

## 1.2大幅提升编程的能力\_SQLServer2005数据库开发详解 PDF

转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/140/2021\\_2022\\_12\\_E5\\_A4\\_A7\\_E5\\_B9\\_85\\_E6\\_c100\\_140351.htm](https://www.100test.com/kao_ti2020/140/2021_2022_12_E5_A4_A7_E5_B9_85_E6_c100_140351.htm)

1.2 大幅提升编程的能力在集成应用程序开发上，SQL Server 2005 最受人瞩目的当属集成 .NET CLR 引擎到 SQL Server 核心，让程序设计师可以通过自己所熟悉的 .NET 语言来开发 SQL Server 内的对象，扩展了程序编写的弹性。另外，支持 ANSI SQL 2003 与 W3C 的 XML 标准，让关系式和 XML 两种最常用的数据处理格式都可以集中到数据库引擎来处理。与 .NET Framework 的集成一致化数据库对象与中间层的程序编写模型。将 .NET 的 Common Language Runtime ( CLR ) 直接集成到 SQL Server 2005 的数据库引擎中，让存储过程、用户自定义函数、触发器、用户自定义数据类型以及聚合函数可以通过 .NET 的语言，如 Visual Basic.NET 或 C# 等等来编写[1]，并且直接与 SQL Server 引擎执行在同一个程序中，以提升运行效率。以往要通过 C 编写扩展存储过程才能扩展 SQL Server 功能，但此种方式困难且容易犯错，往往造成编写的功能不如预期，反倒是增加了系统的不稳定性。如今可以通过 .NET 语言轻易地办到扩展 SQL Server 的功能。例如通过 Regular Expression 验证数据格式正确与否，集成网络上或其他来源的数据一起提供给前端等等。应用系统的架构工程师现在有更大的想象空间，让系统分析员和程序设计员等人可以使用相通的分析设计技巧与工具，如面向对象分析设计、程序编写风格、.NET Framework 类型等来扩展 SQL Server 的功能。Visual Studio 2005 可以直接提供 SQL Server 2005 对象格式的项目程序开发实例，其内提

供程序代码架构、部署设置以及集成调试的 Script 等。当你编写并编译完当作函数库（.dll 格式）的程序集（Assembly）后，可以通过 SQL 语句或是 Visual Studio 2005 提供的功能，直接将其部署到 SQL Server 中。SQL Server 会将该程序集存放在数据库内，而与外部文件系统上的程序集不再有关系。也就是说若你有新的版本，需要先将 SQL Server 内的程序集及其相关的对象删除，重新布署才可以更新。当程序集部署到 SQL Server 后，通过 Create function、Create Procedure 等 T-SQL 语句可以设置存储过程、用户自定义函数等五种对象是来自该程序集。而后就当作一般的 SQL 对象搭配 T-SQL 来使用。前端用户分不出来某个对象是由 T-SQL 还是 .NET 程序语言开发出来的。除了用户自定义数据类型和自定义聚合函数一定要以 .NET 才能编写外，其他的对象用 .NET 和 T-SQL 都可以编写，则何时该用哪种语言架构呢？笔者以为面向数据的函数与程序依然应该采用 T-SQL，也就是说以集合的方式添加、修改、删除和查询数据还是 T-SQL 擅长的部分，若以一般程序语言如 C#/VB.NET 等来逐笔维护数据，反而发挥不了数据库引擎最佳化访问数据的能力。但若复杂的商业逻辑，如加/解密、网络访问、多线程、文件 I/O 等，需要一步步指引 CPU 如何做事，则适合采用 .NET 程序语言。对于复杂的运算，.NET 程序语言采用编译的方式，其编写程序逻辑与执行程序的方式都比解释式的 T-SQL 语言更好。由于是数据库引擎内建 .NET 执行环境，因此若将所有的商业逻辑都搬进到 SQL Server 势必大幅增加服务器的负担，因此仍要慎选程序逻辑的执行位置。一般来说若应用与数据适合结合在一起，例如节省网络来回的负荷，数据与运算需要统一

管理等，才将程序集成到服务器中。在你细心评估后，若觉得放在数据库引擎是最佳的设计，而后才把组件放入到数据库也行，应用的巧妙存乎一心。另外，在编写数据库核心程序需要格外的严谨，毕竟以往若前端程序写得不好，在意外状况发生时只会造成几个人的系统有问题，但若在数据库服务器上执行不稳定，或是有安全疑虑，将会拖累到全部数据库服务器的用户。尤其在大型数据库服务器上，可能还有其他系统同时执行，到时会连累到许多其他系统的用户，因此程序开发者和数据库管理员不可不慎。对 XML 的支持现今应用程序在交换数据或存储设置时，大多采用 XML 格式。在以往我们可能将 XML 数据以文件的方式存放在硬盘目录结构上，然后把该文件相关的管理信息放入到关系式数据库内。在操作 XML 数据时，一方面通过 SQL 查询语句在数据库内找到相关的 XML 文件，一方面用程序语言调用 DOM ( Document Object Model ) 对象来访问解析 XML。当编写这类程序时，开发人员要同时学习 T-SQL 和 XPath 语言，ADO.NET 或其他数据库访问对象以及 DOM 对象类型。若 XML 文件数据很大，由于没有索引结构，用 DOM 解析的效率往往不佳。同时，备份数据既要备份数据库内的 metadata，又要备份文件系统上的 XML，而且安全控制也更加复杂。根据 ANSI SQL 2003 标准的要求，未来关系数据库皆可当作 XML 格式的数据源与集中储存的地方。SQL Server 2005 遵循并扩展该标准，让存放数据的模型可以扩展到 XML 数据格式。在数据定义方面新增 XML 数据类型（每条记录最长 2 GB），你可以用来定义数据字段或是该类型的变量。该类型除了要求符合 XML Well form 的定义外，并可通过 XML Schema

验证 XML 数据的正确性。并通过扩充 XPath 语言而形成的 XQuery 查询语言来查询和修改 XML 数据。以 XML View 将原有的关系式数据转成 XML 层次结构的查看方式，并以专门的 XML 索引格式提升查询效率。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](http://www.100test.com)