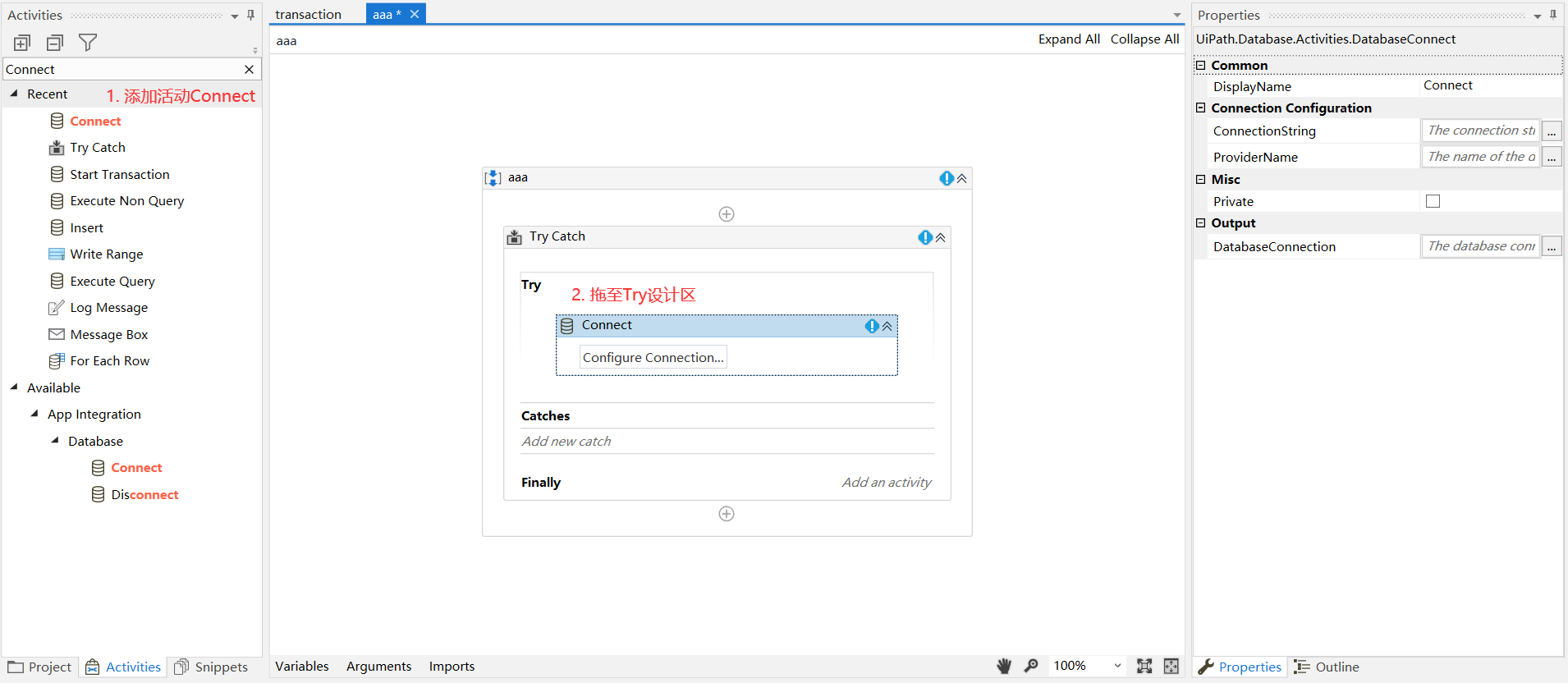
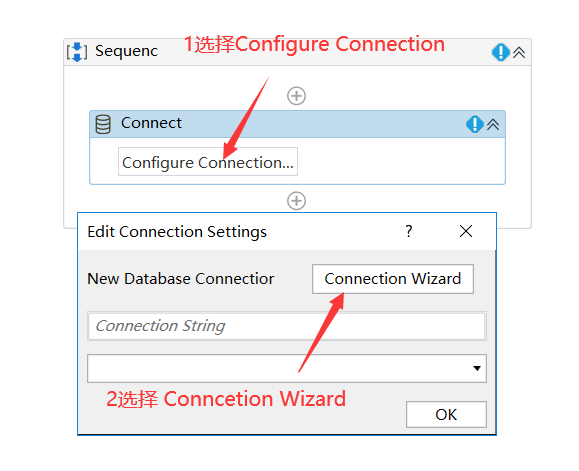
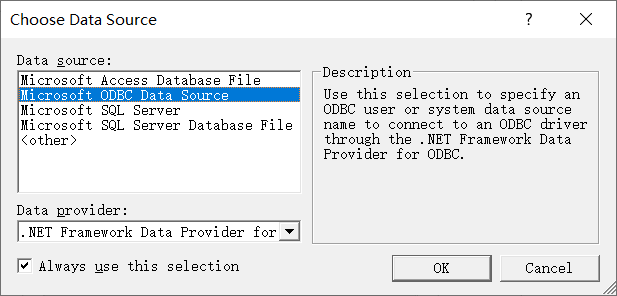
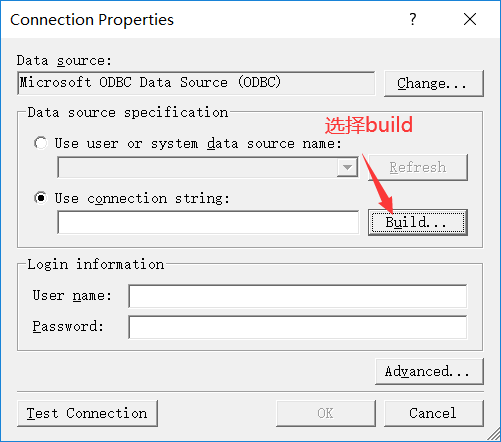
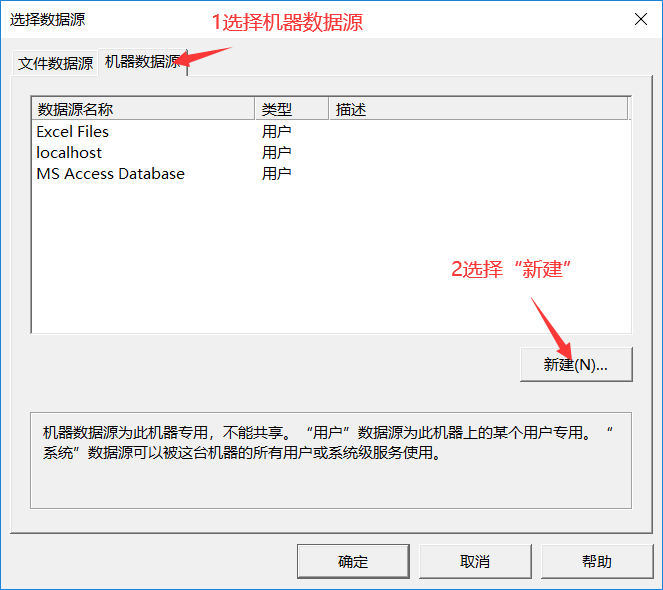
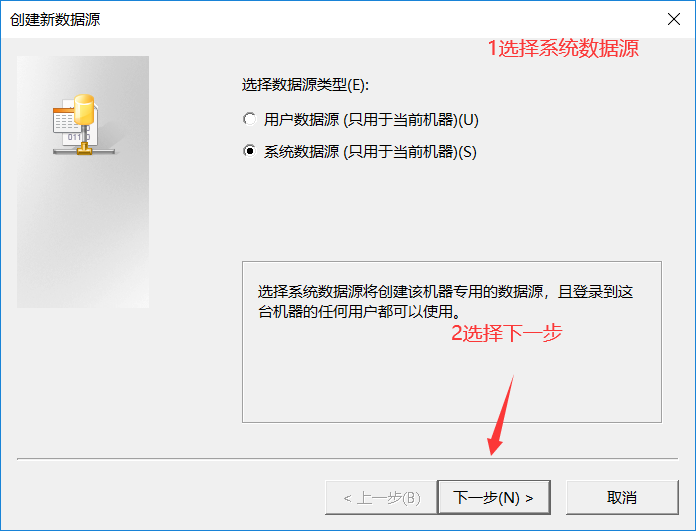
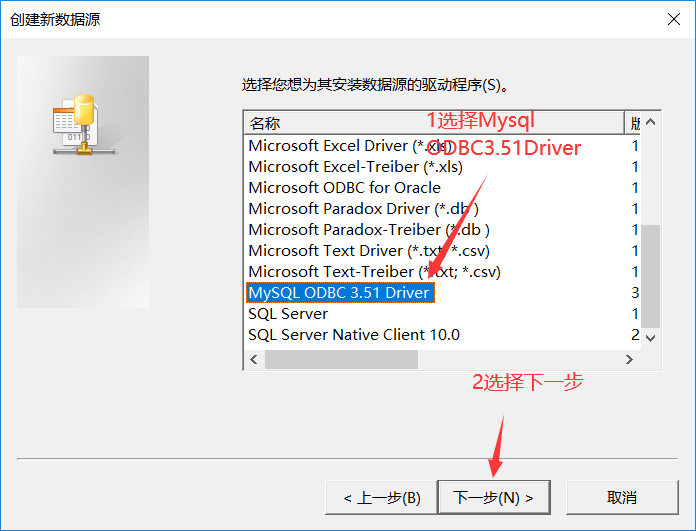
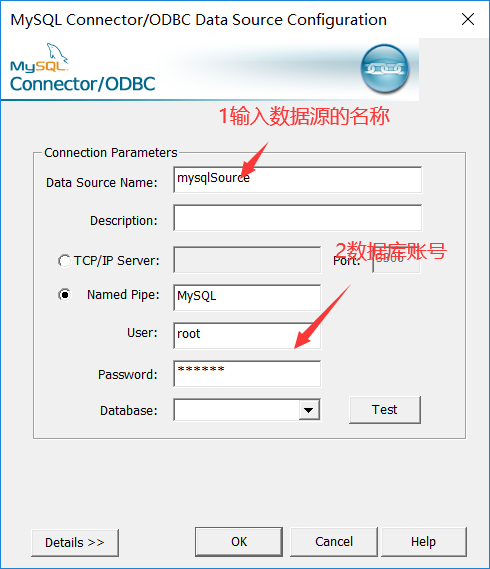
# UiPath 数据库的事务

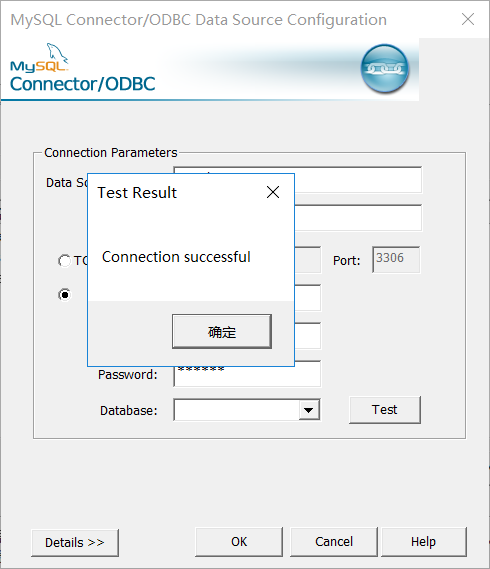
## 1 使用Try Catch

## 

## 2 配置数据库

* 1. 创建一个队列并添加一个Connect, 拖至Try的设计区  
     
  2. 配置Connect  
     
  3. 选择“Microsoft ODBC Data Source”  
     
  4. 配置数据库服务器的连接属性  
     

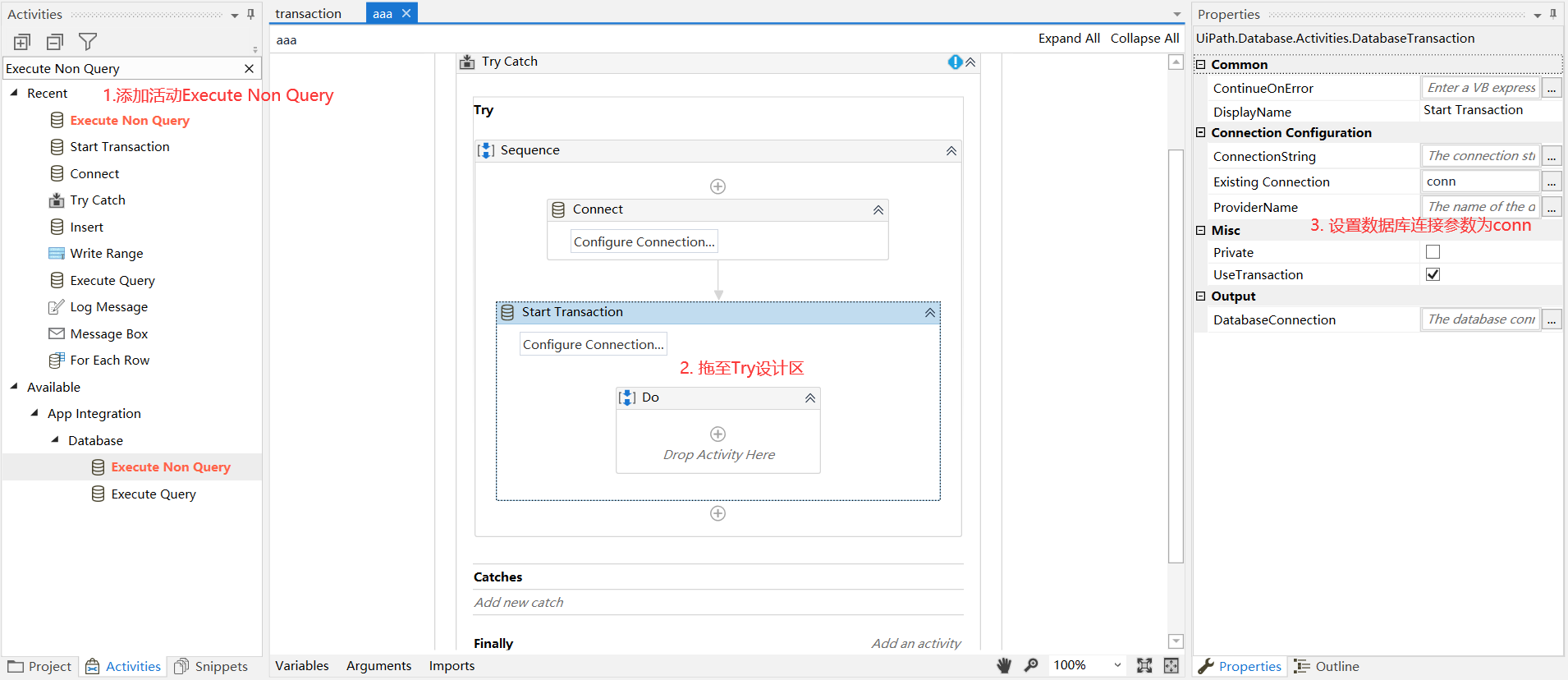
新建数据源  
  
创建新的数据源  
  
选择数据源驱动程序  
  
输入数据库信息  


点击“Test”测试数据库连接  


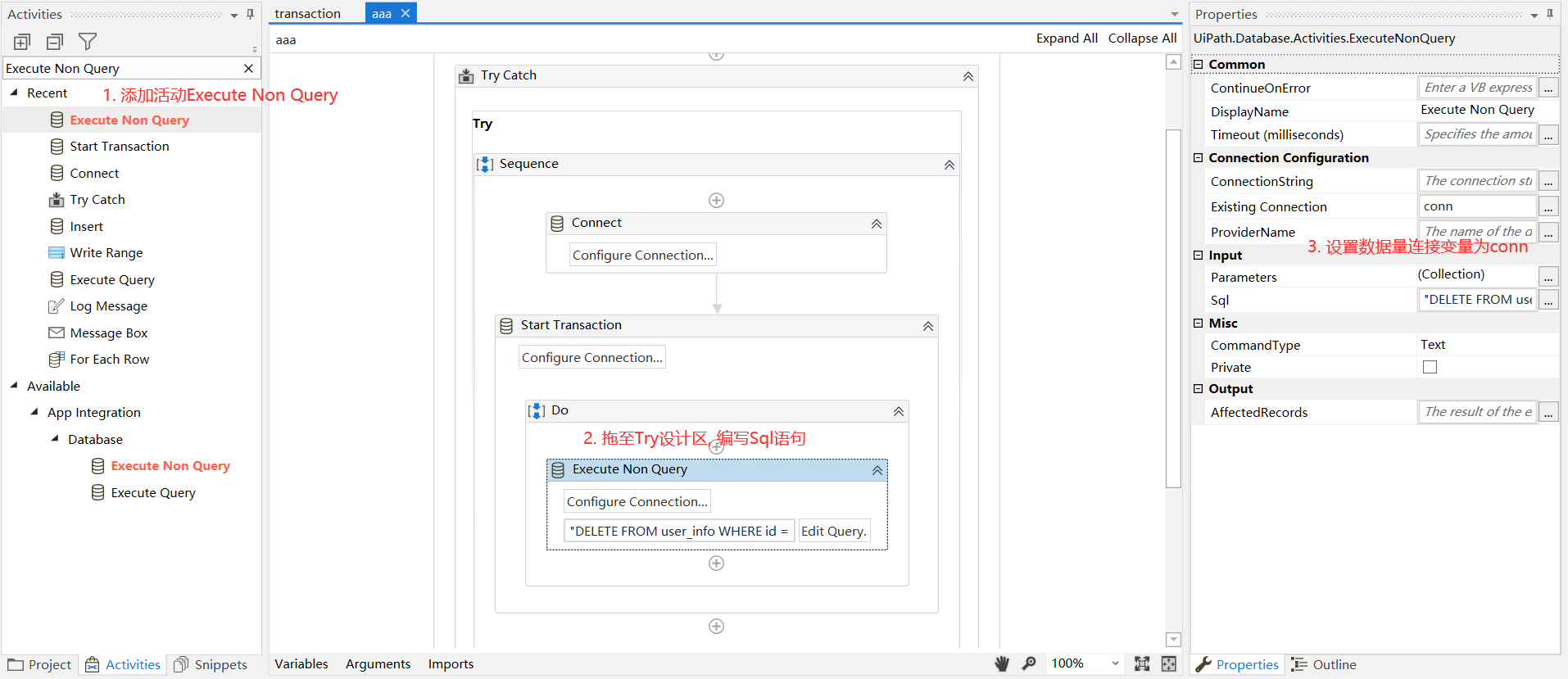
测试成功后，点击“OK“确认

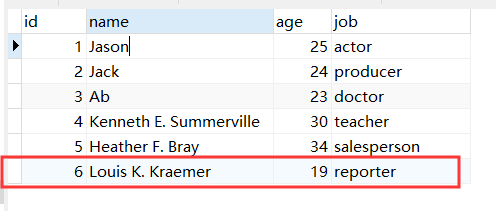
## 3 开启事务

3.1 添加活动“Start Transaction”，并拖至Try的设计区, 设置数据库连接参数为conn

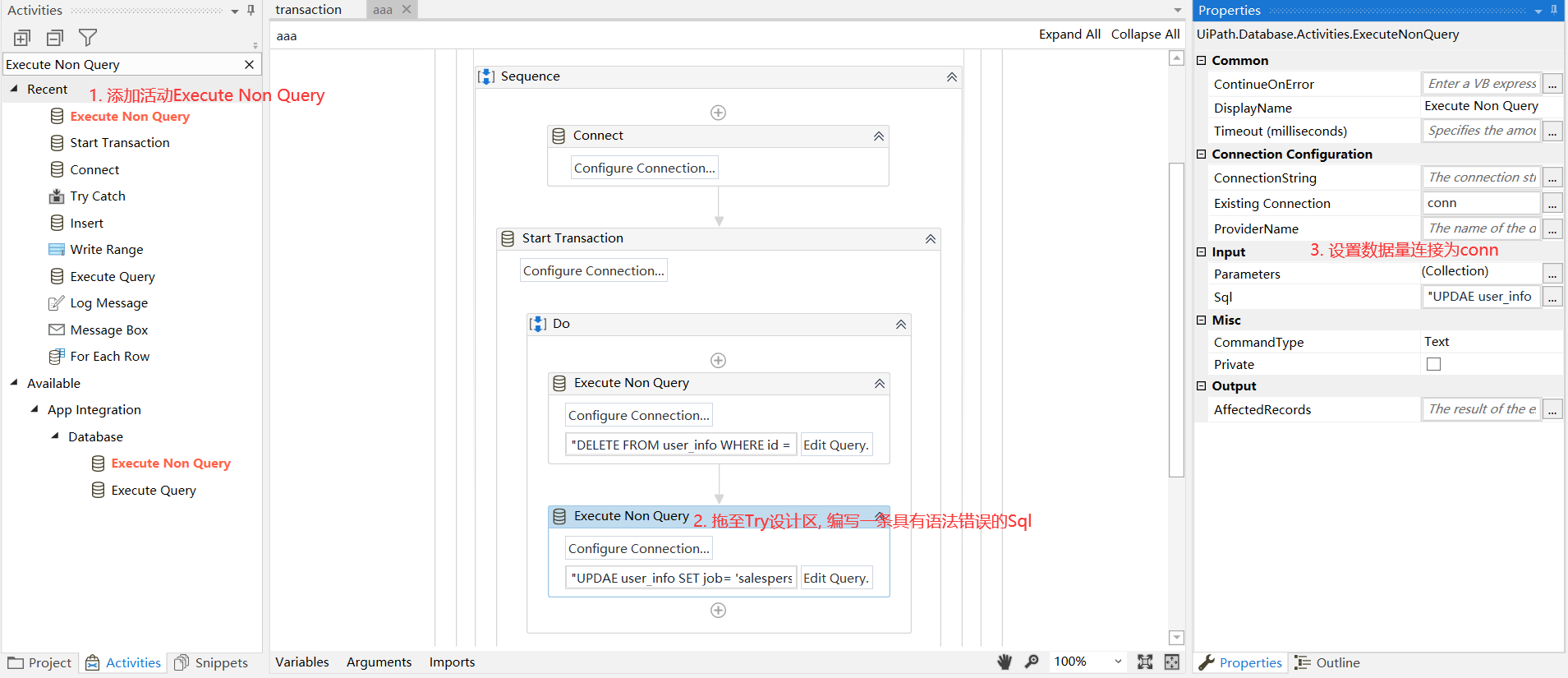


3.2 添加活动“Execute Non Query”，执行一条删除语句(为了方便演示, 假定被删除的记录是存在的), 本案例删除的是id 为6的记录. 设置数据库连接参数为conn

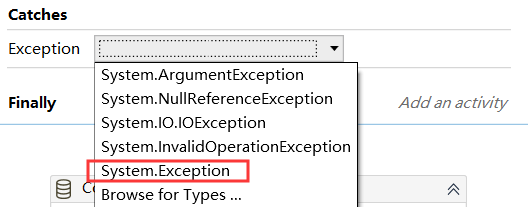


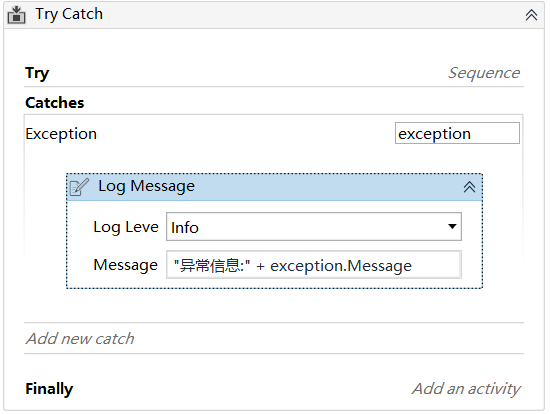


3.3 添加活动“Execute Non Query”，执行一条具有语法错误的Sql语句(为了使流程报错并抛出异常)

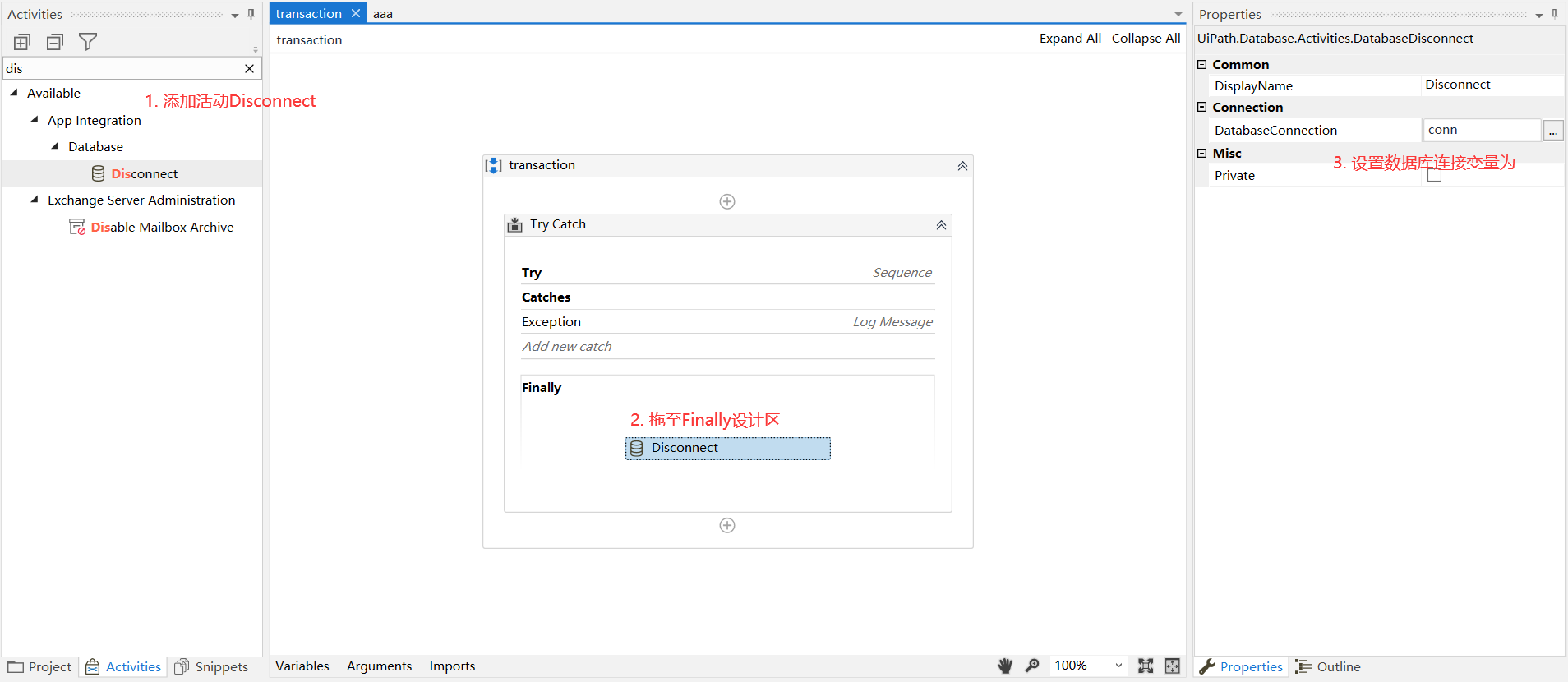


3.4 设置Catches的Exception变量为System.Exception, 表示捕获所有异常类型, 并添加Log Message活动, 打印异常信息



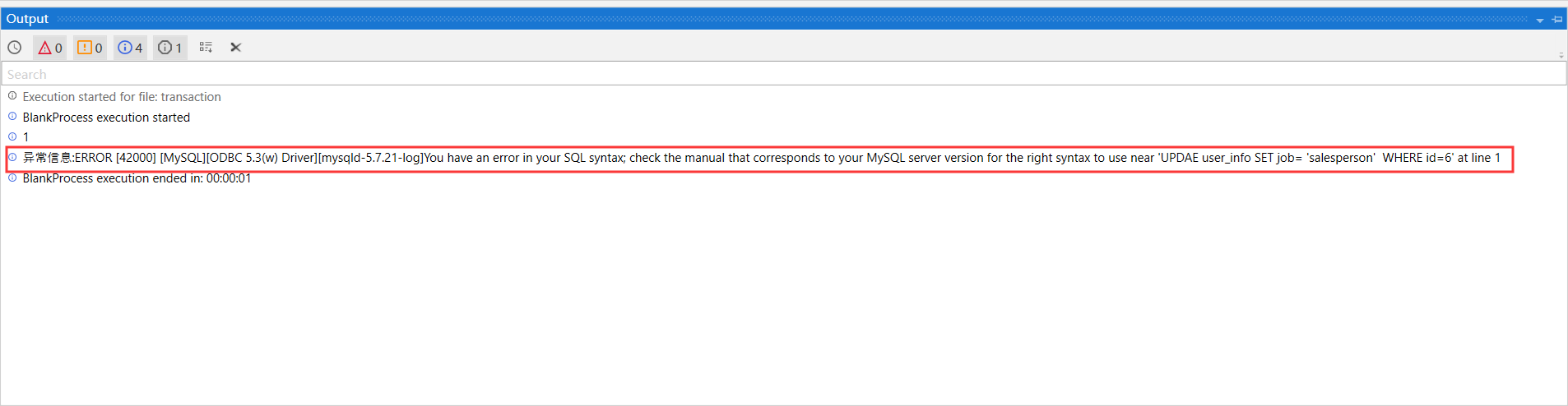


3.5 添加活动”Disconnect”, 拖至Finally设计区, 表示无论以上流程执行成功与否, 都释放数据库连接资源



## 4 事务的回滚

4.1 执行流程, 发现控制台输出了异常信息, 并且id为6的记录并未删除, 因此事务是生效的



4.2 取消Start Transaction活动中UseTransaction的勾选, 再次运行流程, 发现id为6的记录被删除, 因此, 如果未勾选UseTransaction属性, 表示将分别提交所有操作. 是否勾选UseTransaction, 应根据个人需求而定

