事务这个概念呢，在学数据库原理的时候就接触过了，简单的来说就是将一组操作封装起来，要么全部执行，要么全部不执行，举个简单的例子，一个银行卡到另一个银行卡的转账，一个银行卡扣钱了另一个银行卡必须加钱，不能说只执行了一方，这样考虑实际的话要么银行亏损，要么客户亏损，这都是不允许发生的。

事务的4大特性：原子性，一致性，隔离性，持续性

事务的配置

```
<environment id="mysql">
    <!-- 指定事务管理类型，type="JDBC"直接简单使用JDBC的提交和回滚设置 -->
    <transactionManager type="JDBC" />
    <!-- database指数据源配置，POOLED是JDBC连接对象的数据源连接池的实现 -->
    <dataSource type="POOLED">
        <property name="driver" value="com.mysql.jdbc.Driver" />
        <property name="url" value="jdbc:mysql://localhost:3306/mybatis" />
        <property name="username" value="root" />
        <property name="password" value="123456" />
    </dataSource>
</environment>
```

environment定义了连接某个数据库的信息，transactionManager的type决定了用什么类型的事务管理机制，一般采用如图的JDBC

一、使用JDBC的事务管理机制：即利用Java.sql.Connection对象完成对事务的提交（commit()）、回滚（rollback()）、关闭（close()）等

二、使用MANAGED的事务管理机制：这种机制MyBatis自身不会去实现事务管理，而是让程序的容器如（JBOSS，Weblogic）来实现对事务的管理

注意：如果我们使用MyBatis构建本地程序，即不是WEB程序，若将type设置成"MANAGED"，那么，我们执行的任何update操作，即使我们最后执行了commit操作，数据也不会保留，不会对数据库造成任何影响。因为我们将MyBatis配置成了“MANAGED”，即MyBatis自己不管理事务，而我们又是运行的本地程序，没有事务管理功能，所以对数据库的update操作都是无效的。