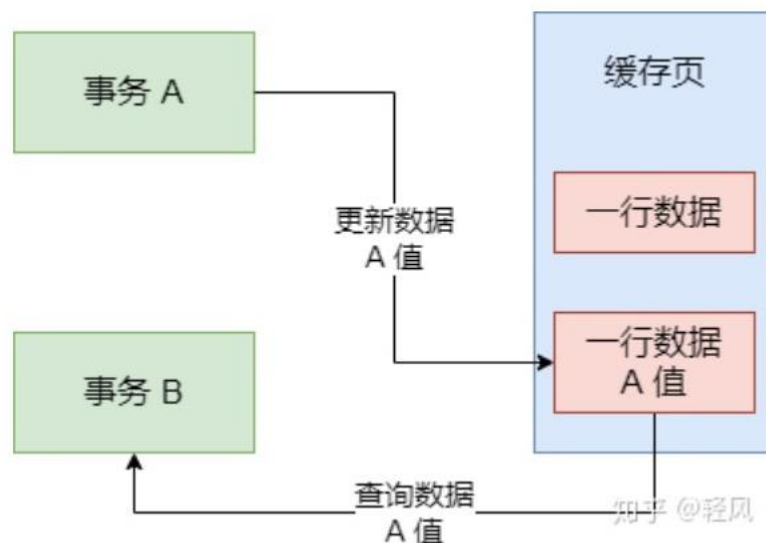
(3) session_1 执行 rollback 后:

mysql> rollback;	mysql> select * from sc where sno= '2005001' and cno= '1';						
	<table><tr><th>sno</th><th>cno</th><th>grade</th></tr><tr><td>2005001</td><td>1</td><td>87</td></tr></table>	sno	cno	grade	2005001	1	87
sno	cno	grade					
2005001	1	87					
	mysql> commit;						



由上述现象可知：隔离级别为未提交读（READ UNCOMMITTED）时会产生脏读。

脏读指的是在不同的事务下，可以读到另外事务未提交的数据，则违反了数据库的隔离性。



而本题中由于二者隔离级别相同，且未设置任何锁，所以会导致 session_2 一直可以读到 session_1 修改但未提交 commit 或回滚 rollback 的数据。

12.2 针对表 12-6

表 12-6 InnoDB 存储引擎避免脏读

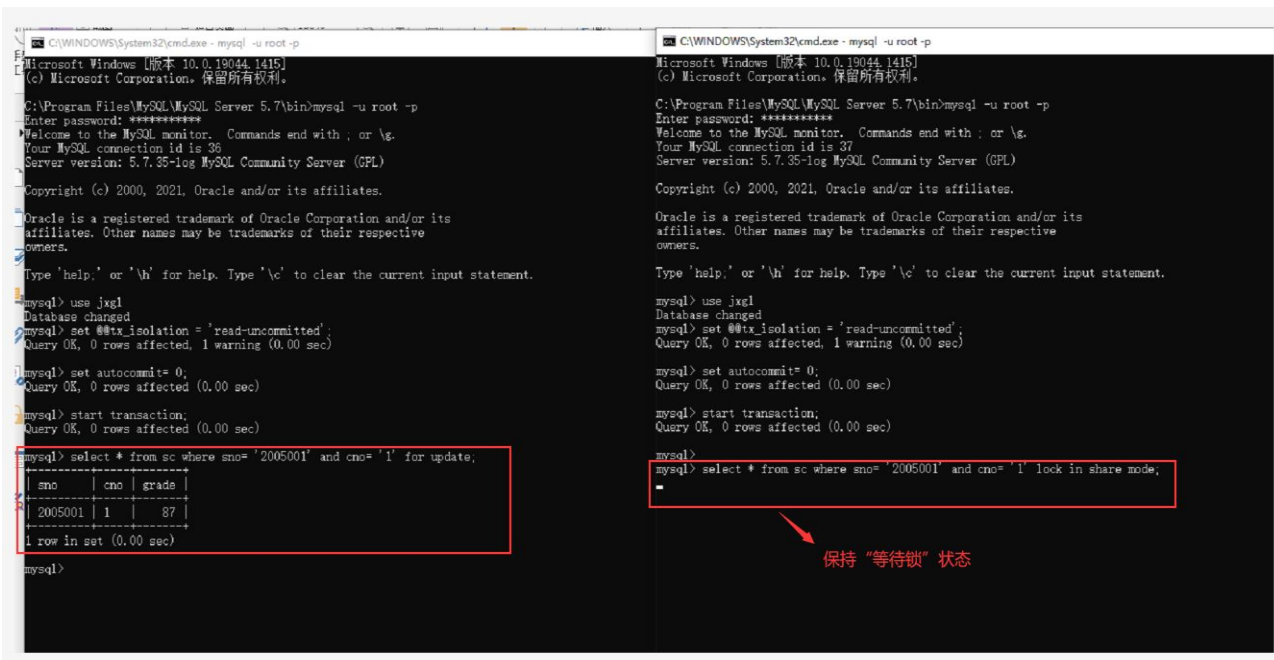
session_1	session_2
<pre>mysql> set @@tx_isolation='read-uncommitted'; Query OK, 0 rows affected (0.00 sec) mysql> set autocommit=0; Query OK, 0 rows affected (0.00 sec) mysql> start transaction; mysql> select * from sc where sno='2005001' and cno='1' for update;</pre> <pre>+-----+-----+-----+ sno cno grade +-----+-----+-----+ 2005001 1 87 +-----+-----+-----+</pre>	
<pre>mysql> update sc set grade=grade+5 where sno='2005001' and cno='1';</pre>	等待
<pre>mysql> select * from sc where sno='2005001' and cno='1';</pre> <pre>+-----+-----+-----+ sno cno grade +-----+-----+-----+ 2005001 1 92 +-----+-----+-----+</pre>	等待
<pre>mysql> rollback;</pre>	等待
	获得锁 <pre>+-----+-----+-----+ sno cno grade +-----+-----+-----+ 2005001 1 87 +-----+-----+-----+</pre>
	<pre>mysql> select * from sc where sno='2005001' and cno='1' lock in share mode;</pre> <pre>+-----+-----+-----+ sno cno grade +-----+-----+-----+ 2005001 1 87 +-----+-----+-----+</pre>
	<pre>mysql> commit;</pre>

解决脏读的办法是指定更高级别的事务隔离级别，如 READ COMMITTED，或者对查询添加共享锁或排他锁。

(1) 两边执行到第一次 select 查询时：

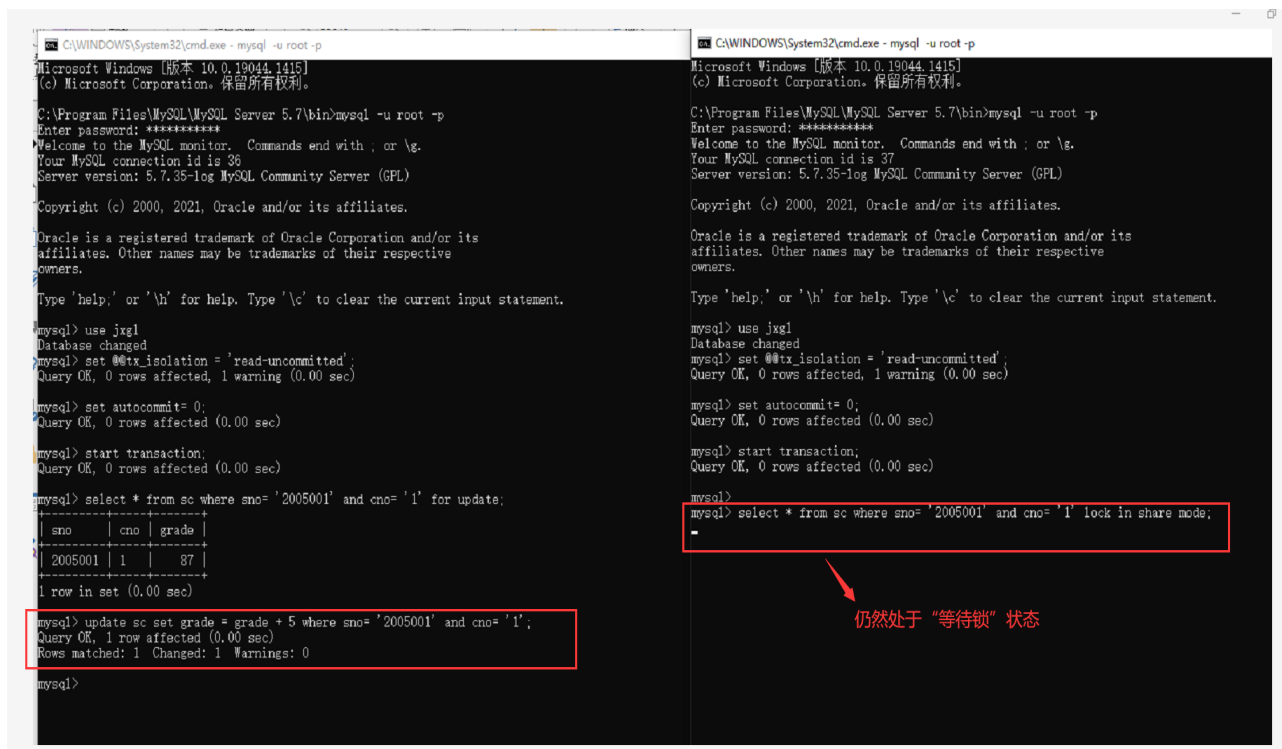
表 12-6 InnoDB 存储引擎避免脏读

session_1	session_2
mysql>set @@tx_isolation='read-uncommitted'; Query OK, 0 rows affected (0.00 sec) mysql>set autocommit=0; Query OK, 0 rows affected (0.00 sec) mysql>start transaction; mysql>select * from sc where sno='2005001' and cno='1' for update; +-----+-----+-----+ sno cno grade +-----+-----+-----+ 2005001 1 87 +-----+-----+-----+	mysql>set @@tx_isolation='read-uncommitted'; Query OK, 0 rows affected (0.00 sec) mysql>set autocommit=0; Query OK, 0 rows affected (0.00 sec) mysql>start transaction; mysql>select * from sc where sno='2005001' and cno='1' lock in share mode; 等待锁



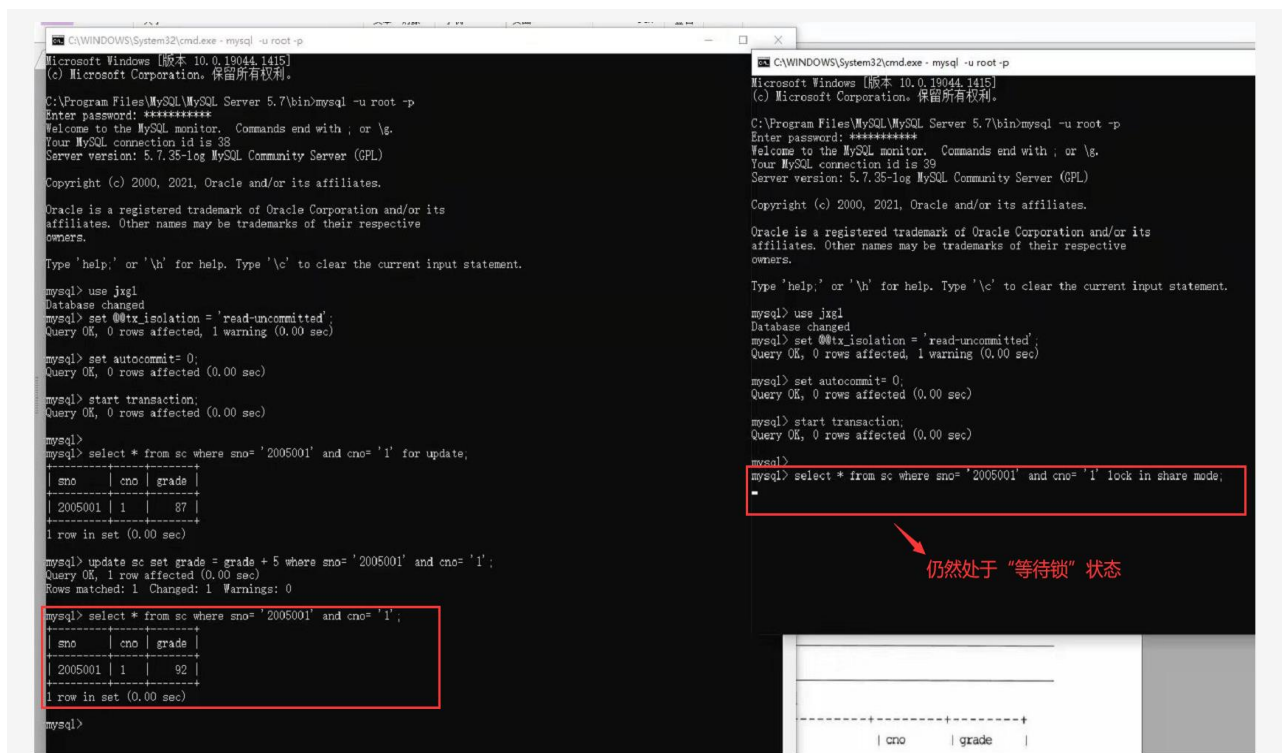
(2) session_1 更新完 grade 后（结果图见下页）：

mysql>update sc set grade=grade+5 where sno='2005001' and cno='1';	等待
--	----



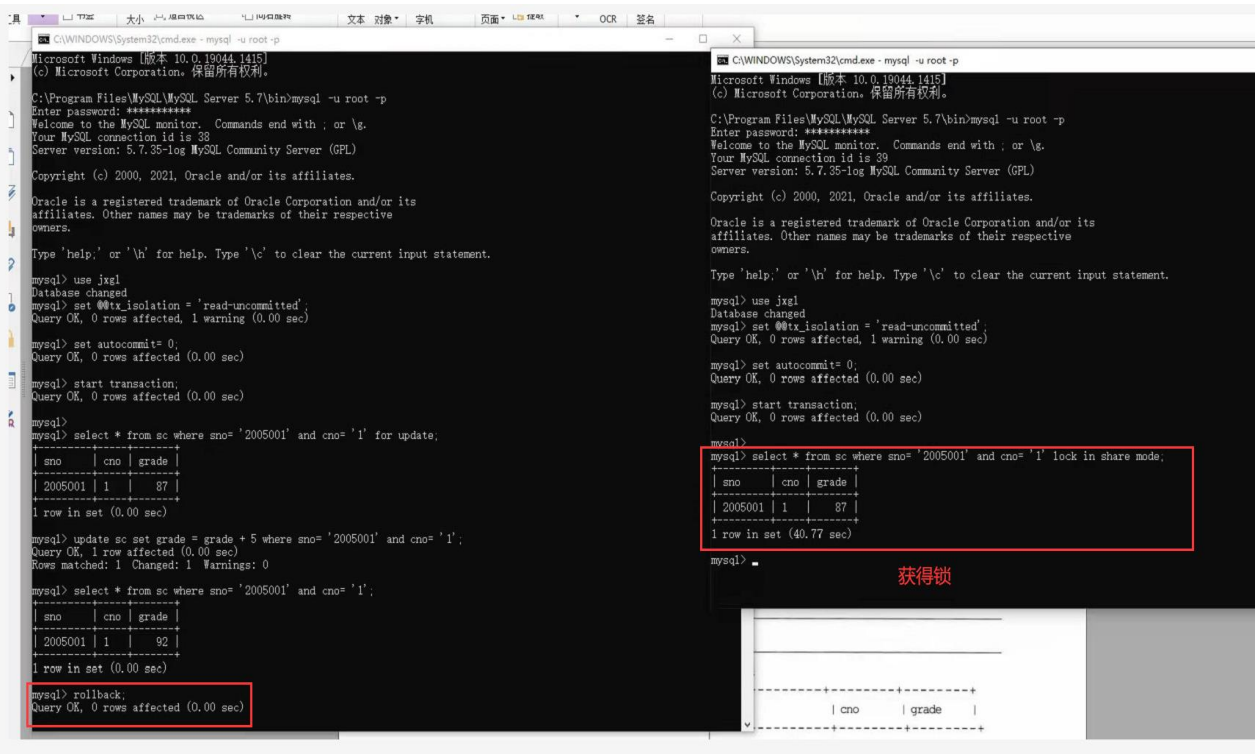
(3) session_1 更新完 grade 重新查询时 (图见下页) :

mysql>select * from sc where sno= '2005001' and cno= '1';	等待
+-----+-----+-----+	
sno cno grade	
+-----+-----+-----+	
2005001 1 92	
+-----+-----+-----+	



(4) session_1 执行 rollback 后（结果图见下页）：

mysql> rollback;	等待						
	获得锁						
	<table><tr><td>sno</td><td>cno</td><td>grade</td></tr><tr><td>2005001</td><td>1</td><td>87</td></tr></table>	sno	cno	grade	2005001	1	87
sno	cno	grade					
2005001	1	87					
	mysql> select * from sc where sno='2005001' and cno='1' lock in share mode;						
	<table><tr><td>sno</td><td>cno</td><td>grade</td></tr><tr><td>2005001</td><td>1</td><td>87</td></tr></table>	sno	cno	grade	2005001	1	87
sno	cno	grade					
2005001	1	87					
	mysql> commit;						



(5) session_2 执行再次查询并 commit（结果图见下页）：

	<pre>mysql>select * from sc where sno='2005001' and cno='1' lock in share mode;</pre> <table><tr><td>+</td><td>-----+</td><td>-----+</td><td>-----+</td></tr><tr><td> </td><td>sno</td><td> </td><td>cno</td><td> </td><td>grade</td><td> </td></tr><tr><td>+</td><td>-----+</td><td>-----+</td><td>-----+</td></tr><tr><td> </td><td>2005001</td><td> </td><td>1</td><td> </td><td>87</td><td> </td></tr><tr><td>+</td><td>-----+</td><td>-----+</td><td>-----+</td></tr></table>	+	-----+	-----+	-----+		sno		cno		grade		+	-----+	-----+	-----+		2005001		1		87		+	-----+	-----+	-----+
+	-----+	-----+	-----+																								
	sno		cno		grade																						
+	-----+	-----+	-----+																								
	2005001		1		87																						
+	-----+	-----+	-----+																								
	<pre>mysql> commit;</pre>																										


```
Microsoft Windows [版本 10.0.19044.1415]
(c) Microsoft Corporation. 保留所有权利。

C:\Program Files\MySQL\MySQL Server 5.7\bin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.7.35-log MySQL Community Server (GPL)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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' for update;
+-----+
| sno | cno | grade |
+-----+
| 2005001 | 1 | 87 |
+-----+
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 (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

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

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

C:\Program Files\MySQL\MySQL Server 5.7\bin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 39
Server version: 5.7.35-log MySQL Community Server (GPL)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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' lock in share mode;
+-----+
| sno | cno | grade |
+-----+
| 2005001 | 1 | 87 |
+-----+
1 row in set (40.77 sec)

mysql> select * from sc where sno='2005001' and cno='1' lock in share mode;
+-----+
| sno | cno | grade |
+-----+
| 2005001 | 1 | 87 |
+-----+
1 row in set (0.00 sec)

mysql> commit;
ERROR 1064 (42000): You have an error in your SQL syntax, check the manual that corresponds to your MySQL server version
for the right syntax to use near 'commit' at line 2
mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

接下来我们对该现象进行分析：

① 排他锁 for update

又称写锁（for update），例如 update, insert, delete, 上锁之后，另一个线程不可以读和修改。

② 共享锁 lock in share mode

又称读锁（lock in share mode），例如 select, 当上锁之后，另一个线程只可以读，不可以修改。

所以在 session_1 添加排他锁 for update 后, session_2 的共享锁 lock in share mode 就一直处于等待状态，无法获取到查询结果（等待另一个事务 session_1 处理完）。直至 session_1 执行 rollback 之后，session_2 才可以获得锁，得到 select 查询结果。

同时，对于共享锁 lock in share mode，若在等待一定时间后未获取结果，则会直接报错。

```
C:\WINDOWS\System32\cmd.exe - mysql -u root -p
Microsoft Windows [版本 10.0.19044.1415]
(c) Microsoft Corporation. 保留所有权利。

C:\Program Files\MySQL\MySQL Server 5.7\bin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 42
Server version: 5.7.35-log MySQL Community Server (GPL)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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' for update;
+----+-----+
| sno | cno | grade |
+----+-----+
| 2005001 | 1 | 37 |
+----+-----+
1 row in set (0.00 sec)

mysql>

C:\WINDOWS\System32\cmd.exe - mysql -u root -p
Microsoft Windows [版本 10.0.19044.1415]
(c) Microsoft Corporation. 保留所有权利。

C:\Program Files\MySQL\MySQL Server 5.7\bin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 5.7.35-log MySQL Community Server (GPL)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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>
mysql> select * from sc where sno= '2005001' and cno= '1' lock in share mode;
ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting transaction
mysql>
```

等待50s后, session_2自动报错

最后再补充解释下 2 个 session 开头相同的 3 个 SQL 语句：

① set @@tx_isolation = 'read-uncommitted';

用于修改事务隔离级别。

② set autocommit=0;

当 autocommit 为 0 时，不管有没有 START TRANSACTION，只有当 commit 数据才会生效，ROLLBACK 后就会回滚。

③ start transaction;

这条命令开始一个新的事务。如果声明了隔离级别或者读写模式，那么新事务就使用这个特性。