数据库事务ACID

# 说明

1. 原子性（Atomicity）

 事务中的全部操作在数据库中是不可分割的，要么全部完成，要么均不执行。

1. 一致性（Consistency）

几个并行执行的事务，其执行结果必须与按某一顺序串行执行的结果相一致。

1. 隔离性（Isolation）

 事务的执行不受其他事务的干扰，事务执行的中间结果对其他事务必须是透明的。

1. 持久性（Durability）

 对于任意已提交事务，系统必须保证该事务对数据库的改变不被丢失，即使数据库出现故障。

# JDBC对事务的支持

1. 在jdbc中，事务操作缺省是自动提交。也就是说，一条对数据库的更新表达式代表一项事务操作，操作成功后，系统将自动调用commit()来提交，否则将调用rollback()来回滚。
2. 在jdbc中，可以通过调用setAutoCommit(false)来禁止自动提交。之后就可以把多个数据库操作的表达式作为一个事务，在操作完成后调用commit()来进行整体提交，倘若其中一个表达式操作失败，都不会执行到commit()，并且将产生响应的异常；此时就可以在异常捕获时调用rollback()进行回滚。这样做可以保持多次更新操作后，相关数据的一致性，

# 项目中实例

**private** Connection createConnection() **throws** Exception { // 以mysql为例　创建数据库连接

String url = "jdbc:mysql://" + server + ":" + port + "/" + dbName + "?useUnicode=true&characterEncoding=utf8";

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

Connection connection = DriverManager.*getConnection*(url, userName, userPasswd);

connection.setAutoCommit(**false**); // 设置事务不自动提交

ConnectionProxy handler = **new** ConnectionProxy(connection); // 创建代理类

Connection proxy = (Connection) Proxy.*newProxyInstance*( // 创建代理

connection.getClass().getClassLoader(), **new** Class[]{Connection.**class**}, handler);

**return** proxy;

}