<http://blog.csdn.net/jjavaboy/article/details/45228939>

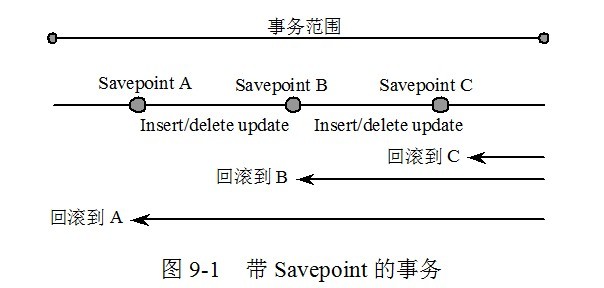
转自：http://my.oschina.net/mushui/blog/161452

并不是所有的数据库都支持事务，即使支持事务的数据库也并非支持所有的事务隔离级别，用户可以通过Connection的getMetaData()方法获取DatabaseMetaData对象，并通过该对象的supportsTransactions()、supportsTransactionIsolationLevel(int level)方法查看底层数据库的事务支持情况。    
  
Connection默认情况下是自动提交的，也即每条执行的SQL都对应一个事务，为了能够将多条SQL当成一个事务执行，必须先通过Connection的setAutoCommit(false)阻止Connection自动提交，并可通过Connection的setTransactionIsolation()设置事务的隔离级别，Connection中定义了对应SQL 92标准4个事务隔离级别的常量。通过Connection的commit()提交事务，通过Connection的rollback()回滚事务。下面是典型的JDBC事务数据操作的代码：

**[java]** [view plain](http://blog.csdn.net/jjavaboy/article/details/45228939) [copy](http://blog.csdn.net/jjavaboy/article/details/45228939)

1. Connection conn ;
2. **try**{
3. conn ＝ DriverManager.getConnection();//①获取数据连接
4. conn.setAutoCommit(**false**); //②关闭自动提交的机制
5. //③设置事务隔离级别
6. conn.setTransactionIsolation(Connection.TRANSACTION\_SERIALIZABLE);
7. Statement stmt = conn.createStatement();
9. **int** rows = stmt.executeUpdate( "INSERT INTO t\_topic ALUES(1,’tom’) " );
10. rows = stmt.executeUpdate( "UPDATE t\_user set topic\_nums = topic\_nums +1 "+  "WHERE user\_id = 1");
12. conn.commit();//④提交事务
13. }**catch**(Exception e){
14. …
15. conn.rollback();//⑤回滚事务
16. }**finally**{
17. …
18. }

在JDBC 2.0中，事务最终只能有两个操作：提交和回滚。但是，有些应用可能需要对事务进行更多的控制，而不是简单地提交或回滚。JDBC 3.0（JDK 1.4及以后的版本）引入了一个全新的保存点特性，Savepoint 接口允许用户将事务分割为多个阶段，用户可以指定回滚到事务的特定保存点，而并非像JDBC 2.0一样只回滚到开始事务的点，如下图所示



下面的代码使用了保存点的功能，在发生特定问题时，回滚到指定的保存点，而非回滚整个事务:

**[java]** [view plain](http://blog.csdn.net/jjavaboy/article/details/45228939) [copy](http://blog.csdn.net/jjavaboy/article/details/45228939)

1. Statement stmt = conn.createStatement();
2. **int** rows = stmt.executeUpdate( "INSERT INTO t\_topic VALUES(1,’tom’)");
4. Savepoint svpt = conn.setSavepoint("savePoint1");//①设置一个保存点
5. rows = stmt.executeUpdate( "UPDATE t\_user set topic\_nums = topic\_nums +1 "+  "WHERE user\_id = 1");
6. …
7. //②回滚到①处的savePoint1，①之前的SQL操作，在整个事务提交后依然提交，
8. //但①到②之间的SQL操作被撤销了
9. conn.rollback(svpt);
10. …
11. conn.commit();//③提交事务

并非所有数据库都支持保存点功能，用户可以通过DatabaseMetaData的supportsSavepoints()方法查看是否支持。 

# 事务类型

数据库事务类型有本地事务和分布式事务：

* 本地事务：就是普通事务，能保证单台数据库上的操作的ACID，被限定在一台数据库上；
* 分布式事务：涉及两个或多个数据库源的事务，即跨越多台同类或异类数据库的事务（由每台数据库的本地事务组成的），分布式事务旨在保证这些本地事务的所有操作的ACID，使事务可以跨越多台数据库；

Java事务类型有JDBC事务和JTA事务：

* JDBC事务：就是数据库事务类型中的本地事务，通过Connection对象的控制来管理事务；
* JTA事务：JTA指Java事务API(Java Transaction API)，是Java EE数据库事务规范， JTA只提供了事务管理接口，由应用程序服务器厂商（如WebSphere Application Server）提供实现，JTA事务比JDBC更强大，支持分布式事务。

Java EE事务类型有本地事务和全局事务：

* 本地事务：使用JDBC编程实现事务；
* 全局事务：由应用程序服务器提供，使用JTA事务；

按是否通过编程实现事务有声明式事务和编程式事务； 

* 声明式事务： 通过注解或XML配置文件指定事务信息；
* 编程式事务：通过编写代码实现事务。