**什么是数据库连接池**

当系统使用JDBC技术访问数据库时会创建一个connection对象，而该对象的创建过程是非常消耗资源的，并且创建对象的时间也比较长，假设系统一天有1万次的访问量，那么一天就会创建1万connection对象，这极大的浪费数据库的资源，而且可能造成数据库服务器内存溢出、宕机。

为了解决上面的问题，就引入了数据库连接池（Connection pooling），它主要用来分配、管理、释放数据库的连接。数据库连接池首先会创建若干（该数量可以配置）个connection对象并将这些对象放入到池中，当系统需要使用connection对象时，数据库连接池会从池中分配一个事先创建好的connection对象给系统，当系统使用完毕或超时后，数据库连接池会将该connection对象重新放入到池中。这样就减少了创建connection对象所耗费的资源和时间，可以提高对数据库操作的性能。

**数据库连接池技术带来的优势：**

**资源重用**

由于数据库连接得到重用，避免了频繁创建、释放连接引起的大量性能开销。在减少系统消耗的基础上，另一方面也增进了系统运行环境的平稳性（减少内存碎片以及数据库临时进程/线程的数量）。

**更快的系统响应速度**

数据库连接池在初始化过程中，往往已经创建了若干数据库连接置于池中备用。此时连接的初始化工作均已完成。对于业务请求处理而言，直接利用现有可用连接，避免了数据库连接初始化和释放过程的时间开销，从而缩减了系统整体响应时间。

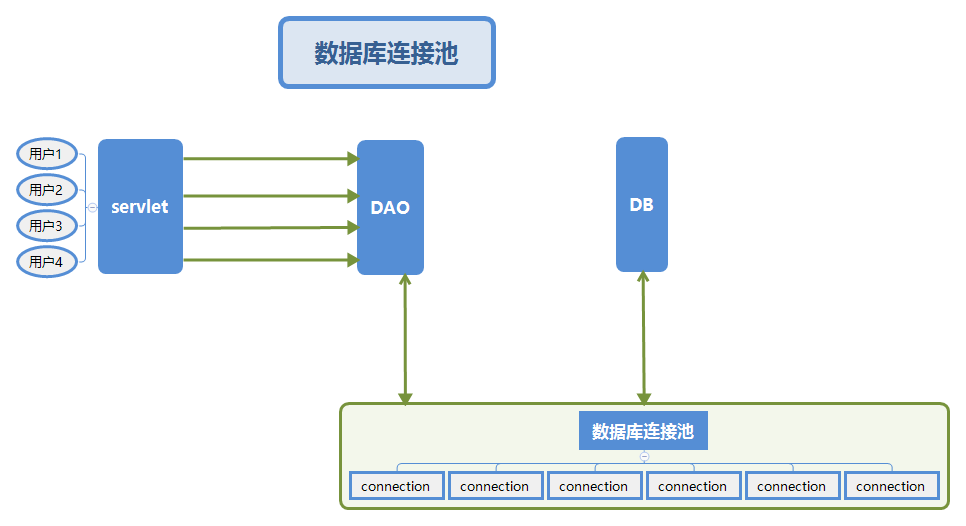
**新的资源分配手段**

对于多应用共享同一数据库的系统而言，可在应用层通过数据库连接的配置，实现数据库连接池技术。某一应用最大可用数据库连接数的限制，避免某一应用独占所有数据库资源。

**统一的连接管理，避免数据库连接泄漏**

在较为完备的数据库连接池实现中，可根据预先的连接占用超时设定，强制收回被占用连接。从而避免了常规数据库连接操作中可能出现的资源泄漏。一个最小化的数据库连接池实现.

连接池类是对某一数据库所有连接的“缓冲池”，主要实现以下功能：①从连接池获取或创建可用连接；②使用完毕之后，把连接返还给连接池；③在系统关闭前，断开所有连接并释放连接占用的系统资源；④还能够处理无效连接（原来登记为可用的连接，由于某种原因不再可用，如超时，通讯问题），并能够限制连接池中的连接总数不低于某个预定值和不超过某个预定值。



**数据库连接池的最小连接数和最大连接数的设置要考虑到以下几个因素**

**最小连接数**：是连接池一直保持的数据库连接,所以如果应用程序对数据库连接的使用量不大,将会有大量的数据库连接资源被浪费.

**最大连接数**：是连接池能申请的最大连接数,如果数据库连接请求超过次数,后面的数据库连接请求将被加入到等待队列中,这会影响以后的数据库操作

**如果最小连接数与最大连接数相差很大**：那么最先连接请求将会获利,之后超过最小连接数量的连接请求等价于建立一个新的数据库连接.不过,这些大于最小连接数的数据库连接在使用完不会马上被释放,他将被放到连接池中等待重复使用或是空间超时后被释放.

**C3P0**

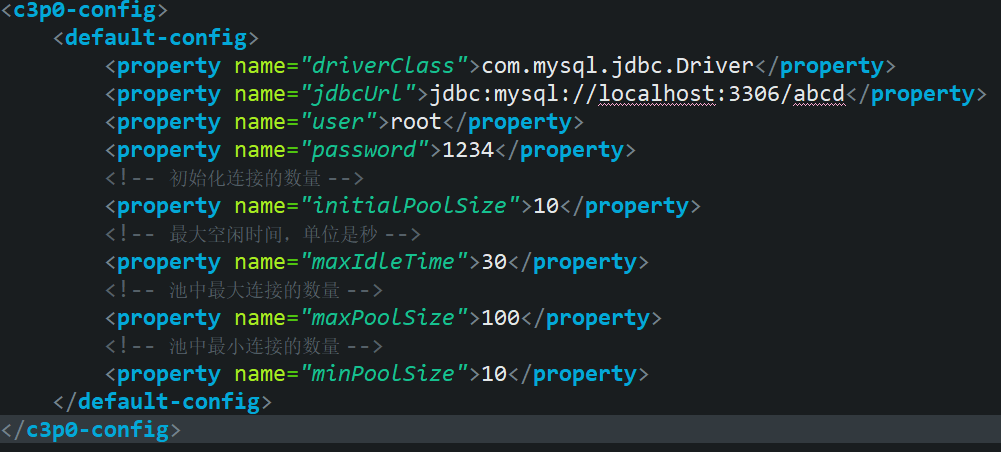
**导入jar包**

c3p0-0.9.5.2.jar

mchange-commons-java-0.2.11.jar

**在项目的src目录下创建c3p0-config.xml文件，也可以创建一个c3p0.properties文件，这两个都支持。**

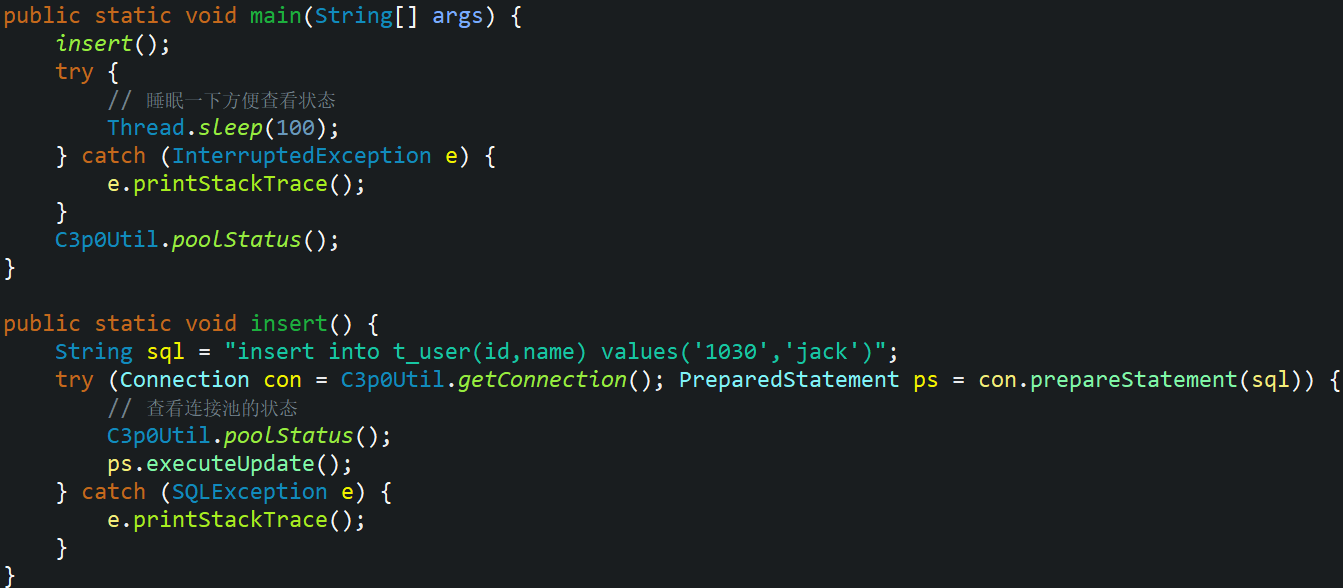
**c3p0-config.xml**



**C3p0Util.java**

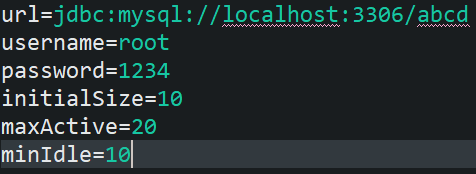
****

**C3p0Test.java**

****

**Druid**

**dbconfig.properties**

****

**DruidUtil.java**

****

**Test.java**

****