

CLR自定义类型存储业务对象 PDF转换可能丢失图片或格式
，建议阅读原文

https://www.100test.com/kao_ti2020/143/2021_2022_CLR_E8_87_AA_E5_AE_9A_E4_c102_143070.htm 问：我可以在SQL Server 2005中用CLR用户自定义类型来存储我的业务对象吗? 答：实现SQL CLR用户自定义类型(UDT)非常简单，就像给.NET类或者结构添加一些额外的片断。其中就包括属性(SqlUserDefinedTypeAttribute)，和接口(INullable)，以及一些额外的方法(Null() and Parse())。这个简单性带来的后果就是，一个有经验的开发人员可以在不到5分钟的时间里把一个业务对象转换为SQL CLR 用户自定义类型。SQL Server 2005的设计目标并不是用于面向对象的数据管理系统。它还是一个标准的SQL 数据库管理系统，并且用户自定义类型的能力也应该被当作是一种系统扩展的类型，而不是一个对象。开发人员在决定是否将现有的业务对象用作CLR UDT的时候，应该仔细权衡他们的选择。每次访问一个类型的实例的方法或者属性的时候，这个实例都应该在这个方法被访问之前串行化。因此，这最好是依靠那些基于他们的串行化字节的可比较的类型。开发人员应该尝试仅仅使用那些可以自动回答问题的用户自定义类型。例如，以下的C#类就不如用户自定义类型：

```
class Product{ public string Name. public string Description. public decimal price.}
```

如果一个查询是针对这样类型的字段，每个行都必须被反串行化以回答如下的问题，“什么产品价值10美元?”这是因为我们不能假设所有的10美元的产品都具有同样的二进制表示。对一个大表(例如一个有上百万产品的表)中的每个行都进行反串行化将会给性能带来严重的考验。

除了性能挑战之外，还有标准化的问题。例如，假设这个类型，一个公司怎么能为同样的产品存储两种描述，并且还要确保产品只有一个有效的价格？最好是坚持使用那些可以回答问题，并且不会带来反串行化负担的类型。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com