Mysql事务管理

## 事务特性

事务就是一组原子性的SQL查询，或者说一个独立的工作单元。如果[数据库](http://lib.csdn.net/base/mysql" \t "_blank" \o "MySQL知识库)引擎能够成功地对数据应用该组查询的全部语句，那么久执行该组查询。如果其中任何一条语句因为崩溃或其他原因无法执行，那么所有语句都不会执行。也就是说，事务内的语句，要么全部执行成功，要么全部执行失败。

事务的四大特性（ACID）：

1.原子性（atomicity）：一个事务必须视为一个不可分割的最小工作单元，整个事务中的所有操作要么全部提交成功，要么全部失败回滚，对于一个事务来说，不可能只执行其中的一部分操作，这就是事务的原子性。(一荣俱荣、一损俱损)

2.一致性（consistency）：数据库总数从一个一致性的状态转换到另一个一致性的状态。

3.隔离性（isolation）：一个事务所做的修改在最终提交以前，对其他事务是不可见的。

4.持久性（durability）：一旦事务提交，则其所做的修改就会永久保存到数据库中。此时即使系统崩溃，修改的数据也不会丢失。

## 二、事务处理

### 1、什么是事务处理

事务处理（transaction processing）可以用来维护数据库的完整性，它保证成批的MySQL操作要么完全执行，要么完全不执行 。

这里就需要使用事务处理了。事务处理是一种机制，用来管理必须成批执行的MySQL操作，以保证数据库不包含不完整的操作结果。利用事务处理，可以保证一组操作不会中途停止，它们或者作为整体执行，或者完全不执行（除非明确指示）。如果没有错误发生，整组语句提交给（写到）数据库表。如果发生错误，则进行回退（撤销）以恢复数据库到某个已知且安全的状态。

### 2、事务处理的相关术语

* 事务（transaction）指一组SQL语句；
* 回退（rollback）指撤销指定SQL语句的过程；
* 提交（commit）指将未存储的SQL语句结果写入数据库表；
* 保留点（savepoint）指事务处理中设置的临时占位符（placeholder），你可以对它发布回退（与回退整个事务处理不同）。

## 二、控制事务处理

### 1、使用rollback

说明：ROLLBACK命令用来回退MYSQL语句

实例：

select \* from student;

START TRANSACTION; #开始事务

delete from student;

select \* from student;

ROLLBACK; #回滚事务

select \* from student;

结果：最后一条SELECT语句显示该表不为空。  
结论：  
　　1.ROLLBACK只能在一个事物处理内使用，即在执行一条START TRANSACTION命令之后。  
　　2.事务处理用来管理INSERT、 UPDATE和DELETE语句。你不能回退SELECT语句。

### 2、使用commit

说明：在事务处理块中，提交不会隐含地进行。为进行明确的提交，使用COMMIT语句 ；

实例：

select \* from student;

START TRANSACTION; #开始事务

delete from student;

select \* from student;

COMMIT; #提交事务

select \* from student;

隐含事物关闭：当COMMIT或ROLLBACK语句执行后，事务会自动关闭（将来的更改会隐含提交）。

### 3、使用保留点

说明：简单的ROLLBACK和COMMIT语句就可以撤销或写入整个事务处理。但是，只是对简单的事务处理才能这样做，更复杂的事务处理可能需要部分提交或回退。

实例：

select \* from student;

START TRANSACTION; #开始事务

delete from student where s\_no = 1;

SAVEPOINT del1:

delete from student where s\_no = 2;

select \* from student;

ROLLBACK TO del1; #回滚到保存点

select \* from student;

### 4、更改默认提交行为

默认的MySQL行为是自动提交所有更改。换句话说，任何时候你执行一条MySQL语句，该语句实际上都是针对表执行的，而且所做的更改立即生效。为指示MySQL不自动提交更改，需要使用以下语句：

Mysql中默认不受此参数的影响，jdbc中可以配置此参数：

set autocommit = 0; #不自动提交

set autocommit = 1; #自动提交

实例：

select \* from student;

set autocommit = 0; #不自动提交

START TRANSACTION; #开始事务

delete from student where s\_no = 1;

delete from student where s\_no = 2;

select \* from student;

COMMIT; #提交事务

select \* from student;