Atitit 事务存储点提供了一种机制，

**使用存储点**  
  
事务只是一种最坏情况下的保障措施，事实上，平时系统的运行可靠性都是相当高的，错误很少发生，因此，在每次事务执行之前都检查其有效性显得代价太高——绝大多数的情况下这种耗时的检查是不必要的。我们不得不想另外一种办法来提高效率。  
  
事务存储点提供了一种机制，用于回滚部分事务。因此，我们可以不必在更新之前检查更新的有效性，而是预设一个存储点，在更新之后，如果没有出现错误，就继续执行，否则回滚到更新之前的存储点。存储点的作用就在于此。要注意的是，更新和回滚代价很大，只有在遇到错误的可能性很小，而且预先检查更新的有效性的代价相对很高的情况下，使用存储点才会非常有效。  
  
使用.net框架编程时，你可以非常简单地定义事务存储点和回滚到特定的存储点。下面的语句定义了一个存储点“NoUpdate”：  
  
myTran.Save("NoUpdate");  
  
当你在程序中创建同名的存储点时，新创建的存储点将替代原有的存储点。  
  
在回滚事务时，只需使用Rollback()方法的一个重载函数即可：  
  
　　myTran.Rollback("NoUpdate");  
  
下面这段程序说明了回滚到存储点的方法和时机：

数据库事务原理和实践(ZZ) - releaseyou - 博客园.html