

# 事务

---

真正管理事务的对象

org.springframework.jdbc.datasource.DataSourceTransactionManager                      使用

SpringJDBC 或 或 iBatis    进行持久化数据时使用

org.springframework.orm.hibernate5.HibernateTransactionManager                      使用

Hibernate

## 一、事务介绍：

### 1、例子：

你去ATM机取1000块钱，大体有两个步骤：首先输入密码金额，银行卡扣掉1000元钱；然后ATM出1000元钱。这两个步骤必须是要么都执行要么都不执行。

如果银行卡扣除了1000块但是ATM出钱失败的话，你将会损失1000元；

如果银行卡扣钱失败但是ATM却出了1000块，那么银行将损失1000元。

所以，如果一个步骤成功另一个步骤失败对双方都不是好事，如果不管哪一个步骤失败了以后，整个取钱过程都能回滚，也就是完全取消所有操作的话，这对双方都是极好的。

### 2、事务概念

事务是逻辑上的一组操作，要么都执行，要么都不执行。

事务是一系列的动作，一旦其中有一个动作出现错误，必须全部回滚，系统将事务中对数据库的所有已完成的操作全部撤消，滚回到事务开始的状态，避免出现由于数据不一致而导致的接下来一系列的错误。事务的出现是为了确保数据的完整性和一致性，在目前企业级应用开发中，事务管理是必不可少的。

获取连接 Connection con = DriverManager.getConnection()

开启事务con.setAutoCommit(true/false);

执行CRUD

提交事务/回滚事务 con.commit() / con.rollback();

关闭连接 conn.close();

## 二、四大特性 (ACID)

- 原子性：事务是最小的执行单位，不允许分割。事务的原子性确保动作要么全部完成，要么完全不起作用；
- 一致性：执行事务前后，数据保持一致，例如转账业务中，无论事务是否成功，转账者和收款人的总额应该是不变的；
- 隔离性：并发访问数据库时，一个用户的事务不被其他事务所干扰，各并发事务之间数据库是独立的；
- 持久性：一个事务被提交之后。它对数据库中数据的改变是持久的，即使数据库发生故障也不应该对其有任何影响。

### 三、事务并发所可能存在的问题：

#### 1.脏读 (Dirty read)

一个事务读到另一个事务未提交的更新数据。也就是说，比如事务A的未提交（还依然缓存）的数据被事务B读走，如果事务A失败回滚，会导致事务B所读取的数据是错误的。

#### 2.不可重复读 (Unrepeatableread)

指在一个事务内多次读同一数据。在这个事务还没有结束时，另一个事务也访问该数据。那么，在第一个事务中的两次读数据之间，由于第二个事务的修改导致第一个事务两次读取的数据可能不太一样。这就发生了在一个事务内两次读到的数据是不一样的情况，因此称为不可重复读。

在第一读的时候，total是100，然后事务B就把total的数据改成 200，事务A再读一次，结果就发现，total竟然就变成200了，造成事务A数据混乱。

#### 3.幻读 (Phantom read)

一个事务执行两次查询，但第二次查询比第一次查询多出了一些数据行。

它发生在一个事务（T1）读取了几行数据，接着另一个并发事务（T2）插入了一些数据时。在随后的查询中，第一个事务（T1）就会发现多了一些原本不存在的记录，就好像发生了幻觉一样，所以称为幻读。

#### 4.丢失更新 (Lost to modify)

指在一个事务读取一个数据时，另外一个事务也访问了该数据，那么在第一个事务中修改了这个数据后，第二个事务也修改了这个数据。这样第一个事务内的修改结果就被丢失，因此称为丢失修改。

不可重复度和幻读区别：

不可重复读的重点是修改，幻读的重点在于新增或者删除。（读错的原因是数据的）

## 四、事务隔离级别

### 1、SQL 标准定义了四个隔离级别：

- READ-UNCOMMITTED(读取未提交)：最低的隔离级别，允许读取尚未提交的数据变更，可能会导致脏读、幻读或不可重复读。
- READ-COMMITTED(读取已提交)：允许读取并发事务已经提交的数据，可以阻止脏读，但是幻读或不可重复读仍有可能发生。
- REPEATABLE-READ(可重复读)：对同一字段的多次读取结果都是一致的，除非数据是被本身事务自己所修改，可以阻止脏读和不可重复读，但幻读仍有可能发生。
- SERIALIZABLE(可串行化)：最高的隔离级别，完全服从ACID的隔离级别。所有的事务依次逐个执行，这样事务之间就完全不可能产生干扰，也就是说，该级别可以防止脏读、不可重复读以及幻读。

### 2、图解

隔离级别	脏读	不可重复读
READ-UNCOMMITTED	√	√
READ-COMMITTED	×	√
REPEATABLE-READ	×	×
SERIALIZABLE	×	×

