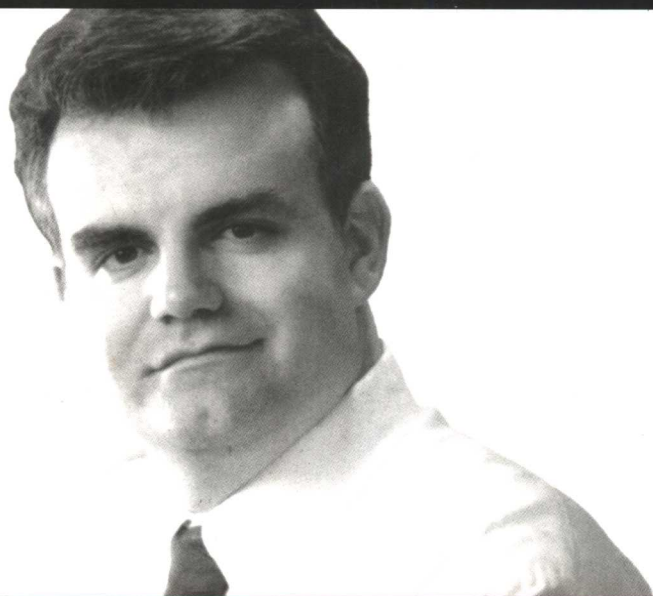




Beginning SQL Server 2005 Programming

SQL Server 2005

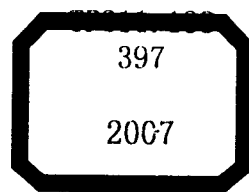
编程入门经典

(第2版)

(美) Robert Vieira 著
叶寒 管贤平 译



清华大学出版社



SQL Server 2005

编程入门经典

(第2版)

(美) Robert Vieira 著

叶寒 管贤平 译

清华大学出版社

北 京

Robert Vieira

Beginning SQL Server 2005 Programming

EISBN: 0-7645-8433-2

Copyright©2006 by John Wiley & Sons, Inc.

All Rights Reserved. This translation published under license

本书中文简体字版由 John Wiley & Sons, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2006-1580

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

SQL Server 2005 编程入门经典(第2版)/(美)维埃拉(Vieira, R.)著; 叶寒, 管贤平 译. —北京: 清华大学出版社, 2007.4

书名原文: Beginning SQL Server 2005 Programming

ISBN 978-7-302-14653-7

I. S… II. ①维… ②叶… ③管… III. 关系数据库—数据库管理系统, SQL Server 2005 IV. TP311.138

中国版本图书馆 CIP 数据核字(2007)第 020713 号

责任编辑: 王 军 于 平

装帧设计: 孔祥丰

责任校对: 成凤进

责任印制: 李红英

出版发行: 清华大学出版社 地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn> 邮 编: 100084

c-service@tup.tsinghua.edu.cn

社 总 机: 010-62770175 邮购热线: 010-62786544

投稿咨询: 010-62772015 客户服务: 010-62776969

印 刷 者: 清华大学印刷厂

装 订 者: 三河市漂源装订厂

经 销: 全国新华书店

开 本: 185 × 260 印 张: 38.75 字 数: 943 千字

版 次: 2007 年 4 月第 1 版 印 次: 2007 年 4 月第 1 次印刷

印 数: 1 ~ 4000

定 价: 69.90 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 020307-01

前 言

写书的经历漫长而奇特！早在 1999 年，当我第一次编写 *Professional SQL Server 7.0 Programming* 一书时，无论书籍或程序开发环境都与今天的情况截然不同。那时还未听说过 .NET，Visual Studio 98 是当时最流行的开发环境，Java 开发环境被广泛使用，而可供选择的开发环境(如 Delphi)也比当今更具竞争力。现在正快速进入所谓的 .NET 时代，数据库管理系统(DBMS，如 SQL Server)的使用逐步递增。

但问题还是存在的。当找到许多有关 SQL Server 的书籍时，所有的书籍都面向数据库管理员。这些书籍的作者花费了大量的时间和精力在普通程序员几乎不关注的内容上。事实上有些无能为力的事情不得不放弃，就如迫切需要解决的全球饥饿或核扩散问题，因为我们根本无力解决，但我们完全能够编写一类面向开发人员的有关 SQL 的新书籍。

当时我写了 *Professional SQL Server 7.0 Programming* 一书，内容包罗万象，适合于各种读者。其结果造就了一本很厚，占用许多人一大块空间的大部头。

SQL Server 2005 是 SQL Server 7.0 以后的第二主修订版本，而当我们计划重新编写这本书时，我们意识到：书实在是太厚了！SQL Server 2005 的新特征在于有太多的内容可轻易编著一本书，因此我们决定将老“高级编程”系列丛书分成“入门经典”和“高级编程”系列丛书。本书为本系列丛书的“入门经典”篇。

本书覆盖了 SQL Server 的主要内容，相信能与我们先前编写的 *Professional SQL Server 7.0 Programming* 一书同样成功。在本套书完成之后，当您希望自己成为一名 SQL Server 2005 高级程序员时，则可以继续学习本套书的“高级编程”篇。

0.1 本书读者对象

本书书名中的“Beginning”容易引起误解。本书的确非常适合初学者，但不要误解，本书也适合中级和高级程序员。本书主要内容覆盖初学者必须了解的内容，但本书的许多内容也适合于中级、甚至是高级用户，他们可在需要时快速浏览并参考相关的内容。

初学者应该从头开始学习。本书主要内容的设计都是以“需要掌握”哪类知识来编排。除了第 16 章、第 17 章和第 18 章之外，本书的每一章都是易于理解的基本内容，这样在遇到 SQL Server 问题时，您可轻易找到相关内容来学习。

对于中级读者，可以跳过本书的前 7 章或 8 章，但仍建议浏览这几章，以弥补这方面知识的漏洞。也可以先浏览一下这几章，然后快速跳过熟悉的内容，而阅读不了解的内容。

高级程序员除了可利用本书作为很好的参考资料之外，还要集中学习第 12 章以及后面的章节。事实上从第 12 章以后的内容更为重要(如新的调试、事务处理、XML、Reporting Services 等)。

0.2 本书内容简介

从本书书名可以看出,本书覆盖的 SQL Server 2005 的内容明确面向程序员。

SQL Server 2005 是发展近 20 年的数据库管理系统的最新版本。它是在对 SQL Server 7.0 版本重新设计的基础上建立的,大大增强了系统的兼容性,并增加了有关 XML、.NET、用户自定义数据类型以及许多特别服务的特征集。本书解决程序员的基本编程需求,而不考虑程序员的编程水平。大量使用 SQL Server 2005 版本是本书的一大亮点,但本书也经常涉及向后兼容性的问题,这可能影响设计或编码时选择哪一种 SQL Server 版本。

0.3 本书内容安排

假如您已经是富有经验的程序员(除了数据库方面),那么从本书的基础部分开始,学习完本书之后,本书会让您的水平有所提高。要学习本书,您必须具有编程的基本知识,比如变量、数据类型以及程序编程。当然您不必曾经使用过数据库查询(尽管假设您已经使用过)。

本书尤其适合于程序员使用。为了简洁起见,本书会掩盖或完全忽略那些面向数据库管理员的内容,而这些内容与程序员编程无关。当然本书也涉及数据库管理的问题,因为这些问题在程序开发过程中需要考虑到,本书将在第 19 章对几个数据库管理相关的问题做一简单的介绍。

本书内容几乎与客户端开发的工具语言无关。通常忽略所使用的语言,如 VB、C#、C++、Java 以及其他语言(本书集中于服务器端),而在涉及时也对不同语言做同等处理。

按照学习进度,本书先安排 SQL 基础对象,再安排与基本查询和连接有关的知识。接下来开始安排数据库的其他对象,并讨论与数据库物理设计有关的内容。然后学习如何使得编写的代码更健壮,包括编写 SQL Server 脚本、存储过程、用户自定义函数与触发器。为了让您的数据库更具合理性和健壮性,本书最后简单安排了数据库管理方面的内容。

0.4 使用本书的条件

使用本书需要安装 SQL Server。本书大量使用 SQL Server 2005 管理工具,而不仅仅是使用了 SQL Server Express,因此推荐安装包含所有模块功能的 SQL Server 版本。本书集中于编程脚本编写,因此即使是 SQL Server Express 用户,也能学习本书的大部分内容。

一套 Visual Studio 对于学习本书也很有用,但是所需要的 Visual Studio 大部分特征都包括在 SQL Server 产品的 Business Intelligence Studio 中。

0.5 用户支持

我们总是想知道您对本书的看法，您喜欢哪些内容，您不喜欢哪些内容，这些信息将有助于我们下一次做得更好。如果您有什么意见和建议，请向 fwkbook@tup.tsinghua.edu.cn 发邮件。但是，请您一定要在您的信中注明本书的书名。

如何下载本书的示例代码

在您登录到 Wrox 站点 <http://www.wrox.com/> 时，只需使用 Search 工具或使用书名列表就可以找到本书。接着在 Code 栏中单击 Download 链接，或单击本书信息页面上的 Download Code 链接，就可以获得本书所有的源代码。另外，您也可以从本书的合作站点 www.tupwk.com.cn/downpage 上下载本书的所有源代码。

从以上站点上下载的文件使用 WinZip 进行了压缩。在把文件保存到硬盘的一个文件夹中时，需要使用解压缩软件(如 WinZip 或 PKUnzip)对该文件解压缩。在解压缩时，代码常常放在各自的章节文件夹中。在开始解压缩过程时，一定要将解压缩软件 WinZip 或 PKUnzip 设置为使用文件夹名。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。请给 wkservice@tup.tsinghua.com.cn 发电子邮件，我们就会检查您的信息，如果是正确的，就把它发送到该书的勘误表页面上，或在本书的后续版本中采用。

要在网站上找到勘误表，可以登录 <http://www.wrox.com/>，通过 Advanced Search 工具或书名列表查找本书，然后在本书的信息页面上，单击 Book Errata 链接。

E-Mail 支持

如果您希望直接就本书的问题向对本书知之甚多的专家咨询，那么，就向 support@wrox.com 发电子邮件，在电子邮件的“主题”(Subject)栏中，加上本书的名称和 ISBN 的最后 4 位号码。典型的电子邮件应该包括下列内容：

- 在“主题”栏加上书的名称、ISBN 的最后 4 位数字(4332)和问题所在的页码。
- 在邮件的正文中加上您的姓名、联系信息和问题。

我们不会发给您垃圾邮件。我们只需要详细的情况以节省您的宝贵时间和我们的时间。当您发送电子邮件时，它会直接链接到以下支持链：

- 客户支持——您的消息会传送到我们的客户支持人员；他们是阅读信息的第一人。他们有常见问题的文件，会迅速回答一般性的问题。他们回答关于本书和网站的一般性问题。
- 编辑支持——更深的问题会转发到负责本书的技术编辑处。他(或)她具有编程或特殊产品的经验，能够回答某个主题的详细技术问题。

- 作者支持——最后，在编辑都不能回答问题的情况下(这种情况很少出现)，这些问题将转发到作者。我们试图保护作者不要从写作中分心，但是，我们也很愿意将特殊的问题转发给他们。所有的 Wrox 作者帮助支持他们的书籍。他们向客户和编辑回复电子邮件，所有的读者都会从中受益。

Wrox 支持过程只能提供直接与已出版的图书相关的问题。对于超出此范围的问题可以通过 <http://p2p.wrox.com>/论坛的团体列表来提供支持。

p2p.wrox.com

P2P 邮件列表是为作者和同行的讨论而设立的。我们在邮件列表、论坛和新闻组中提供“程序员到程序员的支持”(programmer to programmer support)，还包括一对一的电子邮件支持系统。如果把问题发送给 P2P，就可以相信，您的问题不仅仅是由支持专家解答，而且还要提供给我们邮件列表中的许多 Wrox 作者和其他业界专家。在 p2p.wrox.com 上，可以从许多不同的列表中获得帮助，不仅在阅读本书时获得帮助，还可以在开发应用程序时获得帮助。在网站的.NET 类别中，最适合本书的是 `beginning_vb` 和 `vb_dotnet` 列表。

要订阅一个邮件列表，可以遵循下面的步骤：

- (1) 进入 <http://p2p.wrox.com>。
- (2) 从左侧的菜单栏中选择合适的列表。
- (3) 单击想加入的邮件列表。
- (4) 按照指示订阅和填写电子邮件地址和密码。
- (5) 回复接收到的确认电子邮件。
- (6) 使用订阅管理器加入更多的列表，设置自己的邮件设置。

为什么这个系统提供最好的支持

您可加入该邮件列表中，也可以每周分类接收它们。如果您没有时间或设备接收该邮件列表，可以搜索我们的在线文档。垃圾邮件和广告邮件会被删除，您自己的电子邮件地址会被独特的 Lyris 系统保护起来。任何加入或退出列表的查询，或者与列表相关的一般问题，都应发送到 listsupport@p2p.wrox.com。

目 录

第 1 章 RDBMS 基础: SQL Server

数据库构成	1
1.1 数据库对象概述	1
1.1.1 数据库对象	2
1.1.2 事务处理日志	5
1.1.3 最基本的数据库对象: 表	6
1.1.4 文件组	7
1.1.5 关系图	7
1.1.6 视图	8
1.1.7 存储过程	9
1.1.8 用户自定义函数	9
1.1.9 用户和角色	10
1.1.10 规则	10
1.1.11 默认值	10
1.1.12 用户自定义的数据类型	10
1.1.13 全文本目录	10
1.2 SQL Server 数据类型	11
1.3 SQL Server 对象标识符	14
1.3.1 需要命名的对象	15
1.3.2 命名规则	15
1.4 本章小结	16

第 2 章 SQL Server 管理工具

2.1 Books Online	18
2.2 SQL Server Configuration Manager	19
2.2.1 服务管理	19
2.2.2 网络配置	19
2.2.3 协议	21
2.2.4 客户机	23
2.3 SQL Server Management Studio	25
2.3.1 启动 Management Studio	25
2.3.2 Query 窗口	29

2.4 SQL Server 集成服务(SSIS)	34
2.5 Bulk Copy Program	35
2.6 SQL Server Profiler	35
2.7 sqlcmd	35
2.8 本章小结	36

第 3 章 T-SQL 语言基础

3.1 基本 SELECT 语句	38
3.1.1 SELECT 语句与 FROM 子句	38
3.1.2 WHERE 子句	41
3.1.3 ORDER BY 子句	44
3.1.4 GROUP BY 子句	47
3.1.5 HAVING 子句	55
3.1.6 XML 子句	57
3.1.7 OPTION 子句	57
3.1.8 DISTINCT 和 ALL 谓词	57
3.2 使用 INSERT 语句添加数据	60
3.3 用 UPDATE 语句更改数据	65
3.4 DELETE 语句	68
3.5 本章小结	69
3.6 练习	70

第 4 章 连接

4.1 连接	71
4.2 内部连接(INNER JOIN)	73
4.3 外部连接	80
4.3.1 简单的外部连接	81
4.3.2 处理更复杂的外部连接	85
4.4 完全连接(FULL JOIN)	89
4.5 交叉连接(CROSS JOIN)	90
4.6 JOIN 语句的早期语法结构	92
4.6.1 内部连接的早期语法结构	92
4.6.2 外部连接的早期语法结构	93
4.6.3 交叉连接的早期语法结构	94

4.7 联合(UNION).....	94	6.5.2 在已存在的表中添加 DEFAULT 约束.....	159
4.8 本章小结.....	99	6.6 使约束失效.....	159
4.9 练习.....	99	6.6.1 在创建约束时忽略无 效的数据.....	160
第5章 创建和修改数据表.....	101	6.6.2 临时使已存在的约束失效...	162
5.1 SQL Server 中的对象名.....	101	6.7 规则和默认值.....	164
5.1.1 模式名称.....	101	6.7.1 规则.....	164
5.1.2 数据库名称.....	104	6.7.2 默认值.....	165
5.1.3 通过服务器命名.....	104	6.7.3 决定哪个表和数据类型使用 给定的规则或默认值.....	166
5.1.4 总结默认值.....	104	6.8 数据完整性的触发器.....	167
5.2 CREATE 语句.....	105	6.9 选择使用的内容.....	167
5.2.1 CREATE DATABASE.....	105	6.10 本章小结.....	168
5.2.2 创建数据表.....	111	第7章 在查询中添加更多内容.....	169
5.3 ALTER 语句.....	121	7.1 子查询的概念.....	170
5.3.1 ALTER DATABASE.....	122	7.2 相互关联的子查询.....	173
5.3.2 ALTER TABLE.....	126	7.2.1 相互关联的子查询的 工作原理.....	174
5.4 DROP 语句.....	129	7.2.2 在 WHERE 子句中的相互 关联的子查询.....	174
5.5 使用 GUI 工具.....	130	7.2.3 处理 NULL 数据—— ISNULL 函数.....	177
5.5.1 使用 Management Studio 创建数据库.....	130	7.3 派生表.....	178
5.5.2 回到代码: 使用 Management Studio 创建脚本的基础.....	135	7.4 EXISTS 运算符.....	180
5.6 本章小结.....	136	7.5 混合数据类型: CAST 和 CONVERT.....	184
5.7 练习.....	137	7.6 性能方面的考虑.....	186
第6章 约束.....	139	7.7 本章小结.....	188
6.1 约束的类型.....	140	7.8 练习.....	188
6.1.1 域约束.....	140	第8章 做到规范: 规范化和其他 基本设计问题.....	189
6.1.2 实体约束.....	141	8.1 表.....	189
6.1.3 参照完整性约束.....	141	8.2 保持数据“规范”.....	190
6.2 约束命名.....	141	8.2.1 开始之前.....	191
6.3 键约束.....	142	8.2.2 第一范式.....	192
6.3.1 主键约束.....	143	8.2.3 第二范式.....	195
6.3.2 外键约束.....	145	8.2.4 第三范式.....	197
6.3.3 唯一约束.....	155		
6.4 CHECK 约束.....	156		
6.5 DEFAULT 约束.....	157		
6.5.1 在 CREATE TABLE 语句中 定义 DEFAULT 约束.....	158		

8.2.5 其他范式	199	9.4.1 选择性	249
8.3 关系	199	9.4.2 监视成本：什么时候更少 的是更多的	250
8.3.1 一对一关系	200	9.4.3 选择群集索引	250
8.3.2 一对一或一对多关系	201	9.4.4 列顺序问题	253
8.3.3 多对多关系	203	9.4.5 删除索引	253
8.4 图表	205	9.4.6 使用数据库引擎调整向导	253
8.4.1 表	208	9.5 维护索引	253
8.4.2 添加和删除表	208	9.5.1 碎片	254
8.4.3 关系	214	9.5.2 标识碎片与页面拆分 的可能性	254
8.5 非规范化	216	9.6 本章小结	258
8.6 超越规范化	217	9.7 练习	259
8.6.1 保持简单	217		
8.6.2 选择数据类型	218	第 10 章 视图	261
8.6.3 存储方面的错误	218	10.1 简单的视图	261
8.7 规划快速示例	218	10.2 更加复杂的视图	266
8.7.1 创建数据库	219	10.3 使用 T-SQL 编辑视图	273
8.7.2 添加图表和初始表	219	10.4 删除视图	273
8.7.3 添加关系	223	10.5 在 Management Studio 中 创建和编辑视图	273
8.7.4 添加一些约束	225	10.6 审核：显示现存的代码	277
8.8 本章小结	226	10.7 保护代码：加密视图	279
8.9 练习	226	10.8 关于模式绑定	280
第 9 章 SQL Server 存储和 索引结构	229	10.9 使用 VIEW_METADATA 使视图看起来类似于表	280
9.1 SQL Server 存储	229	10.10 带索引的(具体化的)视图	281
9.1.1 数据库	229	10.11 本章小结	284
9.1.2 盘区	229	10.12 练习	285
9.1.3 页面	230		
9.1.4 行	231	第 11 章 编写脚本和批处理	287
9.2 理解索引	231	11.1 脚本基础	287
9.2.1 平衡树(B-树)	232	11.1.1 USE 语句	288
9.2.2 SQL Server 中数据 的访问方式	235	11.1.2 声明变量	288
9.3 创建、改变和删除索引	243	11.1.3 使用 @@IDENTITY	292
9.3.1 CREATE INDEX 语句	243	11.1.4 使用 @@ROWCOUNT	296
9.3.2 创建 XML 索引	248	11.2 批处理	297
9.3.3 使用约束创建的隐含索引	249	11.2.1 批处理中的错误	299
9.4 明智的选择：在何处以及 何时使用何种索引	249	11.2.2 什么时候使用批处理	299
		11.3 SQLCMD	302

11.4 动态 SQL: 以 EXEC 命令 即时产生代码	305	第 13 章 用户自定义函数	379
11.5 本章小结	311	13.1 UDF 的定义	379
11.6 练习	312	13.2 返回标量值的 UDF	380
第 12 章 存储过程	313	13.3 返回表的 UDF	383
12.1 创建存储过程: 基本语法	313	13.4 调试用户自定义函数	393
12.2 使用 ALTER 改变存储过程	315	13.5 数据库中的 .NET	393
12.3 删除存储过程	315	13.6 本章小结	394
12.4 参数化(Parameterization)	315	13.7 练习	394
12.5 流控制语句	321	第 14 章 事务和锁	395
12.5.1 IF...ELSE 语句	321	14.1 事务	395
12.5.2 CASE 语句	331	14.1.1 BEGIN TRAN	396
12.5.3 使用 WHILE 语句循环	338	14.1.2 COMMIT TRAN	396
12.5.4 WAITFOR 语句	339	14.1.3 ROLLBACK TRAN	397
12.5.5 TRY/CATCH 块	339	14.1.4 SAVE TRAN	397
12.6 通过返回值确认成功或失败	340	14.2 SQL Server 记录日志的 工作方式	397
12.7 处理错误	342	14.2.1 失败和恢复	398
12.7.1 以前的方式	343	14.2.2 隐式事务	400
12.7.2 在错误发生前处理错误	349	14.3 锁定和并发	400
12.7.3 手动地激活错误	353	14.3.1 通过锁定可以防止 的问题	401
12.7.4 添加自定义的错误消息	356	14.3.2 可以锁定的资源	404
12.8 使用存储过程的好处	360	14.3.3 锁升级和锁对性能 的影响	404
12.8.1 创建可调用的进程	360	14.3.4 锁模式	405
12.8.2 为了安全性使用 存储过程	361	14.3.5 锁的兼容性	407
12.8.3 存储过程和性能	362	14.3.6 指定一种特有的锁定 类型——优化器提示	407
12.9 扩展的存储过程	364	14.4 设置隔离层	408
12.10 递归简介	364	14.5 处理死锁(A1205)	411
12.11 调试	367	14.5.1 SQL Server 断定 死锁的方式	411
12.11.1 为调试设置 SQL Server	367	14.5.2 选择死锁牺牲者的方式	411
12.11.2 开启调试器	367	14.5.3 避免死锁	411
12.11.3 调试器的组成	371	14.6 本章小结	413
12.11.4 在调试器启动后 使用调试器	372	第 15 章 触发器	415
12.12 .NET 程序集	375	15.1 触发器的概念	416
12.13 本章小结	376	15.1.1 ON 子句	417
12.14 练习	377		

15.1.2	WITH ENCRYPTION 子句	417	15.9	本章小结	433
15.1.3	FOR/AFTER 子句与 INSTEAD OF 子句	417	第 16 章	XML 初级教程	435
15.1.4	WITH APPEND 子句	420	16.1	XML 基础	435
15.1.5	NOT FOR REPLICATION 子句	420	16.1.1	XML 文档的各个 组成部分	436
15.1.6	AS 子句	420	16.1.2	命名空间	444
15.2	为了数据完整性规则 使用触发器	420	16.1.3	元素的内容	445
15.2.1	处理来自于其他表 的需求	421	16.1.4	有效的与良好格式的 ——模式和 DTD	446
15.2.2	使用触发器来检查 更新的增量	422	16.2	SQL Server 为 XML 添加的功能	446
15.2.3	将触发器用于自定义 的错误消息	424	16.2.1	根据 XML 格式检索 关系数据	447
15.3	触发器的其他用途	424	16.2.2	RAW	448
15.4	其他触发器的问题	424	16.2.3	AUTO	450
15.4.1	触发器可以嵌套	424	16.2.4	EXPLICIT	452
15.4.2	触发器可以递归	425	16.2.5	PATH	468
15.4.3	触发器不能防止 修改体系结构	425	16.2.6	OPENXML	473
15.4.4	可以关闭触发器 而不删除它	425	16.3	简要介绍 XSLT	478
15.4.5	触发器激活顺序	426	16.4	本章小结	481
15.5	INSTEAD OF 触发器	427	第 17 章	责任报告: Reporting Services 概述	483
15.6	性能考虑	428	17.1	Reporting Services 101	483
15.6.1	触发器是反应性的 而非主动的	428	17.2	建立简单的报表模型	484
15.6.2	触发器与激活它们的过程 之间不存在并发问题	428	17.2.1	Data Source Views	488
15.6.3	使用 IF UPDATE() 和 COLUMNS_UPDATED()	429	17.2.2	创建报表	494
15.6.4	使触发器简短	431	17.3	Report Server Projects	497
15.6.5	选择索引时不要忘记 触发器	431	17.4	本章小结	501
15.6.6	不要在触发器中进行回滚	431	第 18 章	与 Integration Services 集成	503
15.7	删除触发器	431	18.1	理解问题	503
15.8	调试触发器	432	18.2	使用 Import/Export Wizard 来生成基本的程序包	504
			18.3	运行程序包	509
			18.3.1	使用 Execute Package Utility	510

18.3.2 在 Business Intelligence Development Studio 中 执行程序包	512	19.2.3 恢复	532
18.3.3 在 Management Studio 中执行程序包	512	19.3 索引维护	533
18.4 编辑程序包	513	19.4 数据归档	536
18.5 本章小结	515	19.5 本章小结	536
第 19 章 扮演管理员的角色	517	19.6 练习	537
19.1 调度作业	517	附录 A 练习答案	539
19.1.1 创建运算符	518	附录 B 系统函数	549
19.1.2 创建作业和任务	520	附录 C 找到合适的工具	595
19.2 备份和恢复	528	附录 D 非常简单的连接示例	601
19.2.1 创建备份: 转储	528	附录 E 安装和使用样本数据库	605
19.2.2 恢复模型	531		

RDBMS 基础： SQL Server 数据库构成

数据库包括哪些内容？当然包括数据(不保存任何数据的数据库又有什么作用？)。但实际上关系数据库管理系统(Relational Database Manazent System, RDBMS)要比数据本身重要得多。目前先进的 RDBMS 不仅可以保存数据，而且能够管理数据，如限定输入到系统内的数据类型，还能方便地从系统中读取数据。如果仅仅是想将数据安全保存，则可使用任意数据存储系统。RDBMS 不仅可以保存数据本身，还能定义数据类型和数据业务规则。

不要将“数据业务规则”与更通用的业务规则两者混淆，后者可以应用于整个系统。(例如，用户在登录之前不能看到任何数据，或在每个月的第一天自动调整财务系统中的现金周期)。系统的任一虚拟层可以实施这类规则。(目前，在 n 层系统的中间层或客户层已经很常用)。但这里所涉及的业务规则特指与数据本身有关的规则。例如，不存在为负数额的销售订单。使用 RDBMS，可将这些业务规则与数据库本身的完整性集成。

本章为本书其余的内容提供了一个概述，所介绍的内容将覆盖本书后面的章节，为学习后面的内容提供一个指导。本章内容包括：

- 数据库对象
- 数据类型
- 保证数据完整性的其他数据库概念

1.1 数据库对象概述

RDBMS(如 SQL Server)包含许多对象。Microsoft 选择什么可以称为对象(或什么不能称为对象)是否满足对象的标准定义，这里不做深究，但是对于 SQL Server，它常包括以下重要的数据库对象：

数据库	索引
事务日志	程序集
表	报表
文件集	全文本目录

图表	用户自定义数据类型
视图	角色
存储过程	用户
用户自定义函数	

1.1.1 数据库对象

在给定 SQL Server 中，数据库实际上是最高层对象。(从技术角度来说，服务器本身可以看作一个对象，但从实际“编程”的角度看，不能称其为对象，因此本书不作深究)。在 SQL Server 中，大部分其他对象(但不是所有)为数据库对象的子对象。

注意：

如果熟悉 SQL Server 老版本，您可能要问：“登录、远程服务器以及 SQL 代理服务发生了什么变化？”SQL Server 具有几个其他对象(如前面所列的)，用于支持数据库对象。除了链接服务器和 Integration Services 数据包之外，在设计和编程时通常不需要考虑其他对象，其中链接服务器和 Integration Services 数据包主要由数据库管理员管理。其他对象可通过 SQL 管理对象(SQL Management Object, SMO)进行编程。

数据库通常至少包括一组表对象，一般也包括其他一些对象，如存储过程和视图。存储过程和视图是对保存在数据库表中的数据中的特定分组。

在数据库中存储什么类型的表，而在另一数据库中又可以存储什么类型的表呢？这个问题将在本书的后面章节详细讨论，现在简单地认为：属于同一系统或明显相关的数据要存储在同一数据库中。RDBMS(比如 SQL Server)在一个服务器上可能有多个用户数据库，当然也可能只有一个用户数据库。SQL Server 上可容纳多少数据库的决定因素有容量(CPU 能力、磁盘 I/O 限制、内存等)、意愿(您希望某个人具有系统运行的服务器的管理权，而其他也具有另一不同服务器的管理权)或者仅仅是您公司或客户具有多少数据库。许多服务器仅有一个产品数据库，而其他服务器则有多个数据库。另外记住：使用任何版本的 SQL Server，很可能发现目前该产品仍在生产(SQL Server 2000 在被取代之前已经使用了 5 年，因此假设许多软件店有该版本或更高版本的 SQL Server 销售)，在同一个物理服务器上，能够具有 SQL Server 的多个实例，完成单独登录和管理权限。

注意：

有人会问：“能否在同一服务器上安装不同版本的 SQL Server，比如 SQL Server 2000 和 SQL Server 2005？”答案是肯定的。可以将 SQL Server 2000 和 SQL Server 2005 混合安装。本书不推荐采用这种配置，但是如果有需要的话，可以这样安装。

安装好的 SQL Server 第一次启动时包括 4 个系统数据库：

- 主数据库
- 模型数据库
- msdb
- tempdb

安装了以上所有数据库的服务器才能正常运行。(事实上，如果缺少其中任一数据库，

服务器都无法运行)。根据安装选项的不同,服务器可能包括以下样本数据库:

- AdventureWorks (样本数据库)
- AdventureWorksDW (用于 Analysis Services 的样本)

除系统安装的样本数据库外,本书还大量应用老版本的样本数据库。(参考附录 F——系统安装在线帮助)

- pubs
- Northwind

注意:

在本书设计过程中,是使用新样本数据库还是坚持使用经过验证的老样本数据库,存在不同意见。Microsoft 对保留使用老样本数据库的选择不是很满意,但本书还是使用了老样本数据库。

新的 AdventureWorks 样本数据库当然更加健壮,且为了让您会利用 SQL Server 2005 中的示例,示例的设计费了很大功夫。但是 AdventureWorks 数据库过于复杂,特别是用于数据库培训时。该数据库可能仅在个案中使用,并且使用它们作为显著的特性。我邀请几位教授 SQL Server 以及编写 SQL Server 书籍的朋友投票,而所有朋友赞同我的这个观点:Northwind 和 pubs 数据库非常简单,对理解 SQL Server 相关的基本概念相对容易。我更愿意您能理解数据库的基本概念,并不断深入学习下去,而不愿您因 AdventureWorks 数据库不必要的复杂而受到打击。

1. 主数据库

任一 SQL Server 都有主数据库,不管 SQL Server 是哪一版本,或是定制安装。主数据库保存一组特殊的表(系统表)以用于系统的总体控制。比如在服务器上新建一数据库,则在主数据库中的 sysdatabases 表中将加入该记录项。所有外部存储过程和系统存储过程都存储在主数据库中,不论该存储过程是用于哪一数据库。既然几乎所有描述服务器的信息都存储于主数据库,显然,该数据库对于系统至关重要,不能删除它。

在必要时,主数据库中的系统表可能非常有用。系统表能让您确定某些对象在对其执行操作之前是否存在。比如,如果试图在数据库中建立已经存在的对象,系统会提示出错。如果想要解决该问题,则可以测试 sysobjects 表,确认是否存在有对应该数据库的表对象的记录项。如果有,在重新创建之前要将它删除。

说明:

如果您对系统表非常了解,您可能对自己说:“我不可能破坏系统表!”实际上以任何形式使用系统表都非常危险。Microsoft 推荐至少在 3 个 SQL Server 版本中禁止使用系统表。他们不绝对保证不同版本 SQL Server 的主数据库的兼容性,而事实上,主数据库几乎都被更改。在更新主数据库中的对象时,会碰到最大的错误。注意:以任意方式更改这些系统表可能使 SQL Server 失效!不过有几个对象(如系统函数、系统存储过程和信息模式视图)由于保存于系统表中的元数据中,因而能重新找到。

除上述之外,还有些时候要使用该数据库。其他会讨论一些不能避免要使用系统表的情况,但通常最好不要去碰它。

2. 模型数据库

顾名思义,模型数据库是指可以基于该模型得到一个副本。模型数据库构成新建数据库的模板。也就是说,如果想要改变新建数据库的样式,则可以根据需要更改模型数据库。例如,可以向新建的每一数据库中加入一组 `audit` 表。也可以将一些用户组复制到系统新建的每一个数据库中。注意:由于模型数据库作为其他任意数据库的模板,因此系统中必须保留该数据库,禁止删除它。

在更改模型数据库时要注意的是:第一,任意新建的数据库至少要比模型数据库大。也就是说,如果您将模型数据库大小更改到 100MB,就不能新建比 100MB 小的数据库。另外更改模型数据库还会引起其他问题。同样,对于绝大部分 SQL Server 系统的安装,强烈建议不要修改这一项。

3. msdb 数据库

msdb 数据库是 SQL 代理进程保存任意系统作业的场所。如果计划对一数据库在每夜进行备份,则在 msdb 数据库中有一个记录项。要执行一次计划好的存储过程,则在 msdb 数据库中要有一个记录项。

4. tempdb 数据库

tempdb 数据库是服务器主要工作区域之一。只要执行一个复杂或者大型的查询操作,则 SQL Server 需要建立一些中间表,而建立的中间表就是在 tempdb 数据库中。只要建立临时表,则这些表会建立在 tempdb 数据库中,即使您是在当前数据库中建立的这些表。只要需要临时保存数据,则很可能是将数据保存在 tempdb 数据库中。

tempdb 数据库与其他任意数据库不同。不仅数据库中的对象是临时的,连数据库本身也是临时的。在每一次 SQL Server 启动时,tempdb 数据库是系统中唯一完全重建的数据库。

说明:

从技术上来说,可以在 tempdb 数据库中建立自己的对象,但强烈反对进行该操作。可以在系统中任意有权使用的数据库中建立临时对象,该对象也会存储在 tempdb 数据库中。直接在 tempdb 数据库中建立对象没有任何意义,只是增加了数据库之间的混乱。

5. AdventureWorks 数据库

在有 AdventureWorks 数据库之前,SQL Server 就包括了其他样本数据库。老样本数据库虽然有不足之处,如它们包含了一些不好的设计习惯。(这里不再考虑 AdventureWorks 数据库是否存在相同问题,但是 AdventureWorks 数据库确实试图解决这些问题)。此外,老的例子过于简单,集中阐明某些数据库概念,而不是将 SQL Server 作为一个产品或数据库的整体。

从 Yukon(今天其他所知的 SQL Server 2005 的内部名字)发展的最早期阶段,Microsoft 想要一个更加健壮的样本数据库,使其尽可能作为其产品的一个示例。AdventureWorks 数据库就是这种想法的结果。尽管该数据库对于初学者过于复杂,但是其作为数据库示例确