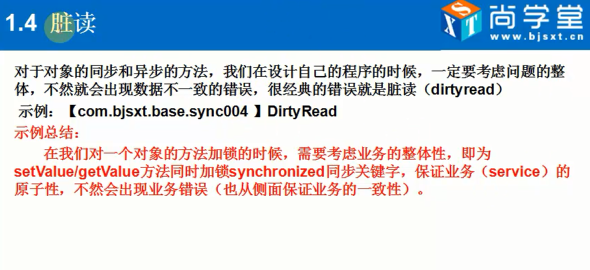




在静态方法上加锁。就是对class 对象类级别的锁





数据库 ACID

**ACID特性**

数据库管理系统中**事务(transaction)**的四个特性（分析时根据首字母缩写依次解释）：**原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability）**

所谓事务，它是一个**操作序列，这些操作要么都执行，要么都不执行，它是一个不可分割的工作单位**。（执行单个逻辑功能的一组指令或操作称为事务）

**详解**

**1. 原子性**

原子性是指事务是一个**不可再分割的工作单元**，事务中的操作要么都发生，要么都不发生。

可采用“**A向B转账**”这个例子来说明解释

在DBMS中，默认情况下**一条SQL就是一个单独事务**，事务是**自动提交**的。只有显式的使用**start transaction**开启一个事务，才能将一个代码块放在事务中执行。

**2. 一致性**

一致性是指在**事务开始之前和事务结束以后**，**数据库的完整性约束没有被破坏**。这是说数据库事务不能破坏**关系数据的完整性**以及**业务逻辑上的一致性**。

如A给B转账，不论转账的事务操作是否成功，其两者的存款总额不变（这是业务逻辑的一致性，至于数据库关系约束的完整性就更好理解了）。

保障机制（也从两方面着手）：**数据库层面**会在一个事务执行之前和之后，数据会符合你设置的**约束**（**唯一约束，外键约束,check约束**等)和触发器设置；此外，数据库的内部数据结构（如 B 树索引或双向链表）都必须是正确的。业务的一致性一般由开发人员进行保证，亦可转移至数据库层面。

**3. 隔离性**

**多个事务并发访问时，事务之间是隔离的**，一个事务不应该影响其它事务运行效果。

在并发环境中，当**不同的事务同时操纵相同的数据**时，每个事务都有**各自的完整数据空间**。由并发事务所做的修改必须与任何其他并发事务所做的修改隔离。事务查看数据更新时，数据所处的状态要么是另一事务修改它之前的状态，要么是另一事务修改它之后的状态，**事务不会查看到中间状态的数据**。

事务最复杂问题都是由事务隔离性引起的。完全的隔离性是不现实的，完全的隔离性要求数据库同一时间只执行一条事务，这样会严重影响性能。

关于隔离性中的事务隔离等级（事务之间影响），参见相应博文

**4. 持久性**

这是最好理解的一个特性：持久性，意味着在事务完成以后，**该事务所对数据库所作的更改便持久的保存在数据库之中，并不会被回滚。**（完成的事务是**系统永久的部分**，对系统的影响是永久性的，该修改即使出现致命的系统故障也将一直保持）

**write ahead logging**：SQL Server中使用了WAL（Write-Ahead Logging）技术来保证事务日志的ACID特性，在数据写入到数据库之前，先写入到日志，再将日志记录变更到存储器中。