

事务

教学导航

教学目标	案例-完成转账
教学方法	

一、事务概述

1. 什么是事务

一件事情有 n 个组成单元 要不这 n 个组成单元同时成功 要不 n 个单元就同时失败
就是将 n 个组成单元放到一个事务中

2. mysql 的事务

默认的事务：一条 sql 语句就是一个事务 默认就开启事务并提交事务

手动事务：

- 1) 显示的开启一个事务：start transaction
- 2) 事务提交：commit 代表从开启事务到事务提交 中间的所有的 sql 都认为有效
真正的更新数据库
- 3) 事务的回滚：rollback 代表事务的回滚 从开启事务到事务回滚 中间的所有的
sql 操作都认为无效数据库没有被更新

二、JDBC 事务操作

默认是自动事务：

执行 sql 语句：executeUpdate() ---- 每执行一次 executeUpdate 方法 代表
事务自动提交

通过 jdbc 的 API 手动事务：

开启事务：conn.setAutoCommit(false);

提交事务：conn.commit();

回滚事务：conn.rollback();

注意：控制事务的 connection 必须是同一个
执行 sql 的 connection 与开启事务的 connection 必须是同一个才能对
事务进行控制

三、DBUtils 事务操作

1. QueryRunner

有参构造：QueryRunner runner = new QueryRunner(DataSource dataSource);
有参构造将数据源(连接池)作为参数传入 QueryRunner，QueryRunner 会从连接池中获得一个数据库连接资源操作数据库，所以直接使用无 Connection 参数的 update 方法即可操作数据库

无参构造：QueryRunner runner = new QueryRunner();

无参的构造没有将数据源（连接池）作为参数传入 QueryRunner，那么我们在使
用 QueryRunner 对象操作数据库时要使用有 Connection 参数的方法

四、使用 ThreadLocal 绑定连接资源

五、事务的特性和隔离级别（概念性问题---面试）

1. 事务的特性 ACID

- + 1) 原子性 (Atomicity) 原子性是指事务是一个不可分割的工作单位，事务中的操作要么都发生，要么都不发生。
- + 2) 一致性 (Consistency) 一个事务中，事务前后数据的完整性必须保持一致。
- + 3) 隔离性 (Isolation) 多个事务，事务的隔离性是指多个用户并发访问数据库时，一个用户的事务不能被其它用户的事务所干扰，多个并发事务之间数据要相互隔离。
- + 4) 持久性 (Durability) 持久性是指一个事务一旦被提交，它对数据库中数据的改变

就是永久性的，接下来即使数据库发生故障也不应该对其有任何影响。

2. 并发访问问题----由隔离性引起

如果不考虑隔离性，事务存在 3 中并发访问问题。

- 1) 脏读：B 事务读取到了 A 事务尚未提交的数据 ----- 要求 B 事务要读取 A 事务提交的数据
- 2) 不可重复读：一个事务中 两次读取的数据的内容不一致 ----- 要求的是一个事务中多次读取时数据是一致的 --- update
- 3) 幻读/虚读：一个事务中 两次读取的数据的数量不一致 ----- 要求在一个事务多次读取的数据的数量是一致的 --insert delete

3. 事务的隔离级别

- 1) read uncommitted：读取尚未提交的数据：哪个问题都不能解决
- 2) read committed：读取已经提交的数据：可以解决脏读 ---- oracle 默认的
- 3) repeatable read：重读读取：可以解决脏读 和 不可重复读 ---mysql 默认的
- 4) serializable：串行化：可以解决 脏读 不可重复读 和 虚读---相当于锁表

注意：mysql 数据库默认的隔离级别

查看 mysql 数据库默认的隔离级别：select @@tx_isolation

```
mysql> select @@tx_isolation;
+-----+
| @@tx_isolation |
+-----+
| REPEATABLE-READ |
+-----+
1 row in set (0.00 sec)
```

设置 mysql 的隔离级别：set session transaction isolation level 设置事务隔离级别

```
mysql> set session transaction isolation level read uncommitted;
Query OK, 0 rows affected (0.00 sec)

mysql> select @@tx_isolation;
+-----+
| @@tx_isolation |
+-----+
| READ-UNCOMMITTED |
+-----+
1 row in set (0.00 sec)
```

总结：

mysql 的事务控制：

开启事务：start transaction;

提交：commit；

回滚：rollback；

JDBC 事务控制：

开启事务：conn.setAutoCommit(false);

提交：conn.commit()；

回滚：conn.rollback()；

DBUtils 的事务控制 也是通过 jdbc

ThreadLocal：实现的是通过线程绑定的方式传递参数

概念：

事务的特性 ACID

并发问题：脏读、不可重读、虚读\幻读

解决并发：设置隔离级别

read uncommitted

read committed

repeatable read （mysql 默认）

serializable

隔离级别的性能：

read uncommitted > read committed > repeatable read > serializable

安全性：

read uncommitted < read committed < repeatable read < serializable