**《数据库原理》实验报告**

**实验名称 数据库的事务创建与运行实验**

**班 级 2014211302**

**学 号 2014211182**

**姓 名 曹桢**

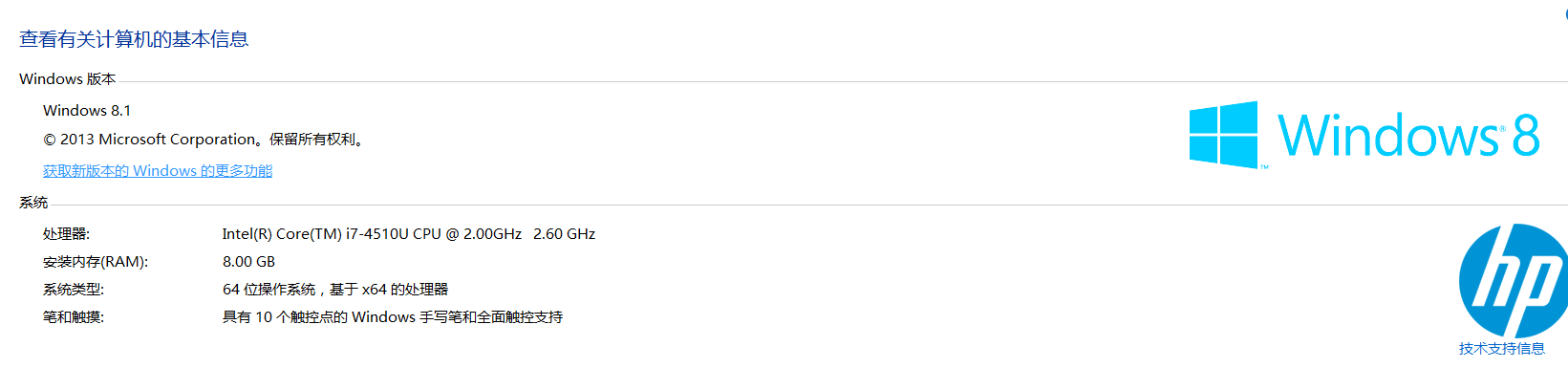
# 实验7 数据库的事务创建与运行实验

## 一、实验目的

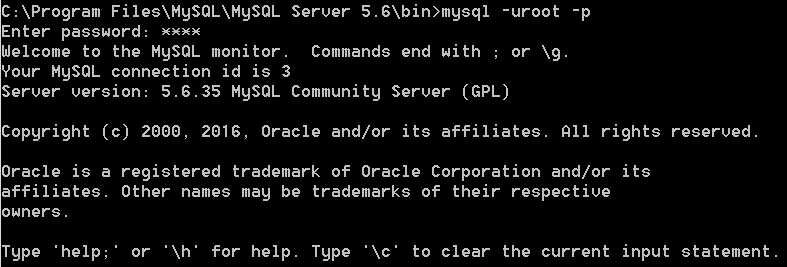
通过实验，了解mysql数据库系统中各类数据库事务的定义机制和基于锁的并发控制机制，掌握mysql数据库系统的事务控制机制。

## 二、实验环境

* 操作系统环境：windows8.1 64位



* 数据库版本：mysql 5.6



## 三、实验内容

１． 定义三种模式的数据库事务

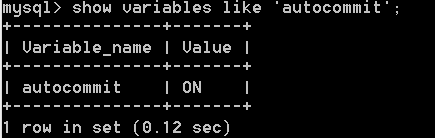
２． 察看事务的隔离级别

## 四、实验步骤

**（一）定义三种模式的数据库事务**

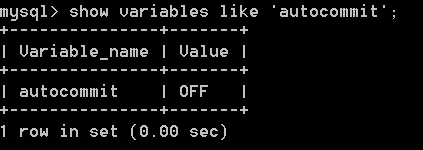
1. **显式事务**

需要修改autocommit变量。mysql默认采用autocommit模式。

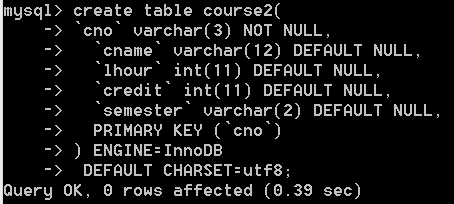


修改变量autocommit





创建一个innodb类型表格course2



insert into course2 values('C06','C',30,2,'春');

select \* from course2;

savepoint p1;

插入一个元组，并设置第一个回滚点p1

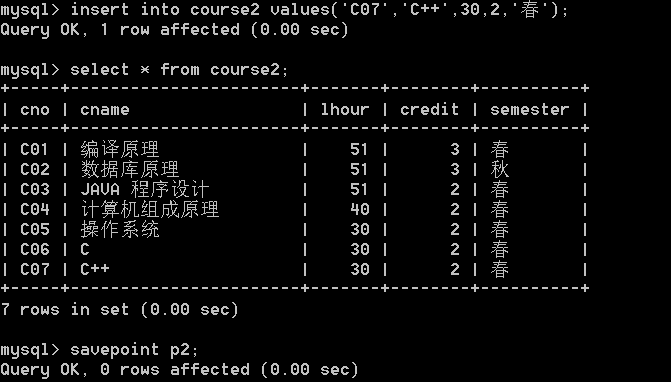


insert into course2 values('C07','C++',30,2,'春');

select \* from course2;

savepoint p2;

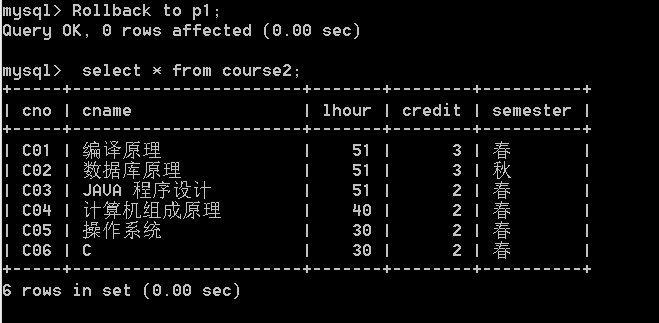
再插入一个元组，并设置第一个回滚点p2



Rollback to p1;

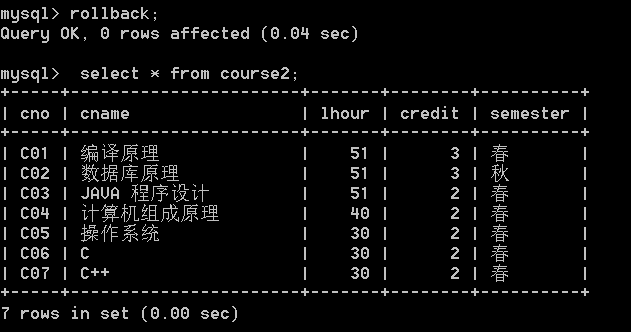
select \* from course2;

回滚到p1



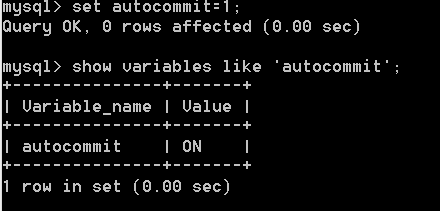
Rollback;

回滚到没有执行上一步的时候。

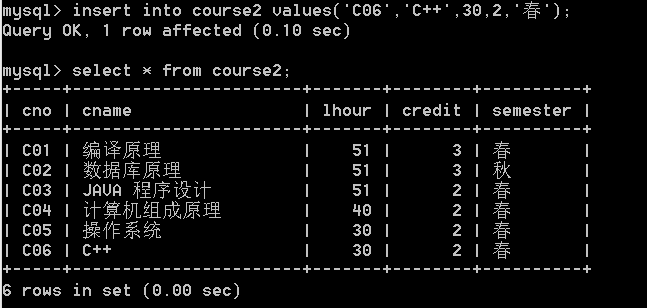


**2 自动提交事务**

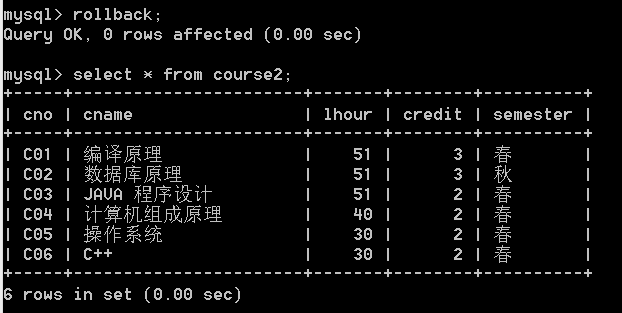
设置autocommit值为自动提交



插入元组，设置回滚点







没有成功回滚。

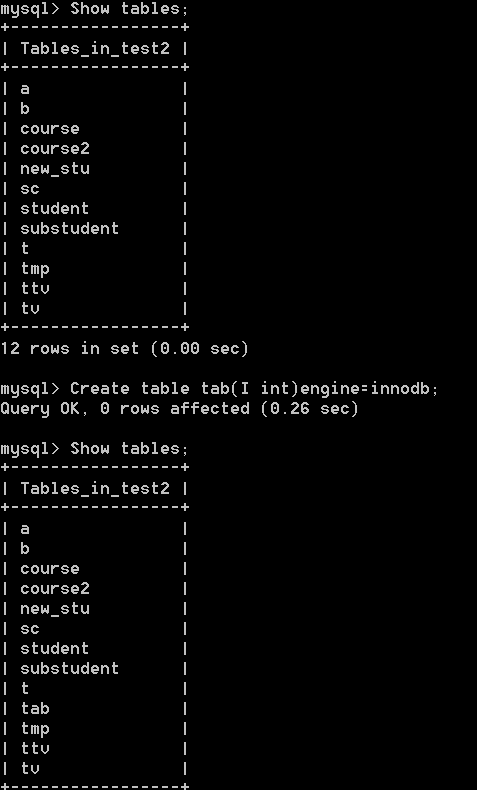
**3隐式事务**



Show tables;

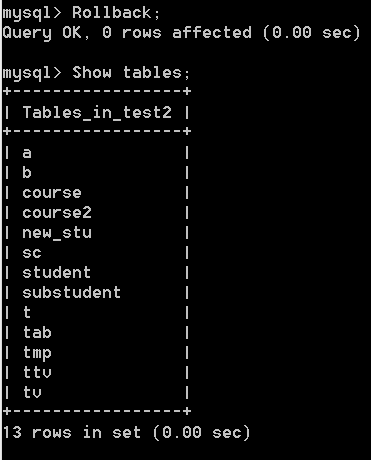
Create table tab(I int)engine=innodb;

Show tables;



Rollback;

Show tables;

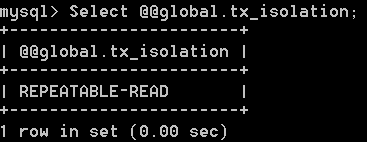


没有成功回滚。

**（二）察看事务的隔离级别**

A查看innodb系统级别的事务隔离级别

Select @@global.tx\_isolation;



B查看innodb会话级别的事务隔离级别

Select @@tx\_isolation;



## 五、实验结果分析

1. 建立显示事务，插入元组，设置回滚点，并回滚，发现操作都被回滚。显式事务执行成功。显式事务即事务不自动提交，可以回滚到原始点，在rollback和commit之前对数据库的修改都可以挽回，而不是永久写入。
2. 建立自动提交事务，插入元组后，设置回滚点，回滚后，发现结果并没有改变，说明回滚失败，数据已经自动提交。

3、在建立隐式事务后，虽然autocommit=off，但某些语句会隐含地结束一个事务，直接commit数据，比如create table，alter funciton，drop index等。

4、查看innodb系统级别的事务隔离级别和会话级别的事务隔离级别，其值均为REPEATABLE-READ，可重复读。

隔离级别：

1）Read Uncommitted（读取未提交内容） ：在该隔离级别，所有事务都可以看到其他未提交事务的执行结果。本隔离级别很少用于实际应用，因为它的性能也不比其他级别好多少。读取未提交的数据，也被称之为脏读（Dirty Read）。

2）Read Committed（读取提交内容） ：这是大多数数据库系统的默认隔离级别（但不是MySQL默认的）。它满足了隔离的简单定义：一个事务只能看见已经提交事务所做的改变。这种隔离级别也支持所谓的不可重复读（Nonrepeatable Read），因为同一事务的其他实例在该实例处理其间可能会有新的commit，所以同一select可能返回不同结果。

3）Repeatable Read（可重读） ：这是MySQL的默认事务隔离级别，它确保同一事务的多个实例在并发读取数据时，会看到同样的数据行。不过理论上，这会导致另一个棘手的问题：幻读（Phantom Read）。简单的说，幻读指当用户读取某一范围的数据行时，另一个事务又在该范围内插入了新行，当用户再读取该范围的数据行时，会发现有新的“幻影” 行。InnoDB和Falcon存储引擎通过多版本并发控制（MVCC，Multiversion Concurrency Control）机制解决了该问题。

4）Serializable（可串行化） ：这是最高的隔离级别，它通过强制事务排序，使之不可能相互冲突，从而解决幻读问题。简言之，它是在每个读的数据行上加上共享锁。在这个级别，可能导致大量的超时现象和锁竞争。

## 六、实验小结

（一）问题和解决

1、不清楚DML和DDL语句的默认事务操作方式

所有的 DML 语句都是要显式提交的，也就是说要在执行完DML语句之后，执行 COMMIT 。而其他的诸如 DDL 语句的，都是隐式提交的。也就是说，在运行那些非 DML 语句后，数据库已经进行了隐式提交，例如 CREATE TABLE，在运行脚本后，表已经建好了，并不在需要你再进行显式提交。

在提交事务（commit）之前可以用rollback回滚事务。

1. 不清楚rollback后，到底回滚到哪里。

回滚到上一步。但可以用”rollback to （savepoint） point\_name;”回滚到指定节点。

（二）心得

这次实验，我学习了mysql数据库系统中三种事务：显示事务、隐式事务和自动提交事务，并了解了如何使用这三种事务，以及他们之间的区别。此外，我还查看了事务的隔离级别，并通过查阅资料，了解了4种事务隔离级别的区别。

这次实验让我实践了课堂学习到的知识，并真正掌握了他们，让我受益匪浅。