学校的理想装备
电子图书·学校专集
校园网上的最佳资源
SQL Server 7
参考手册（一）
BOOK
内部资料 非卖品

本书的结构

如果您是一个用户、开发人员、管理员或经理，会采用不同的方式来使用 SQL Server。为了满足这些不同的需求，本书分为五个部分：

·第一部分“欢迎使用 Microsoft SQL Server 7”是 SQL Server 的总提纲，讲述了 SQL Server 的历史及其发展。本部分包括以下内容：体系结构的改进、新的存储结构、Internet/Intranet、Microsoft Transaction Server，企业、Microsoft Managment Console (MMC)、SQL Server Agent、与 Windows NT 安全性的
集成、Transact-SQL 的改进、新增强的工具、复制的增强和数据库管理员角色的简化。

·第二部分“安装”包含安装 SQL Server、安装 SQL Mail 以及将 SQL Server 升级到版本 7 的指导。这些过程用所支持的实例进行了详细的说明。

·第三部分“管理”讲述的任务有：数据库管理和预防性的维护、备份、远程服务器和链接服务器、数据库监视(其中着重论述数据库的完整性、可恢复性和可靠性)。

·第四部分“开发：编程语言”包括开发主题，如数据库、引用和事务完整性；如何设计和建立数据库；规范化；建立索引；巨型数据库(VLDB)；数据库设计的方法、一般化、编码策略；分区；并发性；编写 Transact-SQL，存储程序和触发器；数据仓库。

·第五部分“参考资料”包括 Transact-SQL 语法、系统表和系统存储程序。这些章节的内容按字母表的顺序安排，这样用户可以快速查找到需要的信息。

译 者 的 话

Microsoft SQL Server 7.0 是 Microsoft 公司推出的数据库软件 Microsoft Windows NT & BsckOffice 家族中的 SQL Server 的最新版本，它在易用性、可靠性和其他性能方面作出了一系列革新，并且为开发者提供了快捷简便的编程模式。它为企业运作、电子商务和移动计算机提供了广泛的可伸缩的解决方案；它具有强大的数据仓库，并提供了一个综合平台，这个平台使用户可以轻松地设计、建立、管理以及使用数据仓库解决方案；它同 Microsoft Office 产品紧密集成，可以对 Microsoft Office 实现完美无缺的数据访问、桌面多维分析等。

本书结合丰富的实例，系统地介绍了 Microsoft SQL Server 7.0 的功能和使用方法。全书共分为五部分：其中，第一部分包括第 1 章到第 10 章，介绍了 Microsoft SQL Server 数据库的概念、历史和新功能；第二部分包括第 11 章到第 13 章，介绍了 Microsoft SQL Server 数据库的安装和升级；第三部分包括第 14 章到第 17 章，介绍了数据库的管理、链接服务器和远程服务器以及监视服务器性能的工具；第四部分包括第 18 章到第 20 章，主要介绍了发展的程序设计语言；第五部分包括第 21 章到第 23 章，向用户提供了参考资料。

本着认真负责的态度，我们在本书的翻译过程中相互协作，并对有关技术问题进行了统一处理。参加本书翻译工作的有薛焱、王军、宋愚、胡东、李鹏、马利涛、冯博、张小蓝、许书明、孔祥丰、袁建华、曹木军、刘利平、王维、刘艳平、周珂和志强等，在此表示衷心的感谢！

由于译者水平有限，译文难免会有错误，欢迎读者提出宝贵意见。

1999 年 10 月 10 日

总　目　录

第一部分　欢迎使用 Microsoft SQL Server 7
第二部分　安装
第三部分　管理
第四部分　开发——编程语言
第五部分参　考　素　材
　　第 21 章 Transact-SQL 编程语言参考
　　第 22 章系统表
　　第 23 章系统存储过程

返回总目录

目　　录

第一部分　欢迎使用 Microsoft SQL Server 7 5
第 1 章什么是 Microsoft SQL Server 6
1.1Microsoft SQL Server 的发展历史 6
1.2Microsoft SQL Server 的将来 9
1.3 体系结构的改进 12
1.4Microsoft Transaction Server 的到来 16
1.5Enterprise 一瞥 17
第 2 章容量的改进 21
第 3 章 Microsoft 管理控制台 22
3.1 增加 Microsoft Management Console Snap-Ins 22
3.2 服务器的 Enterprise Manager 功能 32
第 4 章 SQL Server Agent 68
4.1 定义操作员 72
4.2 创　建　作　业 75
4.3 创　建　警　告 82

4.4 多任务和多方案作业 99
第 5 章与 Windows NT 安全性的集成 101
5.1Security Manager 的消失 102
5.2 身 份 验 证 102
5.3 许可的有效性 111
5.4 授 予 权 限 127
5.5 实现安全性的若干建议 142
5.6 所 有 权 链 143
5.7xp_cmdshell 143
第 6 章 Transact-SQL 增强工具 146
6.1 增强的存储过程 146
6.2 新系统扩展存储过程 157
6.3 触 发 程 序 159
6.4 新数据类型 164
6.5 数据存储增强功能 165
6.6 新的属性功能 165
6.7 新的安全性功能 166
6.8 其他新功能 167
6.9 查询的增强功能 168
6.10 新的优化程序提示 169
6.11 事务处理的增强功能 170

6.12 系统表的增强功能 171
第 7 章新工具和增强工具 173
7.1SQL Server 查询分析工具 173
7.2 数据转换服务工具 177
7.3SQL Server Profiler 185
7.4 数据库维护计划向导 190
7.5Web 辅助向导 193
7.6 可视数据库工具(Database Tools) 198
7.7Replication Monitor（复制监视工具） 204
第 8 章实用程序的变化 206
8.1BCP 206
8.2 备份(Backup) 214
8.3 恢复(Restore) 215
8.4 数据库一致性检查程序(Database Consistency Checker) 216
8.5 OSQL 和 ISQL 225
8.6SQLMAINT 226
第 9 章 SQL Server 7 的复制增强 233
9.1 多点更新复制(Multi-site Update Replication) 234
9.2 合并复制(Merge Replication) 235
9.3 快照和数据转换服务 242
9.4 分区事务复制 244

9.5 更新订阅器 244
9.6 推、拉订阅 247
9.7 因特网复制增强 250
9.8 匿名拉回订阅 251
9.9 数据加密 252
9.10 在因特网上的保密：微软代理服务器 254
9.11 多机种环境数据复制服务 257
9.12 创建复制脚本 257
第 10 章简化数据库管理员的任务 259
10.1 自动和动态配置 260
10.2 动态结构管理 268

第一部分　欢迎使用 Microsoft SQL Server 7

第 1 章什么是 Microsoft SQL Server

Microsoft SQL Server是运行在 NT 操作系统上的关系型数据库。SQL 即结构化查询语(Structured Query Language),是得到广泛公认的工业标准，它用来定义、修改和管理数据，并用表格、索引、关键字、存储数据的行和列来控制如何修改数据库。SQL 是从国际商用机器公司(IBM)的 Edgar(Ted)F.Codd 博士的概念发展而来的，70 年代 Edgar(Ted)F.Codd 博士在 IBM 研究实验室工作时曾协助开发关系模型。

1.1Microsoft SQL Server 的发展历史

1992 年 6 月，在会见《计算机世界》杂志高级编辑 Jean Bozman 时，Codd 博士回忆起 60 年代末出席的一个关于 IBM 竞争对手的数据库产品的讲座。当时他向主讲人提出“谓词逻辑和存在判断及统一量化”的问题，这位数据库专家对关系型技术的描述知之甚少，当时 Codd 确信关系型理论是一个广泛开放的领域，人们一旦了解了关系理论，就可以把它应用到数学和统计领域中。

1969 年，Codd 博士继续探索和倡导关系数据库模型的研究。那时，IBM 宣

布自己对数据库系统的主攻方向只可能是 IMS，这是一种非关系型数据库。值得庆幸的是，随后 IBM 的研究主管 Ralph Gamory 也提倡研究关系型数据库的原型，称为 System R。当时人们普遍反对这个与 IMS 争夺市场的难以驾驭的新关系概念，因 IMS 是“独一无二的数据库管理系统”。Codd 博士说“他曾屡次受到指控，说他试图拆 IBM 的台”，他的关系型数据库被视作 IMS 的竞争对手。

甚至随着 IMS 的发布，许多用户要求检索数据的方法应比该数据库能提供的方法更加方便，他们希望开发出更好的工具，为一般人存储和检索数据消除障碍。他们希望能够用类似人类的语言查询数据库，能够即时重新定义数据关系，而不是每两个星期或召集全组程序员时才能重新定义。他们希望简化修改数据结构和数据定义的过程，甚至希望访问数据时不必使用最初建立或更新数据库时所定义的视图。

在非关系型环境下访问数据有许多障碍。在网状和层次数据库中，无法在文件中增加字段，无法改变字段的长度或数据类型，也无法删除字段，因为最初设计用来存储数据的物理结构不能更改。对数据库的更改将带来对结构的更改，这些改变是成批进行的。这样就可能对开发造成瓶颈，该瓶颈是数据库管理员的门槛，数据库管理员通常必须在休息时间立即对累计的数据库全盘修改。程序通常采用 COBOL 编写，必须编写程序才能从数据库中卸载数据，在重新建立物理结构以后，还必须编写更多的程序才能把该数据重新装入数据库。这是个费时又费力的过程。

非关系型数据库的另一个缺点是，在原始定义的逻辑结构视图之外，网状

或层次数据库不能以自由方式查询数据。随着时间的流逝，查询数据所需要的方式在不断变化，实际的商业场合需要把原始数据及时转变为最新的信息，以提高企业的竞争力。在非关系型数据库中，任何人都不能查看或查询与创建数据库时定义的内容有差别的数据，除非定制程序和进行专项处理。卸载数据、重建方案和重装数据所用的时间和要求的严格性，只可能带来一个不能满足实时信息需求的僵硬不灵活的数据库。

程序员和用户陷入这种耗时而昂贵的过程时，开始考虑由 Codd 博士定义的关系型数据库的技术概念，这并不令人感到意外。1984 年，笔者一直在设计这样的数据库：它允许用户以无数不同的方法查看收入帐单和收支平衡表，而管理员告诉笔者“太糟糕了，您还没用关系型数据库”。这是笔者首次听说这种数据库，以及这种数据库工作原理方面的知识。

关系型数据库克服了访问数据的许多障碍，并为程序员提供了一种轻松的方式。像 alter table 这样简单的一行命令就可以改变数据库的结构。不需要卸载，不需要重装，数据根本不需要打乱次序(卸载是把数据与数据库结构分离的操作)。调整起来非常容易，数据库中的所有数据元素都可以灵活地查询。不必预先定义数据视图，稍加注意就可以建立数据之间的关联。

到了 70 年代中期，人们已认识到关系型数据库最适合传递用户所希望的功能。为了满足用户的要求，IBM 重新实施 System R 项目，到 80 年代初期，关系型数据库开始投放市场。1992 年,Codd 博士在《计算机世界》杂志上撰文指出：“关系型数据库具有坚实的未来，但是必须增强它们的功能，提高运行速度，对终端用户的访问更加开放。它们还需要嵌入新的技术。”

从这些早期的关系数据库开始，逐渐形成了今天的 Microsoft SQL Server。Microsoft 最初从 Sybase 获得 SQL Server 的基本构造部件的许可，使该产品可运行于 OS／2 的 PC 平台，最近更可运行于 Windows NT 的 PC 平台。在 Sybase、Ashton-Tate 和 Microsoft 的共同努力下，1988 年开发出基于个人计算机的第一个数据库产品，正如 Codd 博士当年预想的那样，使关系型数据库对终端用户的访问更加开放。Microsoft 率先提出 SQL Server 项目，在 SQL Server 从 OS／2 转向 NT 操作系统时，Ashton-Tate 退出这一领域，而 Microsoft 和 Sybase 都在销售基于 PC 硬件平台的数据库产品，直到 4.21 版。但是 Microsoft ／ Sybase 的伙伴关系于 1993 年解体,此后 Sybase 以小型机平台为重点，而 Microsoft 则主攻个人计算机平台。

也许 Bill Gates 阅读了 1992 年第 6 期《计算机世界》的那篇文章，由于他和 Codd 博士在同一篇文章中接受采访，并被同时誉为“改变世界的 25 个人”，SQL Server 自那篇文章开始，在 Microsoft 进入发展的鼎盛时期。

Microsoft、个人计算机和 Internet 正在实现 Codd 博士使终端用户以新的方式访问数据的构想。

1.2Microsoft SQL Server 的将来

今天，关系型数据库被认为是存储数据的优选方法，具有各不相同的关系型数据库的许多公司都在竞争，以领导 SQL 软件包。实际上，数据库的未来和

SQL 都是受用户需求和市场驱动的，这些要求与 30 年前一样：更大、更快、更易使用，数据库的可访问性更高。这给关系型数据库开发人员带来长期的持续性压力。

使用简单的要求使大型计算机和小型计算机操作系统与个人电脑不同，大型计算机和小型计算机操作系统价格昂贵，难以使用，没有设计成简单的用户界面。另外，大型计算机数据库系统需要较多的经费和昂贵的专门知识，而很多小型计算机数据库局限于复杂的 UNIX 操作系统中。

个人计算机是市场上最易使用的计算机。Microsoft 公司在微机的 SQL Server 7 中所采用的图形用户界面的广泛发展清楚地表明，Microsoft 在听取用户的意见，并把重点放在易于使用和易于管理的需求上。多年来，笔者常常感到奇怪，为什么编写数据库的程序员一直在重复他们做过的事情。Microsoft 已修复和消除了 80 年代初期传统设计中的很多错误和不必要的复杂性，这种设计早就应该彻底检查了，总之，低级的重复劳动是显而易见的；数据库管理人员将会为在这个领域的提高所陶醉，并会特别喜爱工作流式的管理。Microsoft 称之为追求低端市场导致不再需要数据库管理员，我则称之为常识。

如今，个人电脑与过去的大型计算机一样强大，包括个人平台(如我所用的)到带有 8 到 16 个处理器的高端 SMP 服务器。Codd 博士在与《计算机世界》杂志讨论输入／输出比实际计算慢很多这个古老问题时，正确预言了将来的解决方法会是“基于每个磁盘单元的小电子计算机，可以把整个 CPU 放在一个单独的芯片上，以便管理对该磁盘的查找和更新。“他也明确谈到了分布式计算和个人电脑。

Microsoft 通过个人电脑把数据即信息放入小型企业、中型企业和大型企业中。毕竟，有多少企业能买得起带有复杂数据库的大型计算机呢？只有那些流动资金有百万美金的巨型企业才买得起大型计算机，只有中型或大型企业才买得起小型计算机。另一方面，Microsoft SQL Server 对小型企业来说不算贵，也能满足大型公司的需要。

Microsoft 清楚地说明了大公司如何在 Microsoft SQL Server 上运营自己的事务。他们卖出的每一份软件都记录在 Microsoft SQL Server 的数据库中，已创建了上千万的数据行。Microsoft 称之为“吃自己的狗食”，显然他们的健康从未受到过影响!所有 NT 和 Microsoft SQL Server 新产品的发布均须经 Microsoft 内部实际应用的严格测试，且使小型企业主能用得起这些功能。

Microsoft 已经说明了可伸缩性，尽管有一些不利的谣言，但它还在很好地工作着，大多数人不能想像该产品在 Microsoft 这样的大公司内日复一日地使用，但就我的个人经验，我可以告诉他们这个产品的伸缩性非常好，并且会随着技术的进步不断改善。

当支持分析员需要详细检查数据库的设计，或数据库缺少处理事务的代码时，表明数据库的性能较差。Microsoft Transaction Server 可帮助上述开发者。设计 SQL Server 数据库和 SQL 编码有正确的方法和错误的方法，一个设计不好、编码技术不佳的数据库，其速度会较慢。若有好的设计、好的编码技术和合适的硬件，那么 Microsoft SQL Server 完全能胜任做这些工作。

将来，对小、中、大型企业而言，Microsoft SQL Server 访问数据将更便宜、更快及更容易。他们为满足更广阔市场中的客户要求所做的不断努力，使 SQL

Server 的将来有了保障。随着 Microsoft 不断为小型企业增添技术（这些技术以前只提供给少数有特权的合作者），Microsoft SQL Server 真正的实力和生命力也就变得越来越明显了。

1.3 体系结构的改进

在这一版本中，Microsoft 采用了 SQL Server 数据库的核心技术，并采用更多的数据仓库功能、更好的优化工具和重新编写的数据库存储引擎，对它进行了重新设计。另一个重要的改进是 Microsoft 特别注重的可伸缩性。现在 Microsoft SQL Server 在每个数据库中能存储一特兆（即一千吉）的字节。Microsoft Enterprise 的思想主要是满足分布式计算环境中的高端数据库的市场需求，并提供各种工具来管理这类计算环境中所需的许多服务器,包括可访问多达 8 个 CPU 的能力。Microsoft Management Console(MMC)把所有这些功能集成起来。MMC 的 Snap in 功能允许插入第三方工具来管理环境，并配套提供其中的 Enterprise Manager 所包含的所有功能。第三章将进一步讨论 MMC 及其功能。

体系结构的改进对数据库来说是一个主要的进展，Microsoft 并未停留于此，它将继续充分利用 PC 硬件的发展优势，并将提供在多个 CPU 上进行并行处理

的更好方法。数据库结构将继续改进以确保其性能和易用性。Microsoft 完全集中在降低 TCO(Total Cost of Ownership)方面，这包括减少、简化，直至最终消除数据库管理员的任务。

1.3.1 新的存储结构

在 Microsoft SQL Server 版本 7 中，存储数据的方法与 6.5 版完全不同。它没有使用以往的设备存储方案，即每个物理文件允许使用多个数据库。现在看到的是存储为文件的数据库，每个文件或每组文件使用一个数据库。过去，设备标记隐藏了实际正在发生的事件，从而引发操作系统文件的产生。现在，虽然多个数据库不再共存于一个文件中，但由于文件组的功能，单个数据库能驻留于多个文件中，这就在性能和容量方面获得了巨大的改进。通过简化概念，并消除对设备的混乱引用，以及创建这些引用所用到的多余步骤，也为新用户减轻了入门的理解困难。现在 create database 命令指定了数据的名字和大小，以及与数据库相关联的日志文件；以往版本中的 disk init 语句用于创建实际的数据库物理文件，现在不再需要，也不再需要以 2K 数据页面为单位来表示大小(SQL Server 7 使用 8K 数据页)，现在是用大多数用户都熟悉的千字节、兆字节和吉字节来说明数据库的大小。

除存储结构中的这些改变外，还有用于行、范围、数据文件和日志文件等的新磁盘格式，这些新格式在改善性能的同时，将允许从低端升级到高端系统所需的可伸缩性。为了提高空间的利用率，如今文本和图像数据能被多行共享。

在过去的版本中，若一列包含了几字节的文本或图像数据，这一列将占据整个2K 页。现在新 8K 页上的自由空间将由属于其它行的数据所使用，还提供了当恢复某个数据库时自动改变和创建数据库的措施。仅发出一个恢复数据库文件的命令，将会自动创建和／或改变这个数据库。这样，数据库管理员不必确定接收到的数据库的大小，就可以在恢复之前创建它。以上只不过是一个例子，说明 SQLServer 7 如何更易于管理。

1.3.2SQL Server 和 Internet

Microsoft SQL Server 和 Microsoft Internet Information Server(IIS)的结合提供了将相关的数据库连接到 Internet 上，同时用 Web 浏览器来显示数据的构架，从而开始了数据库终端用户可访问性的暂新阶段，而这在 Internet 可视化浏览器技术成熟之前是不可能的。将 NT 操作系统与其所有 NT Internet／Intranet 性能以及 SQL Server 数据库集成于一体有数不清的好处。通过使用 NT 操作系统内置的 ODBC 连通性，Microsoft SQL Server 和 NT Internet Information Server 数据库开发者将能通过 Internet Web 站点将交互和在线数据库内容传输给众多用户。SQL Server 7 还提供了对改进的分布式事务处理的支持，增强了无缝复制及多平台实施的分布式管理工具。Microsoft 对 Internet 技术的允分利用使得这些在短短几年前不可想象的事情成为可能。现在，可以通过 Internet 复制数据库，并能使世界各地的远程用户访问数据。改进的图像数据类型能为 Internet 用户添加丰富的内容。图像是以数据碎片的树结构来存储和管理的，看起来像是保存在

一张表的行中。为满足不断发展的 Internet 技术，Microsoft 计划支持声音、视频等一些新的数据类型，这也正是 Codd 博士讲到关系型数据库应能支持新技术时所谈到的内容。

1.3.3Intranets: 防火墙后的 SQL Server

Microsoft Exchange Server能与 Microsoft SQL Server的 SQL Mail部分以及 Microsoft BackOffice产品线一起使用，建立 Intranet解决方案。Intranet与 Internet很相似，但在一个公司里，Intranet 用户数目常受到雇员的限制。Intranet 也受到防火墙的保护。一个公司有了 Intranet，就能使它的数据保密，同时获得 Web 环境的好处。

公司能用 Intranet 发布信息或从其雇员处收集信息，很多公司正将它们的内部应用程序移到 Intranet 上，并操作它们。例如，输入在 Intranet 上工作的时间或输入费用偿还数据到 Intranet Web 页上，该 Web 页会连接到 SQL Server 数据库。在 Intranet 上访问下一个部门的 Web 页，找出它们正在干什么，会有各种各样的可能性，尤其是将存储在公司数据库中的数据插入到 Web 页时，就更是如此。Intranet也可用作将应用程序移到 Internet之前的检验地。

可靠性、快速响应时间以及安全性是使用 Intranet 技术的重要原因。虽然现在 Intranet还相当不成熟，但它们正随着公司内部联系需要的增长而向前推进。如果不采用 Internet 和 Intranet 技术，到下个世纪，企业将不具备竞争力。SQL Server 7 通过推出一系列专为此目的设计的工具来支持这一观点。若数据能存储

到 SQL Server 7 的数据库中，就能通过浏览器技术很快地呈现出来。

1.4Microsoft Transaction Server 的到来

可伸缩性就是当数据长度和用户数目的要求增长时添加更多硬件的能力。多年来可伸缩性一直是客户的要求，分布式计算平台要求可伸缩性要好，但不幸的是，个人电脑的硬件直到最近才能执行分布式计算任务。现在，PC 硬件有不同的处理能力和多个 CPU，且由软件负责利用这种新的能力。Microsoft SQL Server 通过在多个 CPU 上并行执行一个查询来完成这个任务。其中有趣的是，Microsoft SQL Server 7 是唯一被设计在分布式计算环境中运行的数据库，且能以某种方式充分利用 PC 平台上特有的很多机会，而其它关系型数据库则不能。

众所周知，客户机／服务器模式是两层的，即包含两台计算机：客户机和服务器。但当服务器处理数千个用户时，该模式就不能很好地符合可伸缩性，所以，为满足硬件的实际要求，三层或多层应用程序的想法就应运而生了。例如，多层是指另外的计算机插入到能履行企业的事务规则并传送数据的混合层中。

Microsoft Transaction Server 是与之相关联的技术，可使开发者避免处理连通性和关联问题。它对 ActiveX 而言是一个完全的两层服务器环境，并带有自动线程管理及数据库连接池，开发者不必担心事务处理的开始部分和结束部分：如果某一组件标记为事务性，Microsoft Transaction Server 就创建一个有关处理

过程的事务，那些涉及到的组件也将包含在这一事务中。该产品使开发者不必执行系统级任务，如阈值管理，并且最好与 DCOM 一起使用。DCOM，有时称为 COM＋，是 COM(Component Object Model)的一个扩展，它用于为分布式环境建立下一代应用程序。客户访问一个 DCOM 对象，并通过该指针调用例程，就好象该对象运行在客户机上，DCOM 具有数据绑定的特点，可将对象和数据库字段绑定在一起。COM 服务器能创建很多对象类的对象实例，COM 对象有多个接口，每个接口代表这个对象起作用的不同方式。COM 对象和其它相关技术的使用加速了 DCOM 对象的使用，这个结构是将来 Web 上的数据库技术的基础。

该产品使分布式服务器应用程序更易推广使用，且能以其他销售商与之不匹配的方式进行调节。

1.5Enterprise 一瞥

NT Server 企业版将包含 Microsoft Transaction Server(MTS)和 Message Queue Server。SQL Server 企业版使用 Cluster Server，最多可达 16 个节点。SQL Server 7 有 64 位寻址能力，以支持由 32 位软件提供的需大于 2GB 内存的应用程序。Windows NT 和 SQL Server 可在机箱外处理至多 8 个处理器，并能使用高达 3GB 的应用程序内存。在 SQL Server 的一个未详细说明的未来版本中，数据库将在较大的簇上实现大规模的并行处理。这是 Codd 博士早已清楚地意识到的一个

方向。

正如学习关系数据库不可能不知道 Codd 博士一样，提到 Enterprise 时也不可能不提到 Bill Gates，他的 Enterprise 版本出现在 NT Server 的工具、Microsoft SQL Server 和推动分布式计算环境发展的组件中，BackOffice Server 4.0 Enterprise Edition 也包括在这个混合包中。它将包括 NT、SQL Server、Exchange 和 Site Server Enterprise Edition(包括 Commerce Server)的企业版。另外，Wolfpack API 允许第三方用 Microsoft 技术集成其系列产品。允许第三方的产品建立在 Microsoft 技术的基础上是一种策略。Component Object Model 中的一部分 OLE 和 Win32 也是这一策略的一部分，且成为其它操作系统上认可的第三方。这一策略的根本目的是从桌面移到 Enterprise。

服务器模型是 Microsoft 软件产品发行中最显著的特点。BackOffice 中有 SQL Server，System Management Server，SNA Server，MS Mail Server 和 Windows NT。Windows、Component Object Model 和 OLE 使 Enterprise 画面趋于完美。OLD 添加的是一个以面向对象的界面为基础、用 Microsoft Interface Definition Language(MIDL)定义的编程模式。OLE 采用所有的组件和对象提供了一个统一的编程模式，并实现了 Microsoft 的 COM。Microsoft 将开始为开发者提供 OLE 接口，而不用 API。对象可以用 OLE 接口包含在应用程序中，即再使用 COM 对象的想法。

Microsoft 的编程界面是 Win32 API，这个界面是策略的核心部分，目的是使 Win32 以连接图书馆的形式（即允许程序员连接应用程序）在其它操作系统上使用。Win32 API 具有抢先式多任务、多线程、32 位线性内存寻址、保护内

存、对称的多处理能力及其它功能。

对 SQL Server 7 的介绍若不提到链簇的能力就不算完整。通过用 Microsoft Cluster Server API 将服务器链接在一起以提供应用承载平衡的方法，链簇将构成 Enterprise 因素的一个主要部分。当一台数据库服务器失败时，链簇就把它们移到另一个服务器上，使这个数据库成为共享盘上的虚拟数据库。万一失败了，Cluster Server 就在线恢复服务，并帮助系统管理员在维护过程中保持系统有效。系统管理员可以等一段时间，再开始失败过后的服务器维护工作。失败过后是指当监视主服务器的从服务器两次努力后没得到任何反应，及从服务器在共享驱动器队列上保持锁住而其它服务器没有意见的时候。当这些发生时，Cluster Server 能将服务移回以恢复主服务器。Cluster Server 观察着重要的服务和应用程序，任何不能使用的受保护的程序都会在同一机器或不同的服务器上重新启动。为处理硬件的失败，Cluster Server 将利用一台服务器监视另一台服务器。NT Server Enterprise 将包含事务处理器和链簇。Microsoft SQL Server 和 Microsoft Transaction Server MTS 将创建 Enterprise 的软件解决方案。Cluster Server 服务器目前是通过 SCSI 总线链接的，两台服务器共享一列驱动器和监示器，Cluster Server 利用轮询确定服务器或子系统的有效性。

在完整的失败过后，从属服务器开始配置主服务器。重新映射 IP 地址，指定主服务器的 NetBIOS 服务器名，将 NetBIOS 共享返回在线。Microsoft 的 SQL Server 被设计成与 Cluster Server 一同工作以达到分布式加载，或者，如果一个服务器失败，其它服务器将支持共享磁盘上的 SQL Server。

Microsoft 的首席执行官 Bill Gates 概括将来的企业计算景象：“不管多大的

企业都能在 Microsoft 软件和工业标准硬件上运行它的企业应用程序。把企业级的可伸缩性与 PC 工业大容量经济结合在一起，将从根本上重建企业市场形象。Microsoft 和工业界将继续对 R&D 进行投资，这在将来会为我们的客户提供更高水平的可伸缩性、互操作性、可用性和可管理性。”

第 2 章容量的改进

HOPE

声　　明

本电子版不包括第二章内容，请看配套图书第二章。

北京希望电子出版社
1999

第 3 章 Microsoft 管理控制台

除了围绕在 Enterprise Manager 周围的 Microsoft Management Console(MMC)新结构之外，Enterprise Manager 仍非常类似于以前的版本，而 Microsoft Management Console(MMC)是一种管理 BackOffice 服务器的新方法。其意图是能够“紧紧扣住”其他工具，以相同的结构管理 BackOffice 服务器，而这种结构为实现可定制的和中心定位的命令中心创造了条件。这种表示方法非常类似于 Microsoft Windows NT Explorer，表现为树状列表结构。

3.1 增加 Microsoft Management Console Snap-Ins

在打开 Microsoft Management Console 时，可以看到 Console Root 项，窗口分为两个部分，左侧是 Scope Pane，右侧是 Results Pane。如果用户第一次看到 MMC，它似乎很陌生，但将其展开并通过类似 Explorer 的项目深入研究以后，就会觉得比较熟悉。在命令栏中有四个指向下方的锥体：Action、View、Tools 和 Help，每一个都代表一个菜单。右击某个对象时可看到 Action 菜单包含的菜单项，同时弹出该对象的快捷菜单。其中的选项根据在 Scope 或 Results Pane 中

选择的对象不同而变化。SQL Server 7 MMC Snap-ins 控制台称为 SQL Server Enterprise Manager，如图 3.1 所示。

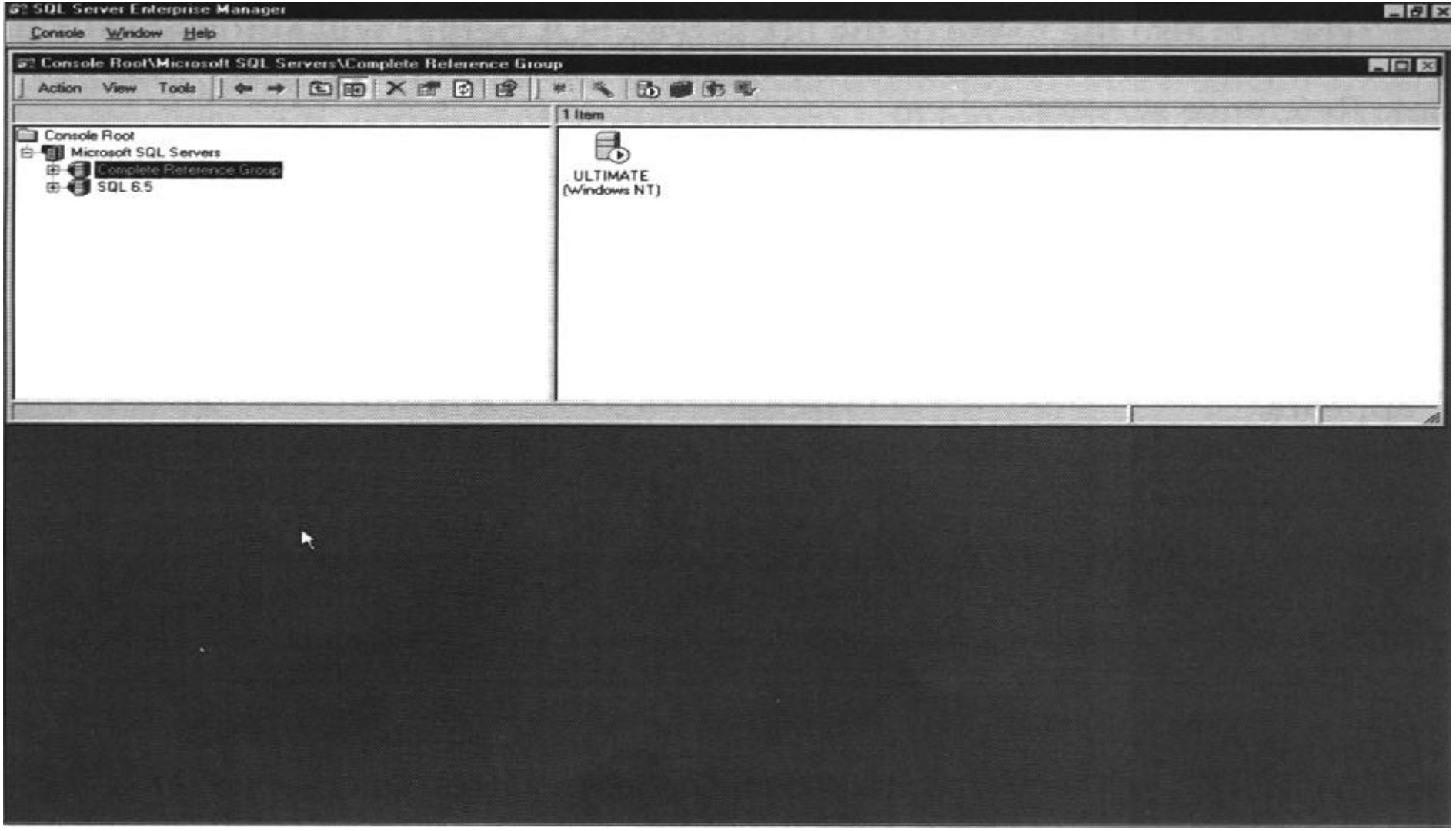


图 3.1Microsoft 管理控制台

3.1.1 增加新的服务器组

要在 Enterprise Manager 的分层树状结构中增加新的服务器组，可进入 Enterprise Manager 的 Command 菜单，单击 Action，就会看到用于增加 New Server Group 的选项。不要将 Action 菜单中的菜单项 New SQL Server Group 与本书前面介绍的 NT 的组和角色弄混。Action 菜单中的 New SQL Server Group 项提供了将服务器组织在一起的方法。该 SQL Server Group 允许在 Enterprise Manager 内部以 SQL Server 的层次建立类似于 Windows NT Explorer 的树状结构，这样用户就可以开始建立并展开该树状结构，看到它的层次。可以把同类型的服务器放到一起，当有数百个服务器时这样做是很方便的。可以把服务器放在易于识别的服务器组中，确定它们在列表层次中的位置，当管理出现问题，多个服务器需要立刻引起注意时可快速访问它们。

下面所示的对话框能建立 New Server Group。注意，该对话框有两个级别，Top Level Group 和 Sub-Group。键入 SQL Server Group 的名称，图中显示了 SQL Servers 的服务器组 Complete Reference Group，该服务器组是在 Manager Server Groups 窗口中创建的。

笔者在计算机上安装了 SQL Server 7，作为本地的 SQL Server，将其命名为 Ultimate，这也是 NT 服务器的名称。SQL Server 会自动将 NT 计算机名用作 SQL Server 名。要将 Ultimate 放入 Complete Reference 服务器组，可以执行下面的操作：

1. 单击 Microsoft Management Console(MMC)的 Console Root 下的 SQL

Servers 项。

2. 单击已创建的服务器组 Complete Reference Group。
3. 从 Action 菜单中选择 Register Server。
4. 当显示如图 3.2 所示的 Register SQL Server Wizard 屏幕时单击 Next。
5. 在可用的服务器列表中选择某个服务器。
6. 按下 Add 按钮。
7. 按下 Next 按钮。显示 Select Connection Option 屏幕，此时允许用户选择用 NT Authe-ntication 或 SQL Server Authentication 与服务器连接。如果选择了 NT Authentication，就将使用登录计算机时附带的 NT 帐户信息进行连接，不必输入密码，因为 SQL Serv-er 委托 NT 进行了受托连接。如果选择了 SQL Server Authentication，就必须键入非受托连接的注册 ID 和 密码。

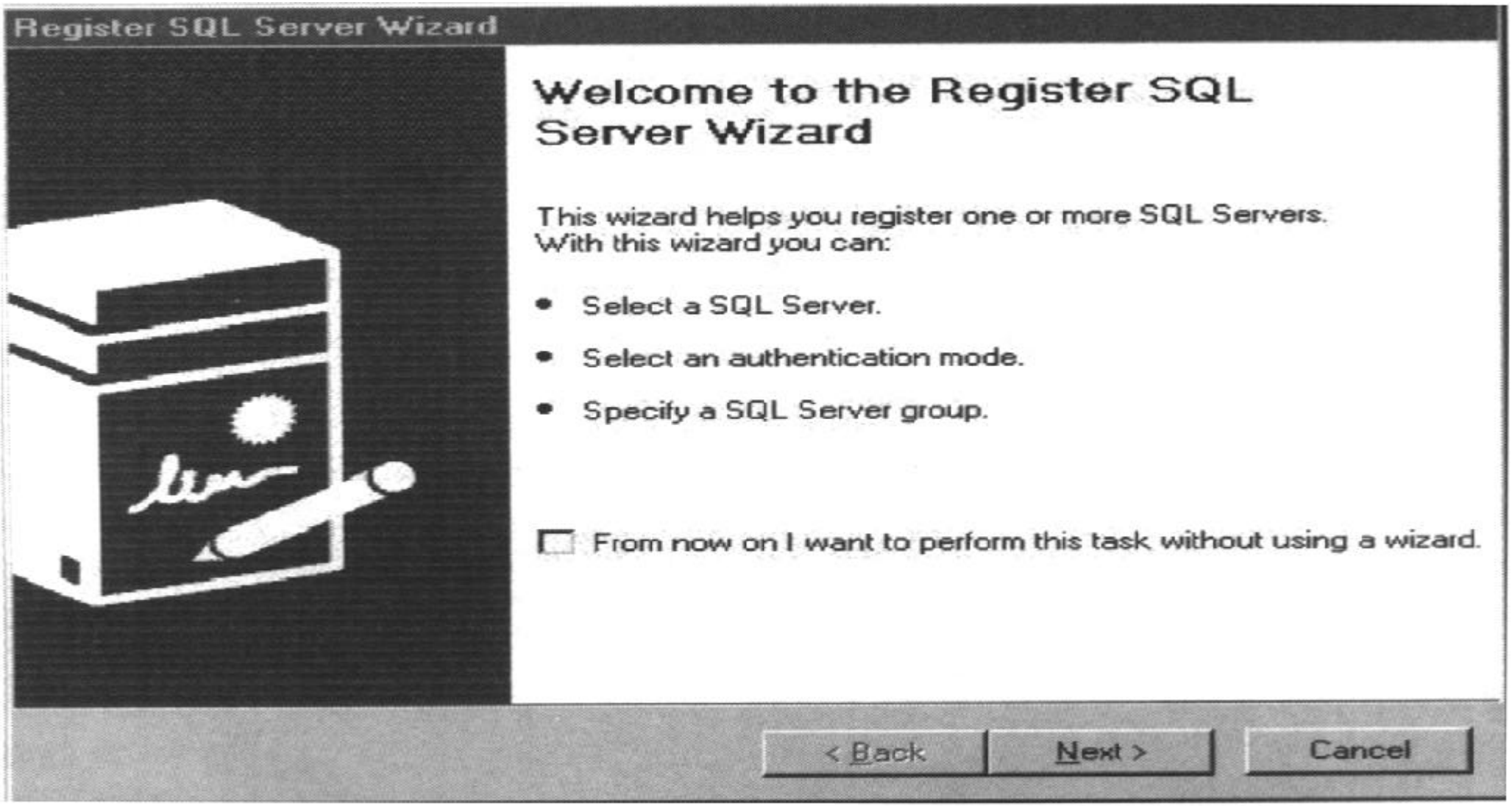


图 3.2Register SQL Server Wizard 的第一个屏幕

8. 选择已选中的连接选项。
9. 按下 Next 按钮。

10. 当显示 Select SQL Server Group 屏幕时，按下 Add 按钮，把＼＼Ultimate 服务器添加到 Complete Reference Group 中。
11. 按下 Next 按钮，进入下一个屏幕。
12. 按下 Finish 按钮。

至此就注册了一个服务器，并把它放到已创建的服务器组中。可继续进行，直到注册完所有的 SQL Servers 为止。一旦完成了这一步，就可利用丰富而功能强大的 GUI 工具管理 SQL Server 了。这些任务在命令行上执行或用 Transact-SQL 命令执行时要占用几十分钟的时间，而用 Enterprise Manager（即 Microsoft Management Console 上的咬接）执行只需几秒钟。

如果选择 SQL Server 后右击鼠标，可从弹出式菜单上选择 Edit SQL Server Registration。如果想将它们应用到服务器上，可检查其他三个用于服务器的选项：

- 在控制台上显示 SQL Server 的状态
- 显示 System Databases 和 System Objects
- 连接时自动启动 SQL Server

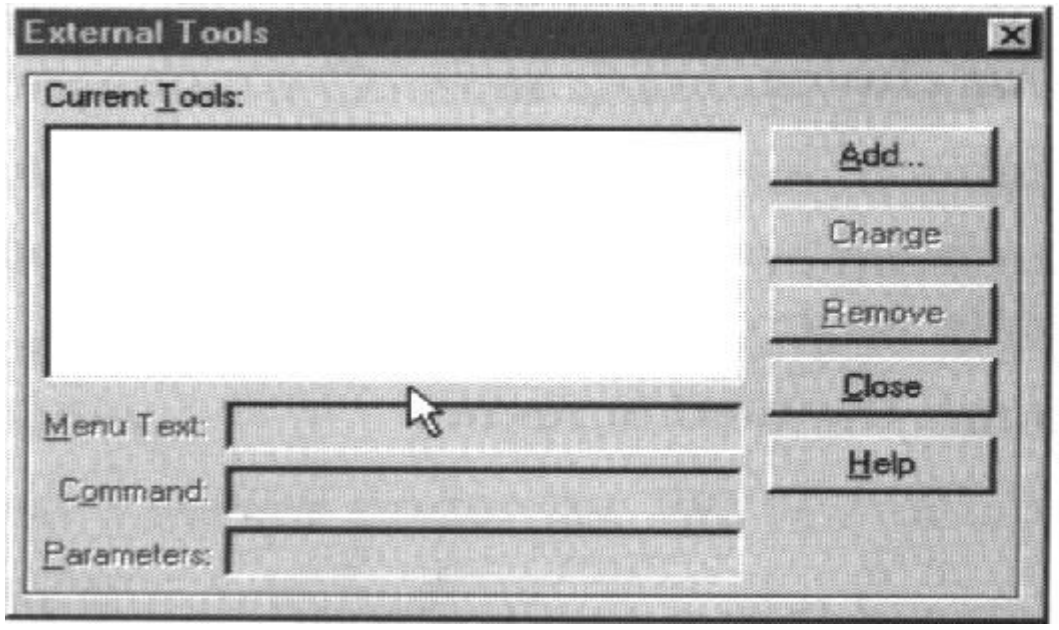
如果 Show System Databases 和 System Objects 复选框未被选中，系统数据库、master、msdb 和 model 在 Enterprise Manager 层次树形结构中都不会出现。

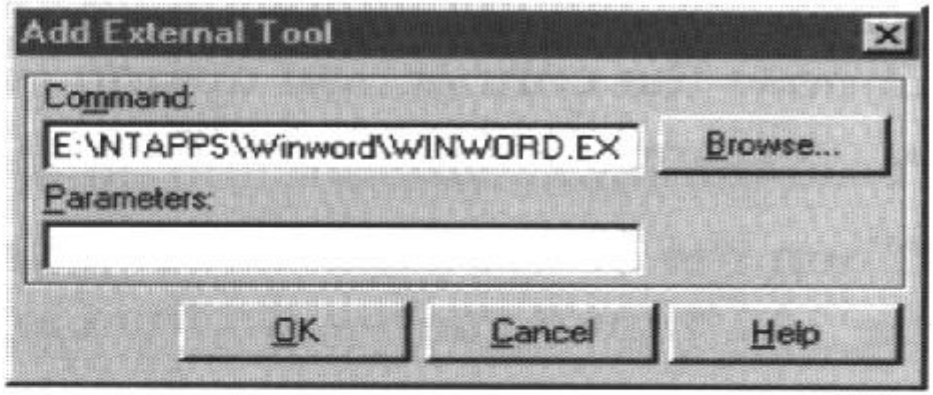
3.1.2SQL Server 与外部工具的集成

可以在 Microsoft Management Console 和 Enterprise Manager 内部运行任何 NT

应用程序。单击 Enterprise Manager 树中服务器名旁边的加号，可展开最新注册的 SQL Server。在 Microsoft Management Console 中选择 SQL Server，可访问 Enterprise Manager 的 Tools 菜单，并在此选择 External Tools。

从 Tools 菜单中选择 External Tools 允许增加外部工具，该外部工具可在 Enterprise Manager 内部调用。下面的对话框用来为 Enterprise Manager 增加外部工具。





在 Enterprise Manager 内部可以调用用户选中的 NT 应用程序工具，在这种情况下，只需点击某些菜单选项，浏览磁盘，查找出想在 Microsoft Management Console 中包含的应用程序，就可以增加 Microsoft Word NT 应用程序。现在就能选择和执行 Microsoft Management Console 中的 Microsoft Word 了。把喜欢的

其余工具增加到这个中心位置上，就可以在 Tools 菜单底部访问它们了。

在 Microsoft Management Console 内部，命令栏上的另一个菜单是 View。该菜单上的菜单项允许以不同的方式查看 Results Pane。如果选择 Large，Results Pane 将显示成如图 3.3 所示的大图标。其他视图选项还有 Small(小图标)、List(图标列表)、Detail(带有列标的列表，提供更详细的信息)和 Taskpad。

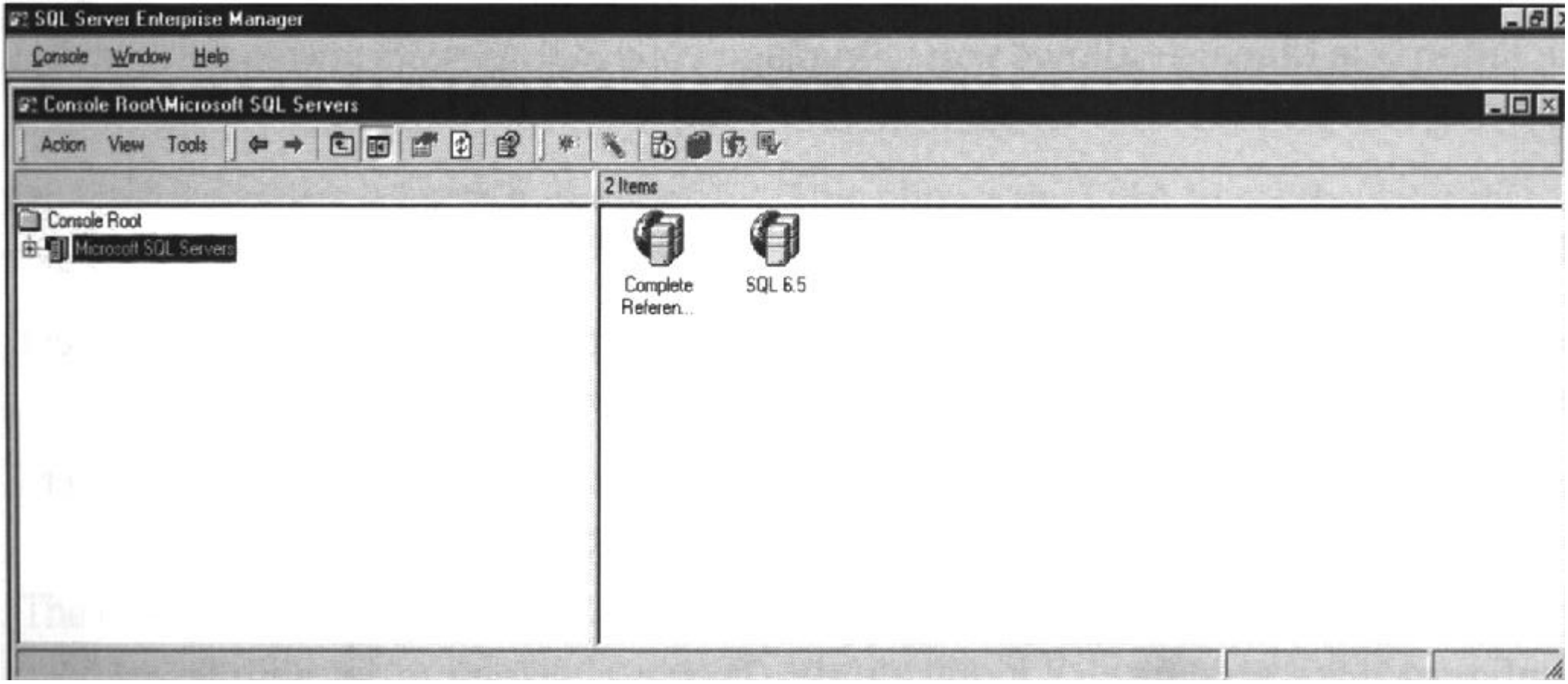


图 3.3 大图标视图中的 Results 面板

Taskpad 是 Enterprise Manager 的一个很有趣的视图，因为它可提供较多的工具和向导以便帮助处理管理数据库环境的过程。(当从 Enterprise Manager 选择 Tools、Wizard 时，可看到 22 个数据库、数据转换服务、管理和复制向导。)如果从 Enterprise Manager 的 TaskpadResults 面板中选择了 Setting Up Your Database Solution，如图 3.4 所示，屏幕上将出现下一个 Taskpad 格式化 Results 面板。Creating databases、Creating views、Creating indexes、Setting up security 和 Creating stored procedures 等 Taskpad 选项都是指导实现这些数据库任务的向导。

图 3.4Taskpad **视图中的** Results **面板**

3.2 服务器的 Enterprise Manager 功能

Enterprise Manager 允许用户管理自己的 SQL Server，并允许执行下列用于服务器的功能：

Enterprise Manager 服务器功能	注释说明
注册服务器(允许在 Enterprise Manager 层次树中看到它们)参见前面的文本和图例	如果使用 SQL Server 权限，指定系统管理员密码安装 SQL Server 后应立即这样做
配置本地服务器	这将更易于处理需要操心的少数几个配置选项，许多配置现在可由服务器自动进行
配置远程服务器	该功能运行良好，并允许 SQL 服务器共享数据并相互通讯
配置多重服务器环境	就像配置本地服务器一样，结果使 Enterprise Manager 名副其实
设置登录安全性	这会更全面地集成到 NT，并且使集成更加方便
创建警告	服务器对错误或事件执行某些动作，如寻呼和接发电子邮件
管理和配置 SQL Mail	SQL Server 能在数据库内部发送电子邮件，这样在完成一批任务后能够(并且很容易)用电子邮件发送报告或发送 SQL Server 警告信息
建立操作员	根据系统管理员安排的工作时间可以很容易地用电子邮件或寻呼机与他们联系

续表

为独立的环境创建和安排作业	该功能得到大大增强
为多重服务器环境创建和安排作业	该功能得到大大增强，包括多重服务器作业安排
创建和管理复制方案	复制功能已进行了许多重要的改进
为 SQL Server Enterprise Manager 设置轮询间隔	该功能允许 DBA 设置 Enterprise Manager 检查的间隔，以便查看远程服务器是否响应，它通过在计算机屏幕上把指示图标变绿或变红，来说明服务器是启动还是停止

下面几节详细解释了执行每项功能的过程。用 Enterprise Manager 管理 SQL Servers 是非常容易的。如果没有这个工具，DBA 就必须编写 Transact-SQL 代码，可能要采用某些第三代的其他编程语言来完成 Enterprise Manager 只需点击界面就可以完成的任务。

这个 GUI 数据库工具远远胜过市场上任何 UNIX 数据库工具或任何 PC 数据库工具。Enterprise Manager 是 DBA 的“专用”工具。这在使用户满意和从网络／层次数据库迁移到关系型数据库的易用性方面是一个巨大的改进。在成本和减少数据库复杂性方面也使用户受益匪浅。Enterprise Manager 代表了数据库工具发展的一个里程碑式的进步。

3.2.1 注册服务器

本章前面详细介绍了服务器的注册过程(参见“增加 Microsoft Management Console Snap-Ins”一节)。注册服务器是使 SQL Server 获得 Enterprise Manager 管理权限的第一步。Enterprise Manager 在允许用户使用 Enterprise Manager 工具集在 SQL Server 上进行操作之前将验证用户的身份。

注意：SQL Server 7 在 Microsoft Management Console Enterprise Manager 内部有一个多姿多彩的 Register Wizard，以帮助用户完成 SQL Server 注册的过程。

3.2.2 指定系统管理员密码

Enterprise Manager 允许用单击方式指定 System Administrator(SA)密码，在安装 SQL Server 之后应立即完成这一工作。如果还没有这样做，系统管理员密码为空值，任何人只需键入 sa 作为登录 ID，并使该密码保持空值，就可以登录 SQL Server，使用系统管理员的特权。

在安装 SQL Server 后立即指定密码就可避免这种安全性风险，这一点应成为书面过程的一部分。然后把密码保存在除了自己别人都找不到的地方，以便在以后需要以 SA 身份登录时不会忘记原来设置的密码。或者，如果用户位于工具装备良好的工作间中，就可以得到一个 SA 的工具，如果查询该工具，该

工具就会提供服务器的 SA 密码。也可以在操作安全的服务器上创建一个表格，并在忘记服务器的 SA 密码时查询该表格。这样比把密码写在一张纸上，放在工作台的某个抽屉里要安全得多。如果要管理大量的服务器，这样做是很必要的。下列步骤说明了改变 SA 密码是多么容易：

注意：相关区域位于 SQL Server 下的 Microsoft Management Console | Enterprise Manager 分层树状结构中。

1. 单击服务器名旁边的加号，展开 Enterprise Manager 分层树状结构中的 SQL Server。
2. 双击 Security 下的 Logins 项。
3. 右击如下所示的 sa 登录 ID。
4. 选择 Properties。
5. 键入 sa(系统管理员)登录 ID 的 SQL Server Authentication 密码，如图 3.5 所示。

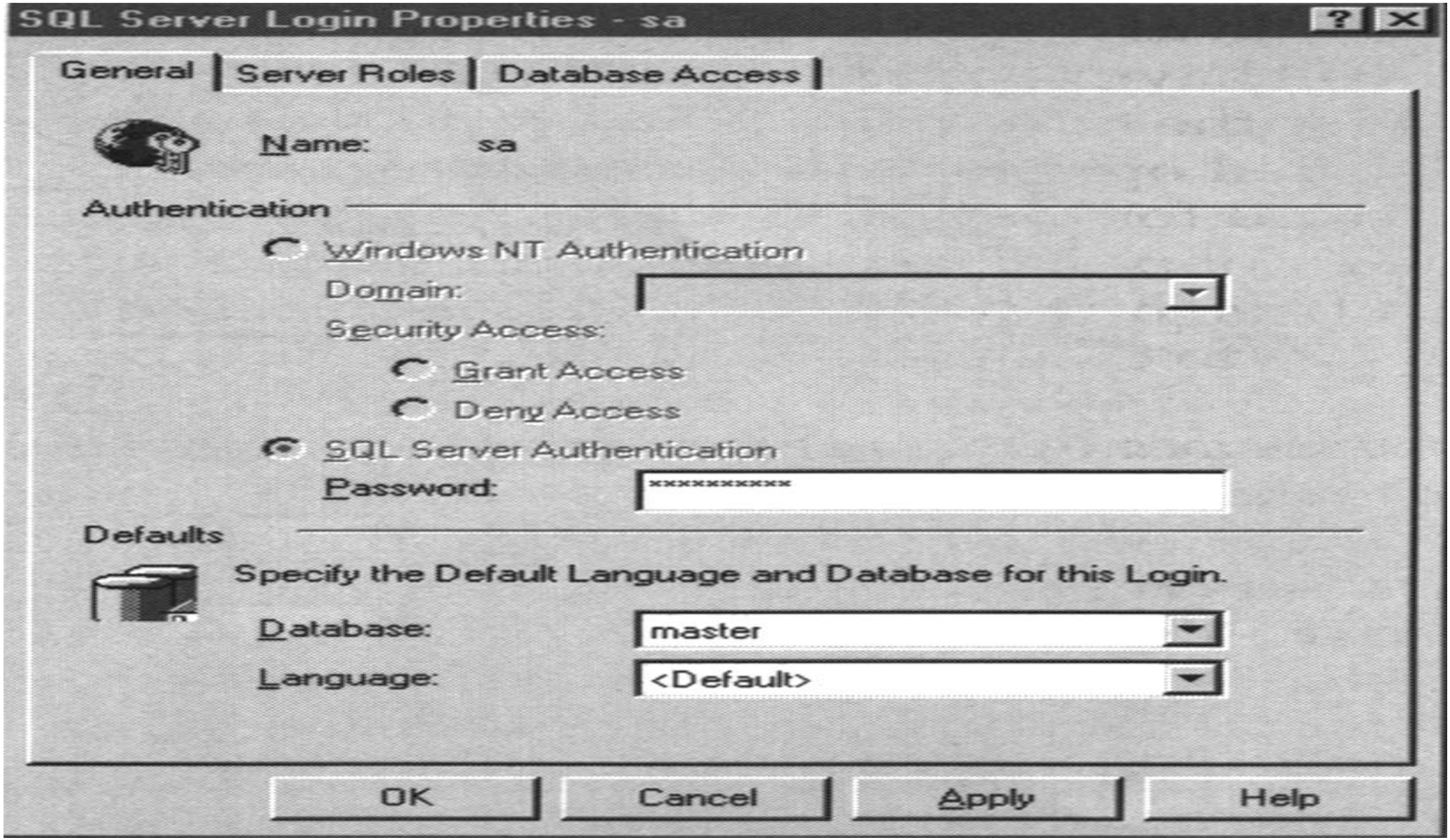


图 3.5 在 sa Properties 对话框中输入密码

6. 为确认密码，再次键入它。

3.2.3 配置本地服务器

配置服务器和数据库从来没有象这样简单。只需访问 Enterprise Manager 下的几个屏幕，修改几个栏目。唯一比较困难的部分是，应对 SQL Server 有充分的了解，知道改变数值后产生的结果。改变配置的实际操作是非常简单的。

注意：相关区域位于 SQL Server 下的 Microsoft Management Console | Enterprise Manager 分层树状结构中。

1. 单击 Enterprise Manager 下的服务器；(Local)表示名为＼＼Ultimate 的本地 SQL Server。
2. 右击服务器名(Local)，显示快捷菜单。
3. 从快捷菜单中选择 Properties。
4. 显示配置服务器的选项标签，如图 3.6 所示。

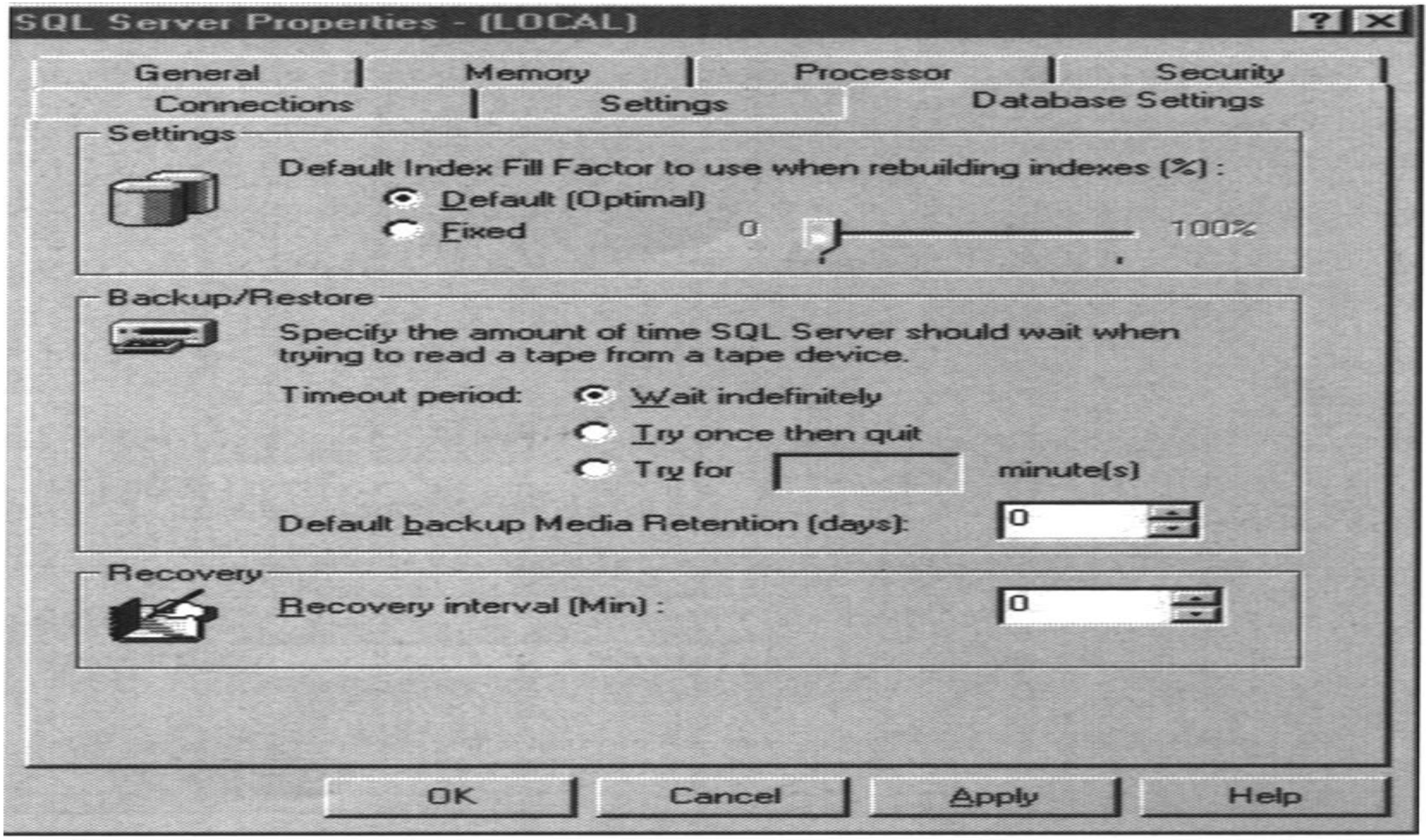


图 3.6SQL Server Properties **标签**

3.2.4 配置远程服务器

DBA 的 Enterprise 管理器的实际功能在此显示出来。配置远程服务器的过程与“配置本地服务器”一节介绍的过程完全相同。Enterprise Manager 必须能识别网络中的远程服务器，以便确定服务器的位置，并允许用户对它进行注册。注册远程服务器之后，同本地服务器一样，该远程服务器就出现在 Enterprise Manager 分层树状结构中。

3.2.5 配置多重服务器环境

多站点管理允许用中心 SQL Server 管理一组 SQL Server。这一技术的应用仅限于用户的想象，它允许创建作业和作业步骤、安排作业，以及改变作业步骤流程。改变作业步骤流程的方法是：成功时报告成功、失败时报告失败、失败时报告成功以及失败时报告成功，以及成功或失败“进入作业步骤”。失败或成功的记录存储在中心 SQL Server 上，在每个本地服务器上安排作业时，这一功能可大大减轻管理的负担。

Make Master Server Wizard 可指导用户创建多重服务器环境。只能在中心主服务器上没有其他 NT 应用程序运行时，才能将中心主服务器定义为管理服务器，并把事件发送到该服务器上。调用支持时创建目标服务器的操作和不支持的操作都称为偏离支持。主服务器也称为 MSX 主服务器。

主服务器上的 msdb 数据库包含 sysdownloadlist 表。该表包含一系列指令，如删除作业、插入作业、启动作业和改变目标服务器上的轮询间隔等。用 sp_post_msx_ operation，或用与作业相关的其他存储过程都可以把这些项目输入到该表中。

要查看主 SQL Server Agent 的错误记录，可右击 SQL Server Agent，选择 MultiSever Administration，再选择 Target Sever 标签上的 Manage Target Servers，然后选择某个服务器和 View Errorlog。

要检查目标服务器的状态，可右击 SQL Server Agent，选择 Multi Sever Administration，再选择 Target Server Status 标签上的 Manage Target Servers，然后选择某个服务器和 CheckState。

使用 SQL Server Enterprise Manager，系统管理员就可以右击 Enterprise Manager 分层树状结构中的 SQL Server Agent，并选择 Multi Server Administration，再选择 Make this a Master 或 Make this a Target，来定义多重服务器配置。可以将一个服务器指定为主服务器，该主服务器把作业、警告和事件消息发布给目标服务器组中的目标服务器。

建立多重服务器环境的概念是用于在目标组中放置服务器，此概念允许将特定作业应用到服务器的目标列表中。下述新存储过程与该功能相关。记住，我们正在讨论的是目标服务器组，而非 Enterprise Manager 分层结构中的用户组或服务器组。

注意:相关区域位于 SQL Server 下的 Microsoft Management Console | Enterpris e Manag-er | Tools Menu | Query Analyzer 分层树状结构中，在此可以

运行这些存储过程。

显示相关的系统存储过程和新系统表，有助于解释这个新概念及其目的。相关的新系统表包括 systargetservers、systargetservergroups 和 systargetserver-group-members，它们都位于 SQL Server 系统数据库 msdb 中。只有系统管理员才能完成这个任务。用户必须位于作为系统管理员注册的 Query Analyzer 的 msdb 数据库中，才能执行下列存储过程：

多重服务器存储过程名	功能说明	举例
sp_add_targetservergroup	增加目标服务器组	EXEC sp_add_targetservergroup ' Servers with Tax database '
sp_delete_targetservergroup	删除目标服务器组	EXECsp_delete_targetservergroup ' Servers with Tax database '
sp_help_ targetservergroup	列出指定目标服务器组中的服务器	EXECsp_help_targetserver ' Servers with Tax database '
sp_add_ targetsvrgrp_member	增加指定的组名，如果给定服务器名，将把指定的服务器增加到指定的组中	EXECsp_add_targetsvrgrp_member ' Servers with Tax database ' , ' UncleSam"
sp_delete_targetsvrgrp_ member	删除指定的目标服务器成员	EXECsp_add_targetsvrgrp_member ' Servers withTax database ' , ' UncleSam；AuntSam"

续表

sp_apply_job_to_targets	在当前作业还未应用的服务器中增加作业	EXECsp_apply_job_to_targets ' Tax table updatejob ' , ' Servers with Tax database ' , ' UncleSam ' , ' APPLY '
sp_remove_job_from_targets	从作业已经应用的服务器中删除指定的作业	EXECsp_remove_job_from_targets ' Tax tableupdate job ' , ' Server with Tax database ' , ' UncleSam,AuntSam '
sp_update_targetservergroup	改变目标服务器组的名称	EXECsp_update_targetservergroup ' Server with Tax database ' , ' Tax Servers '

在 msdb 数据库的 Query Analyzer 中运行的下列选择语句，显示了如何把新目标服务器组的新行添加为存储目标服务器组的系统表中的一行。

```
use msdb
go
select * from systargetservergroups
go
servergroup_id name
------------------------------
1Servers with Tax database
```

(1 row(s) affected)

下图表示在增加目标服务器组后，查询 systargetservergroups 系统表时，Query Analyzer 的输出结果。

3.2.6 设置登录安全性

Enterprise Manager 允许更方便地管理服务器、数据库和表格的安全性特性。通过定义登录 ID 的数据库访问，可以控制访问，如图 3.7 所示。验证的两种类型分别是 NT Authentication 和 SQL Server Authentication。本书第 5 章将详细介绍这方面的内容。

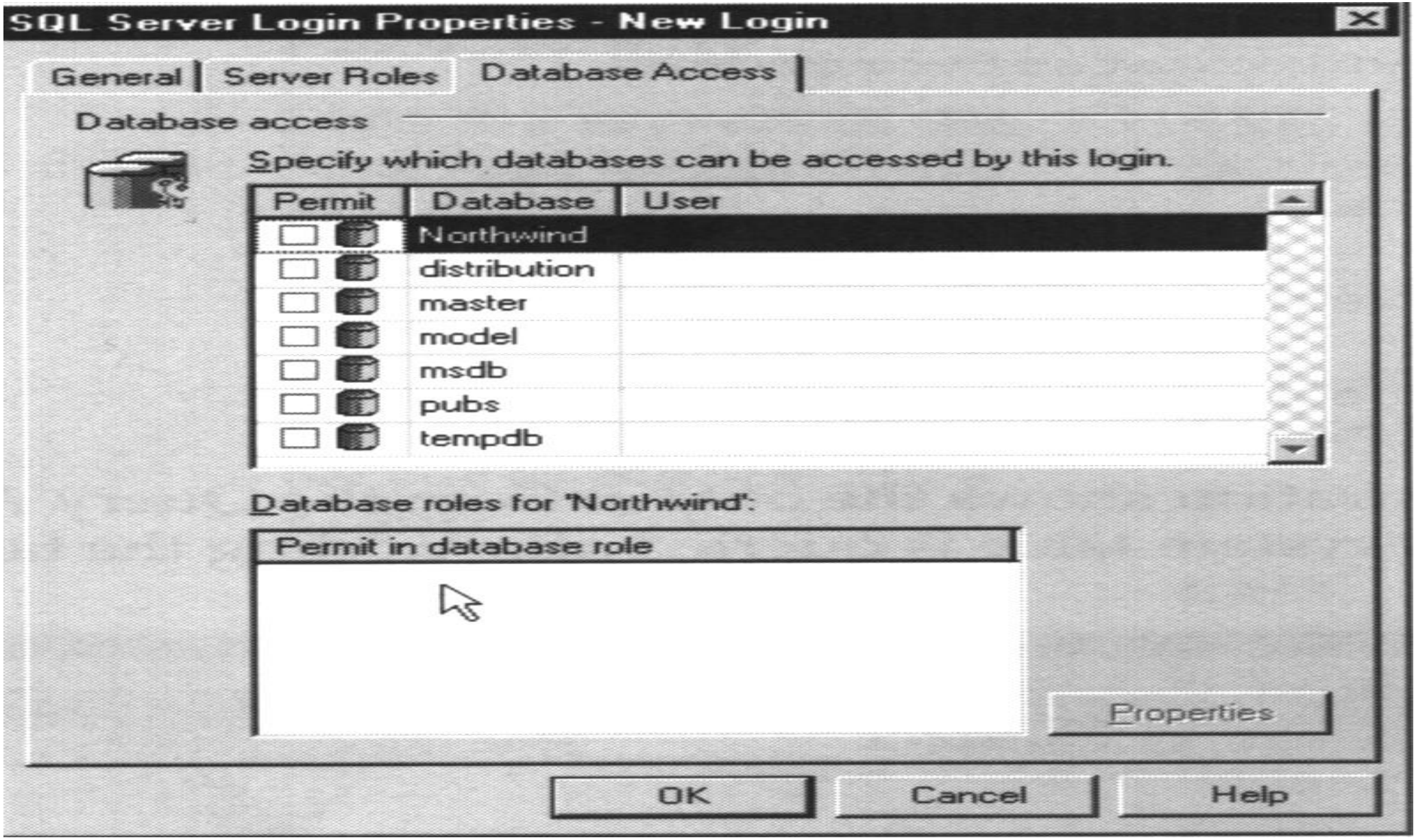


图 3.7 把数据库访问登录到 SQL Server

下面的步骤说明如何用 Enterprise Manager 增加新的登录 ID。

注意：相关区域位于 SQL Server 下的 Microsoft Management Console | Enterprise Manager 分层树状结构中。

1. 单击 SQL Server 下的 Enterprise Manager 分层树状结构中想要的服务器名旁边的加号(＋)，展开服务器。
2. 选择 Logins 并右击，显示快捷菜单。
3. 从快捷菜单中选择 New，显示 SQL Server Login Properties 屏幕，里面有三个标签。
4. 选择 General 标签，如图 3.8 所示。

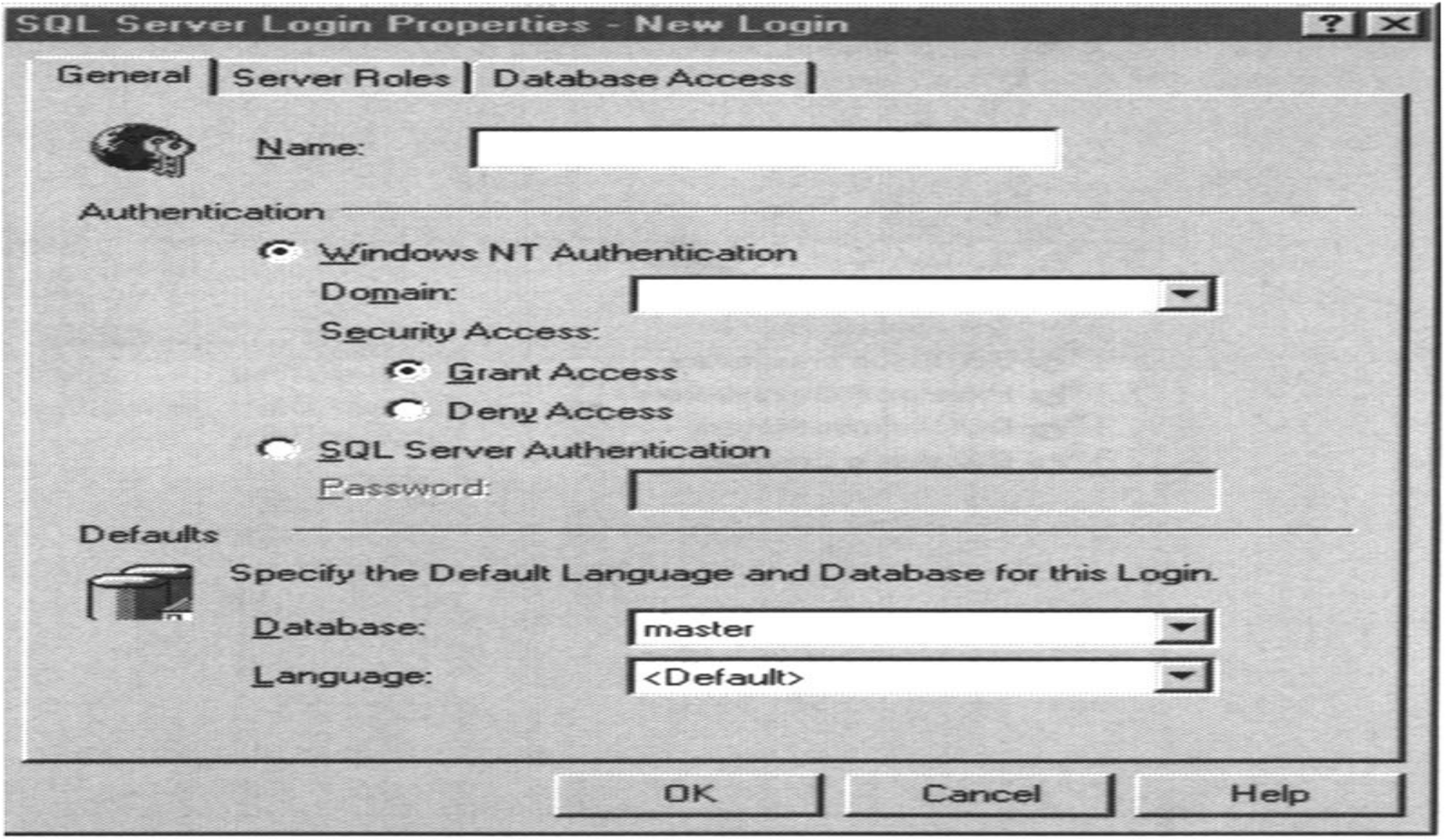


图 3.8SQL Server Login Properties **对话框的** General **标签**

5. 选择想要的验证，即 NT Authentication 或 SQL Server Authentication 二者之一。

6. 如果选择了 SQL Server Authentication，就输入登录 ID 和口令。

7. 如果选择了 NT Authentication，就输入 SQL Server 用于委托连接的 NT 帐户和域。

下面的步骤说明了如何在 Server Role 中增加新创建的登录。角色(Role)将在第 5 章详细介绍。

1. 单击 SQL Server 下的 Enterprise Manager 分层树状结构中想要的服务器名旁边的加号(＋)，展开服务器。

2. 选择 Logins 并双击，显示已有的登录。

3. 右击想要的登录。

4. 从快捷菜单中选择 Properties，显示 SQL Server Login Properties 屏幕，里面有三个标签。

5. 选择 Server Roles 标签，如图 3.9 所示。

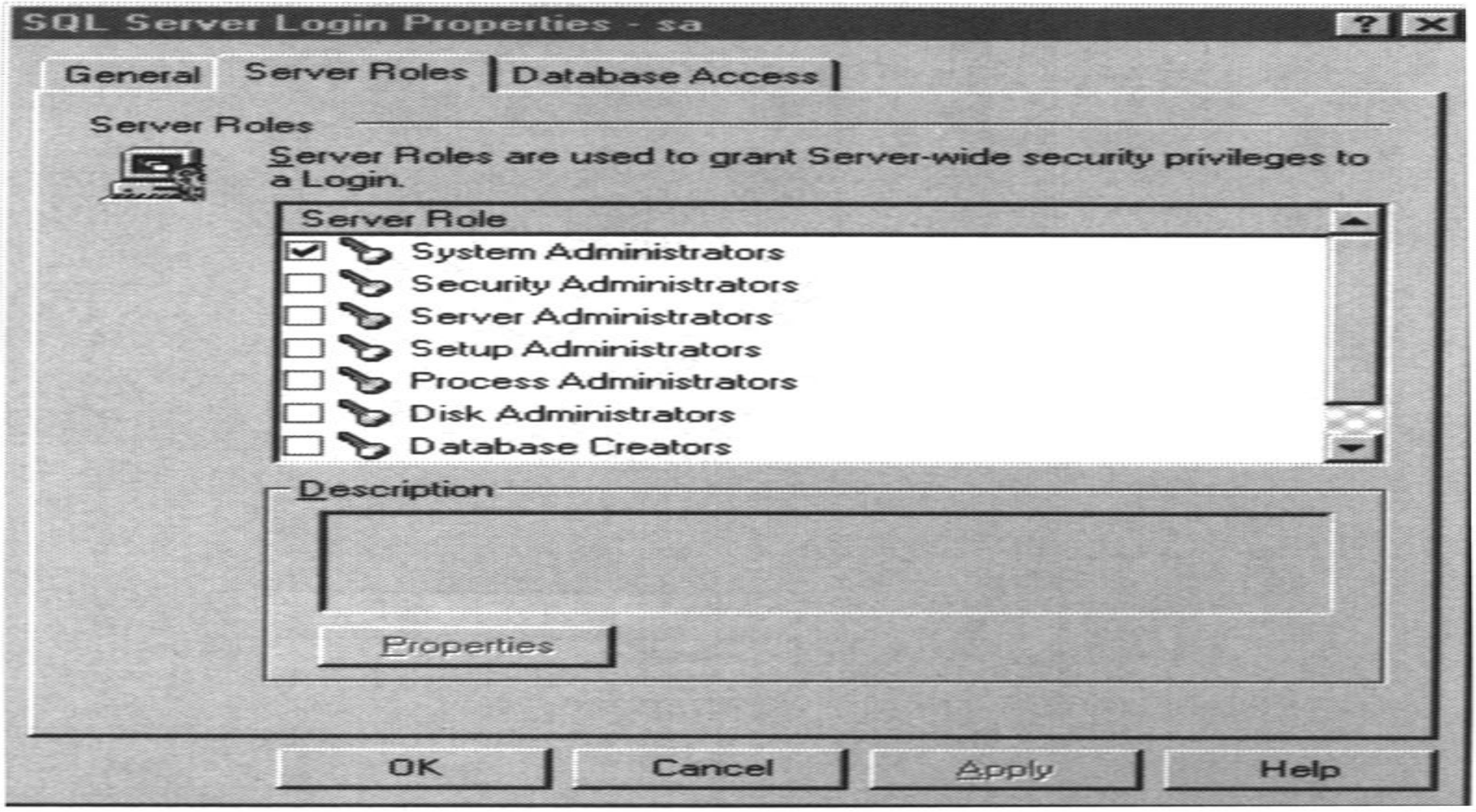


图 3.9sa Properties 对话框的 Server Role 面板

6. 为该登录单击想要的 Fixed Server Role 复选框或用户自定义的 Server Role 复选框。

下面的步骤说明了如何为该登录 ID 提供数据库访问权。用如图 3.10 所示的屏幕可以很容易指定数据库访问。在此可以指定对数据库的访问，并把该登录分配给 Database Roles。

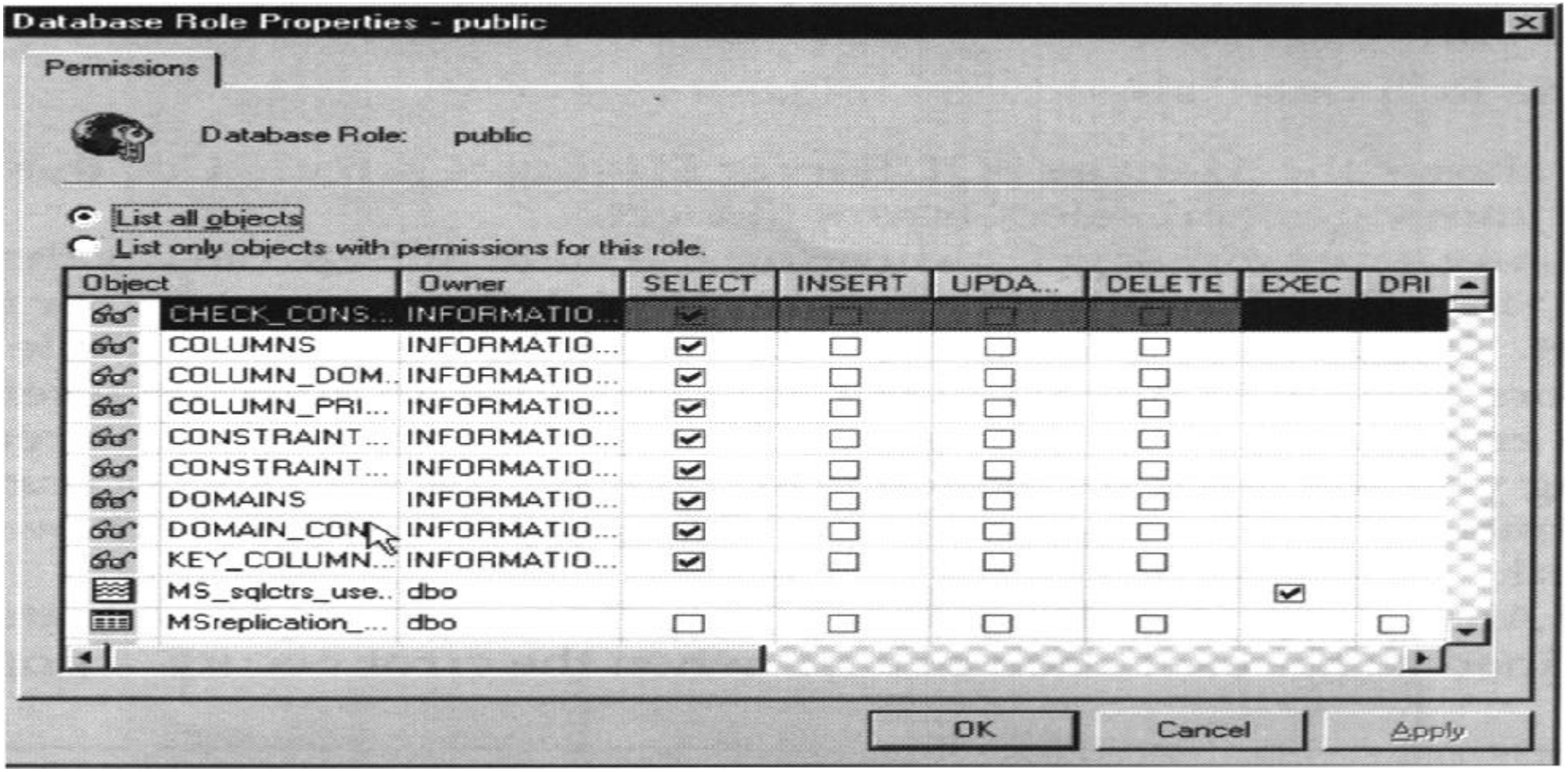


图 3.10 要访问 Permissions 面板，可依次单击 Database Access 标签、Properties 按钮和 Permissions 按钮

1. 单击 SQL Server 下的 Enterprise Manager 分层树状结构中想要的服务器名旁边的加号(＋)，展开服务器。
2. 选择 Logins 并双击，显示已有的登录。
3. 右击想要的登录。
4. 从快捷菜单中选择 Properties，显示 SQL Server Login Properties 屏幕，里面有三个标签。
5. 选择 Database Access 标签。
6. 单击想要的数据库的复选框。
7. 为该登录单击 Database Role 复选框或用户自定义的数据库角色复选框。

3.2.7 创建警告

在 SQL Server 7 中，警告操作员出现 SQL Server 错误的警告的创建过程非常简单。要用 Enterprise 管理器创建新的警告，可执行以下步骤：

1. 单击 SQL Server 下的 Enterprise Manager 分层树状结构中想要的服务器名旁边的加号(＋)，展开服务器。
2. 选择已展开的服务器名下的 SQL Server Agent 项。
3. 单击 SQL Server Agent 旁边的加号，展开 SQL Server Agent。
4. 单击 Alert。
5. 右击 Alert。
6. 从快捷菜单中选择 New Alert，显示 New Alert Properties 屏幕。

7. 输入新警告的名称。

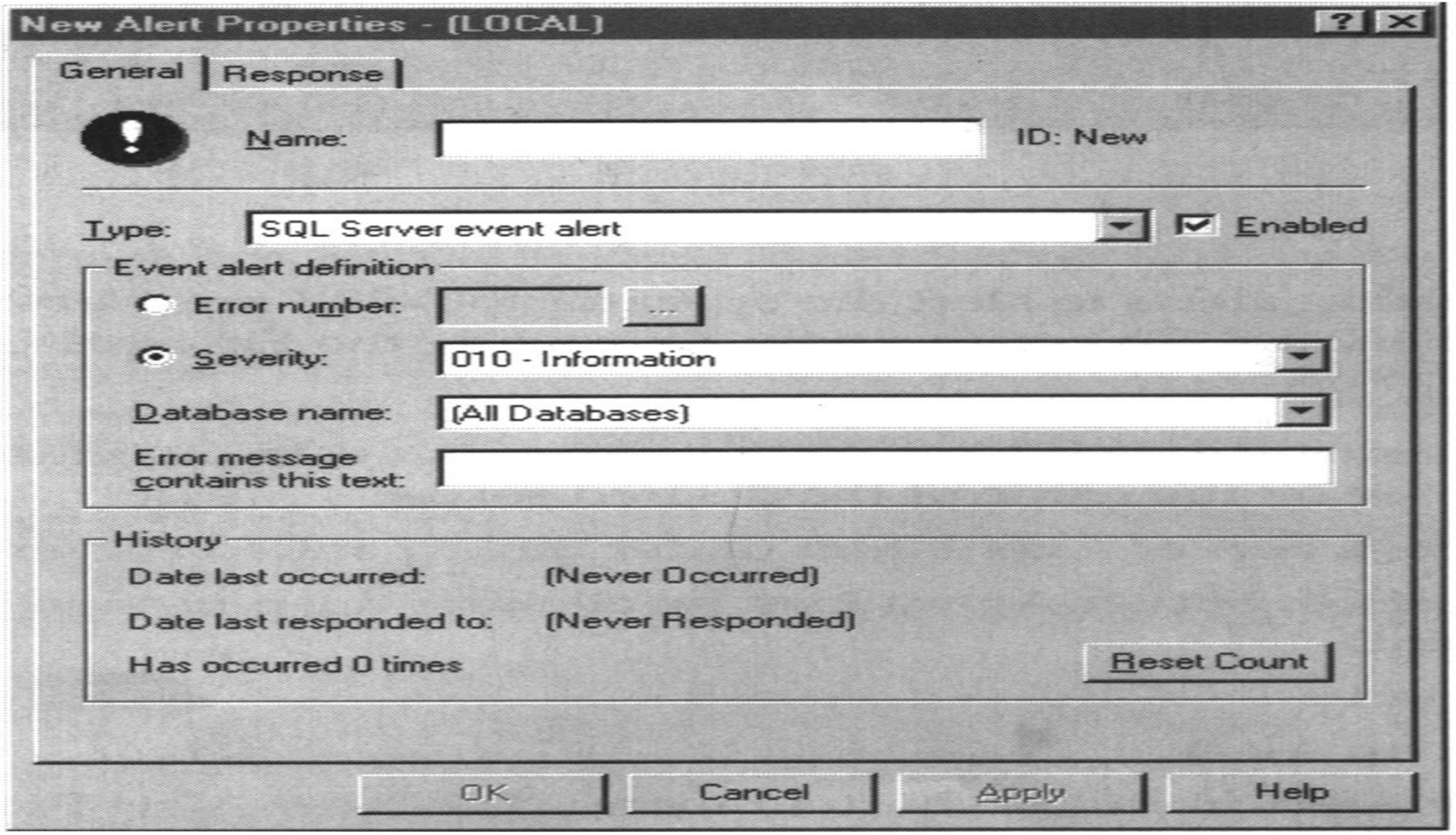


图 3.11 在 Enterprise Manager 中，依次右击 SQL Server Agent、New 和 Alert

8. 选择 Find 并输入选择条件，单击标有...的按钮。显示 SQL Server 中的已有警告列表，如图 3.11 所示。

9. 把游标放到错误上，按下 OK 按钮。这样可把该错误号放到 New Properties Alert 屏幕中。

10. 选择 Response 标签。

图 3.11 显示了 Manage SQL Server Messages 屏幕。使用该屏幕可搜索错误消息，选择要警告的错误。

有许多错误消息指明存储在 sysmessages 表中的 SQL Server 状态。Find 列表允许选取定义警告的消息错误号，它通常是 sysmessages 表中的错误号。只有写到 NT 事件日志中的 sysmessages 错误才能发送警告。

警告一个错误意味着，当发生错误时，用户将接收到一个页面或 e-mail，说明发生了已经设置警告的某个错误，然后可以选择执行某个作业。图 3.12 所示的 Response 屏幕展示了完成这个工作是很容易的。

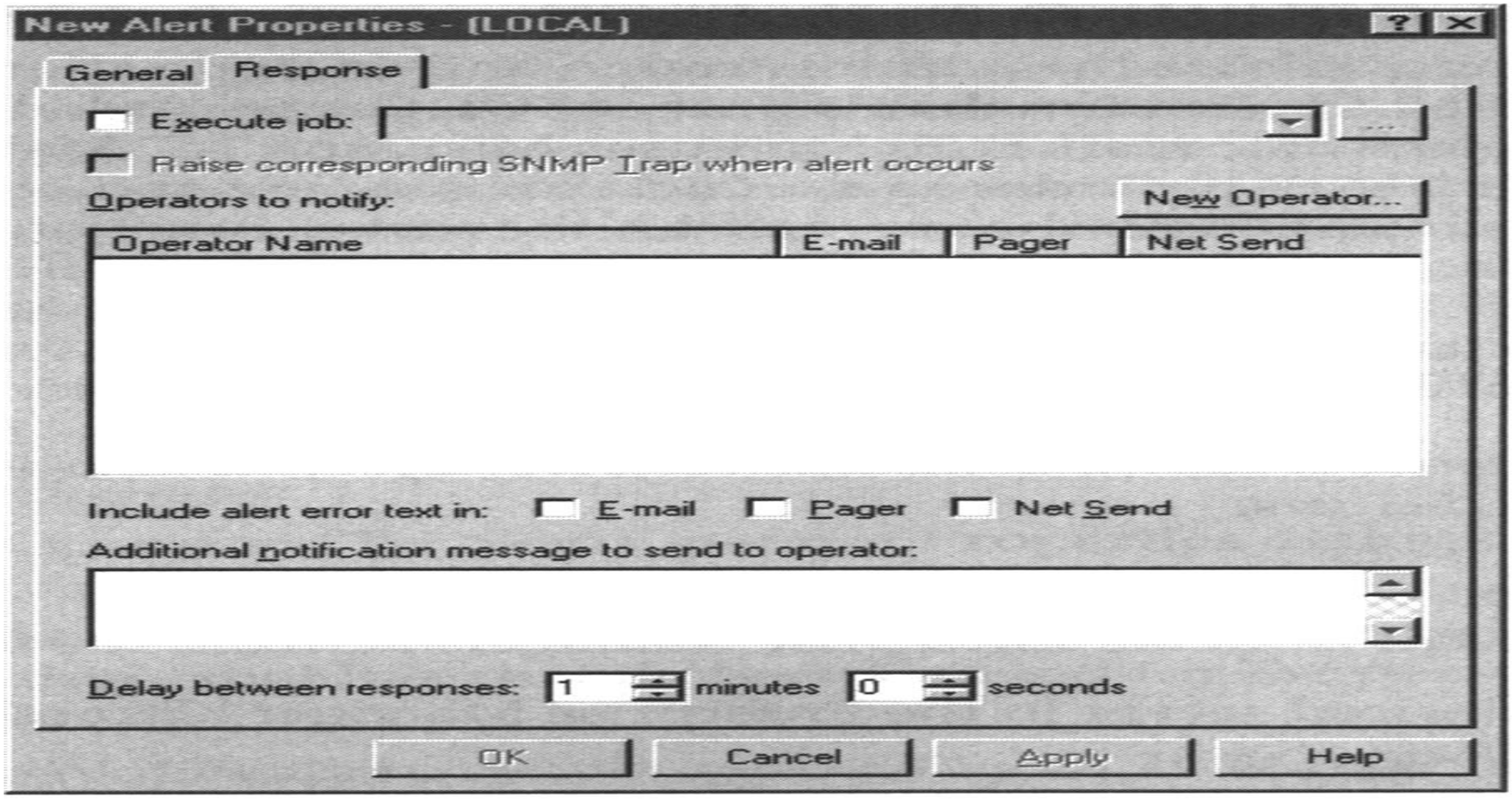


图 3.12New Alert Properties 对话框的 Response 面板

在 New Alert Properties Response 面板上，有一个非常好的特性，当把响应定义为出现错误而执行某个作业时，弹出式消息通知用户没有作业，并询问是

否要增加作业。如果回答要增加，就显示增加作业的屏幕。

注意:如果警告不正常，可查看 SQL Server Agent NT 服务是否正常运行，警告功能是否打开，事件是否记录在 NT 事件日志中。

不需用 Enterprise Manager 创建警告，就能执行下述 msdb 系统数据库中的存储过程。

```
USE msdb
GO
EXEC sp_add_alert @name = '  Unexpected Shut Down' , @message_id=
17061,@severity = 0, @ notification_message = '  Error 17061 has
occurred.The server,Ultimate, has experienced an unexpected shut
down...' ,@ job_name = '  Initialize After Unexpected Shut Down'
GO
```

3.2.8 管理和配置 SQL Mail

Enterprise Manager 允许管理和配置 SQL Mail。SQL Mail 是支持 MAPI 的应用程序，是 SQL Server 的组成部分，它允许应用程序发送电子邮件。SQL Mail 用 Microsoft NT Mail、Microsoft Exchange 或其他 MAPI 供应商作为发送和接收邮件的载体。Microsoft NT Mail 安装在服务器上，作为 Mail 客户机来发挥服务器的功能。SQL Mail 的安装将在第 12 章“SQL Mail”中介绍。SQL Mail 有许多用途，包括服务器上发生警告或事件时发送 e-mail。也可以用它在预定的作

业完成后发送报告。消息可以包含文本串、查询结果或文件形式的附件。

注意:相关区域位于 Microsoft Management Console | Enterprise Manager | SQL Servers 中。

3.2.8.1 管理

要管理 Enterprise Manager 中的 SQL Mail，可执行下面的步骤：

1. 展开 Enterprise Manager 分层树形结构中想要的服务器。
2. 右击 SQL Mail。
3. 选择启动或停止 SQL Mail。

3.2.8.2 配置

要配置 Enterprise Manager 中的 SQL Mail，可执行下面的步骤：

1. 展开 Enterprise Manager 分层树形结构中想要的服务器。
2. 右击 Support Services 下的 SQL Mail。
3. 选择 Properties，显示如下图所示的 SQL Mail Configuration 屏幕：
4. 输入有效的邮件登录 Profile 名。

3.2.9 建立操作员

Enterprise Manager 允许执行建立或定义操作员的服务器功能，操作员在服务器上发生预定义错误时可收到自动下达的通知。SQL Server 7 的这种功能会简化操作员的定义，包括名称、e-mail 名、寻呼 e-mail 名、网络发送地址，以及当该操作员可用时代表工作日开始时间和结束时间的寻呼值班时间表。

定义操作员的步骤如下：

1. 展开所要的 Enterprise Manager 分层树结构中的服务器。
2. 展开 Management 下的 SQL Sever Agent。
3. 右击 Operator。
4. 单击 New Operator，显示 New Operator Properties 对话框，如图 3.13 所示。

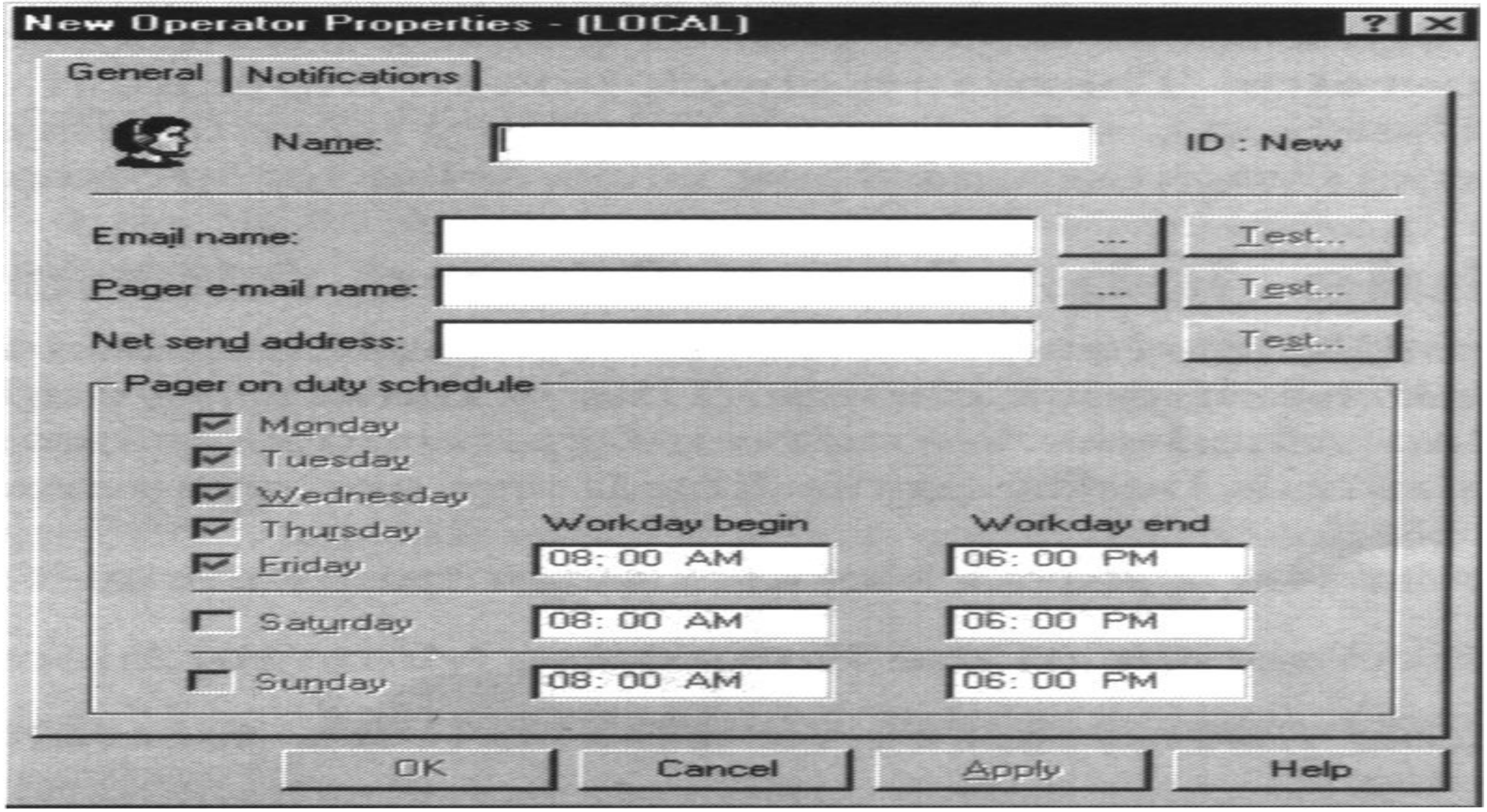


图 3.13New Operator Properties **对话框的** General **面板**

5. **在** General **标签中，输入已为指定的新操作员定义的** e-mail**、寻呼** e-mail

和网络发送地址等信息。

6. 单击 Notification 标签，如图 3.14 所示，输入新操作员的通知时间。图 3.14New Operator Properties 对话框的 Notifications 面板

屏中包含一些按钮，用于测试为操作员选定的通知方法。复选框指明该操作员能否接收到通知，这样，在操作员度假、生病或其他临时事假时，就很容易取消通知。

3.2.10 为独立环境创建和安排作业

Enterprise Manager 允许在独立环境下创建和安排作业，以便在选择的时间运行。如果在线用户和对资源敏感的作业之间对资源的强占减少时，数据库上还有用户，并想将对资源敏感的作业安排在下班时运行，这样做是非常方便的。

在独立环境下创建作业的步骤如下：

1. 展开想要的 Enterprise Manager 分层树状结构中的服务器。
2. 展开 Management 下的 SQL Server Agent。
3. 右击 Jobs。
4. 单击 New Job。
5. 键入作业名，最多可达 100 个字符。
6. 从登录 ID 的下拉列表框中选择负责该作业的拥有者，缺省选择是系统管理员。只有 SA 才能指定作业的拥有者。
7. 键入作业说明，最多可达 512 个字符。

8. 单击 Steps 标签。
9. 单击 New。

每个作业都必须有一个步骤，所以必须在 New Job 步骤屏幕中填写信息，包括键入该作业的命令。可以输入 Transact-SQL 命令。New Job 步骤屏幕中的 Advanced 标签允许输入一个输出文件名，在作业成功或失败时要执行的操作，以及在作业运行的用户许可下要执行的操作。

至此，新的作业已创建好了，就可以执行如下步骤，安排该作业的运行时间：

1. 展开所要的 Enterprise Manager 分层树状结构中的服务器。
2. 展开 SQL Server Agent。
3. 单击 Jobs，在 Result Pane 中显示可用的作业。
4. 右击希望安排的作业。
5. 从快捷菜单中选择 Properties。
6. 选择 Schedules 标签，如图 3.15 所示。
7. 单击 New Schedule。
8. 在 Name 域输入作业名。

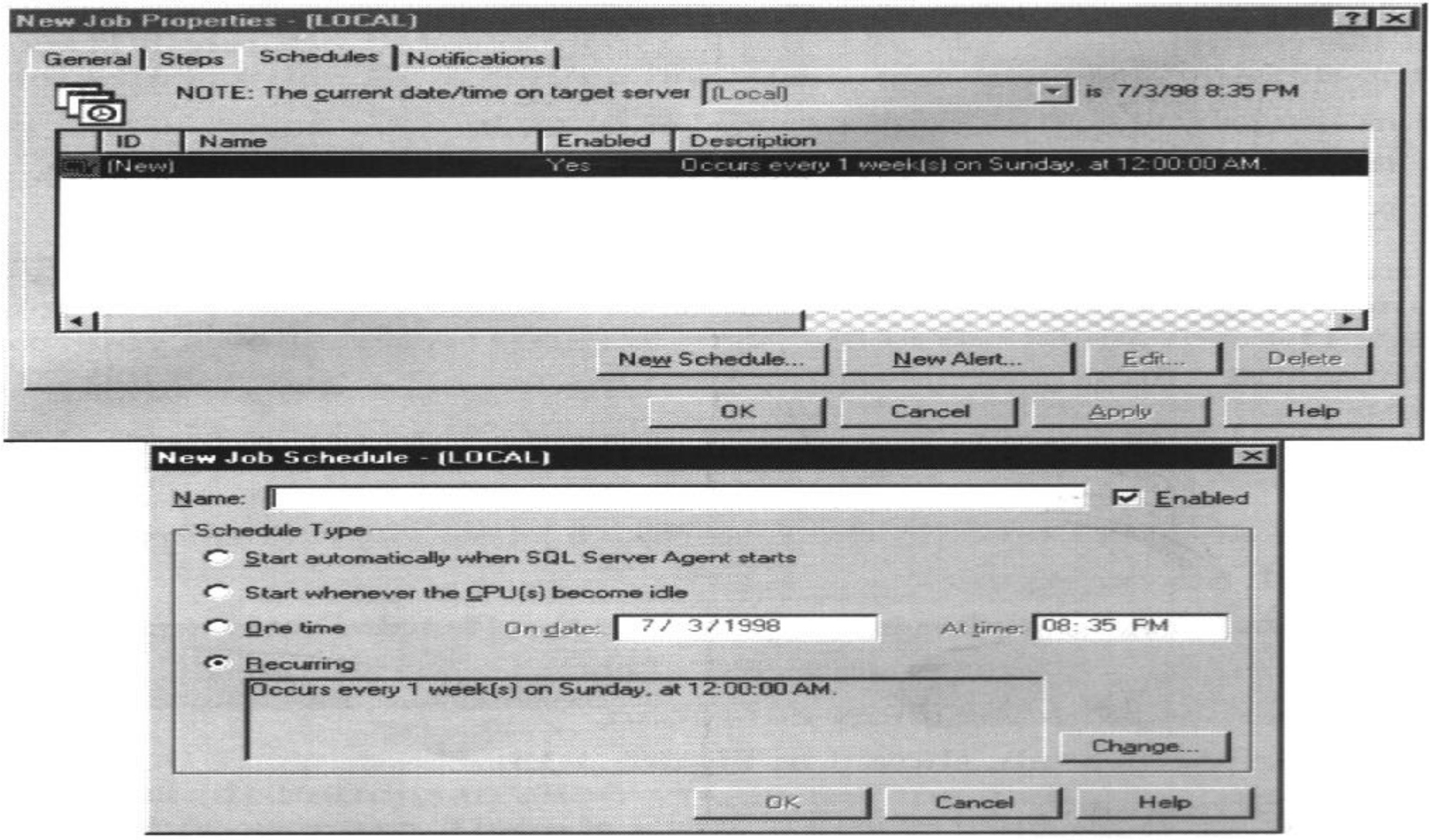


图 3.15 单击 New Schedule 按钮访问 New Job Schedule 对话框

9. 指出何时想从可用的选项中运行作业：SQL Agent 启动时自动启动；只要 CPU 一空闲就启动，一次启动或重复启动。

注意 Enabled 复选框，它允许在不删除时间表的情况下关闭它。新作业步骤的功能非常重要，也非常强大，它可以模拟多年前 PC 环境中没有，而只包含在大型机环境中的功能。Enterprise Manager 不需要编写 Job Control Language(JCL)，而是提供了一个方便易用的点击界面，该界面可以用较少的时间和精力建立作业和作业步骤。

3.2.11 为多重服务器环境创建和安排作业

为多重服务器环境创建作业与在独立的 SQL Server 环境下创建作业所使用的步骤基本相同，二者的主要差别在于，在多重服务器环境下，必须使用以前创建的目标服务器组来决定在多重服务器环境下有哪些服务器(请参见“配置多重服务器环境”一节)。确定这些服务器的屏幕是 New Job Properties 屏幕，以及 Target 本地服务器或 Target 多重服务器的单选按钮选项。如果选择了 Target 多重服务器，屏幕上会显示目标服务器组列表，从中可以选择目标服务器组。

对作业的改变还有一些区别和复杂性，如步骤、时间表或作业与服务器关系的改变等，它们都需要把这些改变记录到 SQL Server Agent 下载列表中。此列表在方法上类似于批处理，可提供通知改变目标服务器的数据，用这些改变执行作业。SQL Server Agent 是下一章讨论的主题。

在多重服务器环境下，用户能够定义主 SQL Server，并使目标服务器与 SQL

Server Agent下载列表同步,在下载列表中可把变化记录到作业中。主 SQL Server Agent把作业和作业步骤的结果记录在本地目标服务器的系统表 sysjobhistroy 和主服务器的系统表 sysjobservers 上。当多重服务器环境中的作业改变时使目标服务器同步，可以确保作业改变在支持服务器的作业指令中。

SQL Server Agent的全部功能和作业的创建将在第四章中详细介绍。

3.2.12 创建和管理复制情形

在 Microsoft Management Console 下运行的 Enterprise Manager 也允许创建和管理复制过程。在任何工作站上都可以用 Enterprise Manager 在应用程序所需数目的服务器之间进行复制。在 Enterprise Manager 中有一个控制复制过程的内置工具 Replication Monitor。另外,任务列表、任务队列和任务历史都可在 Replication Monitor 内部显示和管理，Replication Monitor 可在复制任务中设置警告，以便在发生某个事件时通知操作员，它也用来在分发器上解决疑难问题。Replication Monitor 有一个分发中心，这意味着在把服务器定义为复制分发器时，只能在 Enterprise Manager 分层树形结构中看到它。单击所需的服务器,从命令栏的 Tools 菜单中选择 Replication，就可以使服务器成为复制分发器。

提示:这个向导可以从 Microsoft Management Console | Enterprise Manager | SQL Servers | Tools 的 Tools 菜单中调用。该向导将要求把本地计算机选为分发器，或把另一台注册的服务器配置成分发器。

复制将在第 9 章“Microsoft SQL Server 7 的复制增强”中详细介绍。

3.2.13 为 SQL Server Enterprise Manager 设置轮询间隔

SQL Server Enterprise Manager 允许设置检查 SQL Server、SQL Server Agent、SQL Agen-t、Replication Monitor 和 Distributed Transaction Coordinator 状态的时间间隔，看看它们是否仍在运行。Scope Pane 中的图标指明服务器是在运行还是没有运行。为此，可执行下面的步骤：

1. 展开 Enterprise Manager 分层树形结构中所需的服务器。
2. 从命令栏中的 Tools 菜单上选择 Options。
3. 选择 General 标签，如图 3.16 所示。

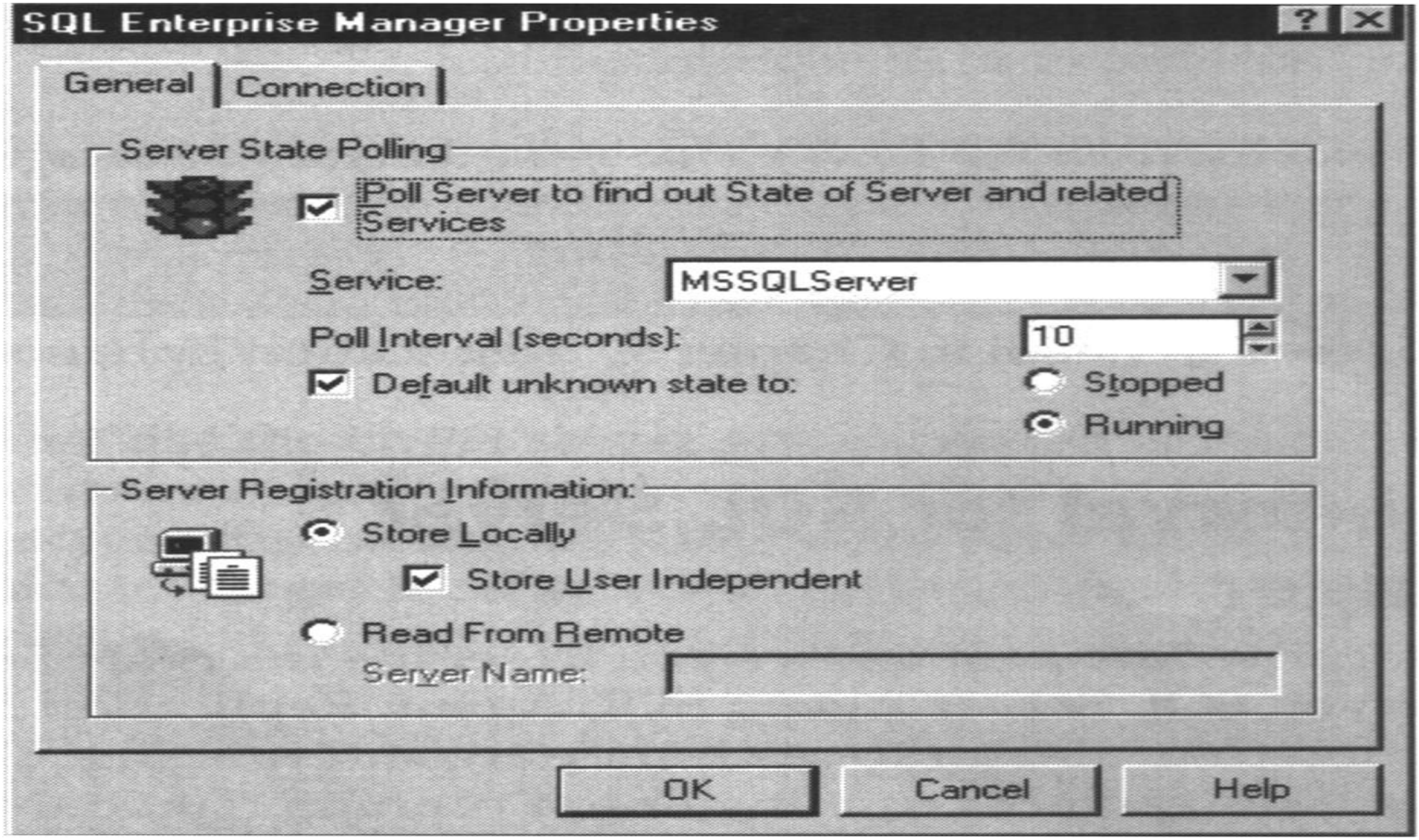


图 3.16 **选择** Enterprise Manager、SQL Servers、Tools **和** Options

4. 服务 SQL Server、SQL Server Agent、SQL Agent、Replication Monitor 和 Distributed Transaction Coordinator 都出现在下拉列表框中。

5. 为上述屏幕中的每个选项设置轮询间隔秒数。

服务器当机轮询间隔的情况稍有差别。它是检查服务器当机的轮询时间间隔，并使 SQL Agent 自动重启 SQL Server。这在另一个屏幕上设置。要设置服务器当机的轮询间隔，可执行下面的步骤：

1. 展开 Enterprise Manager 分层树形结构中所需的服务器。
2. 在 Management 下，为该服务器选择 SQL Server Agent。
3. 右击 SQL Server Agent。
4. 从快捷菜单中选择 Properties。
5. 单击 Advanced 标签，如图 3.17 所示。

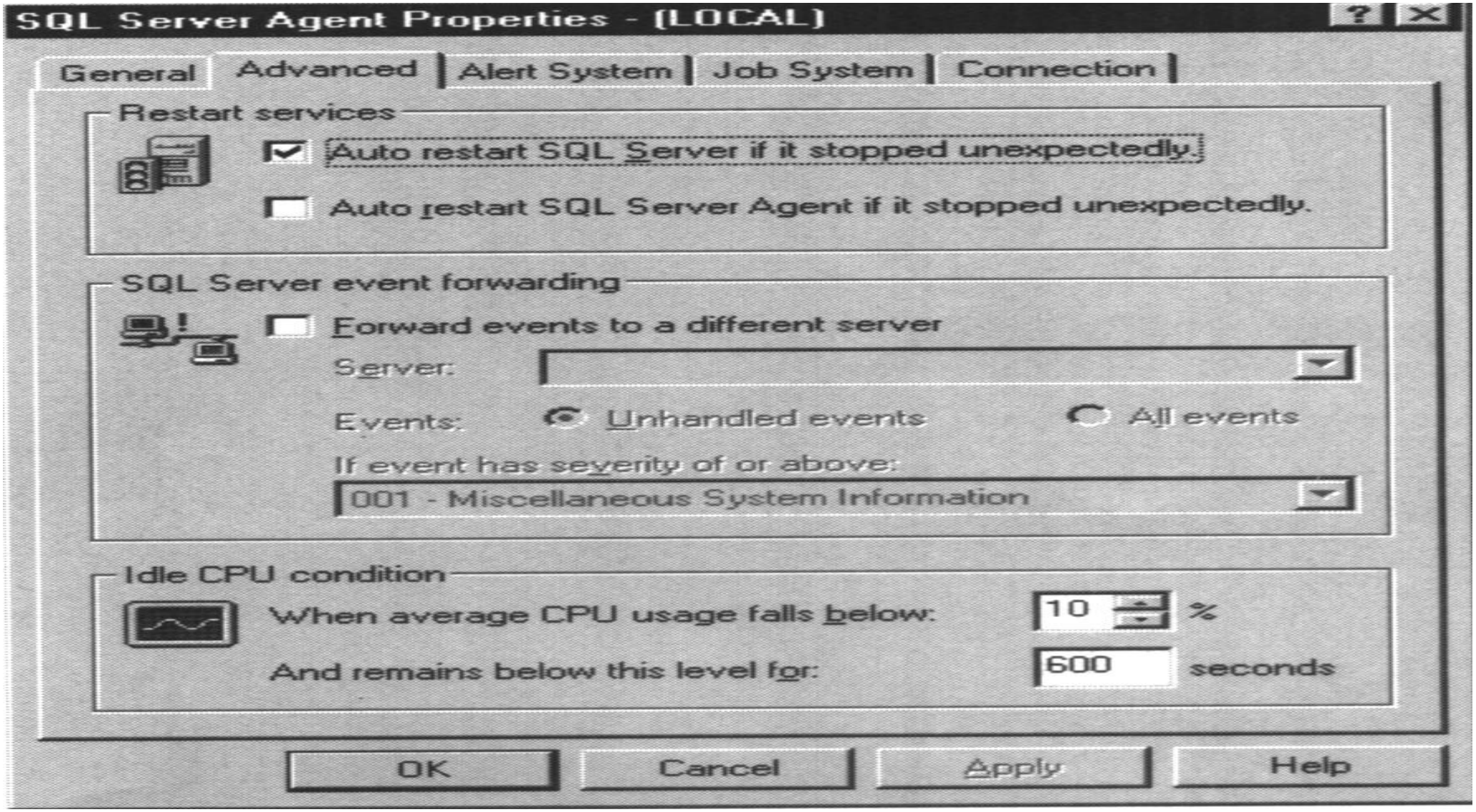


图 3.17 **选择** Enterprise Manager、SQL Servers、SQL Agent(**在** Management **下**)、Properties **和** Advanced

6. 根据用户希望每隔多久检查该服务器是否当机，以便 SQL Agent 重启服务器，设置轮询间隔时间(秒)。

除了用于服务器的功能之外，Enterprise Manager 还提供配置和管理数据库的许多功能。

Enterprise Manager 是数据库管理员和 SQL Server 开发人员的高级工具。在数据库管理工具的发展过程中，只有 Microsoft Management Console Enterprise Monitor 的专业性功能才提供了标准化的用户界面和易用性。

第 4 章 SQL Server Agent

SQL Executive 已更名为 SQL Server Agent，可提供下列功能：

- 安排作业
- 执行作业
- 创建警告
- 通知操作员

注意：相关区域位于 Microsoft SQL Servers | SQL Agent 下的 Microsoft Management Console | Enterprise Manager 分层树状结构中。

SQL Server Agent 是一种 NT 服务，它和 SQL Server 一起运行，向 SQL Server 传送作业调度和警告。所有的调度信息都包含在 msdb 系统数据库中，在安装 SQL Server 时该数据库会自动创建 msdb 系统数据库。SQL Server 的这种扩展增强了许多性能，包括多重服务器作业的管理。图 4.1 表示 Server Manager 服务列表中的 SQL Server Agent 项，图 4.2 表示 Enterprise Manager 层次列表中的 Create Job Wizard。

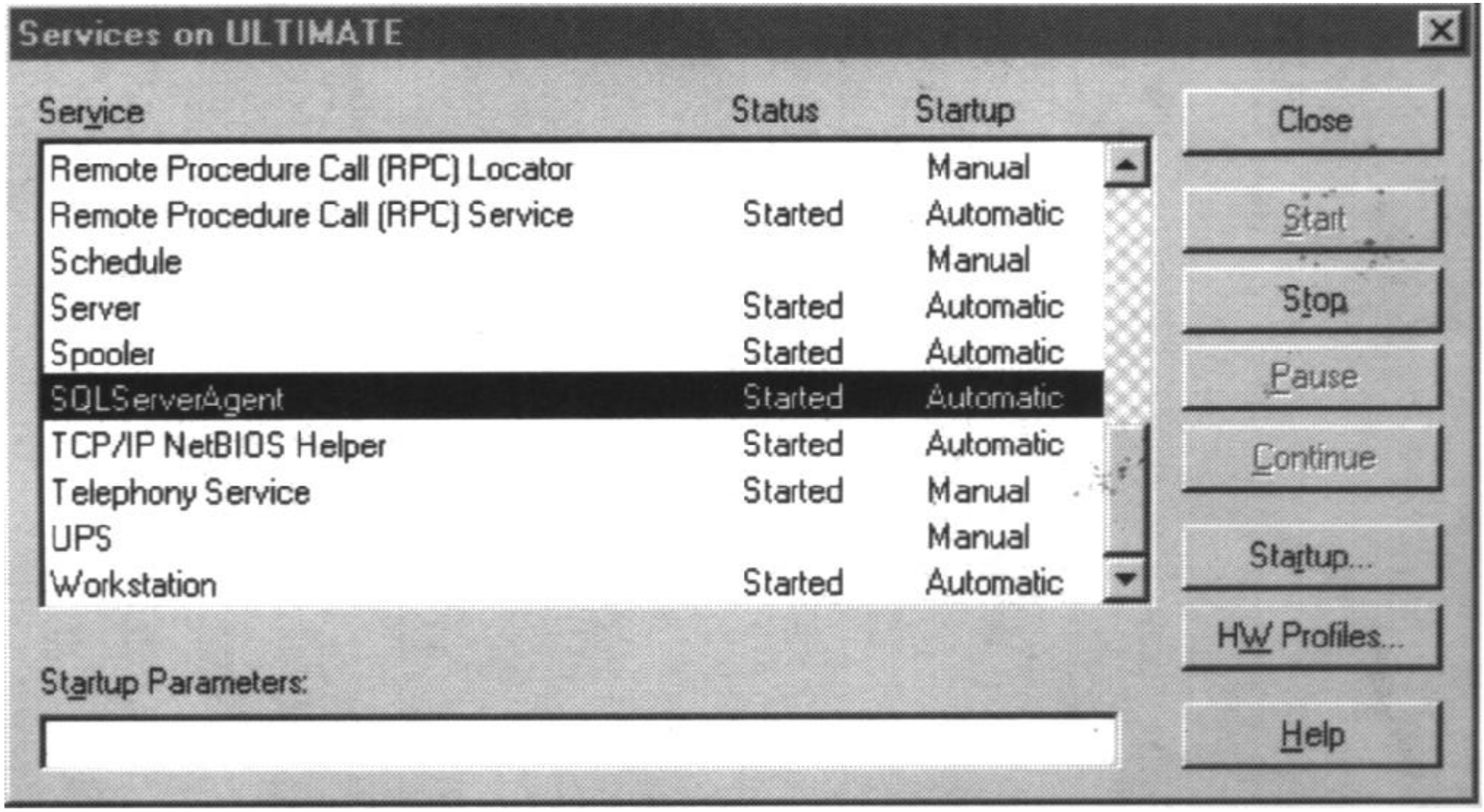


图 4.1 要查看 NT Services Manager 中的层次列表，可进入 Computer 菜单，并选择 Services | SQL Server Agent

注意:该有关区域位于 NT Service Control Manager｜Computer 菜单｜Services 中。

访问 SQL Server Agent 有两种方法:1)Enterprise Manager，它允许创建作业、操作员和警告；2)NT Service Control Manager，用它可手动启动和停止服务，并提供了 SQL Server Agent 的启动配置。

在研究了三个 Enterprise Manager 类别后，将会看到 NT Service Control Manager。SQL Server Agent 下的三个类别分别是:

- Operators(操作员)
- Jobs(作业)
- Alerts(警告)

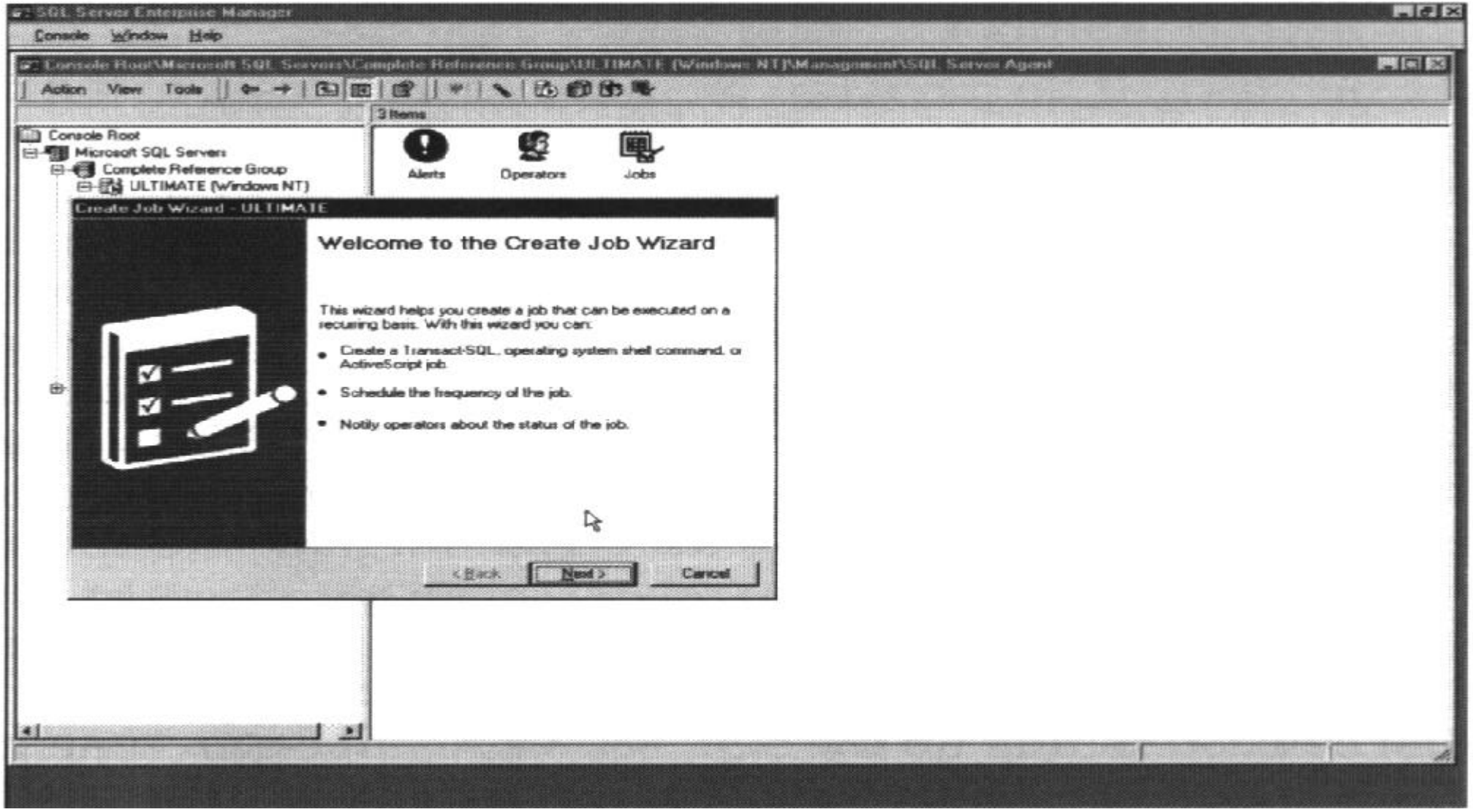


图 4.2 在 MMC 中，选择 Enterprise Manager、Tools、Wizards，然后选择 Create Job Wizard

4.1 定义操作员

SQL Server Agent被创建为一项附加功能，以辅助大规模的数据库操作。SQL Server Agent中的功能可使其安排是管理4个数据库还是4000个数据库。数据库操作员的定义和内置的警告机制是非常有用的。

Database Operators是负责使终端用户能访问服务器和数据库的群体，是数据库操作的关键。当用户访问SQL Server数据库中的数据而发生中断时，就通知数据库操作员。而且一旦发生服务中断，通知操作员往往至关重要。SQL Server Agent将操作员与服务中断相关联，来提供这项服务，方法是在发生错误或失败时创建定制警告，该警告设法通过e-mail、电话寻呼服务或网络消息以通知操作员。通常用计算机警告操作员而不是数据库用户，数据库用户通常只在匆忙完成工作时才发现问题。并不是要数据库用户指出要给谁打电话，并等操作员来找出并解决问题，而是让计算机根据发出寻呼的警告寻呼操作员。只要操作员能访问网络连接，在许多情况下，甚至可以在用户认识到数据的访问发生中断之前就找出并解决该问题。

建立SQL Server Agent环境的前几步之一是定义操作员。为此，可执行以下步骤：

1. 单击Microsoft Management Console(也称为MMC)的Console Root下的Microsoft SQL Server的项。
2. 展开已创建好的Complete Reference Group服务器组。

3. 展开希望使用的服务器。
4. 展开 Management 下的 SQL Server Agent。
5. 右击 Operators。
6. 从快捷菜单上选中 New Operator。
7. 显示 New Operator 屏幕，如图 4.3 所示。

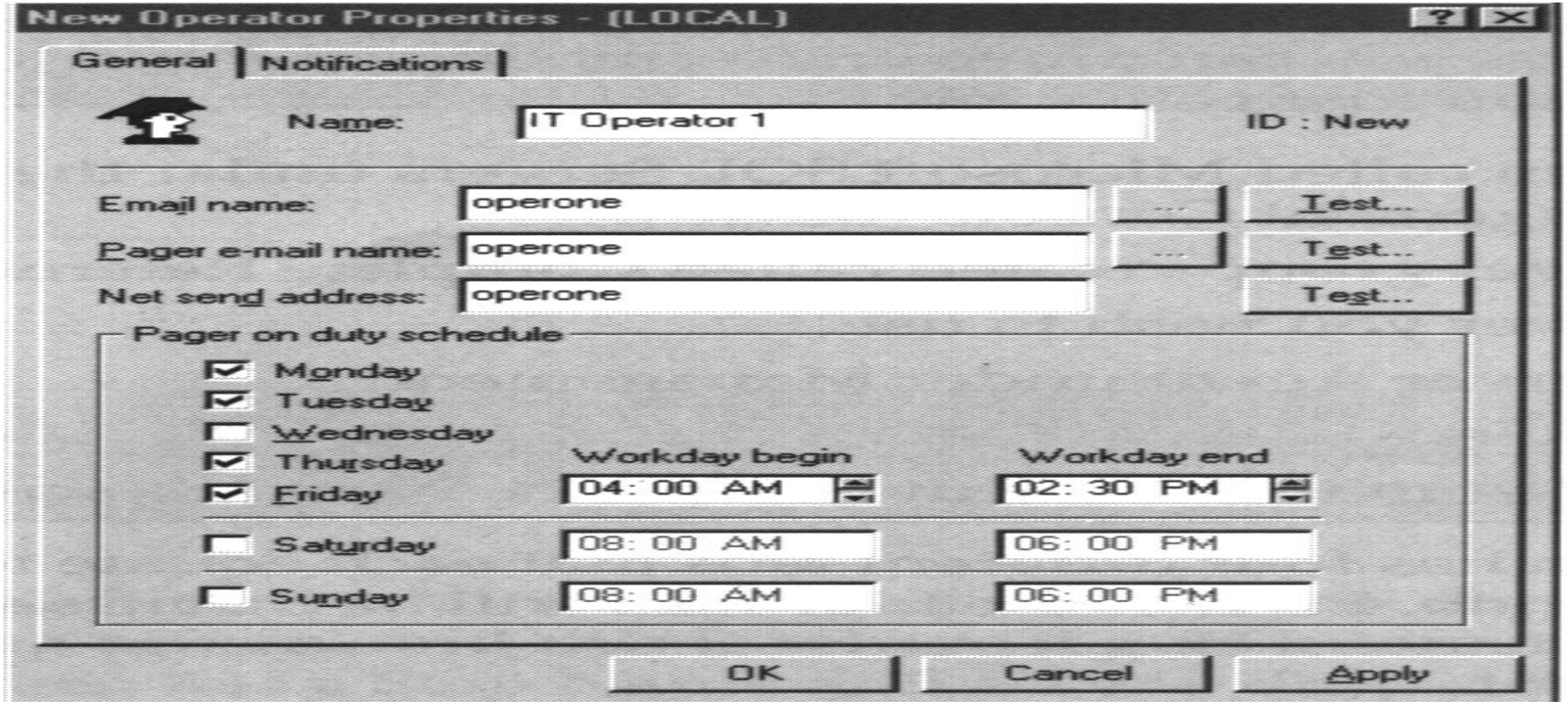


图 4.3 在 MMC 中，选择 Enterprise Manage | Microsoft SQL Server | SQL Agent | Operator | New Operator | General 标签

New Operator 屏幕允许输入名称、e-mail 名和寻呼 e-mail 名，并允许指定操作员接收通知的时间顺序。

New Operator 对话框中的 Notification 标签，用于确定应通知操作员错误的严重程度和通知的方法。甚至还有这样的选择，允许在操作员度假、生病时取消通知，如图 4.4 所示。

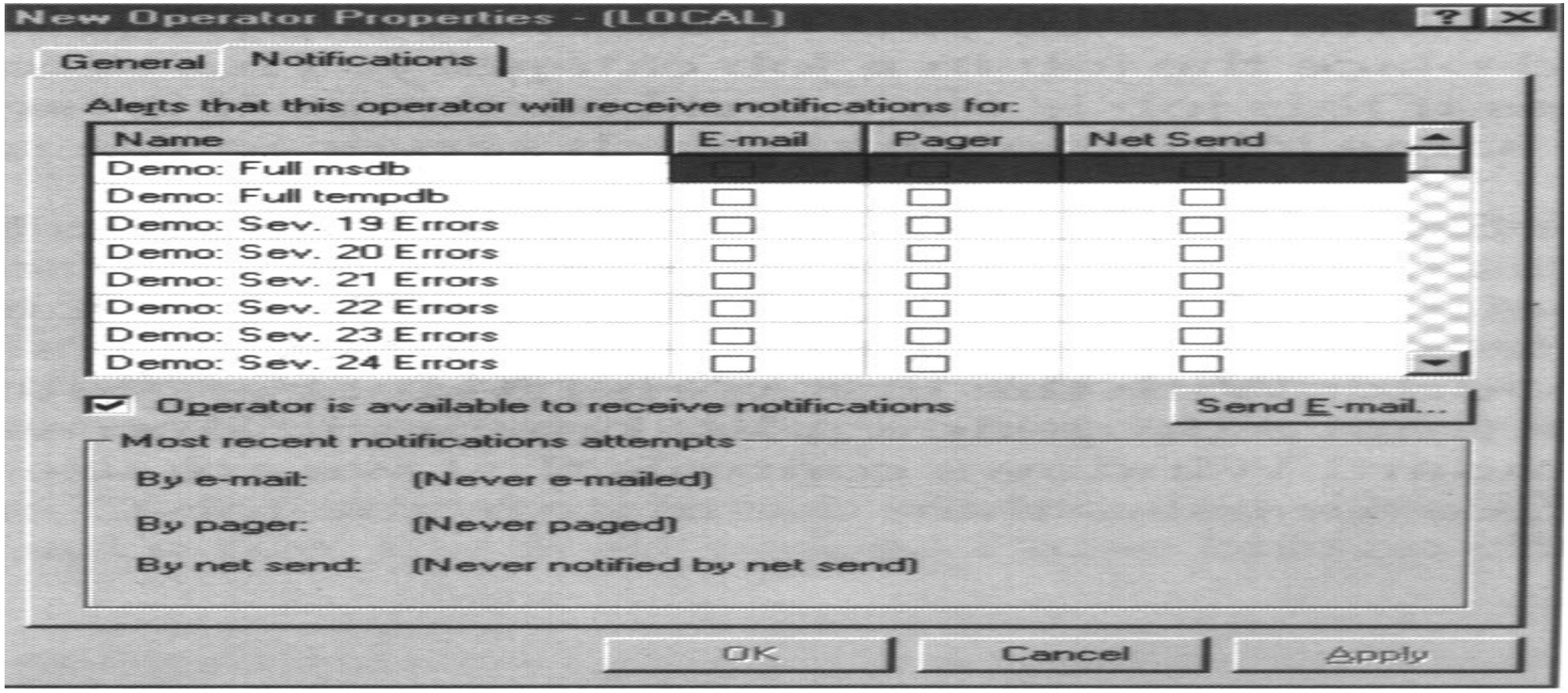


图 4.4 在离开办公室时取消通知

4.2 创 建 作 业

在定义了操作员之后，就可创建作业和作业类别了。创建作业的步骤如下:

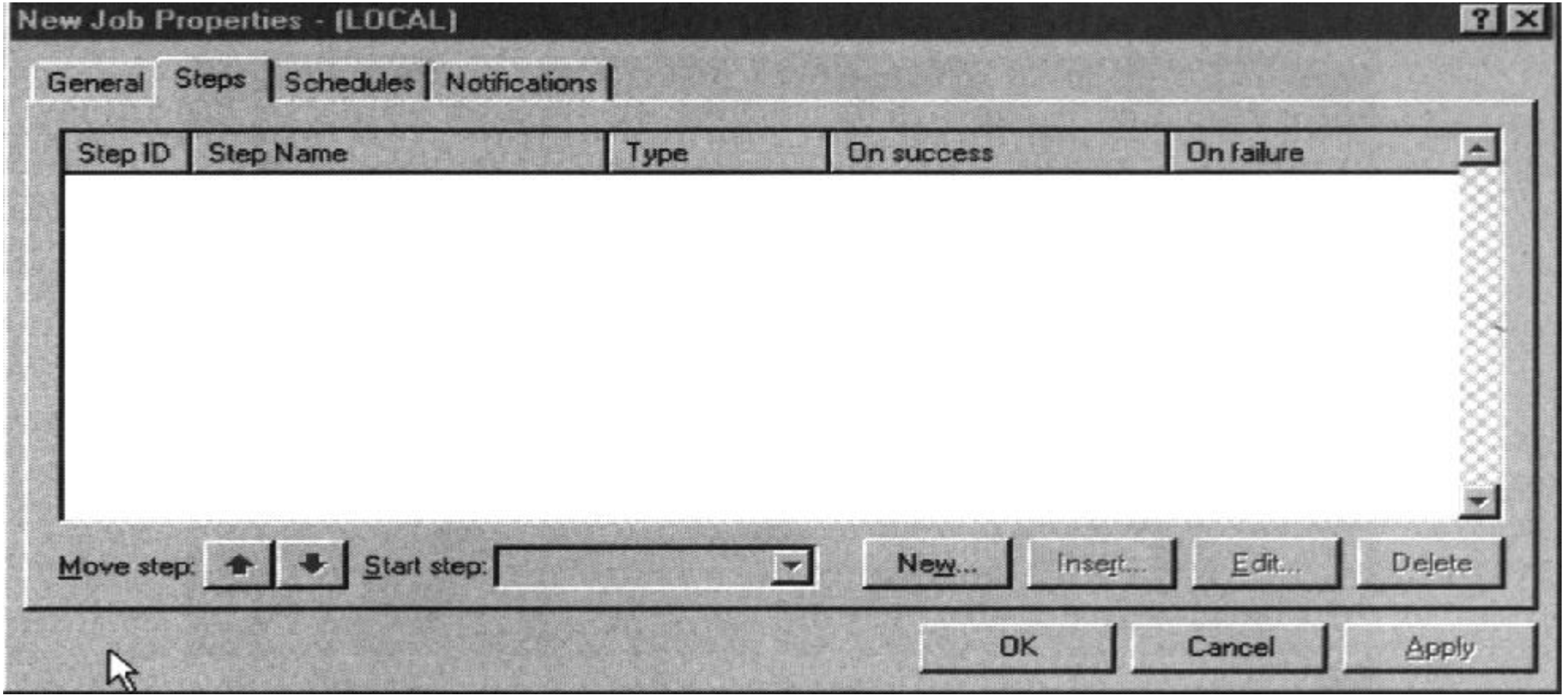
1. 单击 Microsoft Management Console (MMC)的 Console Root 下的 Microsoft SQL Servers 项。
2. 展开已创建好的 Complete Reference Group 服务器组。
3. 展开希望使用的服务器。
4. 展开 Management 下的 SQL Server Agent。
5. 右击 Jobs。
6. 从快捷菜单上选择 New Job，显示 New Job 屏幕。

New Job 屏幕上有 4 个不同的标签，每个标签都等同于新作业的全屏幕功能，下面将对每个标签作详细介绍，这 4 个标签是:

- General(通用)
- Steps(步骤)
- Schedules(调度)
- Notifications(通知)

General 标签下的面板有新作业的输入域，包括 Name、Category、Owner 和 Description。这些输入域允许描述作业、命名作业、指定作业的拥有者，把作业放到作业类别中。这个屏幕包含的选项可确定该作业是用于本地服务器，还是用于多个服务器，以便在多重服务器环境下的每个服务器上运行相同的作业。

Steps 标签下的面板如下图所示，该面板具有创建作业步骤的强大功能。它模拟了大型计算机用户熟悉的作业流环境，并提供一个受欢迎的功能。系统管理员不必再编写和编译在 IBM 大型计算机环境下使用该作业步骤功能时所需的代码。采用 Enterprise Manager，可以填写屏幕上的区域，从下拉列表框选择项目或单击选项，再按下 OK 按钮，这是我们习惯使用的标准 Windows 控件。这是比过去的大型机 Job Control Language(JCL)更柔和的用户界面。



Steps 面板上有一个 New 按钮，可为作业添加步骤。按下 New 按钮可展示 New Job Step 屏幕，这正是引人注意的地方。New Job Step 屏幕的 General 面板上有 Step Name 和 Type 两个下拉列表框，其中包含以下选项：

- Active Scripting
- CmdExec
- Distribution
- Logreader
- Merge
- Snapshot
- TSQL

Type 下拉列表框有些很有趣的功能，就是对于每一种选择 General 面板都不同，就像 Advanced 标签下的面板一样。图 4.5 表示 TSQL 类型(顶部)和 Active Scripting 类型(底部)的 General 标签。每个屏幕都是专为选中的类型制作的。图 4.6 表示 TSQL 类型(顶部)和 Active Scripting 类型(底部)的 Advanced 屏幕。由图可见，每个屏幕都根据类型的不同而不同。以上列出的所有类型都是这样的。

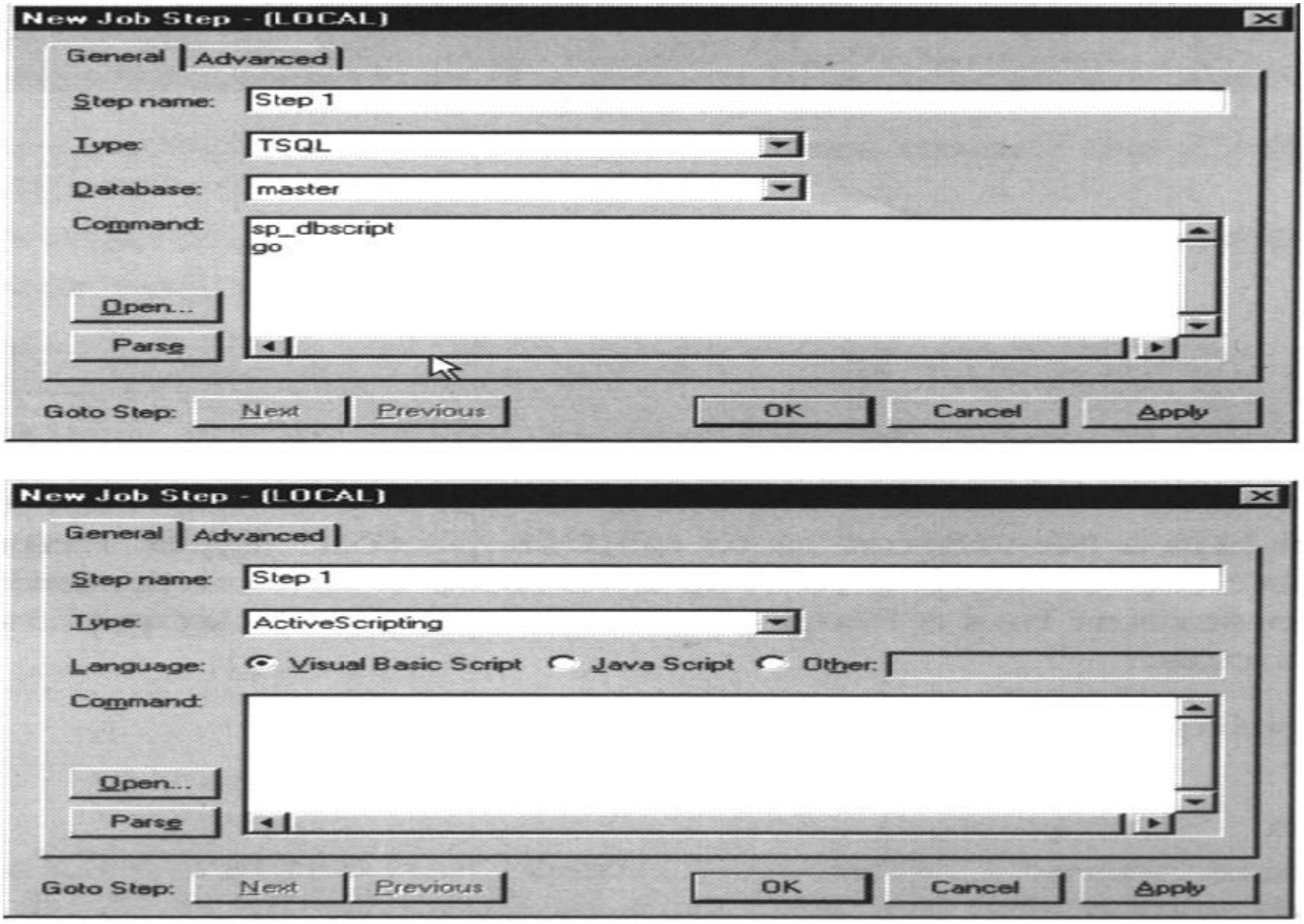


图 4.5TSQL 类型(上图)和 ActiveScripting(下图)的 General 面板

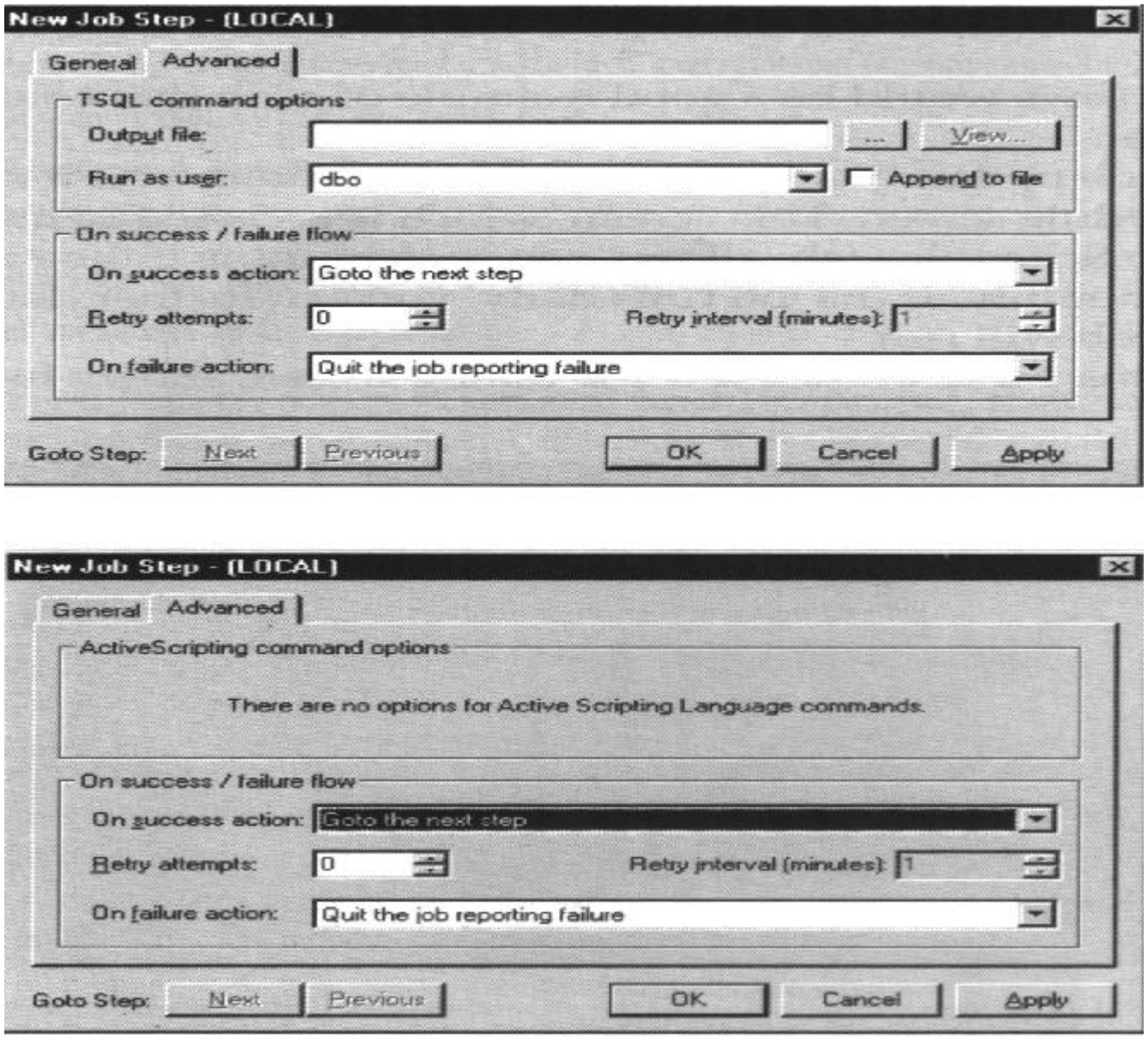


图 4.6TSQL 类型(上图)和 ActiveScripting(下图)的 Advanced 面板

注意，Open 和 Parse 按钮与 TSQL 类型相关联。Open 按钮允许选择文件，位置很恰当的 Parse 按钮可检查 TSQL 代码的错误。

ActiveScripting 的 General 面板上的单选按钮指明 Command 窗口中设置的语言，可选择 Visual Basic 脚本、Java 脚本，或在其他语言名中填上 Other。

TSQL 类型的 Advanced 面板有一个域用于输入输出文件，还有一个选项指明是否想追加到指定的文件中。该面板上有一个 Run as User 输入域，以及 On Success 或 Failure action 下拉列表框形式的 On Success／Failure Flow 逻辑，该下拉列表框中有几个选项，包括退出记录成功的作业、退出记录失败的作业和进入下一步。在 On Success 的背后，有一个真实的逻辑集记录失败，或在 On Failure 背后有一个真实的逻辑集记录成功。

ActiveScripting 的 Advanced 面板与前面 TSQL 类型的 Advanced 面板一样，屏幕上都内置了相同的 On Success／Failure 流逻辑。它与 TSQL 类型的 Advanced 面板具有相同的 Retry Attempts 和 Retry Interval 设置。

New Job 屏幕上的 Schedules 标签允许安排运行作业的时间。它有以下几个调度选项：

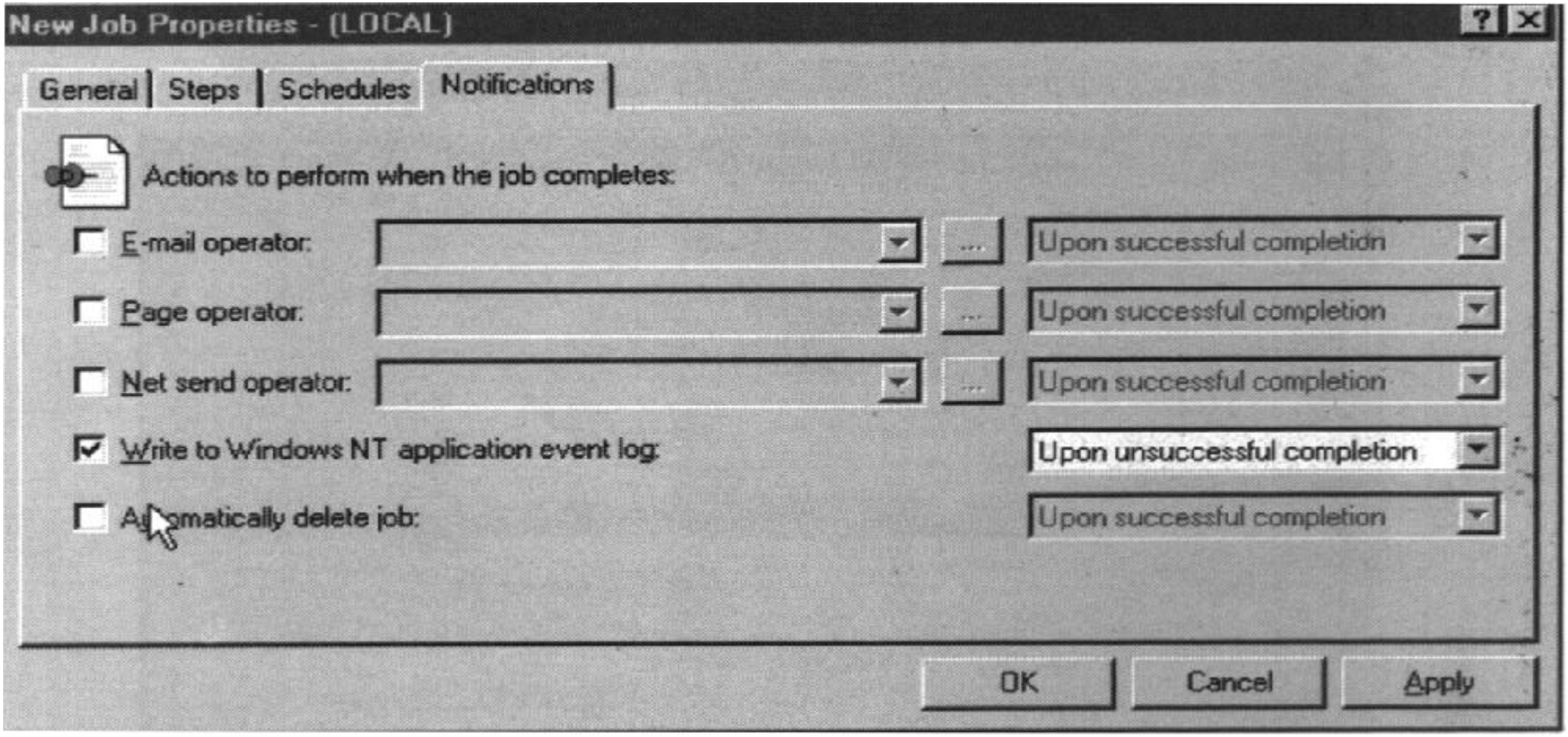
- 在 SQL Server Agent 启动时自动启动
- 只要 CPU 空闲就启动
- 一次启动
- 循环启动

删除行的作业实例可以很好地说明只要 CPU 空闲便运行该作业。

循环作业选项还有另一个与 Edit Recurring Job Schedule 屏幕相关联的屏幕。

该屏幕可通过 New Job Schedules 标签下的 Changes 按钮来访问，它允许确定循环作业是每日、每周还是每月运行。如果该作业每日运行，还可进一步确定该作业每天运行多少次。

New Job 对话屏幕的 Notifications 面板如下图所示：



该面板指明了在作业完成时要执行的操作，包括是谁，用什么方法通知以

前定义的操作员。该屏幕还有一个选项，用来指明是否写入 NT Event 日志。所有的操作都有一个关联的下拉列表框，该列表框描述了这些操作应执行哪些事件。该下拉列表框中的选项如下：

- 成功完成
- 未成功完成
- 一旦作业完成

4.3 创 建 警 告

在创建作业后，就可以创建警告。创建警告的步骤如下：

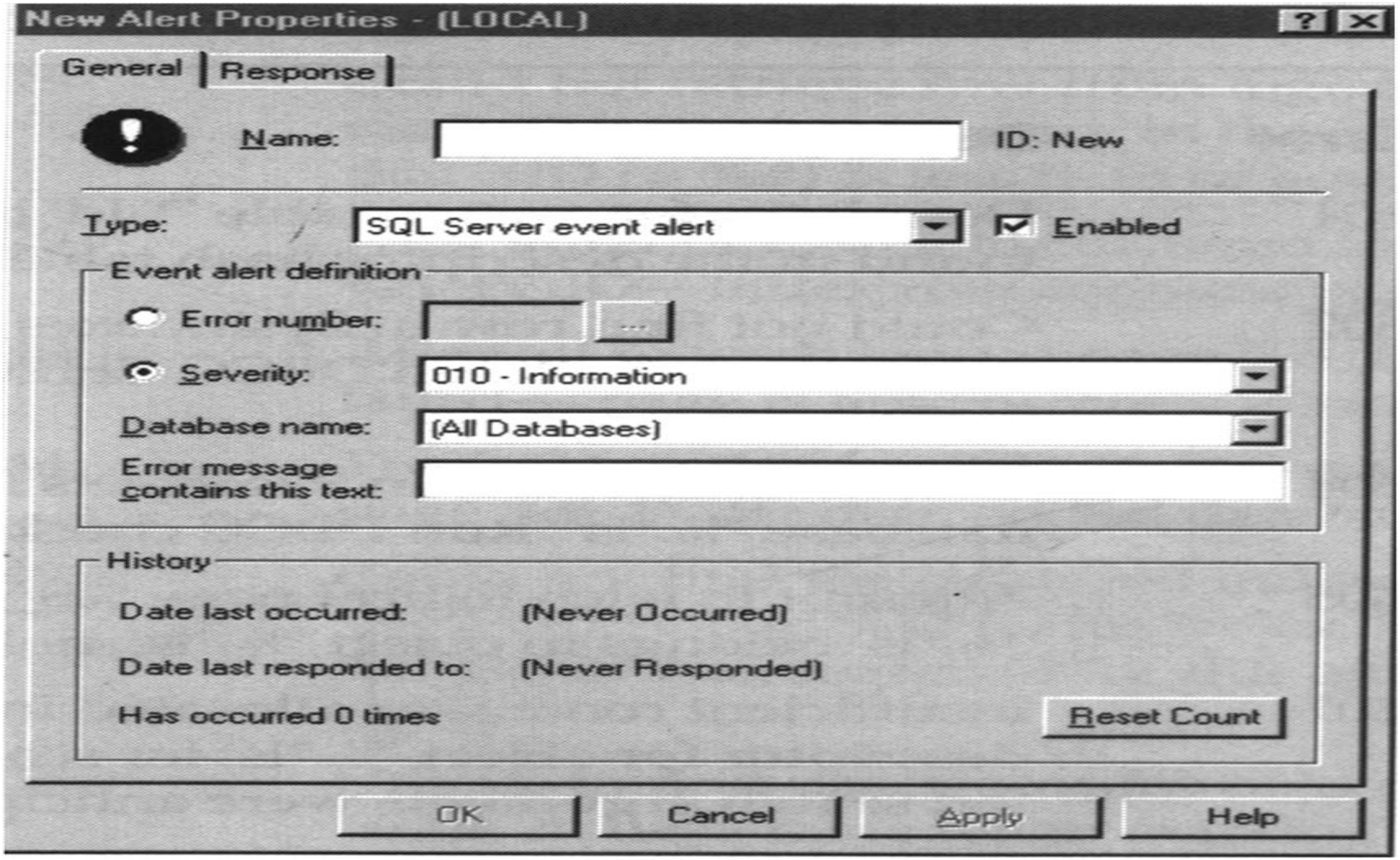
1. 单击 Microsoft Management Console(MMC)的 Console Root 下的 Microsoft SQL Servers 项。
2. 展开已创建好的 Complete Reference Group 服务器组。
3. 展开希望使用的服务器。
4. 展开 Management 下的 SQL Server Agent。
5. 右击 Alerts。
6. 从快捷菜单上选择 New Alert。
7. 显示 New Alert 屏幕。

New Alert 屏幕上有两个标签：General 和 Response，每个标签都相当于新警告的全屏幕功能。在此详细介绍 General 面板，在本章的后面再解释 Response

面板。

New Alert对话框的General面板上有名称、类型以及事件警告定义和历史。可以在区域中输入名称，从下拉列表框中选择类型，但事件警告定义需详细介绍。

该面板上有两个单选按钮：Error号和Serverity(严重程度)。由于SQL Server所生成的错误给定了严重等级，所以可以在某个特殊错误或在具有某个严重程度的一组错误上生成警告。如果选择Error number单选按钮，就可按下...按钮，搜索有效的SQL Server错误。该按钮符号表示可查找更详细的内容，在这种情况下，可搜索有效的错误号。这种访问(以前需编程才可访问)深度错误信息深度的能力，提供了极其有用的特性，它使各地的SQL Server操作员使用起来都非常方便。



由于所有的信息都保存在 sysmessages 表中，所以在企业运作警告策略的设

计中作为辅助手段对此进行查询是非常方便的。前面已经介绍了实例的查询，其查询结果如表 4.1 所示，表中列出了 SQL Server 本身包含的某些错误信息。在主数据库中的 Query Analyzer 下执行该查询，可查看 SQL Server 使用的所有错误消息：

```
SELECT severity, error, description FROM sysmessages
GO
```

表 4.1 显示了可用于警告的错误样本。严重等级为 21 的错误所示的结果集如表 4.1 所示。

表 4.1 错误严重等级 21 的 sysmessages 系统表的输出

严重等级	错误号	描述
21	601	不在描述符杂凑表中的数据库 '%d' 中的系统表 '%ld' 的描述符
21	602	在 dbid '%d'、对象 '%ld'、索引 '%d' 的 Sysindexes 中找不到行。运行 Sysindexes 上的 DBCC CHECKTABLE
21	604	在数据库 '%.*ls' 中对象 '%ld' 的 Sysobjects 中找不到行。运行 Sysobjects 上的 DBCC 检查表
21	605	试图提取属于对象 '%.*ls'，但不属于对象 '%.*ls' 的数据库 '%.*ls' 中的逻辑页面 %S_PGID
21	607	在用于搜索自变量的对象 '%.*ls' 的对话描述符中，分配的空间不足。只预料到有 %d 个搜索自变量严重等级错误号描述
21	608	包含数据库 '%.*ls' 中对象 '%.*ls' 的逻辑页面 %S_PGID 的缓冲区，在扫描期间不能保存于该对象的第一或第二个对话描述符的插槽中

续表

21	611	试图结束空闲事务或处于更新之中的事务
21	613	请求从数据库'%.*ls'中对对象'%.*ls'已进行完的扫描中检索多行
21	615	无法找到数据库表 id=%d,名称='%.*ls'
21	618	将%d 的 varno 传送给打开表——最大的有效值为%d
21	623	不能通过 RID 在页面中检索行，因为逻辑页%S_PGID 不是数据页.%S_RID.%S_PAGE
21	624	由于被请求的 RID 编号比页面上最后一个 RID 更高，试图通过 RID 从页面获取行失败。%S_RID.%S_PAGE。
21	625	不能通过 RID 在逻辑页%S_PGID 中检索行，因为该 RID(=%d)的偏移表(=%d)中的条目小于或等于 0
21	629	不能删除页面%S_PGID 的成簇索引项——索引行包含页面%S_PGID
21	639	不能提取逻辑页%S_PGID dbid %d，该页当前未分配
21	644	在索引页%S_PGID indexid %d 数据库'%.*ls'中未找到 RID'%.*hs'的索引项
21	649	无法找到 Page%S_PGID Objid %ld 状态 Ox%x 的成簇索引项。在索引页%S_PGID 中搜索数据库'%.*ls'中的该项
21	805	在标记缓冲区杂质或刷新 Syslogs 时，找不到杂凑表中数据库'%.*ls'中的对象'%.*ls'的描述符
21	806	找不到数据库'%.*ls'中逻辑页%S_PGID 的虚拟页
21	811	试图抓取位于描述符链上的缓冲区
21	812	试图抓取固定的缓冲区

续表

21	815	无法找到高速缓存中的含有 Sysindexes 页的缓冲区——pageno=%S_PGID dbid=%d
21	820	试图弄脏位于 I／O 中的非日志缓冲区%S_BUF
21P	822	不能启动请求%S_BLKIOPTR 的 I／O
21	824	试图在数据库（Sysdatabases 中没有项）的缓冲区中写入页面，%S_BUF
21	825	试图在无 DBTABLE 结构的数据库中的缓冲区中写入页面，Sysdatabases 行标记为过程%d.%S_BUF 所使用
21	834	在含有杂质的缓冲区上调用缓冲区清理(页面%S_PGID,stat%#x／%#x,objid%#x,sstat%#x)
21	901	在杂凑后，无法找到杂凑表中的数据库'%d'对象'%ld'的描述符
21	909	多于%d 个的项目需要建立数据库'%.*ls'的逻辑虚拟转换表，该数据库的碎片太多
21	912	找不到假定已打开的数据库'%.*ls'的 DBTABLE 描述符
21	915	在数据库'%.*ls'的 DBTABLE 链中找不到系统类别'%.*ls'的描述符——所有的系统类别都应永久驻留在该链中
21	919	数据库'%.*ls'已由恢复程序标记为'suspect'。请运行 DBCC 卸掉该数据库严重等级错误号描述
21	931	因为不能初始化全局时间戳，所以不能打开数据库“%.*ls'。这表明当前数据库的记录有问题。请与 Technical Support 联系以请求帮助
21	1103	数据库'%.*ls'的分配页%S_PGID 中的分段 id 不同于我们正分配的对象分段。请运行 DBCC CHECKALLOC

续表

21	1108	不能再分配范围%S_PGID、数据库%d。范围中的对象%ld、索引 id%d、状态%d 与再分配的对象中的对象 id%ld、索引 id%d、状态%d 不匹配。请运行 DBCC CHECKALLOC
21	1109	不能读取分配页%S_PGID，因对象 ID 不正确(%ld)或页面 ID 不正确%S_PGID
21	1117	对象%ld 的范围链的链接不正确
21	1514	要排序的分配页正忙。页面号%S_PGID.%S_BUF
21	1525	分类失败。分类向前滚动时遇到序列之外的分类描述符时间戳。记录中的旧时间戳为%04x%081x，新的时间戳为%04x%081x。分类描述符中的时间戳为%04x%081x211528 字符数据比较失败。在 SMEL 项［%d］的用于服务器的缺省分类表中有无法识别的 Sort-Map-Element 类型(%d)
21	1529	字符数据比较失败。用于服务器的缺省分类表的 Sort-Map-Elements 列表没有正确结束。此列表开始于 SMEL 项［%d］
21	1601	没有可用的资源以启动'%ls'进程。使用 sp_configure 可增加用户连接数
21	1602	不能初始化网络%d
21	1603	进程优先级无效，或无可用的进程插槽
21	1604	进程不可运行，或 kpid%d 不在范围之内
21	1605	打开新连接的虚拟插槽失败
21	1606	初始化网络接收缓冲器失败
21	1607	初始化网络发送缓冲器失败
21	1608	在向前端发送结果时遇到网络错误。详细信息请查看 SQL Server 错误记录

续表

21	1609	Kpid%d 超出范围
21	1610	不能删除进程%d
21	1611	不能安装退出功能
21	1612	不能安装注意功能
21	1613	不能关闭服务器进程%d 的网络%d 连接
21	1614	不能产生进程
21	1615	进程不能休眠
21	1616	刷新接收流缓冲区失败
21	1617	不能影响进程%d
21	1619	不能打开 TEMPDB，不能继续
21	1620	第一次打开主 db 失败
21	1808	不支持缺省设备
21	1905	在 Sysindexes 中找不到索引'%.*ls'表的'zero'行严重等级错误号描述
21	2202	不能把虚拟读出地址转换到设备，不能搜索 vaddr=0x%x
21	2205	不能将虚拟写入地址转换到设备，不能搜索 vaddr=0x%x
21	2208	试图读取未打开的设备 vaddr=0x%x
21	2209	试图写入未打开的设备 vaddr=0x%x
21	2603	在索引页上插入行时，对象'%.*ls'的索引%d 的逻辑页%S_PGID 未留有空间，这种情况应在转换索引时考虑
21	2620	行号在偏移%d 处的偏移与%S_PAGE 页的偏移表中的项目不匹配
21	2622	在插入长度为%d.%S_PAGE 的行时该页空间不足

续表

21	2623	运行时 PG_DEALLOC 位在该页上，应由恢复程序中的取消分配页.%S_PAGE 关闭它
21	2624	因为行长度%d 小于最小长度%d，所以插入表%S_DES 失败
21	2626	试图在数据库'%.*ls'中的对象'%.*ls'的成簇索引中非法插入重复的关键行
21	2728	找不到表'%.*ls'的 Sysobjects 行
21	3001	内部错误。找不到系统分类'%.*ls'的索引，不能清空数据库
21	3113	数据库'%.*ls'在 DBO 的 SYSUSERS 中没有项目
21	3114	数据库'%.*ls'在 Sysdatabases 中无项目
21	3301	在 Syslogs(log.p%d)中找到无效的日志记录
21	3303	在 xact %S_RID 的日志中找不到 SAVEPT，当前号为%d，查找%d
21	3305	数据库'%.*ls'中的页面%S_PGID 在运行期间读取或装载 xact 未初始化
21	3306	期望过程%d 包含页面%S_PGID 上的逻辑锁，而不是过程%d
21	3307	期望过程%d 包含页面%S_PGID 上的逻辑锁
21	3308	期望页面%S_PGID 带有%ls 锁
21	3309	放弃时过程%d 被选作死锁牺牲品，同时等待过程%d 释放页面%S_PGID 上的锁
21	3310	找不到带有 RID:(%ld,%d)的日志记录
21	3313	在数据库%.*ls 中重新记录操作时发生的错误。该错误位于日志记录 ID%S_LSN 处

续表

21	3314	在数据库%.*ls 中撤消记录操作时发生的错误。该错误位于日志记录 ID%S_LSN 处
21	3401	Rec_init:在 Sysdatabases 中 Rid 上的最后一个检测点的 getnext SCAN_RID 失败.%S_RID
21	3404	Rec_complete:打不开多数据库事务中控制数据库的控制数据库(id %d)
21	3412	数据库%d、表%ld。试图将表标记为可疑。Sysobjects.objid 上的 Getnext SCAN_CLUST 失败
21	3413	数据库%d。试图将数据库标记为可疑失败。Sysdatabases.dbid 上的 Getnext NC 扫描失败
21	3417	不能恢复主数据库，退出
21	3418	打开模型的描述符不够
21	3421	因为不能初始化全局时间戳记，所以不能恢复数据库'%.*ls'。这表明当前数据库的记录有问题。请与 Technical Support 联系请求帮助严重等级错误号描述
21	3423	恢复数据库'%.*ls'错误，在位置%S_RID 处找不到期望的 BEGIN TRANSACTION 记录
21	3424	用于事务(%ld，%d)的事务表中空间不足
21	3425	在事务表中找不到事务(%ld,%d)
21	3429	恢复数据库'%.*ls'错误，不能连接以提交服务，检查 xact:%S_RID 的完整状态
21	3437	恢复数据库'%.*ls'错误，不能连接到 DTC，以检查 xact:%S_RID 的完整状态

续表

21	3440	数据库'%.*ls'(dbid %d):RECOVERDB 命令只能在使用 WITH NORECOVERY 选项的 RESTORE 命令后运行
21	3441	数据库'%.*ls'(dbid %d):访问文件'%ls'时 RECOVERDB 命令失败。错误是'%.ls"
21	3442	数据库'%.*ls'(dbid %d):撤消文件的尺寸不够
21	3443	数据库'%.*ls'(dbid %d):因用于备用／只读而标记，但已经被修改。不能执行 RESTORE LOG
21	3444	数据库'%.*ls'(dbid %d):找不到以前恢复操作的起始点。不能执行 RESTORE LOG
21	3445	数据库'%.*ls'(dbid %d):文件'%ls'为无效的撤消文件
21	3501	在检查点时间%S_RID 得不到数据库'%d'的 Sysdatabases 行
21	3502	在数据库'%.*ls'的 Syslogs 中找不到检查点记录
21	3504	数据库'%.*ls'中的 Sysindexes 页#%S_PGID 在调用 getindex 后不在检查点处的缓冲区高速缓存中
21	3904	不能合并数据库'%.*ls'的对象'%.*ls'中的逻辑页%S_PGID，两个页面合在一起比一个页面包含的数据多
21	3905	不能合并数据库'%.*ls'中对象'%.*ls'中的逻辑页%S_PGID 其行号%d 可用于两个页面
21	3907	从 PROBE 接收结果失败。尚未提交名为%.*ls 的分布式事务，数据库中的数据可能不一致
21	4003	ODS 错误。服务器正在中止该连接
21	4202	清空数据库'%.*ls'中的事务期间，在查找事务页时找不到日志中的 BEGINXACT 记录

续表

21	4203	在清空数据库'%.*ls'中的事务时，裁减页面后找不到检查点记录
21	4802	Bulk_main:在 BULK INSERT 表上，主索引行的 getindex 失败
21	4803	从 bcp 客户机接收了无效的行长度%d。最大行长度为%d
21	4804	从主机上读取当前行时过早地到达消息的结尾。主机程序可能已死机
21	4807	从 bcp 客户机上接收了无效的行长度%d。最小行长度为%d
21	4815	从 bcp 客户机接收了无效的列长度
21	4816	无效的 bcp 对象
21	4905	ALTER TABLE 失败，因为数据库'%.*ls'中系统类别 Sysindexes 的页%S_PGID 不在高速缓存中
21	6501	数据库'%.*ls'中的逻辑页%S_PGID 在提取索引行之后从高速缓存中丢失
21	6901	数据库%d 中时间戳的 High 组件发生溢出。数据库表可能损坏
21	6902	页面时间戳记值在日志中的新旧时间戳记之间。Page#=%S_PGID，对象 id=%ld，页面时间戳记=%04x%081x。Log:旧时间戳记=%04x%081x，新时间戳记=%04x%081x
21	6903	时间戳记的高序号指明时间戳记结构可能不正确
21	6904	请求新的数据库时间戳记时，传递的数据库表和系统结构中的数据库表不匹配。传递的数据库 id:%d。系统数据库 id:%d
21	8646	在数据库'%.*ls'中的表%d 的 indexid %d 中找不到 RID'%.*hs'的索引项
21	9004	数据库"%*ls"的记录受损

在 Severity 下拉列表框中，错误严重等级的某些实例可用于警告，这些错误

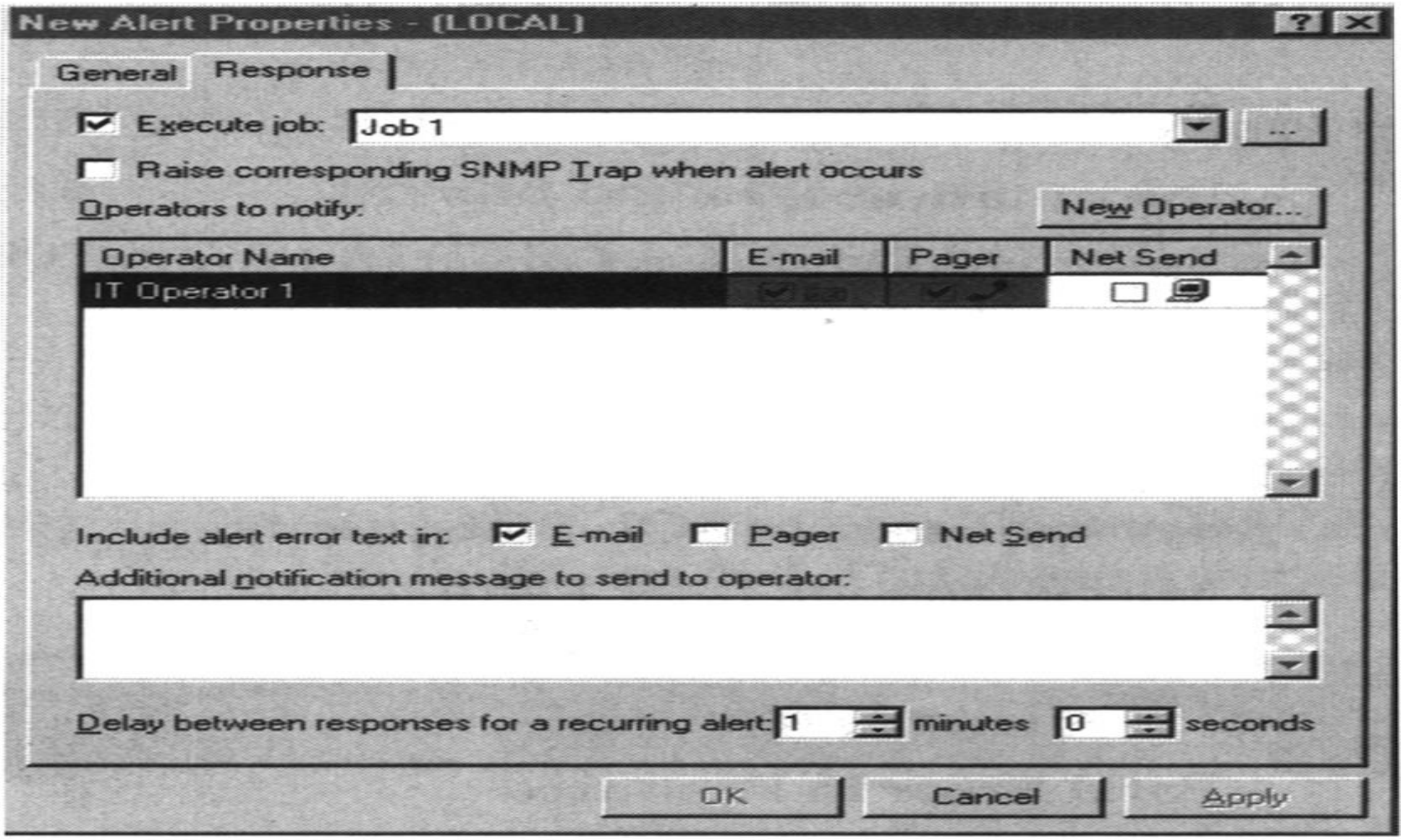
严重等级如下表所示:

错误等级	描述
010	信息
011	未找到指定的数据库对象
012	未用
013	用户事务句法错误
014	许可不足错误
015	SQL 语句中的句法错误
016	其他用户错误
017	资源不足
018	非致命的内部错误
019	资源致命错误
020	当前进程中的致命错误
021	数据库进程中的致命错误
022	致命错误:表的完整性可疑
023	致命错误:数据库的完整性可疑
024	致命错误:硬件错误
025	致命错误
110	非系统消息，服务器信息
120	非系统消息，服务器警告
130	非系统消息，服务器错误
140	非系统消息，服务器异常中止

如错误严重等级的说明所示，17 级和 17 级以上的错误严重等级应引起系统管理员的注意，应向 SQL Server 操作员发出自动警告。如果 SQL Server 包含重

要任务的应用程序，就应寻呼操作员；否则，发出 e-mail 消息就足够了。

New Alert 对话框中的 Response 标签能确定在发出警告时执行什么操作。Response 标签下的面板如下图所示:



用户可以通知操作员或执行某项作业。有些选项包含寻呼或 e-mail 通知消

息中的错误消息。可以向操作员发送附加通知消息，也可以按下 New Operator 按钮并创建新的操作员，也可以指定定期警告响应之间的延迟时间(如果每秒发出一次警告，应防止发出连续寻呼或连续消息)，甚至在发生警告时产生相应的 SQL 陷阱。

至此，用户已了解了如何在 Enterprise Manager 中访问 SQL Server Agent，下面是在 NT Service Control 中访问 SQL Server Agent 的步骤：

1. 在命令行提示下，键入 srvmgr＼＼servername。显示 Server Manager 屏幕。
2. 从菜单栏中选择 Computer。
3. 选择 Services。
4. 向下滚动，单击 SQLServerAgent 服务，如图 4.7 所示。

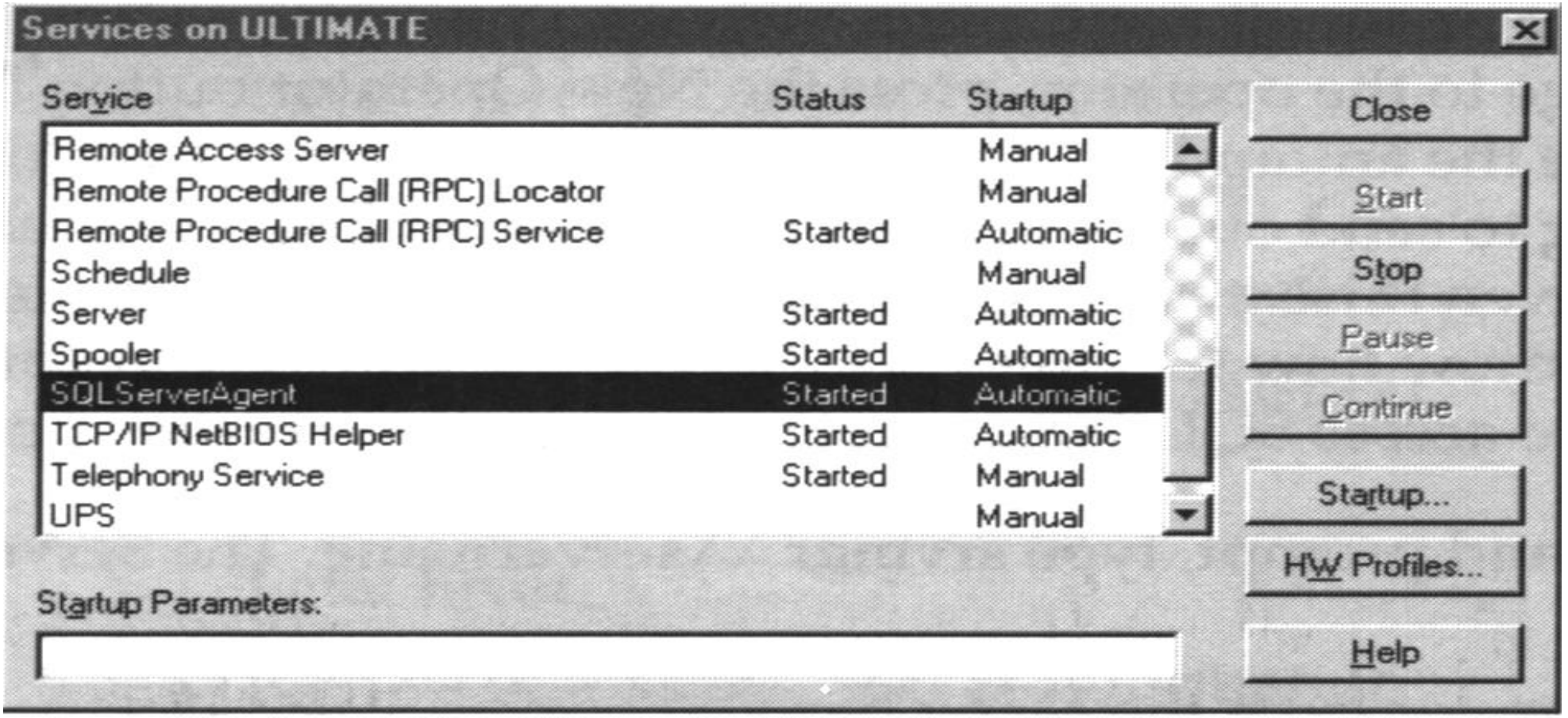


图 4.7NT Service Control Manager | Computer | Services

现在就可以停止和启动 SQLServerAgent 服务，或改变启动参数了。单击 Startup 按钮可进入启动参数屏幕。

启动参数屏幕包含 Startup Type 和 Logon As 两部分。Startup 类型由下列内容组成：

启动类型	描述
Automatic(自动)	重新引导 NT 时，SQLServerAgent 服务自动启动
Manual(手动)	重新引导 NT 时，SQLServerAgent 服务必须手动启动
Disabled(无效)	SQLServerAgent 服务无效，不能启动

Log On As 选项由以下内容组成：

Log On As	描述
系统帐户	由于必须具有网络访问许可，所以在 Local System 帐户中，不能指定运行 MSSQLServer 和 SQLServerAgent 服务。若没有此许可，服务器对服务器的远程过程调用(RPC)、复制作业和备份都将失败
本帐户	指定已特别设置的帐户，以运行 SQL Server Agent。如果要运行 SQL Mail 和 SQL Server Agent，该帐户应有网络访问权和相应的许可

在 NT Service Control Manager 中启动和停止 SQL Server Agent 是很简单的，只需单击 SQLServerAgent 服务，并按下 Start 或 Stop 按钮即可，至于选择哪一项合适，取决于 SQLServerAgent 服务当前是处于停止状态还是启动状态。

4.4 多任务和多方案作业

SQL Server Agent 允许通过创建作业步骤来创建多任务作业，而作业步骤又允许用多个步骤创建一个作业，也允许每个步骤运行一个任务。该逻辑内置于定义步骤的屏幕上，以确定作业步骤成功和失败时各进行什么操作(请参阅本章前面的“创建作业”)。

SQL Server Agent还允许为同一作业创建多种方案，可以决定用每日、每周和每月的方案运行作业。每种方案都可以为同一作业而设置，该作业会在每个预定的时间运行。

Microsoft SQL Server Agent在 Microsoft寻求“lights out（无光）”管理和数据库中，是一个强有力的工具，它可以在晚上没有数据库管理员干预的情况下自动运行任务。关于多服务器作业的详细情况，请参阅第 3 章中“配置多服务器环境”一节。

第 5 章与 Windows NT 安全性的集成

数据库安全性的存在可避免未授权用户查阅、添加、改变或删除数据。随着对 Internet 和各公司上数据库的访问不断增长，最近对访问敏感数据的控制比以往任何时候都更引人注目，甚至连普通人都对此领域很感兴趣，无人希望由未授权用户或随机用户访问或改变自己的信息，如信用报告或 Social Security 信息等。

作为数据库管理员，应对保护数据特别关注，以防止未授权用户查阅或改变数据，还要注意系统使访问对于管理和维护更加方便，因为如果没有合适的过程和标准，就不能管理这种访问，尤其是在负责成千上万用户的访问时就更是如此。

把用户的责任分成几个组和角色可增加安全性，因为工作可以分为多份，这样每个任务可由两个用户分担。只为角色或组指定完成工作所需的权限。例如，可以由一个工人批准某个验证，而由另一个工人打印和邮寄这份验证。

SQL Server 和 NT 在检查用户的访问和许可时集成起来的方法已进行了强化和简化，已经消除了整个管理层。现在有一种分层安全系统，其中角色和组以及用户都可成为其他角色的成员，在许多层都可以指定许可。

5.1Security Manager 的消失

不再使用 Security Manager 并从 Microsoft SQL Server 工具集中删除。现在，为数据库管理可靠的用户访问、为用户提供特定和控制的许可是十分简单的。访问 Microsoft SQL Server 的组成要素是注册、用户、角色和组。下面首先介绍注册，注册就是在 Microsoft SQL Server 中验证用户身份的方法。然后介绍用户、角色和组。

现在，当用 Microsoft SQL Server 工作时，安全性分成两部分:身份验证和许可的有效性。身份验证是安全性的一个部分，它可识别用户，并确认允许用户连接 SQL Server。许可有效性用于控制访问数据库和数据库内的对象。它还控制用户对数据库的访问完成时可以执行的操作。

5.2 身 份 验 证

现在,识别用户和确认用户有访问 SQL Server 的许可只有两种方法:Windows NT 验证和 SQL Server 验证。SQL Server 可以配置为使用验证的 NT Authentication 服务器模式或 Mixed 服务器模式。Mixed 表示 Windows NT 验证注册 ID 和 SQL Server 验证注册 ID 都可用于获得 SQL Server 的访问权。如果采用 NT Authentication 模式，则只允许采用 Windows NT Authentication 注册 ID。

下面的第一个表展示了目前在 Microsoft SQL Server、Mixed 和 NT Authentication 中都可用的验证模式。Mixed Authentication 模式有两种验证类型:NT Authentication 和 SQL Server Authentication，而 NT Authentication 模式只有 NT Authentication 一种类型。下面第二个表展示了验证模式中的两种验证类型，分别称为 NT Authentication 和 SQL Server Authentication。该表还展示了验证类型的特性。

验证模式=Mixed	验证模式=NT Authentication
Windows NT Authentication 注册 ID 和／或 SQL Server Authentication 注册 ID	只有 Windows NT Authentication 注册 ID

Windows NT Authentication	SQL Server Authentication
不必键入注册 ID	必须键入注册 ID
不必键入密码	必须键入密码
NT Administrator 必须创建 NT Account 或 NT Group。	没有相关的 NT 帐户或组
Login ID 就是 NT Account 或 NT Group	注册 ID 是未与 NT 安全性集成的 SQL Server 注册 ID。这只是 SQL Server 安全性功能，与 Windows NT 无关委托 NT(已委托)不委托 NT(未委托)

5.2.1Windows NT 验证模式

NT Authentication 模式下与 SQL Server 的网络连接假定为 SQL Server 和客

户机之间的委托连接(trusted connection)。委托连接通过网络协议(Named Pipes 命名管道、Multi-protocol 等)获得支持，网络协议支持服务器和客户机之间的委托连接。当 NT 建立委托连接并将注册 ID 和密码传送给 SQL Server 时，SQL Server 并不检验密码或帐户，它假设该注册 ID 和密码是正确的，因为在用户进入网络时 NT 已经验证了该用户。SQL Server 使用同一个 NT 网络用户名和密码来验证 SQL Server 中的用户，就像 NT 用来验证网络上的用户或访问操作系统一样；换言之，用 Windows NT Authentication 模式访问 SQL Server 上的数据库，并不需要单独的注册 ID 和密码。因为这种类型的验证允许 SQL Server 充分利用 NT 的安全特性，所以它是很有利的。NT 安全特性由 SQL Server 所提供的极为丰富的验证功能集组成。用户在自己的网络身份下访问 SQL Server 之前，必须进入网络并通过所有 NT 验证功能的验证。例如，可以设置 NT 验证，以使密码在给定时间后失效。

注意:使用 Microsoft Management Console，把 SQL Server 注册到服务器组中，参见图 5.1。

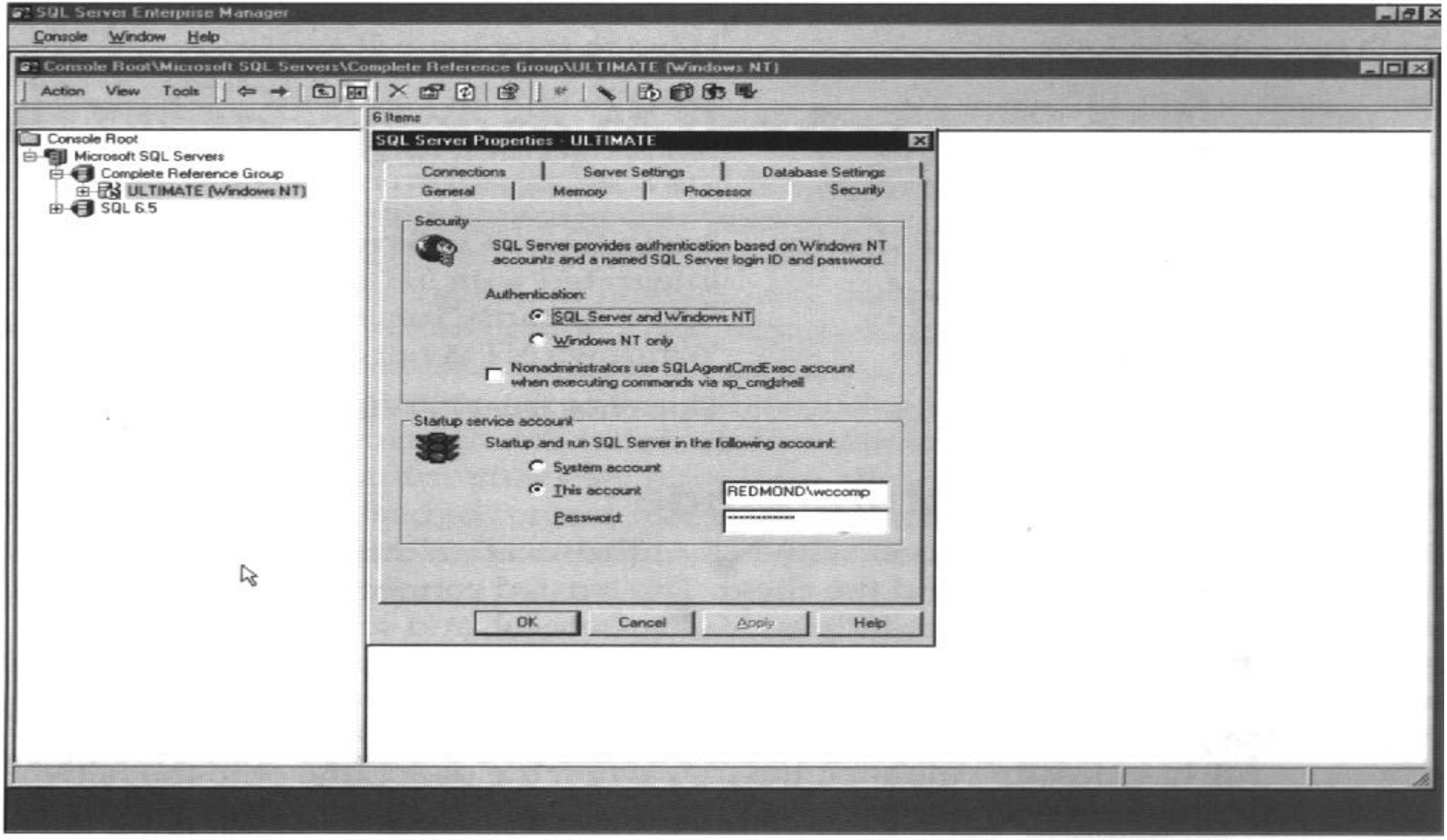


图 5.1 设置 Windows NT Authentication 模式

要充分利用 Windows NT 安全性，应设置 SQL Server 的 Windows NT Authentication Mode:

1. 展开服务器组。
2. 选择一个服务器，右击该服务器，再选择 Properties。
3. 单击 Security 标签。
4. 只选择 Windows NT 或选择 SQL Server 和 Windows NT。
5. 选择 OK 按钮。

要定义 Windows NT 帐户或组和 SQL Server 注册 ID 之间的关系，并设置 Windows NT 和 SQL Server 之间的委托连接，可使用下列语法:

sp_grantlogin ' DomainName＼NTusername'

也可使用下面的语法:

sp_grantlogin [DomainName＼NTusername]

注意:如果游标位于服务器组中的相应服务器上，可使用 Microsoft Management Console 上的 Tools 菜单中的 SQL Query Analyzer Tool。在该窗口和主数据库中找到自己。

Windows NT 用户或组需要用户或组许可，以连接到 Microsoft SQL Server 上。如果 Windows NT 组的所有成员都连接到 SQL Server 上，可以为该组的所有成员添加一个 SQL Server 注册帐户。如果并非全部组员都应获得许可，则可以为单个 Windows NT 用户添加 SQL Server 注册。

另外，还可以用 Microsoft Management Console 为 Windows NT 验证和 SQL Server 验证添加 SQL Server 注册。图 5.2 表示 Windows NT 验证。

注意:使用 Microsoft Management Console**,把** SQL Server **注册到服务器组中,参见图** 5.2。

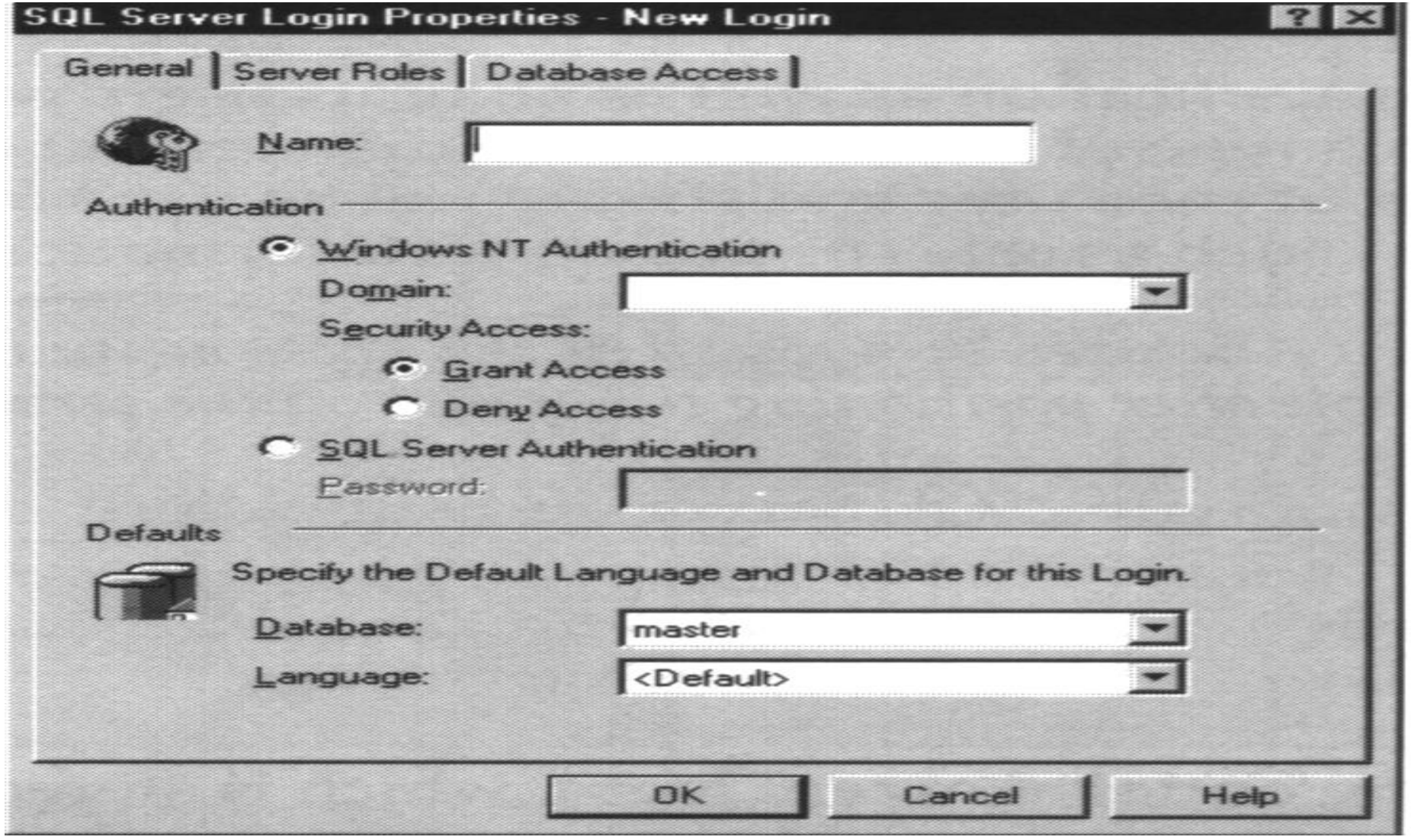


图 5.2 **创建** Windows NT **验证注册** ID

1. 展开服务器名。
2. 右击 Logins，然后选择 New Login。
3. 选择 Windows NT Authentication。
4. 键入 Windows NT 帐户。
5. 设置用户的缺省数据库。

5.2.2SQL Server 验证模式

SQL Server Authentication 模式下与 SQL Server 的网络连接假定为 SQL Server 和客户机之间的非委托连接。用户键入已设置为 SQL Server 标准注册的注册名和密码。SQL Server 将验证注册 ID 和密码本身，并与系统管理员已经预定义成 SQL Server 注册 ID 的注册 ID 相比较。如果系统管理员尚未设置指定的注册 ID，则连接失败。

SQL Server 验证有几种比较好的用途，可以编写许多应用程序来使用它。许多人都喜欢使用这种方法，因为他们了解注册和密码，而 NT 验证可以十分复杂而模糊。SQL Server 验证还可用于 Internet 客户机，以及不是基于 Windows NT 的操作系统。

有时，整个应用程序及其所有用户使用同一个 SQL Server 注册和密码。这种类型的访问易于管理，但由于不跟踪每个用户的标识，而带有安全性风险，所以不应用于包含机密信息的数据库，如销售或人事信息。

注意:将游标定位在服务器组中的服务器上后，可使用 Microsoft Management

Console 的 Tool 菜单上的 SQL Query Analyzer Tool。

要创建 SQL Server 注册(SQL Server 验证)，可执行下列程序：

```
sp_addlogin ' login ' , ' password ' , ' database ' , ' language ' ,
' login_pid ' , ' sid '
```

除了注册外，所有自变量都是可选的，这些自变量都作为系统名存储在主数据库的 syslogins 表中。自变量将在第 23 章“系统存储过程”中详细讨论。

此外，还可以用 Microsoft Management Console 为 Windows NT 验证和 SQL Server 验证添加 SQL Server 注册(图 5.3 展示了 SQL Server 验证)，其步骤如下：

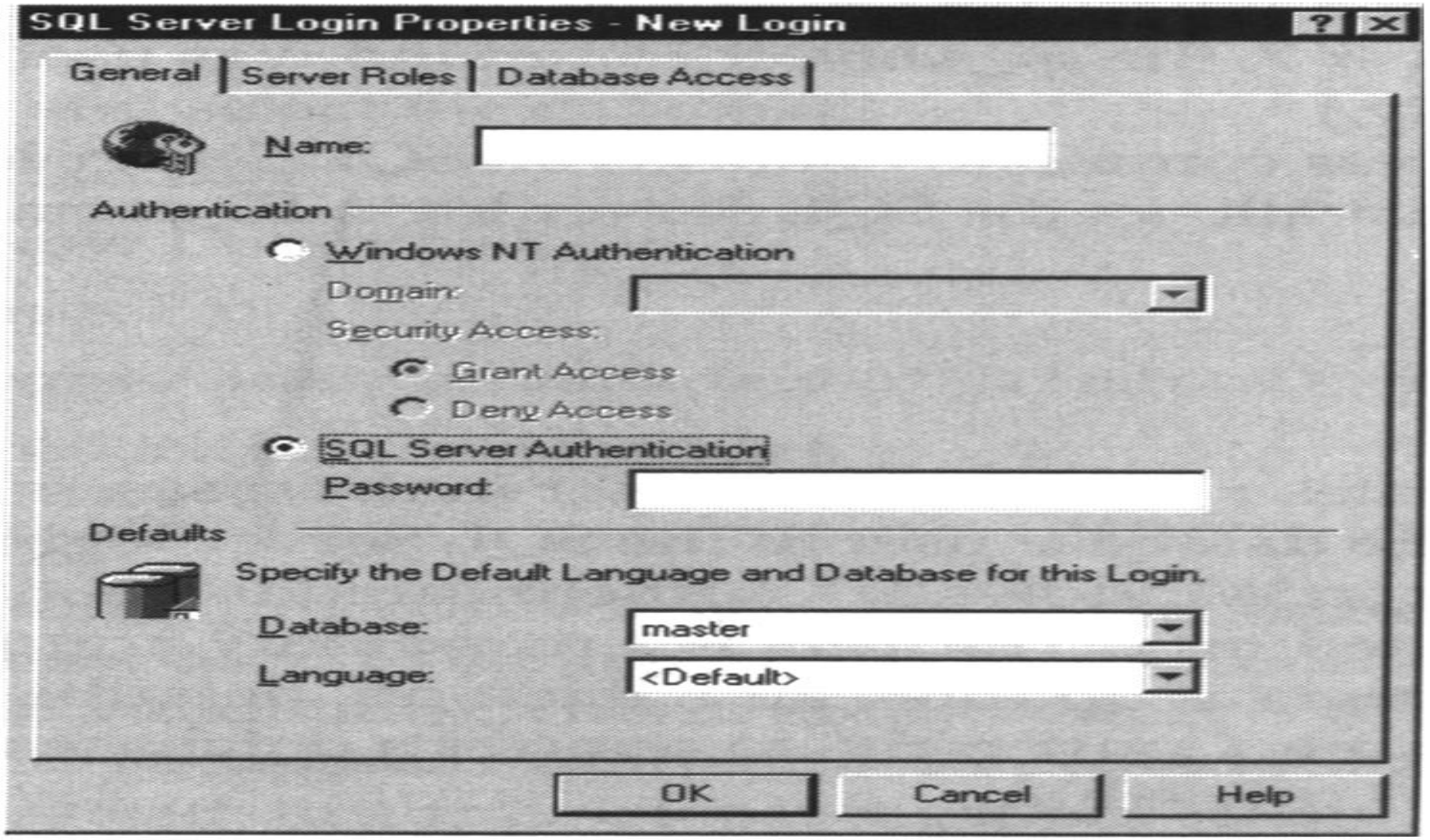


图 5.3 创建 SQL Server 验证注册 ID

1. **展开服务器名。**
2. **右击 Logins，然后选择 New Login。**

3. 选择 SQL Server Authentication。

4. 键入注册名和密码。

5. 为用户选择缺省数据库。

两种类型的注册 ID 之间主要的、可立即看出的差别是，当显示连接到 SQL Server 的屏幕时，采用 Windows NT 验证注册 ID 的用户不必键入注册 ID 和密码，而采用 SQL Server 验证注册 ID 的用户则必须键入。因为 NT 把注册 ID 和密码传递给 SQL Server，再由 SQL Server 检查 NT 帐户或组是否为 syslogins 表的成员，所以 Windows NT 验证发生作用。另一个差别是，如果在选择 Windows NT Authentication 后再输入域名，该域名将自动显示为 Name 域的第一部分(NT 用户必须提前创建)。

总之，注册有两种类型，即 NT 验证注册和 SQL Server 验证注册。注册 ID 用于验证，即识别用户和确认数据库访问。注册 ID 有一个缺省数据库，该数据库在建立连接时使用，并且在创建时为注册 ID 而定义。

5.3 许可的有效性

至此，用户已了解了如何使用两种类型 SQL Server 注册 ID 中的一种来获得 Microsoft SQL Server 的访问权，下面讲述注册 ID 如何与 SQL Server 数据库用户相关联。

5.3.1 数据库用户

SQL Server注册ID和数据库用户ID之间的区别之一是,SQL Server注册ID可用于整个SQL Server。一个注册ID可以与多个数据库用户相关联，所有数据库都可用存储过程sp_grantdbaccess在SQL Server注册ID和数据库用户ID之间建立关系。另一方面，数据库用户ID则只能在SQL Server上的特定数据库中正常工作。

关系必须由用户来定义，该用户在SQL Server注册ID和数据库用户ID之间扮演着db_accessadmin或db_owner的角色。可以在数据库内部创建用户，并把已建立的SQL Server注册ID作为参数传递给sp_grantdbaccess存储过程，来完成上述定义。在这种方式中，如果数据库用户ID没有与有效的SQL Server注册ID相关联，就不能创建数据库用户ID。(在SQL Server 7以前的版本中，使用sp_adduser命令;它仍支持向后的兼容性)。

sp_grantdbaccess命令可把SQL Server注册ID与数据库用户ID连接起来，这就是为什么在特定的数据库中允许使用SQL Server注册ID许可。

sp_grantdbaccess ' login ' , ' databaseusername '

该命令可在数据库的sysuser系统表中创建新的数据库用户，并使数据库用户ID与SQL Server注册ID建立关系,该SQL Server注册ID可以用sp_grantlogin为Windows NT验证用户而创建，也可以用sp_addlogin为SQL Server验证用户而创建。login是一个必选自变量，而databaseusername则是一个可选自变量，如果省略这个自变量，就使用缺省值login，并使两个参数相同。相反，要删除

数据库用户，可使用 sp_revokedbaccess 存储过程。

使用 sp_grantdbaccess 规则如下:

- 在用户定义的事务内不能使用
- SA 注册不能添加到数据库中
- Windows NT 帐户或组需要用域名验证
- 只能由作为某种数据库角色（db_accessadmin 或 db_owner）的用户来执行使用 sp_revokedbaccess 规则为：
- 在用户定义的事务内不能使用
- 不能删除数据库中的公共角色、DBO 用户或固定角色
- 不能删除主数据库和 tempdb 数据库中的 Guest User 帐户
- 不能从 NT 组中删除 Windows NT 用户
- 只能由作为某种数据库角色（db_accessadmin 或 db_owner）的用户来执行
- 如果用户在数据库中拥有对象，就不能删除该用户;必须在使用 sp_revokedbaccess 之前，用存储过程 sp_changeobjectowner 改变该对象的拥有者

注意:使用 Microsoft Management Console，并把 SQL Server 注册到服务器组中。

作为 sp_grantdbaccess 的替代程序，可以使用 Microsoft Management Console 为 Windows NT 验证和 SQL Server 验证创建新的数据库用户(请参阅图 5.4):

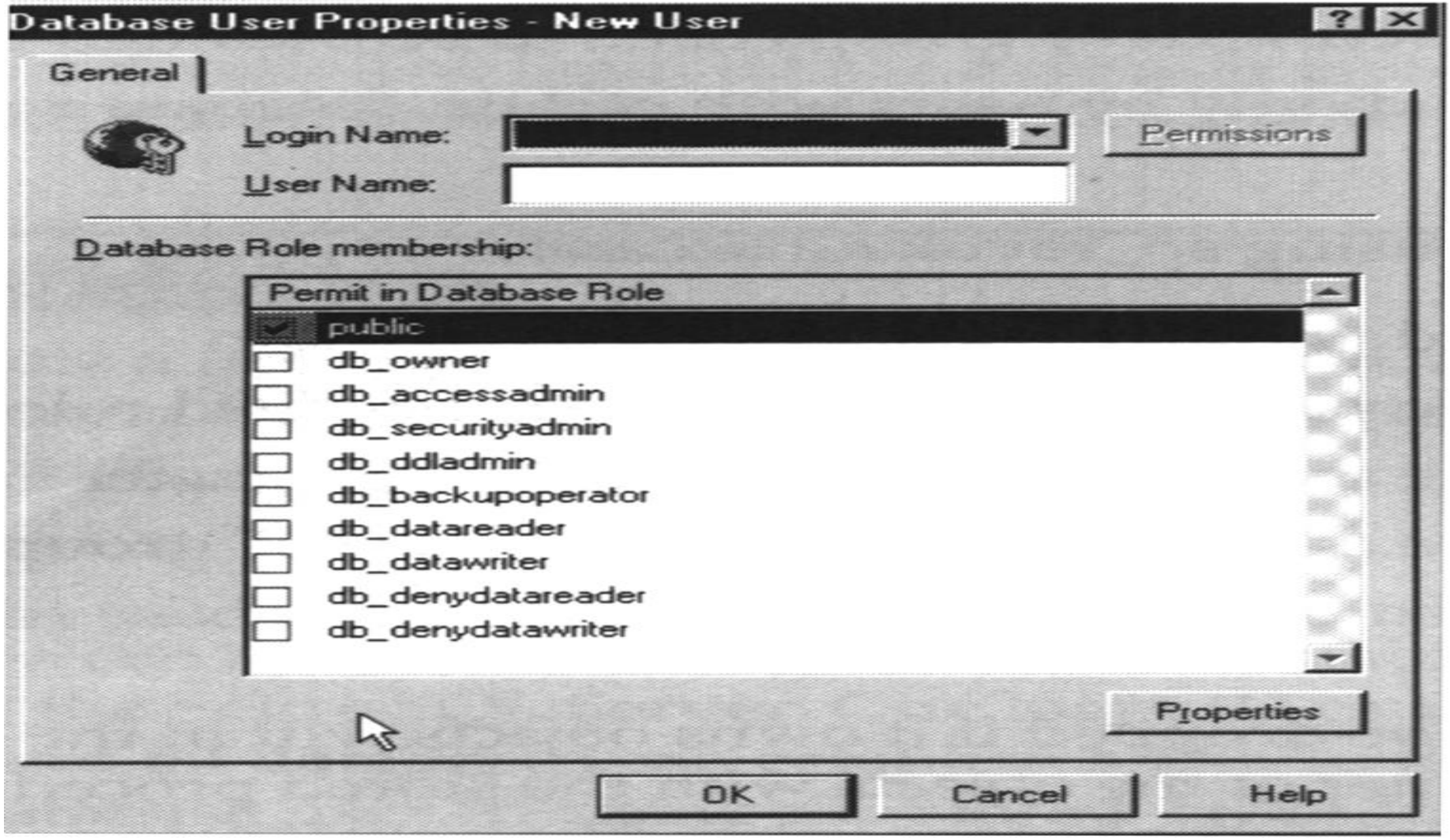


图 5.4 创建新的数据库用户

1. **展开服务器，并选择数据库。**
2. **右击该数据库**;

3. 选择 New，然后再选择 New Database User。
4. 选择注册名。
5. 键入用户名。
6. 选择要放入用户的角色。
7. 单击 OK。

该 sp_grantdbaccess 存储过程是把注册(验证)映射到使用的 SQL Server 数据库用户上的方式，以控制对数据库中对象的访问和执行的操作(许可有效)。这实际上是把验证和许可有效这两部分安全性连接起来，以提供安全访问和特定许可。如果注册 ID 和数据库用户 ID 同名，管理起来会更方便。用户应考虑把这种设置作为系统中的标准设置。

要进一步改进这种访问，而且准许一组用户集体获得特定许可，可使用组和角色。

5.3.2NT 组和 SQL Server 角色

把用户放到组和角色中，将更易于一次管理多用户的许可。组(group)是管理 Windows NT 内的用户或其他组的一种方法。而角色(role)则是管理 SQL Server 内部用户的方法，SQL Server 可包含 SQL Server 注册、Windows NT 注册、组或其他角色。以前版本的 SQL Server 中，数据库组由角色所代替，角色的功能是非常强大的。Sp_addgroup、sp_changegroup、sp_dropgroup 已经包含了向后的兼容性，但它们现在只是处理角色的次要方法。

5.3.2.1NT 组

通过创建组的 SQL Server 注册，可以访问 SQL Server 上整个 NT 组的注册。而 GRANT 语句用于允许用户、角色或组获得访问数据或执行数据库中特定操作的许可。这种许可只准许当前数据库中的用户使用。相反，REVOKE 语句用于剥夺已授予的许可，而 DENY 语句用于阻止用户从发出 GRANT 的人手中获得许可。为 NT 组设置 SQL Server 注册的一个重要方面是一旦进行了设置，就只需为 NT 组增加新的用户。而 SQL Server 注册和数据库许可已经就绪。

许可有两种类型:object(对象)和 statement(语句)许可。对象许可为用户提供访问许可，以便使用诸如数据库中已有的表、视图、程序和扩展程序等对象。而语句许可允许访问特定的 Transact_SQL 语句，如 GREATE TABLE 或 EXECUTE PROCEDURE。

当某个组是某个高级组的成员时，除了为该组本身或用户帐户定义的安全设置外，该组的所有成员还继承了高级组的安全设置。

5.3.2.2SQL Server 角色

如果没有 NT 组，仍可以把用户组织到一个共同单元中，并可使用角色(roles)集中管理这些用户。可以创建用户的功能集(NT 验证和 SQL Server 验证)，以及其他角色。这是简化数据库许可管理的强大工具。

注意:使用 Microsoft Management Console(MMC)，可把 SQL Server 注册到服

务器组中，请参见图 5.5。

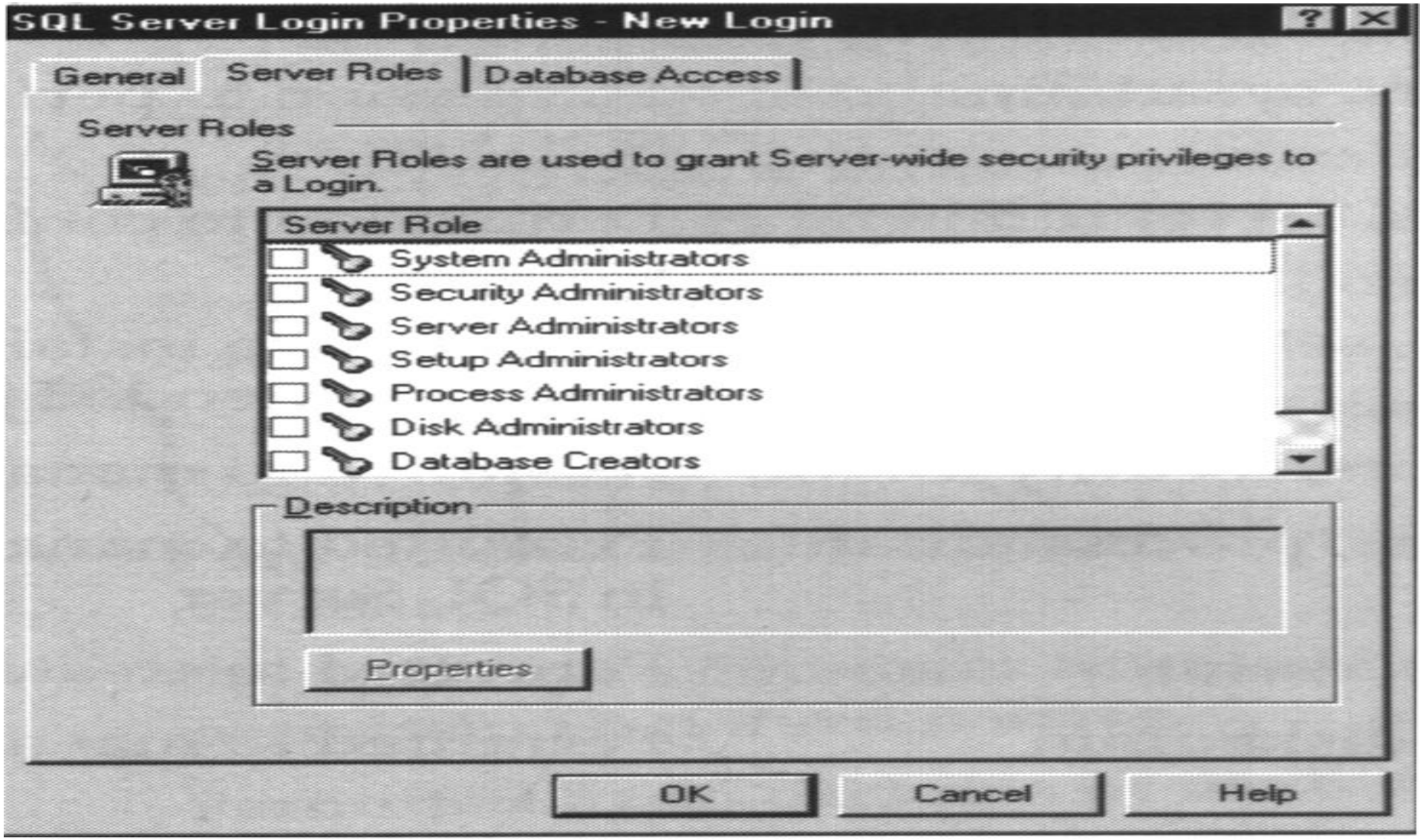


图 5.5 分配固定的服务器角色

使用 Microsoft Management Console 可以为 SQL Server 验证注册 ID 把用户分配给固定角色:

1. 展开 MMC 中的服务器名。
2. 右击 Logins，然后选择 New Login。
3. 选择 SQL Server Authentication。
4. 键入 SQL Server Authentication 或 NT 帐户或 Windows NT Authentication 用户的注册名和密码。
5. 设置用户的缺省数据库。
6. 单击 Server Roles 标签。
7. 单击提供给用户的 Roles。

固定服务器角色　固定服务器角色存在于服务器级并处于特定数据库之外。用户必须拥有 NT 注册帐户或 SQL Server 注册帐户，才能添加到固定服务器角色中。固定服务器角色的任何成员都可以把其他注册添加到角色中。

下表表示 Fixed Server Role 名和 Fixed Server Role 的许可：

固定服务器角色名	该角色的功能
系统管理员，sysadmin	执行 SQL Server 中任何动作的许可
服务器管理员，serveradmin	配置用于服务器的设置的许可
设置管理员，setupadmin	允许安装复制程序，管理扩展程序
安全管理员，securityadmin	允许管理服务器注册
进程管理员，processadmin	允许管理 SQL Server 中运行的进程
数据库创建员，dbcreator	允许创建和改变数据库

续表

磁盘管理员，diskadmin	允许管理磁盘文件

要为固定服务器角色添加成员，请参见图 5.5 的 Microsoft Management Console，Enterprise Manager 方法，也可使用下面的存储过程：

sp_addsrvrolemember ' security_account ' , ' role '

用户将继承正在扮演的角色的所有许可。固定服务器角色表示它们的确切含义，不能删除或改变它们。只有得到执行 sp_addsrvrolemember 许可的人才能成为固定服务器角色的其他成员。syslogins 系统表可更新以指明为哪个固定服务器角色分配哪个注册 ID。

要从固定服务器角色中删除某个成员，请参照图 5.5 所示的 Enterprise Manager 方法，或使用下面的存储过程：

sp_dropsrvrolemember ' security_account ' , ' role '

不能从固定服务器角色中删除 SA(系统管理员)。

固定数据库角色　固定数据库角色存在于数据库级，且处于特定数据库内部。用户必须拥有 NT 注册帐户或 SQL Server 注册帐户才能添加到固定数据库角色中。角色也可以添加到固定数据库角色中。固定数据库角色的任何成员都可以把其他人添加到该角色中。

下表表示 Fixed Database Role 名和 Fixed Database Role 的许可。

固定数据库角色名	该角色的功能
Db_owner	作为 DBO，除了其他 Database Owner 任务外，还允许处理所有其他数据库角色的所有操作
Db_accessadmin	允许和禁止对数据库中的 Windows NT 组／用户和 SQL Server 用户的访问
Db_datareader	允许查看数据库中所有用户表中的所有数据
Db_datawriter	允许修改数据库中所有用户表中的数据(插入、更新、删除)
Db_ddladmin	允许创建、更改和／或删除数据库中的对象
Db_securityadmin	允许管理数据库中的角色和语句以及对象许可
Db_dumpoperator	允许备份数据库
Db_denydatareader	不许查看数据库中的任何数据
Db_denydatawriter	不许修改数据库中的任何数据

要为固定数据库角色添加成员，请参照图 5.6，或使用下面的存储过程:

sp_addrolemember ' role ' , ' security_account '

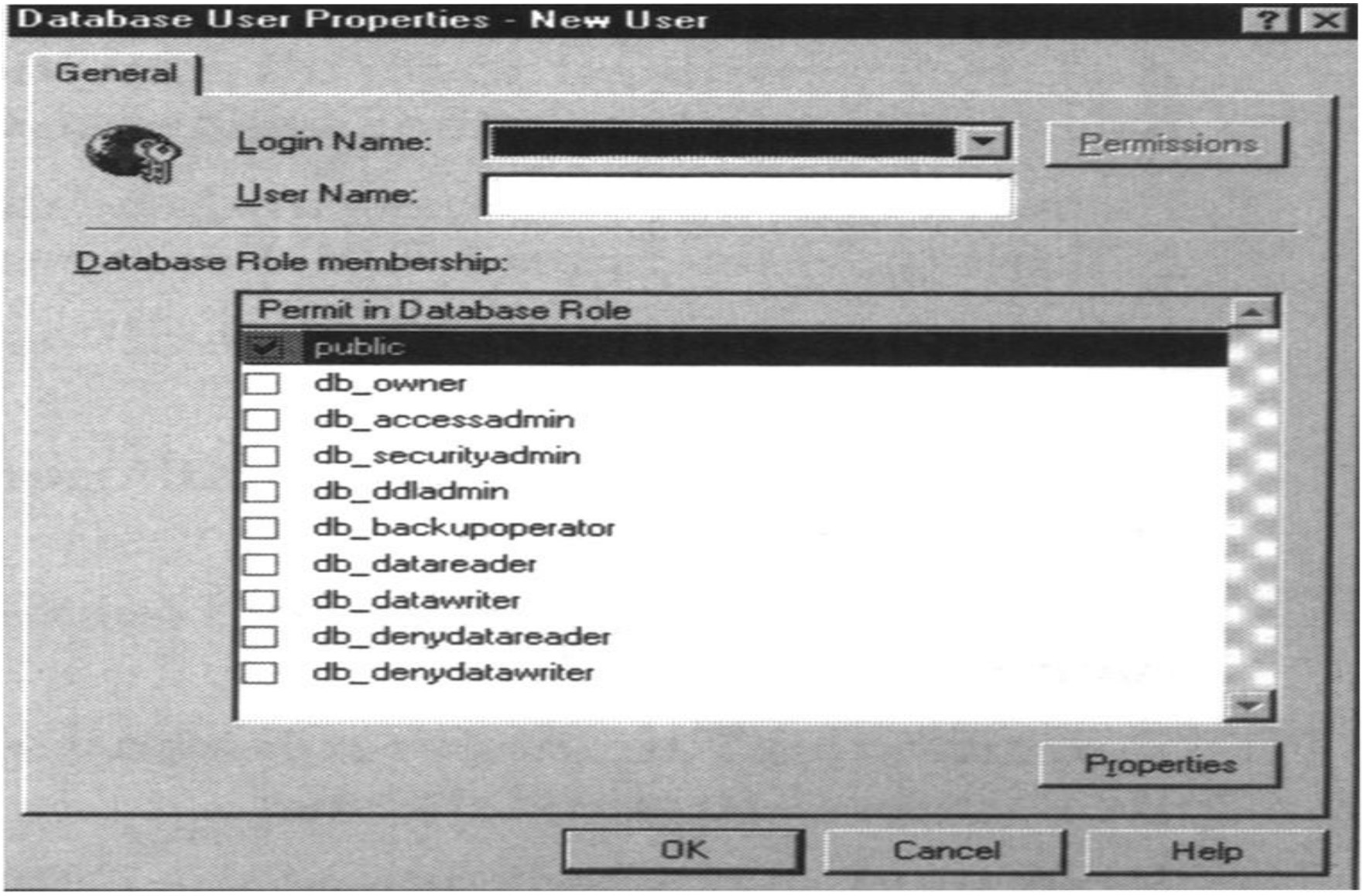


图 5.6 分配固定服务器角色

用户将继承正在扮演的角色的所有许可。如果安全性帐户用于 Windows NT

用户或组，并且不是数据库中的用户，他们就将自动添加为用户。使用 Windows NT 用户或组的安全性帐户前面的域名，例如，DomainName＼securityAccount，可用于安全性帐户自变量。如果安全性帐户用于 SQL Server 用户或角色，那么这些用户或角色必定处在执行命令的数据库中。当安全性帐户代表 SQL Server 用户时，可以用没有域名前缀的安全性帐户作为安全性帐户参数。如果有用户定义的角色，不要试图在数据库级或服务器级上把固定角色添加到用户定义的角色上，因为这样做是无效的。固定数据库角色是其本身的确切含义，不能删除或改变它们。sysusers 系统表可更新以指明哪个数据库用户在扮演哪个固定数据库角色。

除了如图 5.6 所示的固定数据库角色外，Microsoft SQL Server 还有两个特殊的用户／角色许可：

· Guest user 如果用户试图访问数据库，而且尚未成为数据库用户，则该用户将成为 Guest 用户。这意味着他们可以访问数据库中 Guest 用户允许使用的任何对象

· Public role 如果希望为数据库中的每个人提供某些对象的许可，或执行某些操作，就可以为 Public 角色提供许可。为 Public 角色提供的许可可以授予每个人使用下列系统支持的存储过程，可显示每个固定数据库角色的许可：

sp_dbfixedrolepermission ＇role＇

注意：Query Anglyzer 位于 Microsoft Management Console 的 Tools 菜单中。可以在 Query Analyer 窗口中执行存储过程。

从 Query Analyzer 中输出下列信息：

DbFixedRole	许可(Permission)
Db_owner	数据库中的每一项内容
Db_accessadmin	为 db_accessadmin 增加成员
Db_accessadmin	sp_adduser
Db_accessadmin	sp_dropuser
Db_accessadmin	sp_grantdbaccess
Db_accessadmin	sp_revokedbaccess
Db_securityadmin	为 db_securityadmin 增加成员
Db_securityadmin	DENY
Db_securityadmin	GRANT
Db_securityadmin	REVOKE
Db_securityadmin	sp_addrole
Db_securityadmin	sp_addrolemember
Db_securityadmin	sp_droprole
Db_securityadmin	sp_droprolemember
Db_ddladmin	为 db_ddladmin 增加成员
Db_ddladmin	所有 DDL，但 GRANT、REVOKE、DENY 除外
Db_ddladmin	sp_adduserobject
Db_dumpoperator	为 db_dumpoperator 增加成员
Db_dumpoperator	CHECKPOINT
Db_dumpoperator	DUMP DATABASE
Db_dumpoperator	DUMP TRANSACTION
Db_datareader	为 db_datareader 增加成员
Db_datareader	任意对象的 SELECT 许可
Db_datawriter	为 db_datawriter 增加成员
Db_datawriter	任意对象的 DELETE 许可

续表

Db_datawriter	任意对象的 INSERT 许可
Db_datawriter	任意对象的 UPDATE 许可
Db_denydatareader	为 db_denydatareader 增加成员
Db_denydatareader	任意对象都无 SELECT 许可
Db_denydatawriter	为 db_denydatawriter 增加成员
Db_denydatawriter	任意对象均无 DELETE 许可
Db_denydatawriter	任意对象均无 INSERT 许可
Db_denydatawriter	任意对象均无 UPDATE 许可

访问数据库的每个人都可执行存储过程 sp_dbfixedrolepermission 和 sp_helpdbfixedrole。

使用下列系统提供的存储过程，可以显示每个固定数据库角色的名称和说明：

sp_helpdbfixedrole ' role '

注意：Query Analyzer 位于 Microsoft Management Console 的 Tools 菜单中，可以在该窗口中执行存储过程。

从 Query Analyzer 输出的信息如下：

DbFixedRole	描述
Db_owner	DB Owners(拥有者)
Db_accessadmin	DB Access Administrators(访问管理员)
Db_securityadmin	DB Security Administrators(安全管理员)
Db_ddladmin	DB DDL Administrators(DDL 管理员)

续表

Db_dumpoperator	DB Dump Operator(清空操作员)
Db_datareader	DB Data Reader(数据读入员)
Db_datawriter	DB Data Writer(数据写入员)
Db_denydatareader	DB Deny Data Reader(拒绝数据读入员)
Db_denydatawriter	DB Deny Data Writer(拒绝数据写入员)

注意:使用 Microsoft Management Console，可以选择 Server Group 中的 Server，然后选择 Database，并右击 Database Role。

要在数据库中创建新用户定义的角色，请参阅图 5.7 或使用下列存储过程:

sp_addrole ′role′ , ′owner′

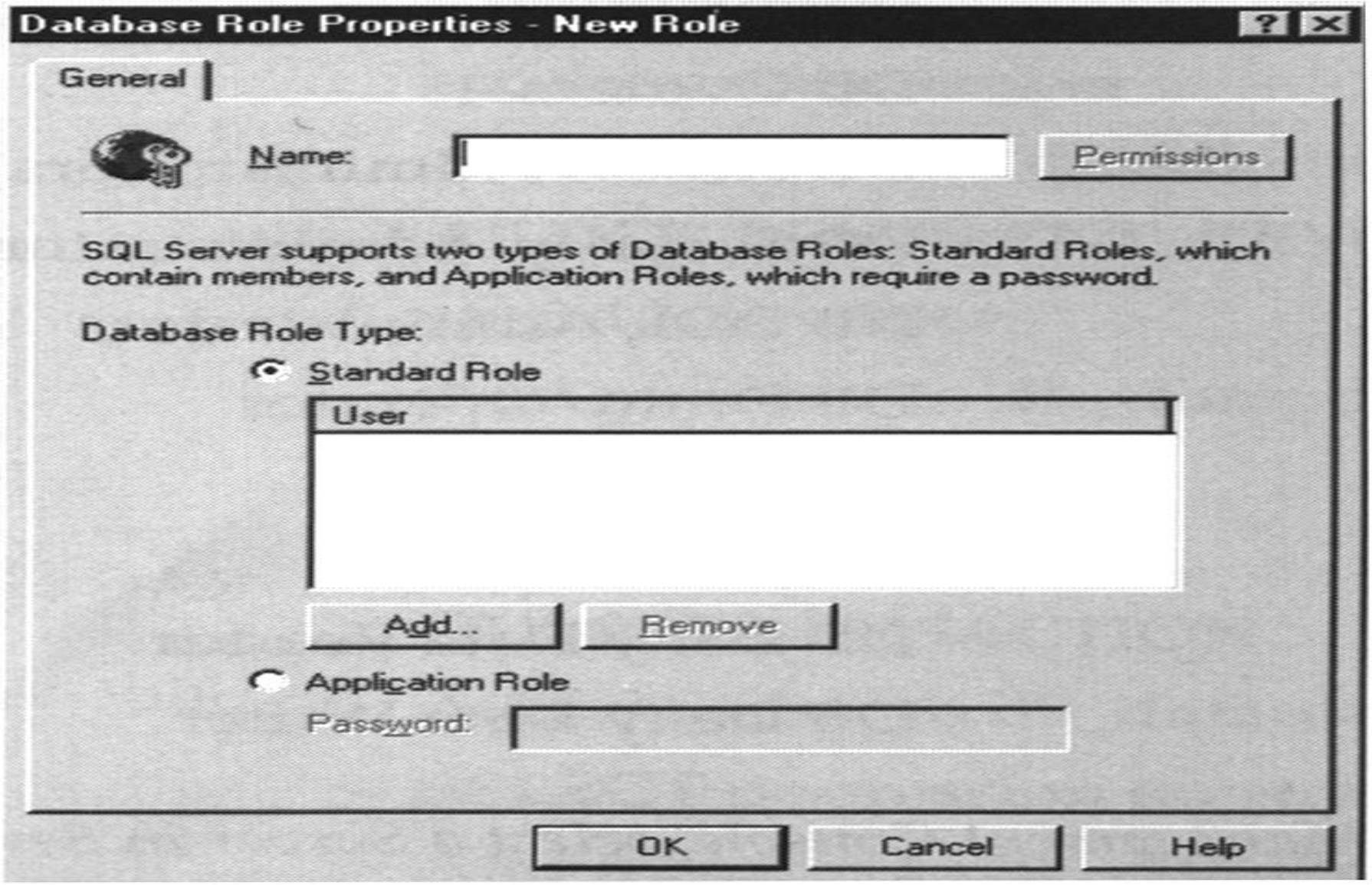


图 5.7 增加用户定义的数据库角色

owner **参数缺省为** DBO **拥有者，然而，如果指定拥有者，则拥有者必须是**

执行命令的数据库中 sysusers 系统表中的数据库用户或数据库角色。只有 db_securityadmin 或 db_owner 角色才允许使用 sp_addrole 存储过程。可以用 GRANT 或 DENY 语句为新的数据库角色分配许可。

要从数据库中删除数据库角色，可在数据库内部执行下列存储过程:

```
sp_droprole 'role'          --删除标准数据库角色
sp_dropapprole 'role'       --删除应用程序数据库角色
```

插入 sysusers 表中的行可从 sysusers 系统表中删除。不能删除固定角色和 public 角色；必须先从该角色中删除数据库用户。使用存储过程 sp_droprolemember 可删除角色中的数据库用户。不要在用户定义的事务内部执行该存储过程，只有 db_owner 或 db_securityadmin 角色的成员才允许使用 sp_droprole。

5.4 授 予 权 限

授予组或角色的许可可由该组或角色的成员继承下来。许可能在不同的层次上授予。给用户授予许可有可能与给另一个角色（该用户是其一个成员）授予的许可冲突。如果用户处于多个组或角色中，或与单个用户许可有冲突，那么在角色之间发生许可冲突时，可应用有一定限制的拒绝和撤消防止用户访问

对象的许可。

许可有以下三种状态:

- Grant(授予)
- Revoke(撤消)
- Deny(拒绝)

5.4.1GRANT 语句

Grant(授予)意味着一个数据库用户可以为另一个数据库用户提供许可，以便查看数据、改变数据和执行数据库操作。只能在当前数据库中为用户授予当前数据库的许可。现在 GRANT 语句又新增了一个 WITH GRAN 选项，它允许给数据库用户的授予许可，也允许该数据库用户给其他用户授予许可。John 向 Jane 提供许可 WITH GRANT，这将允许 Jane 将同样的许可授予 Jay。如果 Jane 为 Jay 提供许可 WITH GRANT，那么 Jay 就能向 Jerry 授予相同的权限。通过给组和角色授予许可 WITH GRANT，也可为一组用户提供这种能力。然而，使用 WITH GRANT 许可时一定要注意，否则将可能不能控制谁向谁授予许可。

许可有以下两种类型:

- 语句许可
- 对象许可

5.4.1.1 语句许可

Transact-SQL GRANT 语句可提供语句许可，语句许可为安全帐户提供了包括 GRANT 语句在内的其他 Transact-SQL 语句的访问权，。在提供语句许可时，可以向用户提供使用下列 Transact-SQL 语句的许可：

ALTER DATABASE
ALTER TABLE
BEGIN TRANSACTION
CHECKPOINT
COMMIT TRANSACTION
CREATE DATABASE
CREATE DEFAULT
CREATE INDEX
CREATE PROCEDURE
CREATE RULE
CREATE TABLE
CREATE TRIGGER
CREATE VIEW
DBCC
DELETE
DISK INIT
EXECUTE
GRANT
对象的 GRANT
INSERT
KILL
RESTORE DATABASE
LOAD TRANSACTION
RAISERROR
READTEXT
RECONFIGURE
REFERENCES
REVOKE
对象的 REVOKE
ROLLBACK TRANSACTION
SAVE TRANSACTION
SELECT

DISK MIRROR　　SET
DISK REFIT　　SETUSER
DISK REINIT　　SHUTDOWN
DISK REMIRROR　　TRUNCATE TABLE
DISK UNMIRROR　　UPDATE
DROP 任意对象　　UPDATE STATISTICS
BACKUP DATABASE　　UPDATETEXT
DUMP TRANSACTION　　WRITETEXT

语句许可的语法如下:

```
GRANT { ALL | statement [,...] }
TO security_account [,...]
```

下表表示语句许可的 Grant 语句自变量及其说明:

语句许可的 GRANT 自变量	描述
ALL	语句:当用作语句许可时能使用所有语句。在提供了所有语句许可的情况下，只有 System Administrator(系统管理员：SA)和 Database Owner(数据库拥有者：DBO)才能提供全部权限，因为只有这些人才拥有全部权限。当然，SA 的权限多于 DBO，因为 SA 涉及服务器范围，而 DBO 只涉及数据库范围

续表

Statement(语句)	如果授予许可的用户是 SA，便可使用 CREATE DATABASE。如果用户不是 SA，则可以为另一个用户提供的语句有 CREATE DEFAULT、CREATE PROCEDURE、CREATE RULE、CREATE TABLE、CREATE VIEW、DUMP DATABASE 和 DUMP TRANSACTION。这种操作能够使被授予人使用该语句。另外，对象拥有者可以执行插入、更新、删除、选择、引用和执行等操作
Security Account(安全帐户)	许可可提供给这种用户，该用户可以是 Microsoft SQL Server User、SQL Server Role、Windows NT User 或 Windows NT Group

下面是语句许可的 GRANT 语句的一个实例：

```
GRANT CREATE DATABASE, CREATE TABLE
TO security_account
GO
```

对于 Windows NT 验证用户，可以使用 DomainName＼SecurityAccount：

```
GRANT CREATE DATABASE, CREATE TABLE
TO DomainName＼SecurityAccount
GO
```

```
GRANT CREATE TABLE
TO rolename
GO
```

5.4.1.2 对象许可

对象许可允许用户拥有下列对象的应用权限：数据库中已有的表、视图、程序和扩展存储过程。如果用户是 sysadmin **和** db_securityadmin **角色的成员，可以在所有数据库中授予所有的许可。如果是对象拥有者，可以给拥有者的任何对象授予许可。**db_owner **角色可以在自己的数据库中授予任何对象的任何许可。**

对象许可的语法如下:

```
GRANT
 {{ALL | permission [,...]}   [(column [,...] )]  ON  { table | view}
 | ON  { stored_procedure  |  extended_procedure}}
To security_account  [,...]
 [WITH GRANT OPTION]
 [AS  { group | role}]
```

对象可的 GRANT **自变量如下**:

对象许可的 GRANT **自变量**	**描述**
ALL	**对象**:**在** ALL **用于提供某个对象的用户许可时，传递该对象的所有权限**

续表

Statement(语句)	如果授予许可的用户是 SA，则可使用 CREATE DATABASE。如果用户不是 SA，可以为另一个用户提供的语句有：CREATE DEFAULT、CREATE PROCEDURE、CREATE RULE、CREATE TABLE、CREATE VIEW、DUMP DATABSE 和 DUMP TRANSACTION。这种操作可使被授予者使用该语句。另外，对象拥有者还可使用插入、更新、删除、选择、引用和执行
TO Security Account(到安全帐户)	为该用户提供许可，该用户可以是 Microsoft SQL Server User、SQL Server Role、Windows NT User 或 Windows NT Group
WITH GRANT OPTION	使用户能把指定的许可授予其他用户
AS {group | role}	该自变量指明授予许可的用户位于哪个组或角色。如果授予人位于多个角色中，则可能发生许可冲突，授予人可以指明允许哪个角色或组授予这些许可
Permission(许可)	对象许可由授予人授予给被授予人。参见下一个表可了解哪些对象有哪些许可
ON Table	用户获准用指定的许可访问的表格名称
ON Column	用户获准用指定的许可访问的列名
ON View	用户获准用指定的许可访问的视图名
ON Stored_procedure	用户获准用指定的许可访问的存储过程名

续表

ON Extended_stored procedure	用户获准用指定的许可访问的扩展存储过程名

下表指定了当用户提供与语句许可相对应的对象许可时的特定对象的许可。

对象名	对象名的许可
表或视图许可	SELECT、INSERT、DELETE、UPDATE、REFERENCES
扩展存储过程许可	EXECUTE
列许可	SELECT、UPDATE
存储过程许可	EXECUTE

下面是使用对象许可的 GRANT 语句的实例：

```
GRANT EXECUTE
ON storedprocedurename
TO security_account
WITH GRANT OPTION
GO
GRANT SELECT，UPDATE
ON tablename
TO public
```

```
GO
GRANT SELECT ON tablename
TO user AS rolename
GO
```

5.4.2 授予许可的特殊规则

在授予许可时需注意以下特定规则:

· 如果用户要用 FOREIGN KEY 约束在表中插入或修改行，且对该表或相应列没有 SELECT 许可，则对此表必须使用 REFERENCES 列(column)许可

· 并非所有语句都需要有许可才能使用。确实需要许可的语句可为数据库增加对象，或管理数据库

· 用户不能授予某些 Transact-SQL 语句的许可。这些语句必须获得特定系统角色(如 sysadmin)中的成员的许可，只有这些人能用 SHUTDOWN 语句关闭数据库

· 语句许可由角色打破。某些角色拥有执行某些语句的许可，例如，CREATE DATABASE 许可是 sysadmin 角色的缺省许可

· 选项 WITH GRANT 只用于对象许可

· 如果有 sysadmin 或 db_owner 角色，可使用 SETUSER 测试许可，以便扮演另一个用户来查看是否已提供了正确的许可

· 在授予许可时可将行插入到 sysprotects 系统表中

· 使用 sp_helprotect 存储过程，可查看某个对象或用户的许可

5.4.3REVOKE 语句

REVOKE 语句可用于删除授予的许可，或撤销对被否定许可的否定。
语句许可的 REVOKE 语法如下:

```
REVOKE {ALL | statement [,...]}
FROM security_account [,...]
```

对象许可的 REVOKE 语法为:

```
REVOKE [GRANT OPTION FOR]
 {ALL [PRIVILEGES] | permission [,...]}
 {{[(column [,...n] )] ON {table | view}
 | {procedure | extended_procedure}}
FROM security_account [,...] [CASCADE]
 [AS { group | role}]
```

下表为 REVOKE 语句的自变量及其说明。

REVOKE 自变量	描述
ALL	对象(Object):当 ALL 用于撤消某个对象的许可时，可删去对该对象的所有许可语句(Statement):当用作语句许可时，可删去语句的所有授予或拒绝的许可。在撤消所有语句许可的情况下，只有 System Administrator(SA)和 Database Owner(DBO)才能撤消所有的许可，因为只有他们才拥有 SA 的所有权限，当然，SA 拥有的许可比 DBO 更多，因为 SA 适于服务器范围，DBO 适于数据库范围
Statement	如果撤消许可的用户是 SA，则可使用 CREATE DATABASE。如果用户不是 SA，则可从另一个用户撤消 CREATE DEFAULT、CREATE PROCEDURE、CREATE RULE、CREATE TABLE、CREATE VIEW、DUMP DATABASE 和 DUMP TRANSACTION。该操作可撤消授予的或拒绝的许可
Security_account	从该用户中撤消许可。安全帐户可以是 Microsoft SQL Server User、SQL Server Role、Windows NT User 或 Windows NT Group
GRANT OPTION FOR	删去 WITH GRANT OPTION 的效果。用户不再能给其它用户授予 WITH GRANT OPTION，但将保留自己的许可
PRIVILEGES	该自变量不做要求，但对 ANSI 兼容机可包含该自变量
AS{group | role}	该自变量表明撤销权限的用户处于的组或角色。如果撤消人处在多个角色中，可能会发生权限冲突，该用户可以指明哪个角色或组拥有撤消这些许可的权限

续表

Permission	对象许可被撤消 Table 许可被撤销的表格名称
Column	许可被撤销的列名
View	许可被撤销的视图名称
Stored procedure	许可被撤销的存储过程名称
Extended stored procedure	许可被撤销的扩展存储过程名
FROM	指明安全帐户列表
CASCADE	从安全帐户中删除许可，同时从已授予特权的所有安全帐户中删除许可

在撤消许可时要注意下列特殊规则:

· 不要试图从系统角色中撤消许可

· 如果在从已授予 WITH GRANT OPTION 许可的用户中撤消许可时 CASCADE 不用作自变量，则返回错误消息

· 在发出撤消语句时，sysprotects 系统表中有一行被删除

REVOKE 语句的几个实例如下:

```
REVOKE CREATE TABLE FROM security_account
GO
REVOKE CREATE TABLE
FROM DomainName＼SecurityAccount
GO
```

要撤消(REVOKE)某个拒绝的许可，可执行下面的程序:

```
REVOKE INSERT ON tablename
```

```
FROM security_account
GO
```

5.4.4DENY 语句

发出拒绝命令可防止用户获得组或角色成员的许可，并在 sysprotects 系统表中增加一行，表示用户在撤消 DENY 之前不能获得许可。

语句许可的 DENY 语法如下:

```
DENY {ALL | statement [,...]}
TO security_account [,...]
```

对象许可的 DENY 语法如下:

```
DENY { ALL [PRIVILEGES] | permission [,...n]}
 {[(column [,...] )] ON { table | view}
 | ON { table | view} [( column [,...] )]
 | {procedure | extended_procedure}}
TO security_account
 [CASCADE]
```

下表表示 DENY 语句的自变量及其描述:

DENY 自变量	描述
ALL	对象(Object):当 ALL 用于拒绝某个对象的用户许可时，可拒绝该对象的所有许可语句(Statement):当用作语句许可时，可消除使用所有语句的能力。在拒绝所有语句许可的情况下，只有 System Administrator(SA)和 Database Owner(DBO)才能拒绝所有许可，因为只有他们才拥有 SA 的所有权限，当然，SA 拥有的权限比 DBO 要多，因为 SA 适于服务器范围，DBO 适于数据库范围
Statement	如果拒绝许可的用户是 SA，则可使用 CREATE DATABASE。如果用户不是 SA，则对另一个用户可拒绝 CREATE DEFAULT、CREATE PRO-CEDURE、GREATE RULE、CREATETABLE、CREATE VIEW、DUMP DATABASE 和 DUMP TRANSACTION。该操作可拒绝使用该语句的用户许可
Security_account	对该用户拒绝许可。该用户可以是 Microsoft SQL Server User、SQL Server Role、Windows NT User 或 Windows NT Group
PRIVILEGES	该自变量可不作要求，但对 ANSI 兼容机可包括该自变量
AS {group | role}	该自变量表明拒绝权限的用户所处的组或角色。如果拒绝人处于多种角色中，则可能出现许可冲突，该用户可以指明哪个角色或组有权拒绝这些许可
Permission	被拒绝的对象许可或语句许可
Table	许可遭到拒绝的表格名
Column	许可遭到拒绝的列名
View	许可遭到拒绝的视图名称
Stored_procedure	许可遭到拒绝的存储过程名
Extended_procedure	许可遭到拒绝的扩展存储过程名

续表

TO	指明用户列表
CASCADE	拒绝对该用户的许可，对已授予许可的用户拒绝所有用户的许可，以及拒绝对授予许可的所有用户的许可，等等

在拒绝许可时要注意如下规则:

· 如果拒绝单个用户，即使这些用户以后添加到具有许可的组中，这些用户也将得不到许可

· 使用 REVOKE 可删去用户的拒绝许可

· 在发出拒绝语句时，sysprotects 系统表插入一行

使用 DENY 语句的实例如下:

```
DENY SELECT, INSERT, UPDATE, DELETE
ON tablename
TO security_account
GO
DENY CREATE TABLE TO rolename
GO
```

要改变某个对象在 Microsoft Management Console 中的许可，可执行以下步骤:

1. 展开服务器名。
2. 展开数据库，然后展开特殊数据库名。
3. 在范围面板中选择对象类型。
4. 在结果面板中右击该对象。

5. 选择 Task，再选择 Manage Permissions。
6. 选择 List All Users／DB Roles。
7. 选择想要改变的许可。

5.5 实现安全性的若干建议

数据库管理员要实现安全性，可采取的措施如下:

· 确定每个用户有唯一的注册 ID。创建一个注册 ID，并允许许多用户用该注册 ID 访问数据库，但并不是一个很好的安全系统，因为无法跟踪是谁在数据库中正在做什么

· 设置组和角色，统一管理用户。这将削减管理安全性的额外开销，更有利于确保访问的精确性和标准化

· 使 DBO 成为所有数据库对象的拥有者

· 最好通过使用存储过程使用户执行所有的插入、更新和删除操作。只需通过 Transact-SQL 就拥有更高级的控制，并准确了解用户对数据所做的操作

· 触发程序可用来防止对数据的意外访问，从而防止不理想的写入操作

· 对于 SA 访问，可把单个用户放在 sysadmin 角色中

· 用公司的组织图表把访问许可分成多个功能角色

5.6 所 有 权 链

对象可依赖于其他对象，这种依赖性称为 ownership chain(所有权链)。如果用户拥有全部对象及其从属对象，那就万事俱备。然而，正是在用户还未拥有视图或存储过程所引用或需要的全部对象时，事情开始变得复杂，这种情况称为断开的所有权链。SQL Server 可检测链中每个对象的许可，该链的下一个链接由另一个用户所拥有。

如果用户授予其他人许可，以使用内含同一个用户拥有的表的视图及存储过程，那么可不必为该视图或存储过程中所引用的单个表格授予许可。

5.7xp_cmdshell

xp_cmdshell 可将给定的命令串作为操作系统命令的外壳来执行，并允许用户执行 SQL Server 内部的 NT 命令。

xp_cmdshell {'command_string'} [,no_output]

下表表示 xp_cmdshell 自变量及其描述:

xp_cmdshell 自变量	描述
'command_string'	最多允许 255 个字符的操作系统命令
no_output	不将输出返回给用户的可选参数

应在某种环境下评价 xp_cmdshell 的使用，在这种环境下，目的是控制服务器上或 NT 域中对操作系统文件的访问。下列各项目指明可能发生的安全性风险：

· 如果运行 Microsoft SQL Server 的帐户被授予了许多许可，如域管理员，那么任何被授予执行 xp_cmdshell 的许可的用户，都将能用运行 Microsoft SQL Server 的帐户特权来执行任何操作系统命令。缺省值是具有本地管理特权的用户帐户，但是，如果该帐户具有域管理员特权，则整个域都可显示出来

· 要限制 xp_cmdshell 访问，可在 Microsoft SQL Server 设置程序中设置 xp_cmdshell-Impersonates Client 选项。当该复选框处于复选状态时，只有已通过受托连接连接到 SQL Server 的用户，和该计算机上本地 Administrators 组的成员才允许使用 xp_cmdshell。当某个非 SA 用户用 Impersonates Client 选项集执行 xp_cmdshell 时，就在 SQLServerAgentCmdExec 用户帐户中运行要求的命令，该用户帐户与 SQL Server Agent 为由所有非 SA 用户输入的调度任务而使用的帐户相同

· 如果选择了选项 xp_cmdshell-Use SQLServerAgentCmdExec Account for NonAAs,并且用户是作为 SA 注册到 SQL Server 中的,则该命令可在 SQL Server 的安全性下执行，而不是在 SQLServerAgentCmdExec 的安全性下执行的使用 QueryAnalyzer 可执行 xp_cmdshell 扩展存储过程,Query Analyzer 可从 Enterprise Manager Tools 菜单中调用(如图 5.8 所示)。

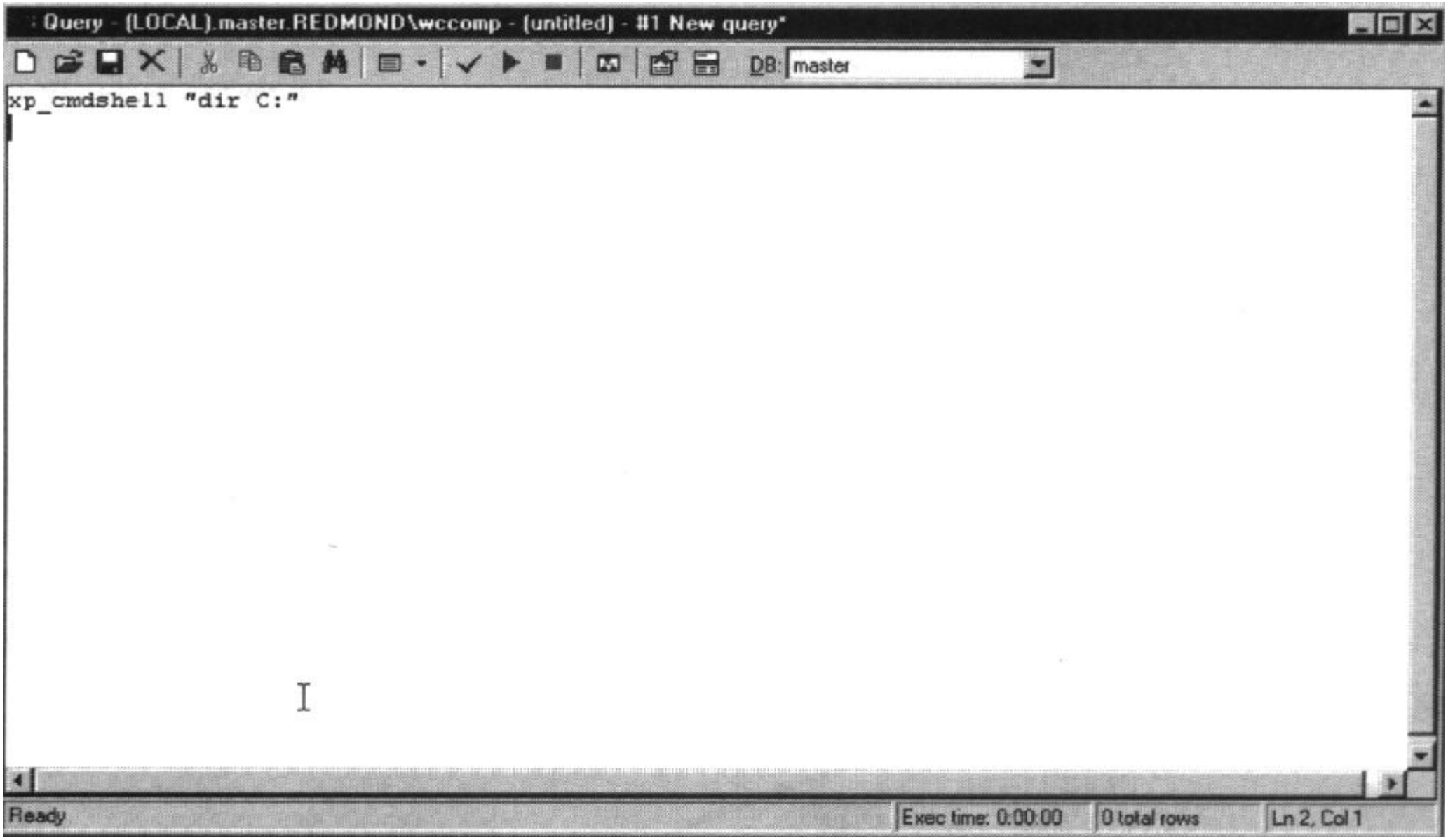


图 5.8 用来执行 xp_cmdshell 的 Query Analyzer

第 6 章 Transact-SQL 增强工具

本章将介绍 Transact-SQL 增强工具的内容，Transact-SQL 增强工具是 Microsoft SQL Server 7 版本的组成部分。

6.1 增强的存储过程

存储过程是 Transact-SQL 语句编译而成的对象，可以用 EXECUTE storedprocedurename 语句运行它。存储过程可以返回一个或多个结果集，整型返回代码、OUTPUT 自变量或参数。

存储程序有许多优点：高速度、可重用性、商业逻辑封装、保护尚未优化的查询、减少网络拥挤、以及安全性(因为只需对存储过程授予许可，而对它访问的对象不需授予许可)。存储程序是 Transact-SQL 开发人员与数据库交互操作、提高数据库性能和可靠性的基础。

Microsoft SQL Server 能提供永久或临时存储过程。临时存储过程存储在 tempdb 系统数据库中。

6.1.1 存储过程中的延迟名解析性能(Delayed Name Resolution Behavior)

由于允许SQL Server开发人员创建在Transact-SQL代码中并不存在的存储过程、批处理文件或触发程序及引用对象，延迟名解析可节省Transact-SQL编程人员的大量时间和精力。在以前的版本中，Transact-SQL编程人员必须按特定的顺序创建对象。例如，在6.x版本中，当创建存储过程时，存储过程在创建脚本的Transact-SQL代码中所引用的对象必须在分析该存储过程(即句法分析)和创建存储过程对象时存在于数据库上。如果在数据库上找不到存储过程所引用的对象，就无法创建存储过程对象。在以前的版本中，SQL Server会返回“引用对象不存在”的错误声明，为了创建引用某个对象的存储过程(也可以是触发程序)，就必须创建该对象。这是非常费时的，也不符合逻辑的循环。

在Microsoft SQL Server 7中，创建存储过程时若引用对象不存在，当分析该存储过程(分析语句的合法性)时，即使在数据库上找不到存储过程或触发程序中引用的对象，SQL Server也允许创建存储过程，并将它放到数据库上。在以后的解析运行阶段会解析名称，解析阶段是另一个合法性分析阶段，它发生在运行期间，而不是对象创建期间。如果存储过程在运行期间引用了对象（这可能是几天之后的事情），而被引用的对象仍未创建，SQL Server将发出一个运行错误的信息，因为存储过程通不过解析阶段。

这样会使Transact-SQL的开发更加容易，因为在创建存储过程时可能有许多对象不存在，但是它们将在执行存储过程的运行期间创建。在解析阶段的运

行期间，可以阐明查询计划(Query plan)。

Microsoft SQL Server 用表中的数据量、表中出现的索引等信息来建立查询表。现有引用对象的数据必须在解析阶段存在，但是因为在这一阶段并不实际执行存储过程对象，所以引用对象名不必存在于对象创建语法验证阶段。

SQL Server Optimizer 必须有关于现有(existing)对象的信息，才能阐明该查询计划，并把该计划放入 procedure cache(程序高速缓存)的内存中(参见图 6.1)。然后 Microsoft SQL Server 使用该查询计划执行存储过程对象。

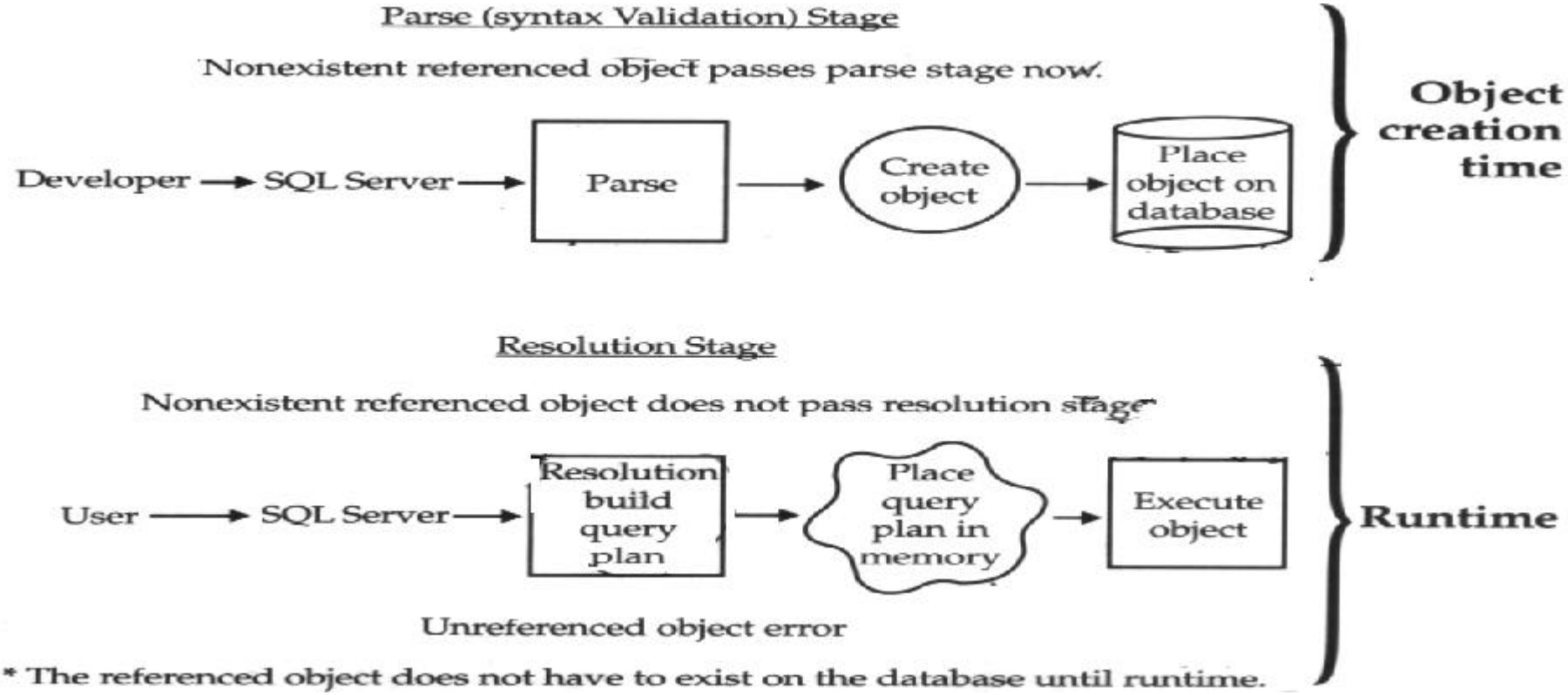


图 6.1 存储过程和触发器的新延迟名称解析

6.1.2Alter Procedure 语句

Microsoft SQL Server 7的另一个受欢迎的变化是：它能改变存储过程对象，而不必从数据库和 syscomments 系统表中删去它来改变存储过程对象。改变存储过程的能力是非常重要的，因为存储过程和触发程序都带有其他引用和许可，在删除对象时会销毁它们。只改变对象而不是删除它将保留与该对象相关的引用和许可。下表展示了谁能使用 Alter Procedure 语句：

6.1.2.1ALTER PROCEDURE 的用法

Stored Procedure Owner
Data Definition Language Administrators
DBO
ALTER PROCEDURE 语句的语法如下：

```
ALTER PROC［EDURE］  procedure name  ［;number］
 ［({@argument_name } data_type  ［= default］   ［OUTPUT］ )］
 ［,...］
WITH {RECOMPILE | ENCRYPTION | RECOMPILE, ENCRYPTION}］
 ［FOR REPLICATION］
AS
Transact-SQL statements  ［...］
```

表 6.1 给出并解释了 ALTER PROCEDURE 的每个参数。

表 6.1ALTER PROCEDURE 的参数

Alter Procedure 的参数	描述
Procedure Name	该参数是用户希望改变的程序名
Number	该参数用来组合同名的存储过程。它是一个可选参数
Argument Name	此参数是用来把值传递给存储过程的自变量。在该自变量名前使用@符号。这表明用户在执行时必须把值传递给存储过程。用户自定义的存储程序可以有 0 至 1024 个自变量
Data Type	指定该参数的数据类型。唯一不能传递的数据类型是 image 数据类型
Default	可以为该自变量提供一个缺省值
OUTPUT	指定自变量可用于执行存储过程后的返回值
RECOMPILE	查询计划不能保存在内存中，以便下一次执行该存储过程时使用，用户每次要求执行存储过程时都会生成新的查询计划。必须使用 RECOMPILE 参数，如果没有给定重新编译参数，则创建存储过程时不会采用重新编译设定的缺省值
ENCRYPTION	存储过程用加密格式存储于数据库的 syscomments 系统表中。如果要使用加密，必须在 alter 中使用该参数。必须明确给定 ENCRYPTION 参数，如果没有给定加密参数，则不会采用创建存储过程时加密设定的缺省值
FOR REPLICATION	在创建或改变作为过滤器的存储过程时，可使用 FOR REPLICATION，且只有通过复制才能使用。该参数不能与 WITH RECOMPILE 选项一起使用

续表

AS	单词 AS 指明一个或多个 Transact-SQL 语句，它们组成随后的存储过程。
SQL Statements	这些语句是存储过程的实际 Transact-SQL 代码

6.1.3 新的系统存储过程

Microsoft SQL Server 7 创建了许多新的系统存储过程。这些新系统存储过程分为如下类别：

- Distributed Query Related
- Cursor Related
- Security Related
- Web Related
- Job Related
- Other

系统存储过程的详细内容将在第 23 章中介绍。

6.1.3.1 新分布式查询相关系统存储过程

新分布式查询相关系统存储过程可用于分布式查询，并包含新的 OLE DB 连接工具。许多工具都已转换为使用 OLE DB 的功能。表 6.2 描述了新的分布

式查询相关系统存储过程。

表 6.2 新分布式查询相关系统存储过程

新分布式查询相关系统存储过程	新功能的描述
sp_addlinkedserver	创建链接服务器
sp_droplinkedserver	删去用于分布式查询的链接服务器
sp_addlinkedsrvlogin	创建或更新映射到链接服务器的 SQL Server 注册
sp_droplinkedsrvlogin	创建或更新映射到链接服务器的 SQL Server 注册
sp_linkedservers	显示本地服务器的链接服务器列表
sp_catalogs	显示命名链接服务器的类别列表
sp_columns_ex	显示目的远程表的列信息
sp_foreignkeys	显示目的远程表的外来键
sp_primarykeys	显示目的远程表的主键列
sp_indexes	显示目的远程表的索引信息
sp_tables_ex	显示匹配远程表的表信息
sp_serveroption	显示分布式查询的选项

6.1.3.2 新的游标相关系统存储过程

表 6.3 描述了新的游标相关存储过程的功能。当需要以每次一行的方式处理结果集时，可使用游标。

表 6.3 新的游标相关系统存储过程

新的游标相关系统存储过程	新功能的描述
sp_cursor_list	sp_cursor_list 存储过程显示用于连接的当前打开的服务器游标的信息，如游标的类型。
sp_describe_cursor	sp_describe_cursor 存储过程显示服务器游标的属性
sp_describe_cursor_columns	sp_describe_cursor_columns 存储过程显示服务器游标结果集中列的属性，例如每个游标列的名称和数据类型
sp_describe_cursor_tables	sp_describe_cursor_tables 存储过程显示服务器游标的结果集中表的属性

6.1.3.3 新的安全性相关系统存储过程

表 6.4 描述了新存储过程的功能。这些程序可用于 Microsoft SQL Server 7 中新的安全特性，如角色。

表 6.4 新的安全性相关存储过程

新的安全性相关存储过程	新功能的描述
sp_dbfixedrolepermission	sp_dbfixedrolepermission 存储过程显示每个固定数据库角色的许可
sp_helpdbfixedrole	sp_helpdbfixedrole 存储过程删去用于分布式查询的连接服务器

6.1.3.4 增强的 Web 相关系统存储过程

表 6.5 表示的是 Microsoft SQL Server 中的新增强型 Web 相关的存储过程。

表 6.5 新的 Web 相关的存储过程

新的 Web 相关的存储过程	新的功能性描述
sp_savehtmltable	sp_savehtmltable 存储过程把 HTML 表输入到特定的 Microsoft SQL Server 表中
sp_makewebtast	sp_ makewebtast存储过程通过创建任务生成一个 HTML 文档，该文档包含由已执行的查询返回的数据

6.1.3.5 新的作业相关系统存储过程

SQL Server Agent 使用新的作业相关存储过程，该程序带有大大扩充的作业处理调度工具。使用 SQL Server Agent 可避免与这些存储过程合并在一起的复杂性，这在本书第 4 章通过 SQL Server Agent 来介绍。表 6.6 描述了新的作业相关系统存储过程的功能。

表 6.6 新的作业相关系统存储过程

新的作业相关存储过程	新功能的描述
sp_add_alert	增加一个警告
sp_add_category	增加作业、警告或操作员的类别
sp_add_job	增加一个作业
sp_add_jobschedule	增加一个作业调度

续表

sp_add_jobserver	增加可用于作业的服务器列表
sp_add_jobstep	增加一个作业步骤
sp_add_notification	增加一个操作员通知
sp_add_operator	增加一个操作员
sp_add_targetservergroup	增加用于多重服务器作业的目标服务器组
sp_add_targetsvrgrp_member	增加一个目标服务器组成员
sp_apply_job_to_targets	把作业应用到目标服务器
sp_delete_category	删除一个类别
sp_delete_job	删除一个作业
sp_delete_jobschedule	删除一个作业调度
sp_delete_jobserver	删除一个可用于作业的服务器
sp_delete_jobstep	删除一个作业步骤
sp_delete_notification	删除一个通知
sp_delete_operator	删除一个操作员
sp_delete_targetserver	删除一个目标服务器
sp_delete_targetservergroup	删除一个目标服务器组
sp_delete_targetsvrgrp_member	删除一个目标服务器组成员
sp_downloaded_row_limiter	用特定服务器的下载状态 1 来限制 sysdownloadlist 系统表中的行
sp_help_category	显示类别信息
sp_help_downloadlist	列出命名作业的 sysdownloadlist 系统表中的所有行。如果没有指定作为参数的作业，则列出所有行
sp_update_notification	更新通知

续表

sp_update_targetservergroup	改变 targetservergroup 的名称
sp_help_job	显示作业的信息
sp_help_jobhistory	显示作业的历史信息
sp_help_jobschedule	显示作业调度信息
sp_help_jobserver	显示作业服务器信息
sp_help_jobstep	显示作业步骤信息
sp_help_operator	显示操作员信息
sp_help_targetserver	显示目标服务器信息
sp_help_targetservergroup	显示目标服务器组信息
sp_manage_jobs_by_login	管理指定注册的作业
sp_msx_defect	从多重服务器操作中删除服务器
sp_msx_enlist	把服务器增加到多重服务器操作中
sp_post_msx_operation	把行插入 sysdownloadlist 系统表，通知目标服务器下载已公布的作业操作
sp_purge_jobhistory	删除作业的历史行
sp_remove_job_from_targets	从目标服务器或目标服务器组中删除已命名的作业
sp_resync_targetserver	在多重服务器环境下使已命名的服务器同步运行
sp_start_job	运行一个作业
sp_stop_job	中止一个作业
sp_update_category	更新类别名
sp_update_job	更新作业的属性
sp_update_jobschedule	更新作业的设置
sp_update_jobstep	更新作业步骤的设置

续表

sp_update_targetservergroup	更新目标服务器组名称
sp_help_notification	显示通知的信息
sp_update_operator	更新操作员信息

6.1.3.6 其他新系统存储过程

表 6.7 描述了 Microsoft SQL Server 中的其他几个新存储过程。

表 6.7 其他新存储过程

新存储过程	新功能的描述
sp_validname	sp_validname 存储过程决定 SQL -Server 标识名是否有效
sp_dbcmptlevel	sp_dbcmptlevel 存储过程是购买期间与新版本 7 特性的桥梁，直到可以修改应用程序以充分利用版本 7 的新功能为止

6.2 新系统扩展存储过程

新系统扩展存储过程很少在 Microsoft SQL Server 的日常工作中使用，除非它通过一个 Microsoft SQL Server GUI 工具来使用。下述列表中的扩展存储过程可由以前称为 SQL Trace 的 SQL Profiler GUI 工具来使用：

xp_trace_addnewqueue　　xp_trace_pausequeue

xp_trace_deletequeuedefinition
xp_trace_destroyqueue
xp_trace_enumqueuedefname
xp_trace_enumqueuehandles
xp_trace_eventclassrequired
xp_trace_getappfilter
xp_trace_getconnectionidfilter
xp_trace_getcpufilter
xp_trace_getdbidfilter
xp_trace_getdurationfilter
xp_trace_geteventfilter
xp_trace_geteventnames
xp_trace_getevents
xp_trace_gethostfilter
xp_trace_gethpidfilter
xp_trace_getindidfilter
xp_trace_getntdmfilter
xp_trace_getntnmfilter
xp_trace_getobjidfilter
xp_trace_getowneridfilter
xp_trace_getpercentfilter
xp_trace_restartqueue
xp_trace_savequeuedefinition
xp_trace_setappfilter
p_trace_setconnectionidfilter
xp_trace_setcpufilter
xp_trace_setdbidfilter
xp_trace_setdurationfilter
xp_trace_seteventclassrequired
xp_trace_seteventfilter
xp_trace_sethostfilter
xp_trace_sethpidfilter
xp_trace_setindidfilter
xp_trace_setntdmfilter
xp_trace_setntnmfilter
xp_trace_setobjidfilter
xp_trace_setowneridfilter
xp_trace_setpercentfilter
xp_trace_setqueueautostart
xp_trace_setreadfilter
xp_trace_setserverfilter
xp_trace_setseverityfilter

xp_trace_getqueueautostart
xp_trace_getreadfilter
xp_trace_getseverfilter
xp_trace_getseverityfilter
xp_trace_getspidfilter
xp_trace_getspidfilter
xp_trace_getuserfilter
xp_trace_getwritefilter
xp_trace_loadqueuedefinition
xp_trace_setspidfilter
xp_trace_setuserfilter
xp_trace_setwritefilter
xp_trace_setqueuedestination
xp_trace_getqueuedestination
xp_trace_setqueuedestination
xp_trace_startconsumer

这些扩展存储过程可由 SQL Server Profiler 使用，下一章介绍新数据库工具时将论述 SQL Server Profiler。

6.3 触 发 程 序

在 Microsoft SQL Server 中，对触发程序的功能和特性进行了改进。触发程序已经增加了下列特性：

- 改变触发程序
- 可递归调用本身的触发程序
- 多个触发程序对应一个触发行为

6.3.1 改变触发程序语句

现在用户能改变触发程序对象，而不必从数据库和 syscommands 系统表中删除该对象。这是非常重要的，因为触发程序携带了在删除对象时被销毁的其他引用和许可。改变该对象而不是删除它，可以保留与该对象相关的引用和许可。如果使用 WITH APPEND 选项，则对每个 Update、Insert 和 Delete 而言，可存在多个触发程序，这意味着对每个 Update、Insert 和 Delete 而言，可激活多个触发程序。WITH APPEND 的功能存在于由 sp_dbcmptlevel 设置的 7 级兼容性。下表描述了谁可使用 Alter Trigger 语句：

Alter Trigger 的用法

Table Owner

Data Definition Language Administrators

DBO

其语法如下：

```
ALTER TRIGGER trigger_name
ON table
 [WITH ENCRYPTION]
{FOR { [,]  [INSERT] [,]  [UPDATE]  [,]  [DELETE] }
 [NOT FOR REPLICATION]
AS
sql_statement  [...n]
```

```
}
|
{FOR { [,]  [INSERT]  [,]  [UPDATE] }
 [NOT FOR REPLICATION]
AS
IF UPDATE (column)
 [{AND |OR} UPDATE (column)  [,...n]]
sql_statement  [...n]
}
```

表 6.8 **描述了** Alter Trigger **的自变量。**

这是如何用 Alter Trigger **语句改变插入触发程序的实例：**

```
ALTER TRIGGER mytrigger
ON tablename
FOR INSERT
AS RAISERROR ('mytrigger error', 1, 2)
```

表 6.8Alter Trigger Statement **的自变量**

Alter Trigger **参数**	**描述**
TriggerName	**要改变的触发程序名**
TableName	**触发程序将执行的表名**
WITH ENCRYPTION	**在** syscomments **系统表中加密编码关键词**

续表

{ [,] [INSERT] [,] [UPDATE] [,] [DELETE] [,] [INSERT] } | { [,] [UPDATE] }	表示哪个语句将激活该触发程序
NOT FOR REPLICATION	如果与复制相关的 sqlrepl 注册 ID 修改了表格，则不能激活该触发程序
AS	下一个是 Transact-SQL 语句
Transact-SQL	语句是触发程序的 Transact-SQL 语句
IF UPDATE(ColumnName)	根据对指定的列是 INSERT 还是 UPDATE，来提供 IF 逻辑
ColumnName	检查 INSERT 或 UPDATE 操作的列名

6.3.2 在触发程序内部不允许使用的语句

在触发程序中可使用的 Transact-SQL 语句是有限的。在触发程序中不能使用下列 Transact-SQL 语句：

ALTER DATABASE
ALTER TABLE
ALTER TRIGGER
CREATE DATABASE
CREATE DEFAULT
CREATE PROCEDURE
LOAD LOG
RESTORE LOG
REVOKE
SELECT INTO(因为它可创建表格)
TRUNCATE TABLE
ALTER PROCEDURE

CREATE RULE
CREATE TABLE
CREATE TRIGGER
DENY
DISK INIT
DROP DATABASE
DROP DEFAULT
DROP PROCEDURE
DROP RULE
DROP TRIGGER
DROP VIEW
LOAD DATABASE
ALTER VIEW
CREATE INDEX
CREATE TABLE
CREATE VIEW
DISK RESIZE
DROP INDEX
DROP TABLE
GRANT
RESTORE DATABASE
RECONFIGURE
UPDATE STATISTICS

6.3.3 使触发程序能递归调用它本身

如果设置了 sp_dboption 递归触发程序选项，并且更新了触发程序内部的表格，该触发程序就可以递归调用自身。

下面是一个实例：

```
sp_dboption mydb,' recursive triggers ' ,  true
```

在这个实例中，如果更新了触发程序内部的表，就把 recursive trigger 数据库选项设置为真，而且允许该触发程序递归调用其自身。

6.3.4 多个触发程序对应一个触发动作

如果指明了 WITH APPEND 或把 sp_dbcmptlevel 的兼容性等级设置为 70，就有多个触发程序对应于数据库修改操作 Update、Insert 和 Delete。

6.4 新数据类型

Microsoft SQL Server 中的数据类型包含新功能，以支持新的增强性能。表 6.9 描述了数据类型的新功能。

表 6.9 新数据类型的功能

新数据类型的功能	描述
CAST 转换功能	CONVERT 转换功能的 ANSI／ISO 同义词
字符数据和二进制字符串	扩展到 8000 个字节
唯一的标识符	全局唯一标识数的数据类型,GUID
nchar、nvarchar 和文本	统一代码数据类型
＋(Add 和 String Concatenation)和 -(Subtract)	数据和时间的加法和减法
Cursor 行为	游标变量

6.5 数据存储增强功能

对数据库上的数据存储进行了某些改变，其中最大的改变是能用 ALTER TABLE 删除一列和 IDENTITY 属性的 NOT FOR REPLICATION 从句。表 6.10 表示受影响的存储语句及其增强功能。

表 6.10 数据存储语句功能的增强

功能项	增强功能的描述
ALTER TABLE	DROP COLUMN 从句，新增了列的非 NULL 缺省项
ALTER VIEW	全局变量
CREATE TABLE	可为空的位列 IDENTITY 属性的 NOT FORREPLICATIONINSERT 不必执行 SET IDENTITY_INSERT ON INSERT 语句，通过一个复制连接保存 IDENTITY 值
用户自定义的索引	最大为 1024 个字节

6.6 新的属性功能

Microsoft SQL Server 7 引入了几个新的属性功能。表 6.11 描述了新的属性功能。

表 6.11 新的属性功能

属性功能	描述
COLUMNPROPERTY	显示列或程序参数的信息
OBJECTPROPERTY	显示数据库中对象的信息
TYPEPROPERTY	显示数据类型的信息
DATABASEPROPERTY	显示数据库的信息
INDEXPROPERTY	显示特殊索引的信息

6.7 新的安全性功能

Microsoft SQL Server 7 中，在增强安全性方面，增加了与角色和新的 Security Identification(SID)数有关的新安全性功能。表 6.12 描述了新的安全性功能。

表 6.12 新的安全性功能

安全性功能	描述
IS_MEMBER	显示当前用户是否为特定的 Windows NT 组或 SQL Server 角色的成员
IS_SRVROLEMEMBER	显示当前注册 ID 或另一个注册 ID（可选）是否为已命名服务器角色的成员
SUSER_SID	显示注册名的安全标识号(SID)

续表

SUSER_SNAME	显示安全标识号中的注册用户名(SID,如未给出 SID 则为当前用户)

6.8 其他新功能

Microsoft SQL Server 7 版本中与新功能相关的其他新特性如表 6.13 所示。

表 6.13 其他新功能

功能	描述
LEN	返回字符串中的字符数
NEWID	产生唯一的标识号(GUID)
REPLACE	用第三个字符串代替第一个字符串中第二个字符串的字符
SUBSTRING	提供文本数据的 SUBSTRING 功能
SYSTEM_USER niladic 功能	显示 Windows NT 域和用户中已注册的用户名(只用于 Windows NT Authentication)
FILE_ID	显示已命名文件的文件标识号
FILE_NAME	显示已命名文件标识号的文件名
FILEGROUP_ID	显示已命名文件组的文件组标识号
FILEGROUP_NAME	显示已命名文件组标识号的文件组名
NCHAR	给出指定字符代码值的 Unicode 字符
UNICODE	给出已命名 Unicode 字符的 Unicode 整数值
PARSENAME	给出一个对象名的命名部分

续表

SET @local_variable	将某个值赋给已命名的局部变量，可以和游标变量一同使用
DECLARE @local_variable	创建游标变量
OPENROWSET	提供 OLE DB 功能以创建分布式查询
QUOTENAME	显示已命名的字符串是有效的引用标识符
VAR	显示统计变化
VARP	显示人口的统计变化
STDEV	显示标准偏差
STDEVP	显示人口的标准偏差
CURSOR_STATUS	显示程序是否返回游标，结果是否为给定的自变量而设置，其中是否有行
GETCHECKSUM	显示校验和的值

6.9 查询的增强功能

Microsoft SQL Server 7 还对查询的处理进行了一些改变，包括新的优化程序提示、查询的并行处理和 SET 语句中的显示计划增强功能等。

表 6.14 描述了查询的增强功能。

表 6.14 查询的增强功能

查询的增强功能	描述
ANSI 视图	显示 SQL Server 元数据的独立于系统表的视图,在改变了系统表后应用程序仍可正常工作
SET ANSI_NULLS	只用于批处理，不能用于存储过程内部
SET SHOWPLAN_ALL	显示将如何执行 Transact-SQL 语句，并估计该语句的资源需求
SET SHOWPLAN_TEXT	SET SHOWPLAN_ALL 的子集
各个查询的并行执行	在多个处理器上并行运行查询
优化程序提示	LOOP、HASH、MERGE、ROBUST PLAN 和 FASTFIRST-ROW 等都是新的提示 READPAST 优化程序提示可用于 SELECT 语句。该优化程序提示用于给优化程序提供信息，以帮助其更好地执行
已计算的列	能在数据库中存储和检索
SELECT	TOP、PERCENT 和 WITH TIES 选项

6.10 新的优化程序提示

应注意某些新优化程序提示，它们可用于 SELECT、INSERT、UPDATE 和 DELETE 语句，并可与 Query Analyzer 进行通信，以便指明如何执行查询。表 6.15 描述了这些新的优化程序提示。

表 6.15 新的优化程序提示

优化程序提示	描述
READPAST	读取过去锁定的行，再跳过它们。这与把@@LOCK_TIMEOUT 设置为 0 是相同的。该提示可用于 SELECT
{HASH|ORDER} GROUP	查询的 GROUP BY 或 COMPUTE 子句中的集合使用散列或排序
{MERGE|HASH|CONCAT}UNION	表示 UNION 应通过合并、散列或连接这些设置来执行。如果命名了多个 UNION 提示，则优化程序将选择最佳策略
FAST	对于前几行的快速检索进行优化查询。发送前几行之后，该查询可产生整个结果集
FORCE ORDER	表示查询的连接排序应在查询优化期间保持不变
ROBUST PLAN	在查询优化期间使用数据资源的上限

6.11 事务处理的增强功能

在 Microsoft SQL Server 的事务领域中也增加了一些新功能。现在 COMMIT 和 ROLLBACK 后可以带用户定义的事务名称，并增加了优化程序提示，详细内容请参阅上一节。表 6.16 描述了各个事务处理的增强功能。

表 6.16 事务处理的增强功能

事务处理的增强功能	描述
COMMIT [WORK]	指明事务的结束，现在可以接受用户定义的事务名
DELETE	优化程序提示增加了：LOOP、HASH、MERGE、ROBUST PLAN 和 FASTFIRST-ROW 是新的提示。优化程序提示可用于 SELECT、INSERT、UPDATE 和 DELETE 语句
INSERT INTO...SELECT	优化程序提示增加了：LOOP、HASH、MERGE 、ROBUST PLAN 和 FASTFIRSTROW 是新的提示。优化程序提示可用于 SELECT、INSERT、UPDATE 和 DELETE 语句
ROLLBACK [WORK]	回卷到用户指定的事务
UPDATE	优化程序提示增加了：LOOP、HASH、MERGE 、ROBUST PLAN 和 FASTFIRSTROW 是新的提示。优化程序提示可用于 SELECT、INSERT、UPDATE 和 DELETE 语句

6.12 系统表的增强功能

Microsoft SQL Server 7 增加了新的系统表以支持其中的新功能，有许多新系统表都与版本 7 的新作业功能相关。表 6.17 描述了新的系统表。

表 6.17 新系统表

系统表	内容
Sysfiles	每个数据库文件都有行
Sysfilegroups	每个数据库文件组都有行
Syscategories	各类别都有行
Sysdownloadlist	作业下载列表都有行
Sysjobhistory	各作业历史都有行
Sysjobs	各作业都有行
Sysjobschedules	各作业调度都有行
Sysjobservers	各作业服务器都有行
Sysjobsteps	各作业步骤都有行
Systargetservergroupmembers	各目标服务器组成员都有行
Systargetservergroups	各目标服务器组都有行
Systargetservers	各目标服务器都有行
Systaskids	各任务都有行
Sysvolumelabel	用于备份的各递增卷标都有行
Sysoledbusers	每个用户和命名链接服务器的密码映射有一行

如前所述，Transact-SQL 在版本 7 中进行了许多修改。在下一章中，将回顾一下 Microsoft SQL Server 7 中这些由旧版本转化来的新增强工具。

第 7 章新工具和增强工具

本章介绍 SQL Server 7 的新工具和改进的工具，其中包括：
SQL Server Query Analyzer(SQL Server 查询分析工具)
Data Transformation Services(数据转换服务工具)
SQL Server Profiler(SQL Server Profiler 工具)
Index Tuning Wizard(索引调节向导)
Replay SQL Server(重新运行 SQL Server)
Database Maintenance Plan Wizard(数据库维护计划向导)
Web Assistant Wizard(Web 辅助向导)
Visual Database Tools(可视数据库工具)
Replication Monitor(复制监视工具)

7.1SQL Server 查询分析工具

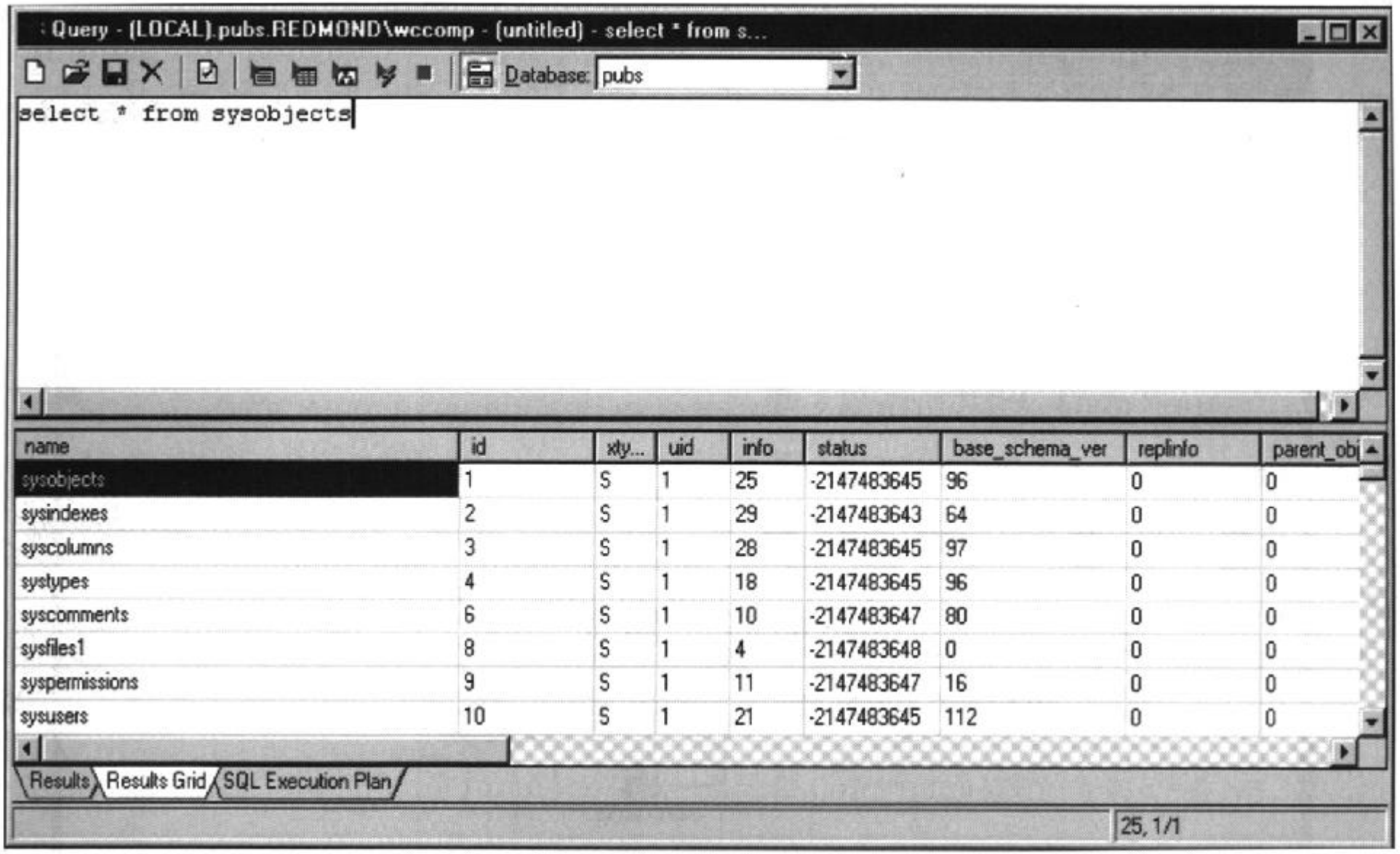
ISQL／w 最明显的变化是现在称为 SQL Server Query Analyzer。这是编写 Transact-SQL 语句的最佳环境之一。现在，在窗口中输入查询时，每个字的颜

色都会变化，就象 C++或 Visual Basic 程序编辑器为不同的语言结构设置不同的代码颜色一样。Query Analyzer 与 SQL Server 的接口使用 ODBC。查询结果根据 Query 菜单中所作的选择以下列三种格式显示：

- 执行(Execute)
- 以网格执行(Execute with Grid)
- 显示执行计划(Display Execution Plan)

Execute 在以前的版本中是输出窗口，也就是说，它可显示列。这种视图的问题是，较宽的列有时难以阅读，必须水平滚动才能看到所有列的结果。

Execute with Grid 选项以易于阅读的栏状格式排列数据，它看上去类似于 Microsoft Excel 电子表格，允许在屏幕上迅速显示更多的数据。要在 Microsoft Management Console 中使用 Execute with Grid 选项，可进入 Enterprise Manager 的 Tools 菜单，选择 SQL Server Query Analyzer | Query | Execute with Grid。图 7.1 表示使用该选项时输出结果的外观。



name	id	xty...	uid	info	status	base_schema_ver	replinfo	parent_obj
sysobjects	1	S	1	25	-2147483645	96	0	0
sysindexes	2	S	1	29	-2147483643	64	0	0
syscolumns	3	S	1	28	-2147483645	97	0	0
systypes	4	S	1	18	-2147483645	96	0	0
syscomments	6	S	1	10	-2147483647	80	0	0
sysfiles1	8	S	1	4	-2147483648	0	0	0
syspermissions	9	S	1	11	-2147483647	16	0	0
sysusers	10	S	1	21	-2147483645	112	0	0

图 7.1Execute with Grid **选项**

Display Execution Plan 查询视图以图表方式显示查询的情况。该选项有一项有趣的功能，即当游标移到查询计划图表中的对象上时，将显示某些信息。该信息包括：

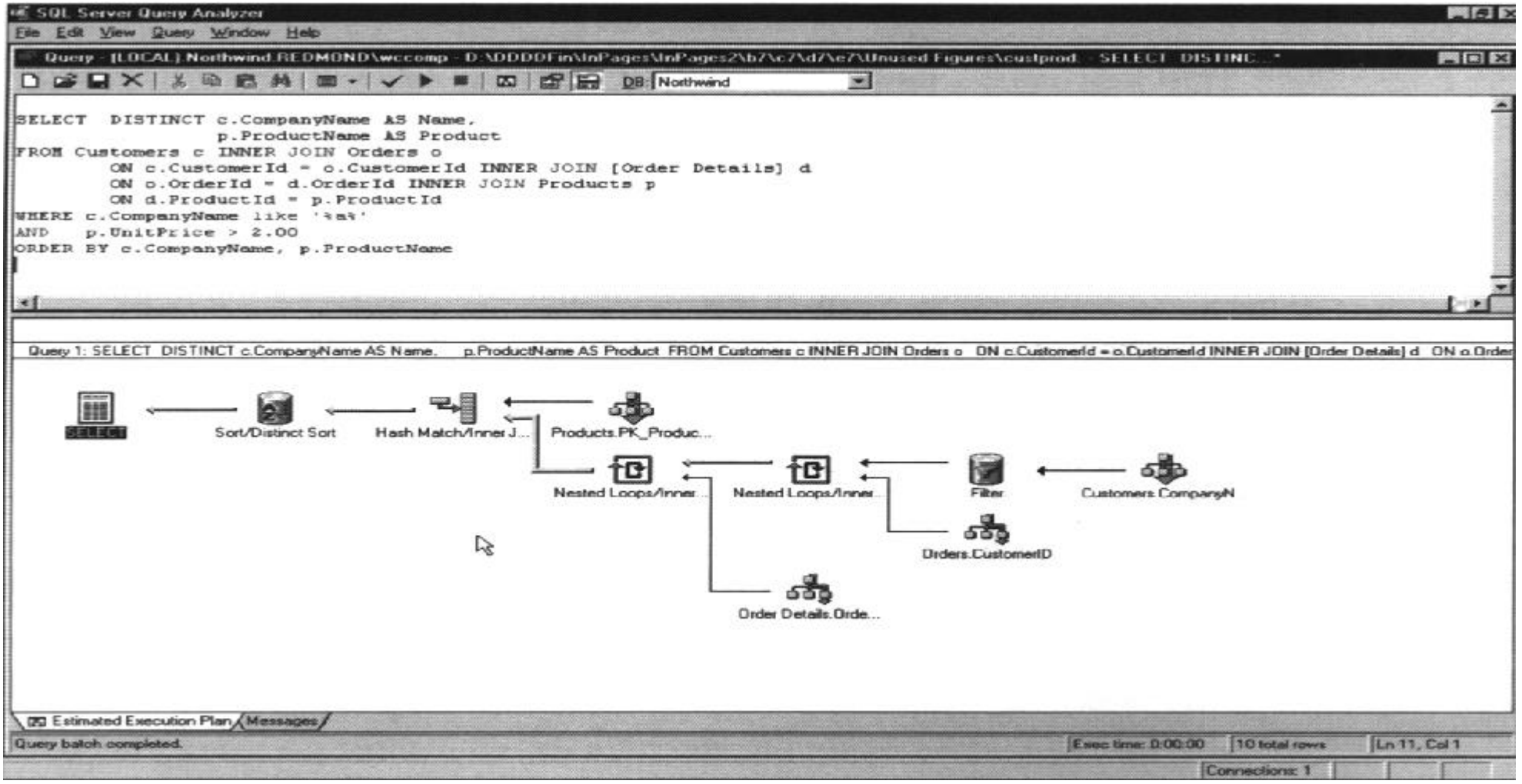


图 7.2Query Analyzer 执行计划

- Type of Scan(扫描类型)
- Physical Location(物理位置)
- Logical Location(逻辑位置)
- Estimated Rows(估计行数)
- Estimated IO(估计 IO)
- Estimated CPU(估计 CPU)
- Average Row Size(平均行宽)
- Total Subtree Cost(占用的总子树数)
- Argument(自变量)

简单查询的执行计划如图 7.2 所示。

7.2 数据转换服务工具

能够把数据从一个环境移到另一个环境在公司中是至关重要的，SQL Server 7 中最重要的新特性就是能象 Microsoft Access 和 Microsoft Excel 那样容易地完成这项工作。Data Transformation Service(DTS)利用 DTS Import Wizard 和 DTS Export Wizard 实现这一功能，它提供的功能甚至比任何 Microsoft 数据输入和输出程序更为强大。

要调用 DTS Import／Export Wizard，可以在 Enterprise Manager 中展开服务器，在展开的服务器下右击 Data Transformation Service 项，并选择 New Package。

(要调用 DTS 向导，可以从 Enterprise Manager 的命令栏中选择 Tools，再选择 Wizards，列出所有的向导，从列表中选择 DTS Wizards。)

DTS Import Wizard 允许选择数据源，如图 7.3 所示。Source 下拉列表中包含下列选项：

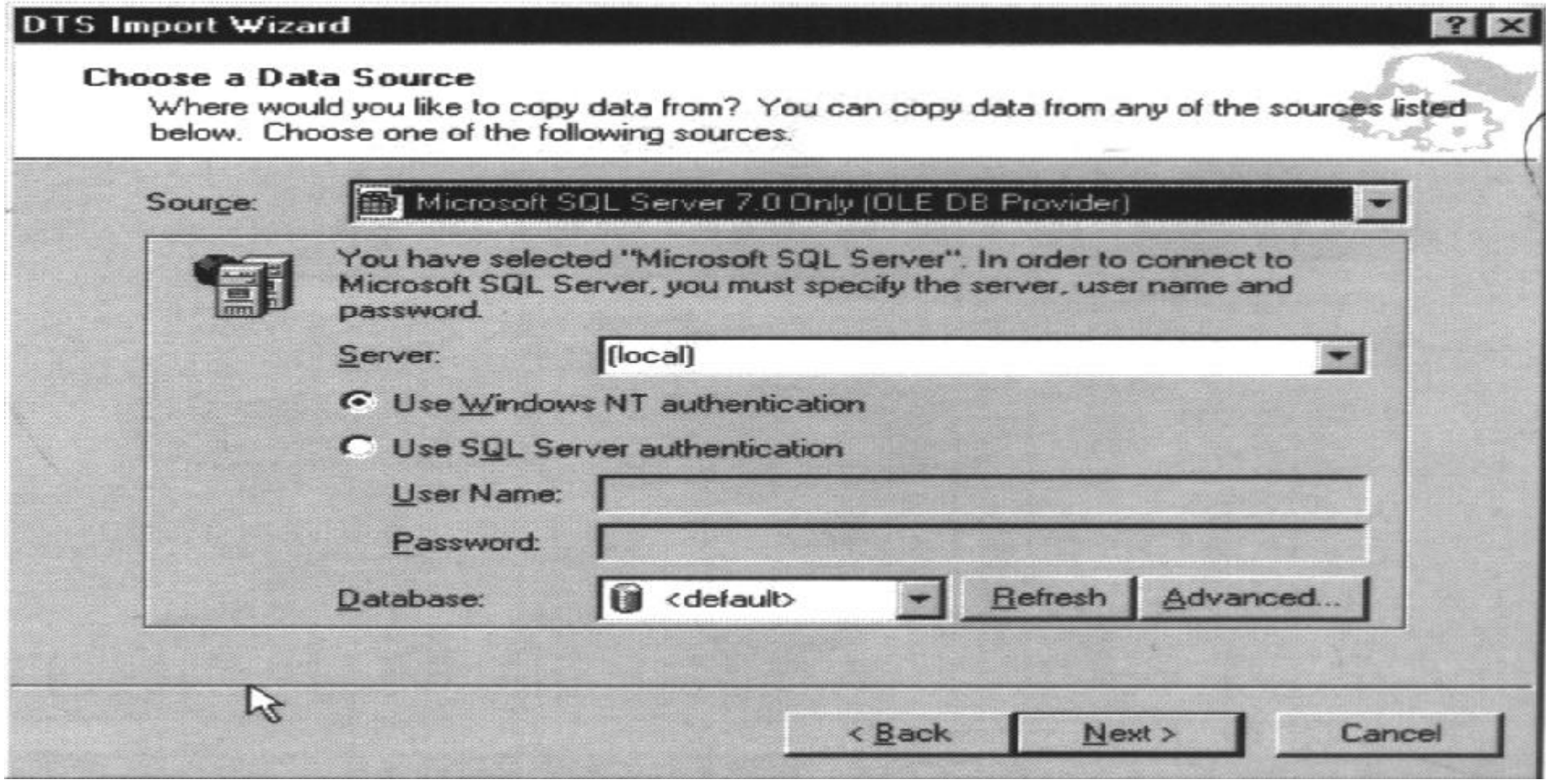


图 7.3 选择数据源

- dBase 5、III 和 IV
- Microsoft Access
- Microsoft Data Link
- Microsoft Excel 3.0、4.0、5.0 和 8.0
- Microsoft Jet OLE DB Provider
- Microsoft OLE DB Provider for Oracle
- Microsoft OLE DB Provider for Plato
- Microsoft OLE DB Simple Provider
- Microsoft Remotes
- Microsoft SQL Server(ODBC Driver)
- Microsoft SQL Server 7.0 only(OLE DB Driver)
- Microsoft Visual Fox Pro Driver
- Oracle (Microsoft Driver)
- Other (ODBC Data Source)
- Paradox 3.x、4.x 和 5.x
- Text File

屏幕区域根据选中的数据源而变化。如果调用了某个数据库，该区域会显示服务器、数据库、注册、密码和／或数据集名。如果调用了文件，则要求输入文件名。

Import Wizard 的下一个屏幕如图 7.4 所示，它要求选择数据的目的地。显示的下拉列表框和 Data Source 屏幕中的列表框一样，也带有同样的选项。

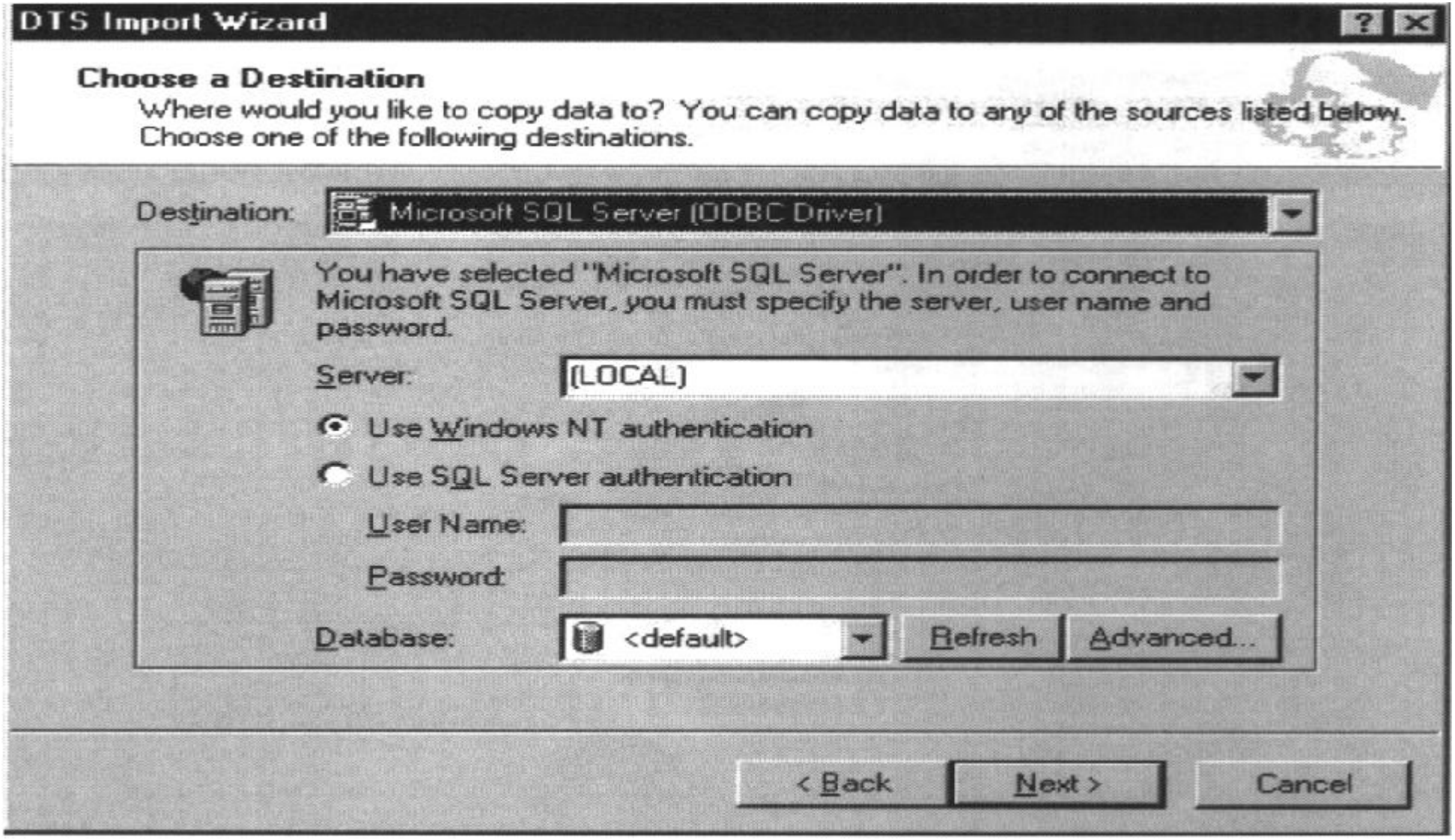


图 7.4 选择数据的目的地

当进入下一个屏幕时，会看到 Source Database 或 Query the Source Database

中的 Copy Table 选项。如果 Query the Source Database 被选中，显示的屏幕上允许输入一个查询。该查询窗口有一个 Query Builder 按钮和一个 Parse 按钮。对于想要输入的数据，可以输入查询以接受来自源数据库的数据。

Select Source Tables 屏幕允许输入目标表名，如图 7.5 所示。

当在 Select Source Table 屏幕上再次按下...按钮，并移到 Data Transformation Service 的内部时，会看到 Column Mapping and Transformation 屏幕。该屏幕有两个标签：Column Mapping 和 Transformation。Column Mapping 面板允许改变列名、数据类型、无效能力和数据输入时的尺寸。Transformations 面板如下图所示，它包含 Active X 脚本，该脚本将把数据转换成输入时所需的格式。

Transformation 面板上的 Advanced 按钮可显示 Transformation Flags 的设置，该设置可确定数据如何从数据源移到目的地，并要求选择可应用到数据输入的 Transformation Flags。

当完成 DTS Import Wizard 中的屏幕时，就创建了所谓的 DTS package，它是定义好的步骤模板。使用 Save and Schedule Package 屏幕可以保存自己的工作。

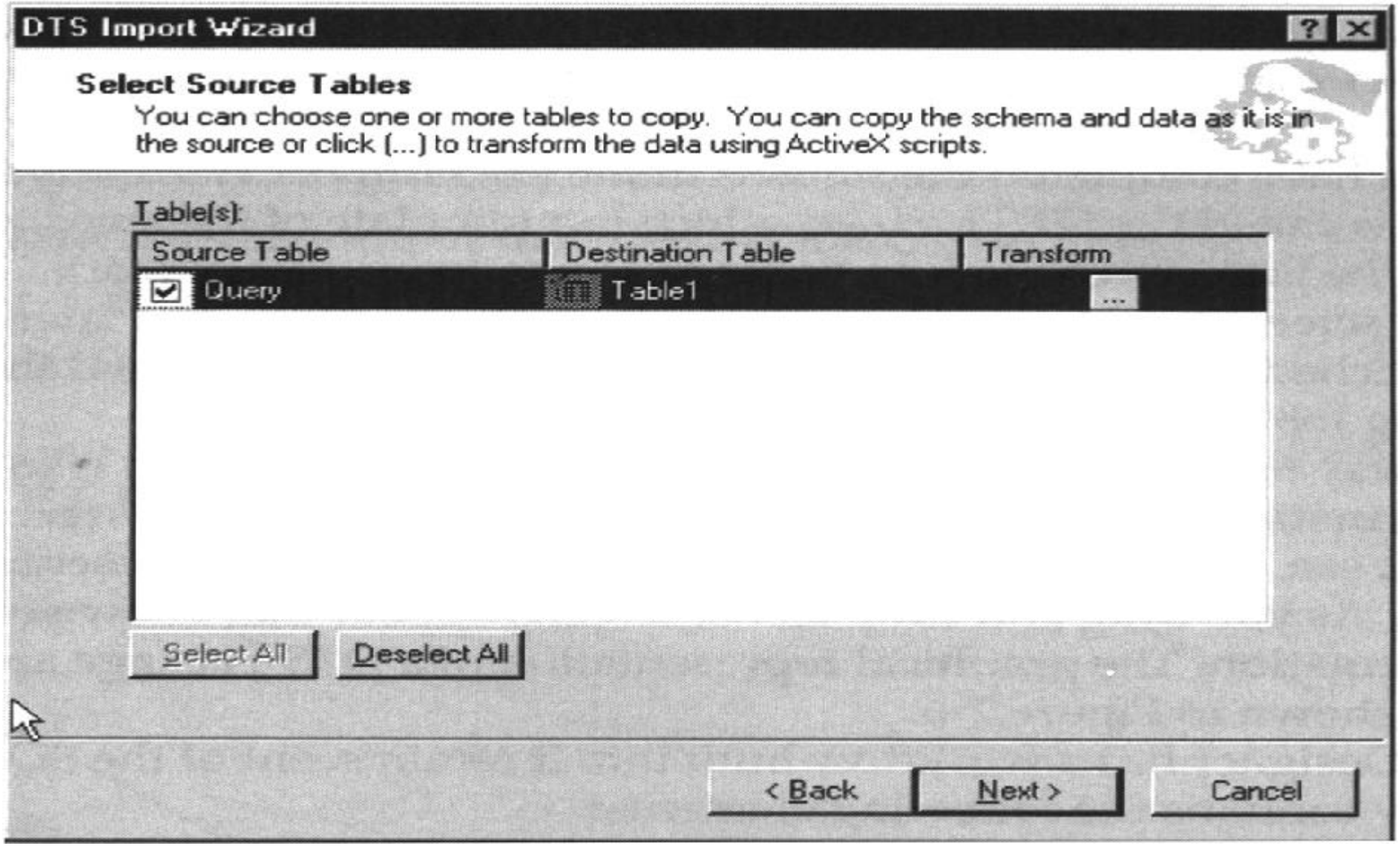


图 7.5 用此屏幕命名目标表

从该屏幕上，通过选中用于以后执行的复选框 Schedule DTS Package，再使

用 Edit Recurring Job Schedule 屏幕，就可调度 DTS 数据包，使其在以后的时间运行。

创建 DTS Package 的另一种方法(使用 DTS Designer)是右击 Enterprise Manager 层次树中的 Data Transformation Services。使用 DTS Designer 可以拖放图标，创建 DTS Package 的图形表示。拖放图标时，显示的屏幕上允许输入需要的信息。DTS Package 的图形表示和 DTS Designer 如图 7.6 所示。

DTS Designer 有一个内置的工作流，让人回忆起 SQL Server Agent。该工作流包含下列元素：

Connection(连接, 创建 Open Connection)

Task(任务, Operations 集)

Precedence Constraint(优先约束, Steps with On Success On Failure)

Package(程序包, Collection of Tasks)

Data Transformation(数据转换, Movement of Data)

建立数据包的同时用这些元素创建工作流，这就是 DTS Designer 的目的，用下列方法可实现这一目的：

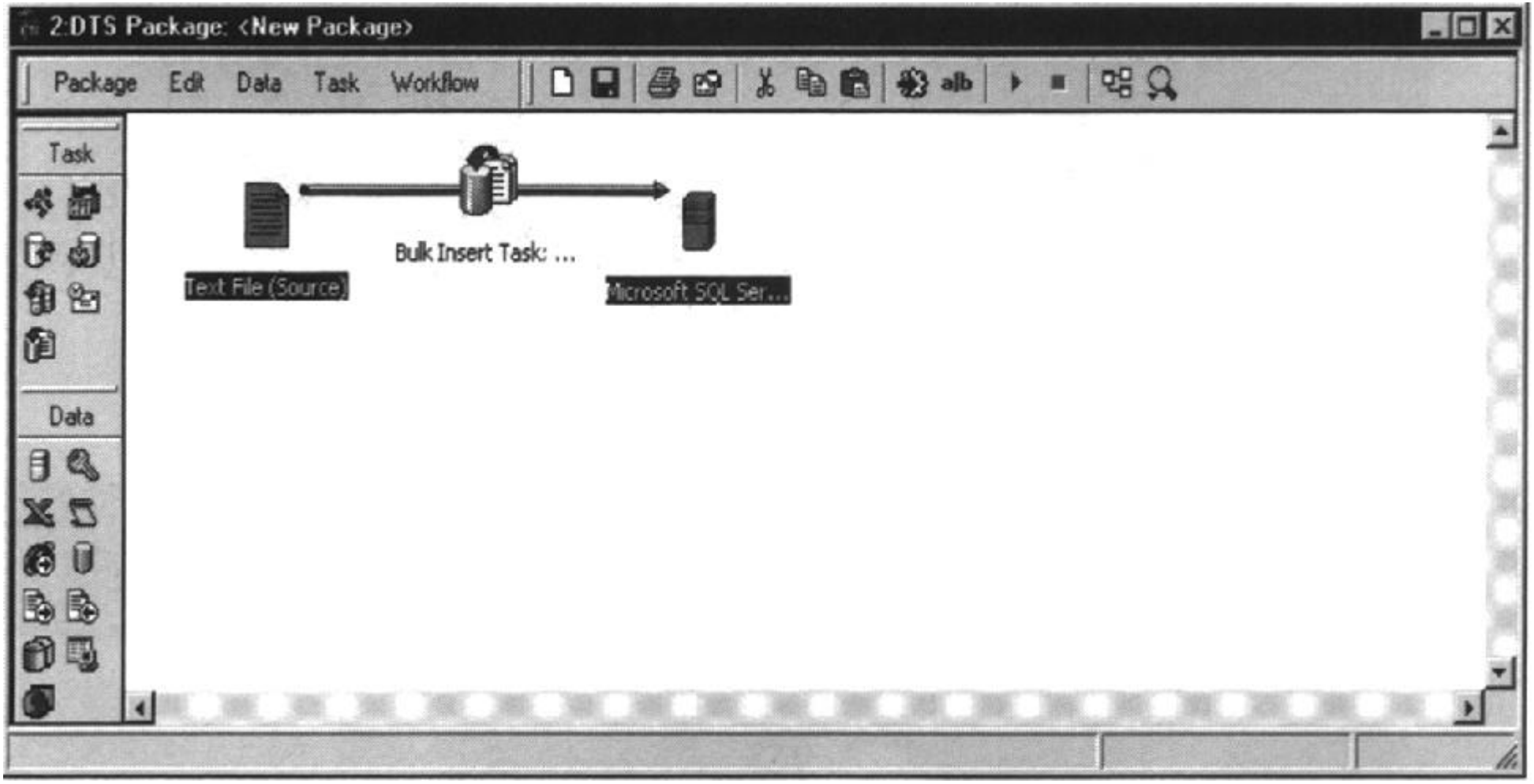


图 7.6DTS Designer **中的** DTS Package:< New Package>

1.**在** DTS Designer **中创建连接**

2.**输入连接信息**

3.**在源连接和目标连接之间创建任务**

4.增加定制任务或 Microsoft ActiveX 脚本

5.使用程序约束

使用 OLE DB ODBC 供应商，通过 SQL Server Data Transformation Services 可访问的 ODBC 数据源是：

ODBC Data Source

Oracle

DB276

Microsoft Access

Microsoft Word

DTS 可实现快速 BCP，并能创建 Excel 主元表。DTS 能使用 COM 对象提供数据的附加转换，如有必要，可用于创建数据仓库。Data Transformation Services(DTS) 不会在不同的关系数据库或源之间创建诸如存储过程、视图或触发程序这类对象。

7.3SQL Server Profiler

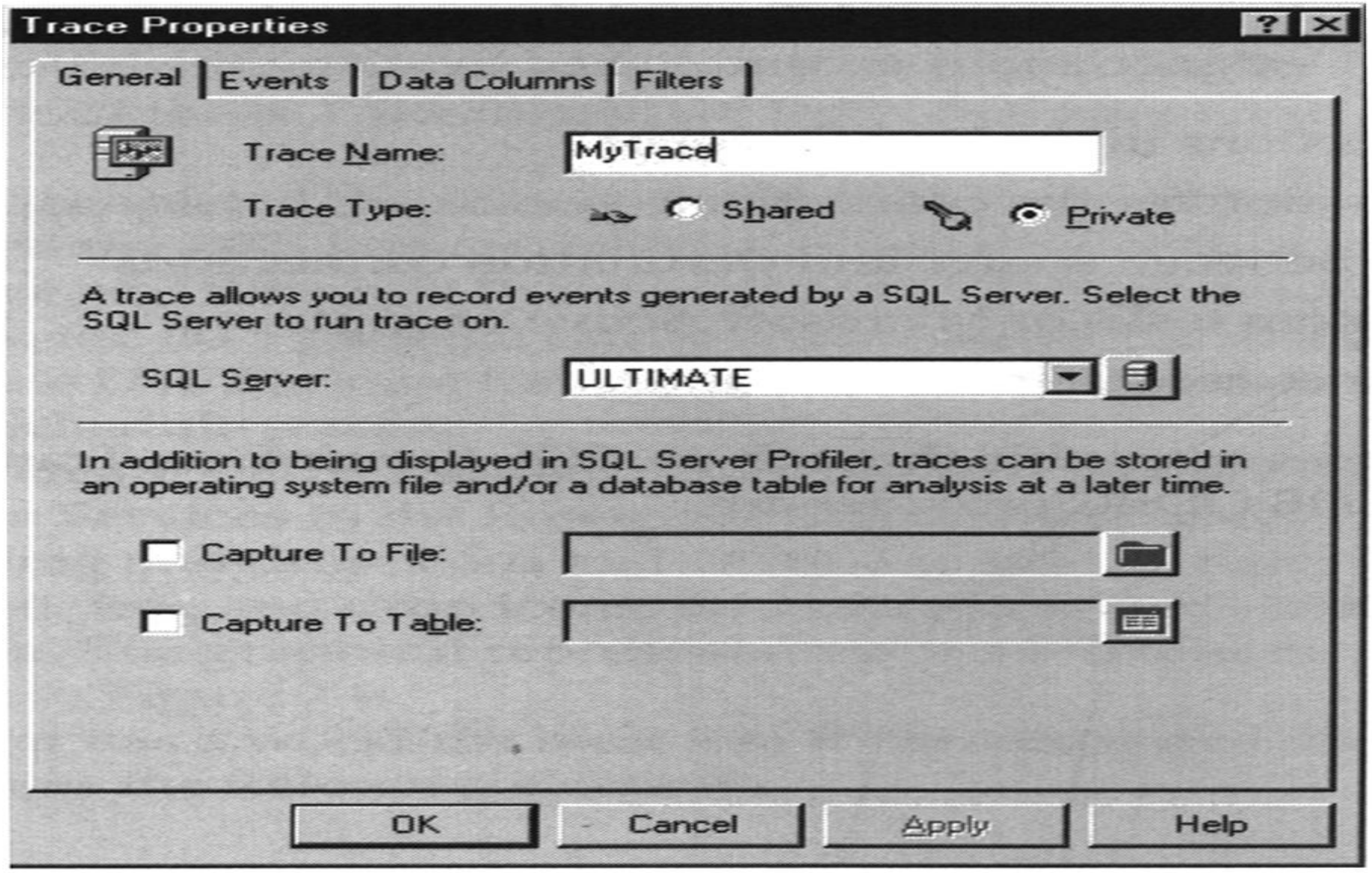
SQL Server Profiler 是一个数据库工具，它可监视 Microsoft SQL Server 上的事件，并代替以前称作 SQL Trace 的工具。用户可以创建跟踪以搜集关于诸如 Transact-SQL 语句和连接这类事件的数据。它是一种调试工具，目的是找出有问题的查询，监视 SQL Server 的性能，以及调试应用程序。SQL Serer Profiler

的详细信息请参阅第 17 章。

注意:在 NT Start 按钮下面的 Microsoft SQL Server 菜单上选择 SQL Server Profiler，或从 Enterprise Manager 的 Tools 菜单中选择 SQL Server Profiler 就可以调用 SQL Server Profiler。

7.3.1 跟踪输出文件

要把跟踪输出保存为文件，可使用 Trace Properties 对话框的 General 标签下的选项，如下图所示：



该文件称作跟踪输出文件(trace output file)，它有以下几种重要的用途：

- 捕捉 SQL Server 错误的解析事件
- 调试扩展存储过程
- 建立一段时间内服务器事件的量度
- 辅助创建 Index Tuning Wizard 的工作量文件
- 在 SQL Server Profiler 中创建输入，以便重新执行 SQL Server 功能

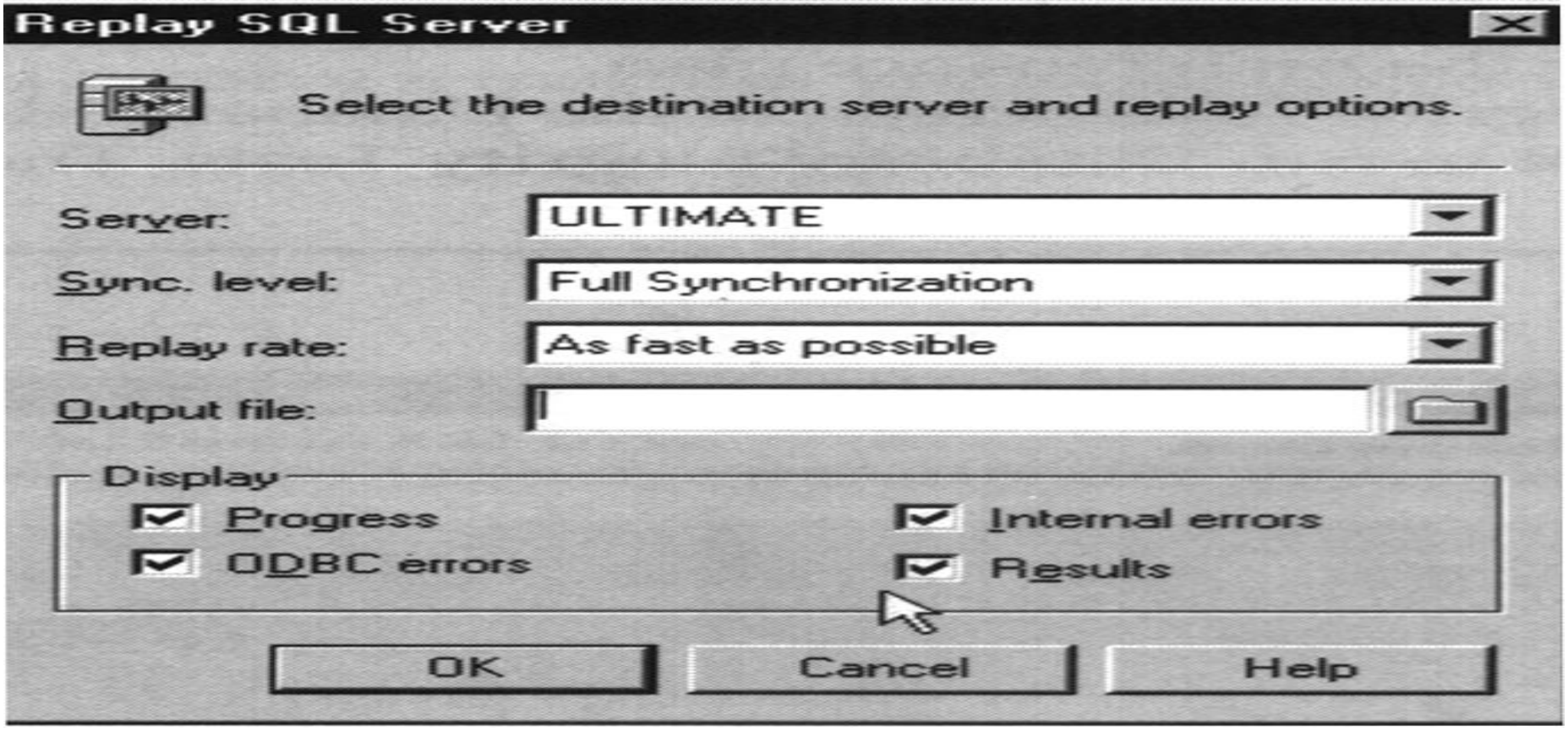
7.3.2Index Tuning Wizard(索引调整向导)

Index Tuning Wizard 的目的是自动为整个数据库创建索引。Index Tuning Wizard 可分析生成 what if 索引的概要事件，并推荐工作量的最佳索引混合。用户可不必填全带有 what if 方案的索引，可在当时生成索引，也可以安排在以后创建索引,并按周或按月自动调整数据库。如果愿意,可以使 Microsoft SQL Server 一早给 DBA 发送 e-mail，推荐如何重新生成该索引以提高工作量的性能。

从 Enterprise Manager 中的 Tools 菜单或从 SQL Server Profile 中的 Tools 菜单选择 Query Analyzer，就可以在 Perform Index Analysis 选择集内部调用 Index Tuning Wizard，Index Tuning Wizard 也可以推荐索引的创建过程来优化数据库。它是通过分析工作量 (Transact-SQL 脚本或来自运行 Microsoft SQL Server Profiler 的跟踪文件)，使用 SHOWPLAN 成本公式选择最佳索引，并推荐更高效的索引混合，来实现这一点的。CREATE INDEX 语句可由推荐自动生成，并可当时执行，或安排在以后执行，该语句将为工作量删除和创建更有效的索引。

7.3.3 重新运行 SQL Server

SQL Server Profiler 的 Replay SQL Server 能以实时模式或压缩模式重新运行数据文件的跟踪捕捉。要使用 SQL Server Profiler 重新运行数据的捕捉跟踪，可打开某个跟踪文件，选择 Replay 菜单，然后再选择 Settings，以查看下列 Replay 屏幕：



Replay SQL Server 可以读取下列由 SQL Server Profiler 保存的文件类型：

SQL Server Profiler 文件类型	描述
filename.sba	SQL Server Profiler BCP ANSI
filename.scu	SQL Server Profiler Compact Unicode
filename.sbu	SQL Server Profiler BCP Unicode
filename.srw	SQL Server Profiler_70_raw

SQL Server Profiler_70_raw 是占据大多数磁盘资源的原始事件数据，并且读取速度最快。

7.4 数据库维护计划向导

Database Maintenance Plan Wizard 的目标是创建可安排在 SQL Server Agent 下运行的维护计划。要使用 Database Maintenance Plan Wizard，可展开 Enterprise Manager 中的服务器，右击已展开服务器下的 Database Maintenance Plan 项，并选择 New Plan。使用该维护计划可以进行数据库的一致性检查、备份数据库和事务记录、更新统计以及重建索引。上述工作就像创建一个维护计划，并安排它在休息时间运行一样简单。

Database Maintenance Plan Wizard 的 Update Data Optimization Information 屏幕中的复选框允许重新组织数据和索引，并更新统计。此外，该屏幕上的调度信息将会传给 SQL Server Agent，以安排维护计划的执行时间。

在 Database Maintenance Plan Wizard 的 Run Data Integrity Test 屏幕上(如图 7.7

所示),可以选择运行 Database Integrity Test。这些操作也可以用 SQL Server Agent 在该屏幕上调度。在备份数据库或事务记录之前选中相应的复选框，就可以进行这些测试。

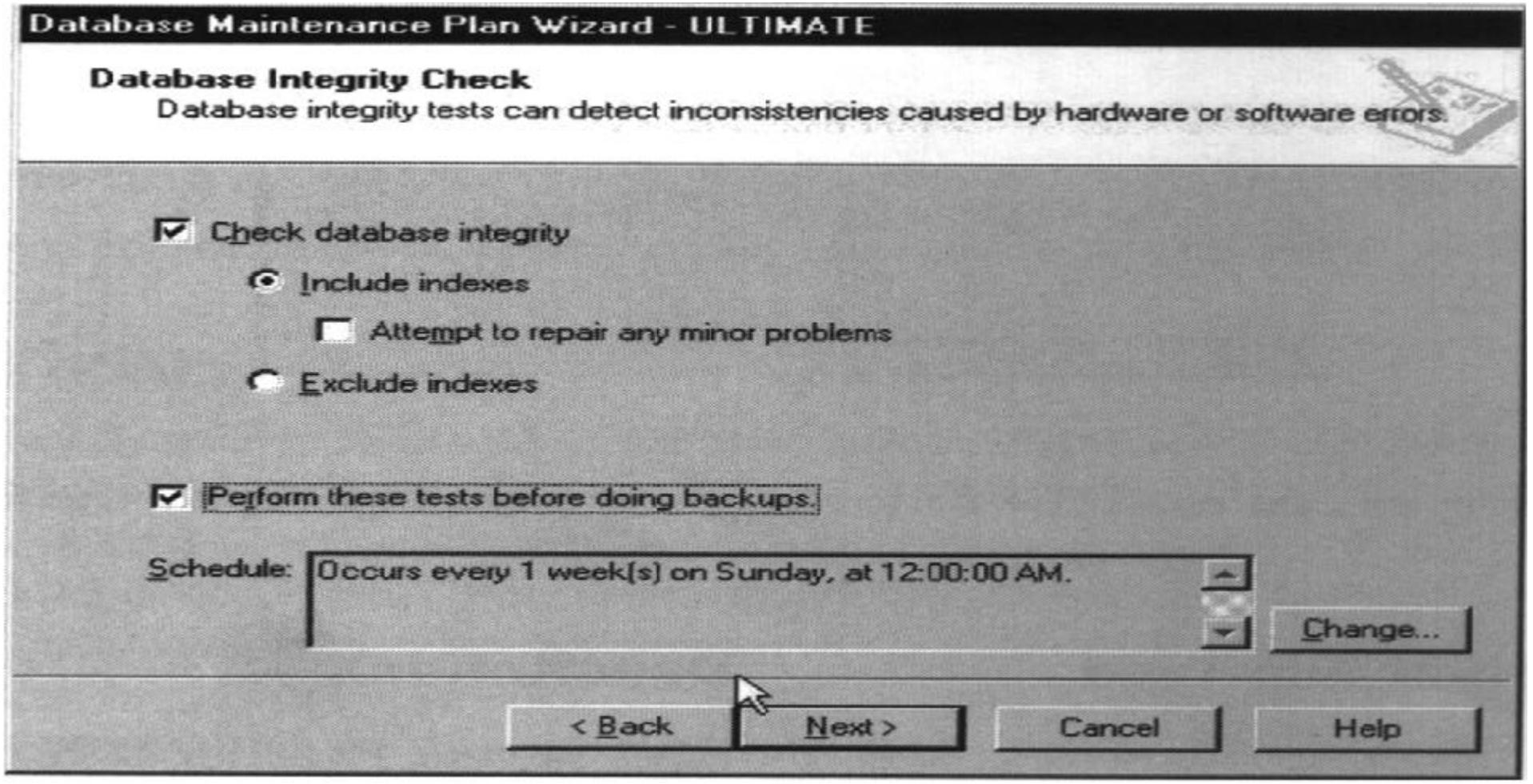


图 7.7Database Maintenance Plan Wizard 的 Run Data Integrity Test 屏幕

利用 Specify the Database Backup Plan 屏幕(如图 7.8 所示)，可以设置备份数据库的方式和时间。选择相应的复选框，就可以在完成时验证备份的完整性。所用的介质可以是磁带或磁盘，并有注释说明磁盘备份文件名是自动创建的。

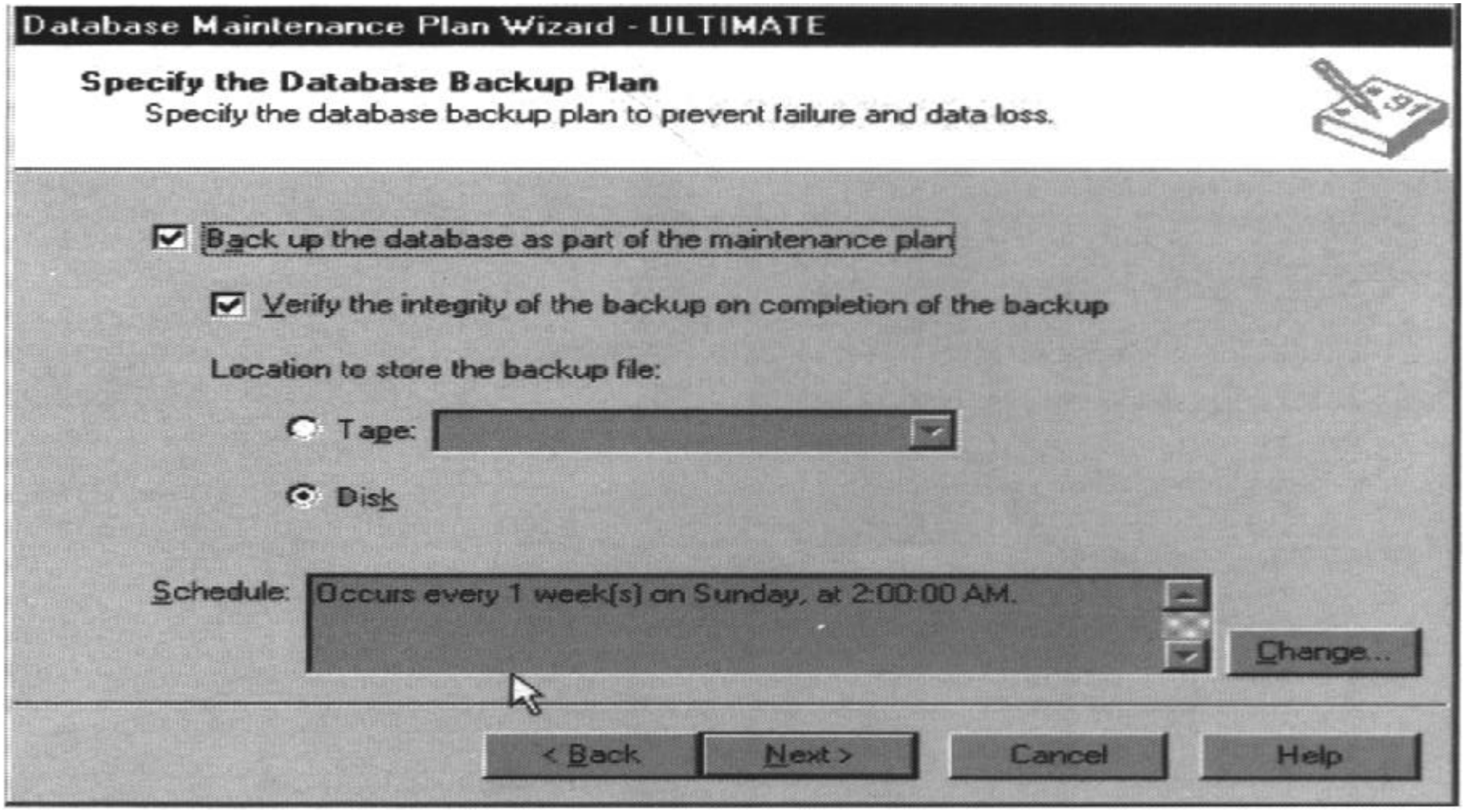


图 7.8 使用 Database Maintenance Plan Wizard 指定数据库备份计划

利用 Specify the Transaction Log Backup Plan 屏幕，可以设置备份事务记录的方式和时间。另外，选择相应的复选框，就可以在完成时验证备份的完整性，所用的介质可以是磁带或磁盘，并有注释说明磁盘备份文件名是自动创建的。

利用 Reports to Generate 屏幕，可以将报告向发送给 Web 页、文件或 Database Administra-tor(以前在 SQL Server Agent 中设置的一个操作员)。

Database Maintenance Plan Wizard 的最后一个屏幕以段落格式显示用户创建的维护计划。当所有一切都创建成功时，会收到计划已完成的消息。

7.5Web 辅助向导

Web Assistant Wizard 是一个方便的数据库工具，它提供了一种建立 Web 页显示查询结果的简便方法。要调用该工具，可展开 Enterprise Manager 中的服务器，右击 Web Assistant Jobs 项，并选择 New Web Assistant Job。Web Assistant Wizard 的第一页展示了该工具的功能。

Web Assistant Wizard 的下一个屏幕要求选择数据库。在此可打开一个窗口，输入 Transact-SQL Query，为 Web 页从所要的表格中选择数据。Schedule the Web Assistant 屏幕可以把更新 HTML Web 页的作业安排在以下时间执行：向导完成后执行，根据用户的要求执行，在某一日期或时间执行，SQL Server 的数据发生改变时执行，或在定期安排的时间间隔执行。

Publish Web Page 屏幕上包含发布 Web 页的路径的位置。该位置可以是物理

目录、网络目录或 FTP 路径。该屏幕进一步说明了，MSSQLServer 启动帐户必须能访问该位置。

Format Web Page 页询问用户是想要 SQL Server 帮助格式化 Web 页，还是想选择模板文件，如图 7.9 所示。该模板文件定义了 HTML 文档的格式，引用 <%insert_data_here%>指定了放置查询结果的位置。在该模板文件中，也可以指定如何使用<%begindetail%>和<%enddetail%>格式化行，包括<TR>、<／TR>、<TD>和<／TD>HTML 标记。对于结果集里的每一列，必须插入<%insert_data_here%>标记。

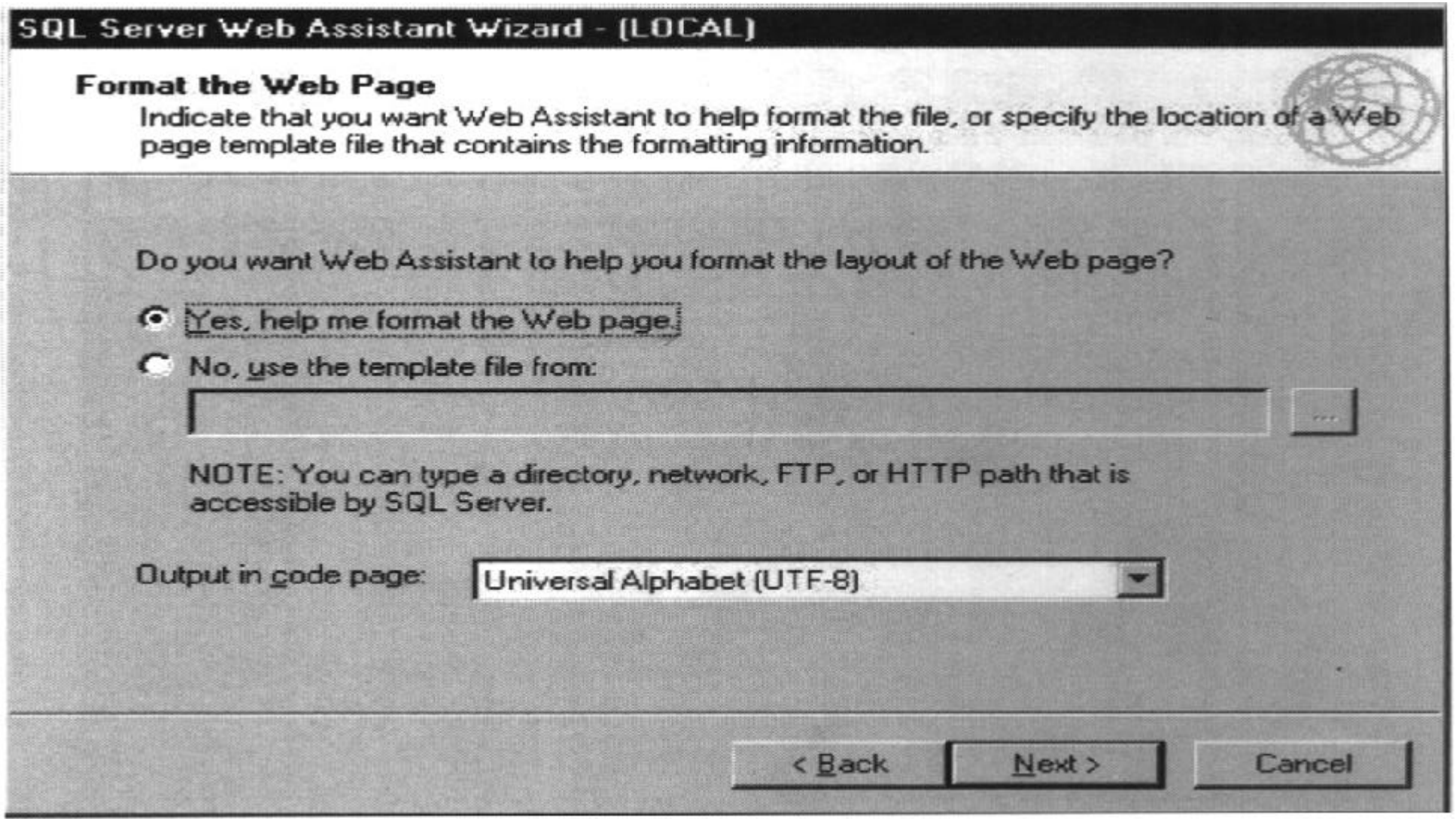


图 7.9 格式化 Web 页

下一个屏幕是 Specify Titles，其中的输入域可接受 Web 页的标题和带有数据的 HTML 表格的标题，也可以指定 HTML 表格标题字体的尺寸。

用 SQL Server Web Assistant 格式化表格时，会询问用户是否要在 HTML 表格中显示列名，以及应用于表格数据的字体具有什么属性：固定、成比例、粗体或斜体。用户也可以选中复选框以在表格周围画上边框。

Add Hyperlinks 屏幕如图 7.10 所示，该屏幕询问是增加一个超级链接、没有超级链接，还是从 SQL Server 表格（以前存放超级链接的位置）中选择超级链接列表。

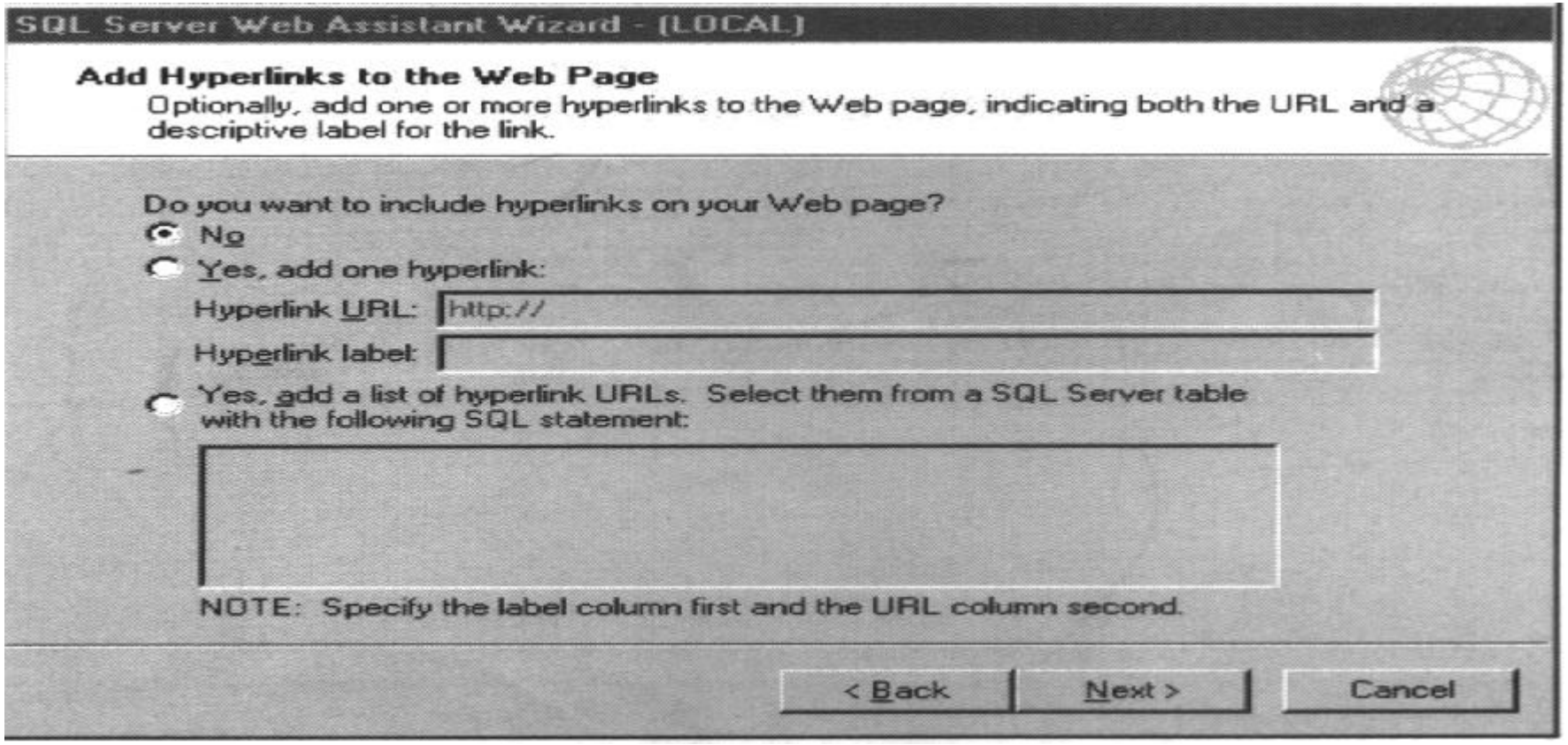


图 7.10 为 Web 页增加超级链接

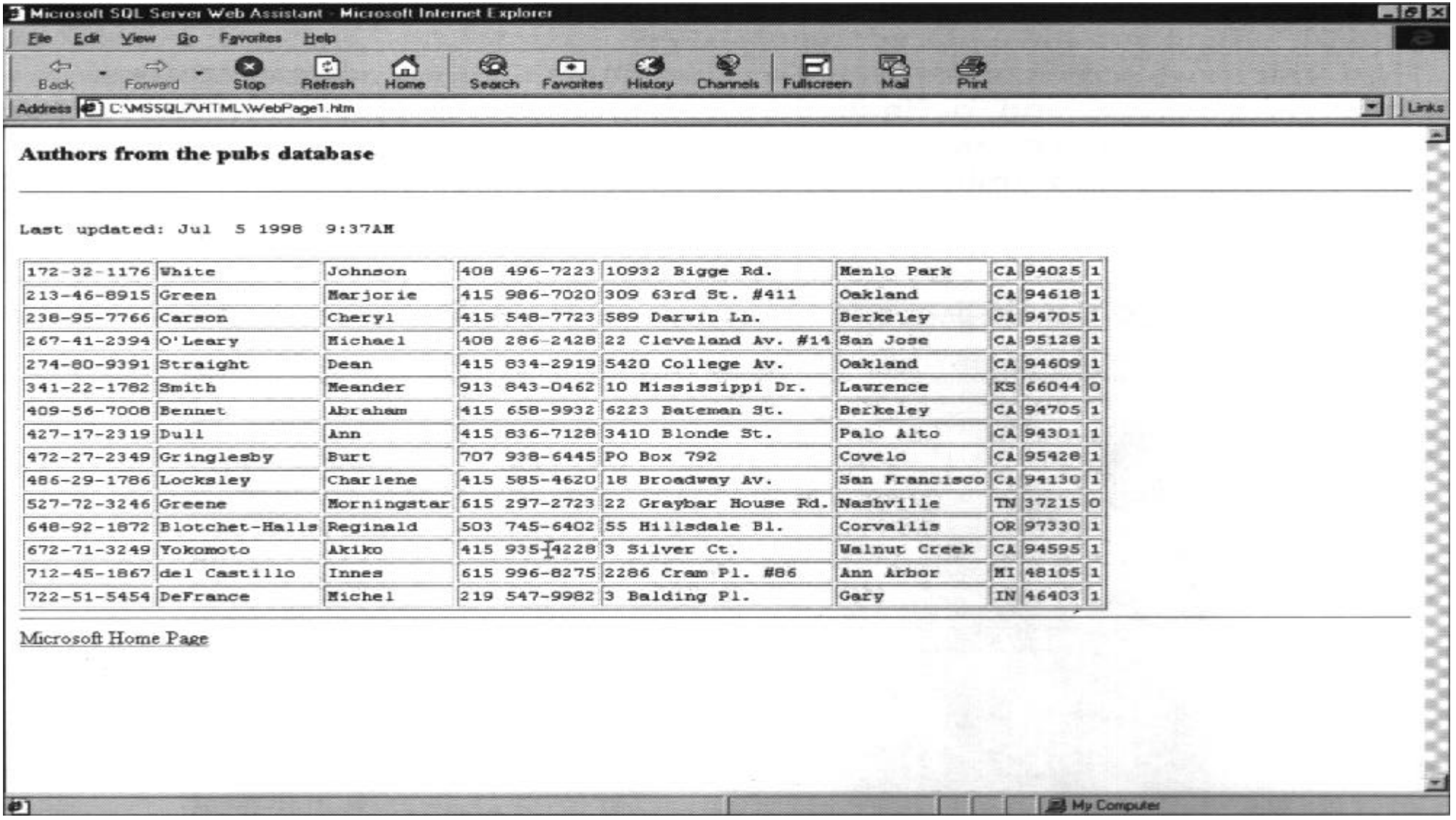


图 7.11 用 Web Assistant Wizard 创建的 Web 页

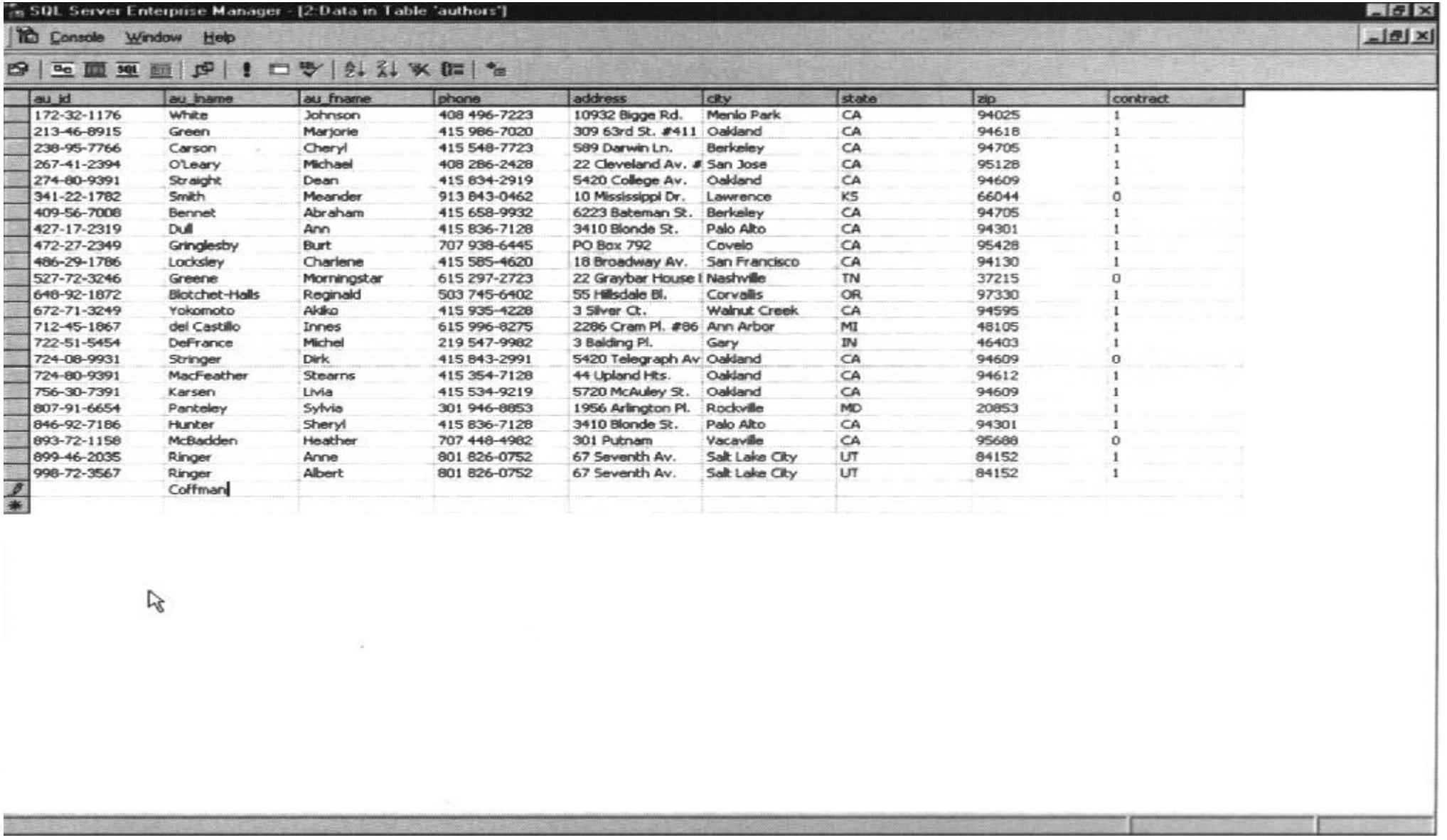
Web Assistant 的 Limit Rows 屏幕限制 SQL Server 返回的总行数，和／或限

制 Web 页上显示的行数。如果选择了限制 SQL Server 返回的行数，将返回用户指定行数的前几行。通过指定每一页显示多少行，还可以调节 Web 页上的行的位置，把所有的行都放到一个滚动页上，或把连续的页链接到一起。

Complete the Microsoft SQL Server Web Assistant Wizard 屏幕将目前所做的工作以段落方式显示出来。最后，Web Assistant 通知用户已经成功地完成了 Web 页。现在，使用 NT Explorer，并双击.htm 文件可进入新创建的 Web 页所在的目录。图 7.11 所示为带有 Web 页的 Internet Explorer，就好象它是用 Web Assistant 创建的。

7.6 可视数据库工具(Database Tools)

Visual Database Tools 已完全集成到 Microsoft Management Console 的 Enterprise Manager 中，可以在 Enterprise manager 在各个位置处调用。例如，如果用 Enterprise Manager 展开服务器、展开数据库、选择 Tables 并选择某个表格，则可以右击该表格，选择 Open Table，查看和更新表格中的数据，如图 7.12 所示。这在以前的 Enterprise Manager 中是绝对不可能的。用户还可以从同一个菜单中选择 Design table，用 Visual Database Tool 设计表格。



au_id	au_lname	au_fname	phone	address	city	state	zip	contract
172-32-1176	White	Johnson	408 496-7223	10932 Bigge Rd.	Menlo Park	CA	94025	1
213-46-8915	Green	Marjorie	415 986-7020	309 63rd St. #411	Oakland	CA	94618	1
238-95-7766	Carson	Cheryl	415 548-7723	589 Darwin Ln.	Berkeley	CA	94705	1
267-41-2394	O'Leary	Michael	408 286-2428	22 Cleveland Av. #	San Jose	CA	95128	1
274-80-9391	Straight	Dean	415 834-2919	5420 College Av.	Oakland	CA	94609	1
341-22-1782	Smith	Meander	913 843-0462	10 Mississippi Dr.	Lawrence	KS	66044	0
409-56-7008	Bennet	Abraham	415 658-9932	6223 Bateman St.	Berkeley	CA	94705	1
427-17-2319	Dull	Ann	415 836-7128	3410 Blonde St.	Palo Alto	CA	94301	1
472-27-2349	Gringlesby	Burt	707 938-6445	PO Box 792	Covelo	CA	95428	1
486-29-1786	Locksley	Charlene	415 585-4620	18 Broadway Av.	San Francisco	CA	94130	1
527-72-3246	Greene	Morningstar	615 297-2723	22 Graybar House I	Nashville	TN	37215	0
648-92-1872	Blotchet-Halls	Reginald	503 745-6402	55 Hillsdale Bl.	Corvallis	OR	97330	1
672-71-3249	Yokomoto	Akiko	415 935-4228	3 Silver Ct.	Walnut Creek	CA	94595	1
712-45-1867	del Castillo	Innes	615 996-8275	2286 Cram Pl. #86	Ann Arbor	MI	48105	1
722-51-5454	DeFrance	Michel	219 547-9982	3 Balding Pl.	Gary	IN	46403	1
724-08-9931	Stringer	Dirk	415 843-2991	5420 Telegraph Av	Oakland	CA	94609	0
724-80-9391	MacFeather	Stearns	415 354-7128	44 Upland Hts.	Oakland	CA	94612	1
756-30-7391	Karsen	Livia	415 534-9219	5720 McAuley St.	Oakland	CA	94609	1
807-91-6654	Panteley	Sylvia	301 946-8853	1956 Arlington Pl.	Rockville	MD	20853	1
846-92-7186	Hunter	Sheryl	415 836-7128	3410 Blonde St.	Palo Alto	CA	94301	1
893-72-1158	McBadden	Heather	707 448-4982	301 Putnam	Vacaville	CA	95688	0
899-46-2035	Ringer	Anne	801 826-0752	67 Seventh Av.	Salt Lake City	UT	84152	1
998-72-3567	Ringer	Albert	801 826-0752	67 Seventh Av.	Salt Lake City	UT	84152	1
	Coffman							

图 7.12 在表中使用 Enterprise Manager 更新一行

从 Enterprise Manager 内部调用 Visual Database Tools 功能的另一种方法是展开服务器、展开数据库，再选择 Database Diagram。用户会看到一个已有的数据库图表，并可以通过定义数据库中表格之间的关系来创建新的数据库图表。用 Create Database Diagram Wizard 创建 pubs 数据库的 Database Diagram 如图 7.13 所示。

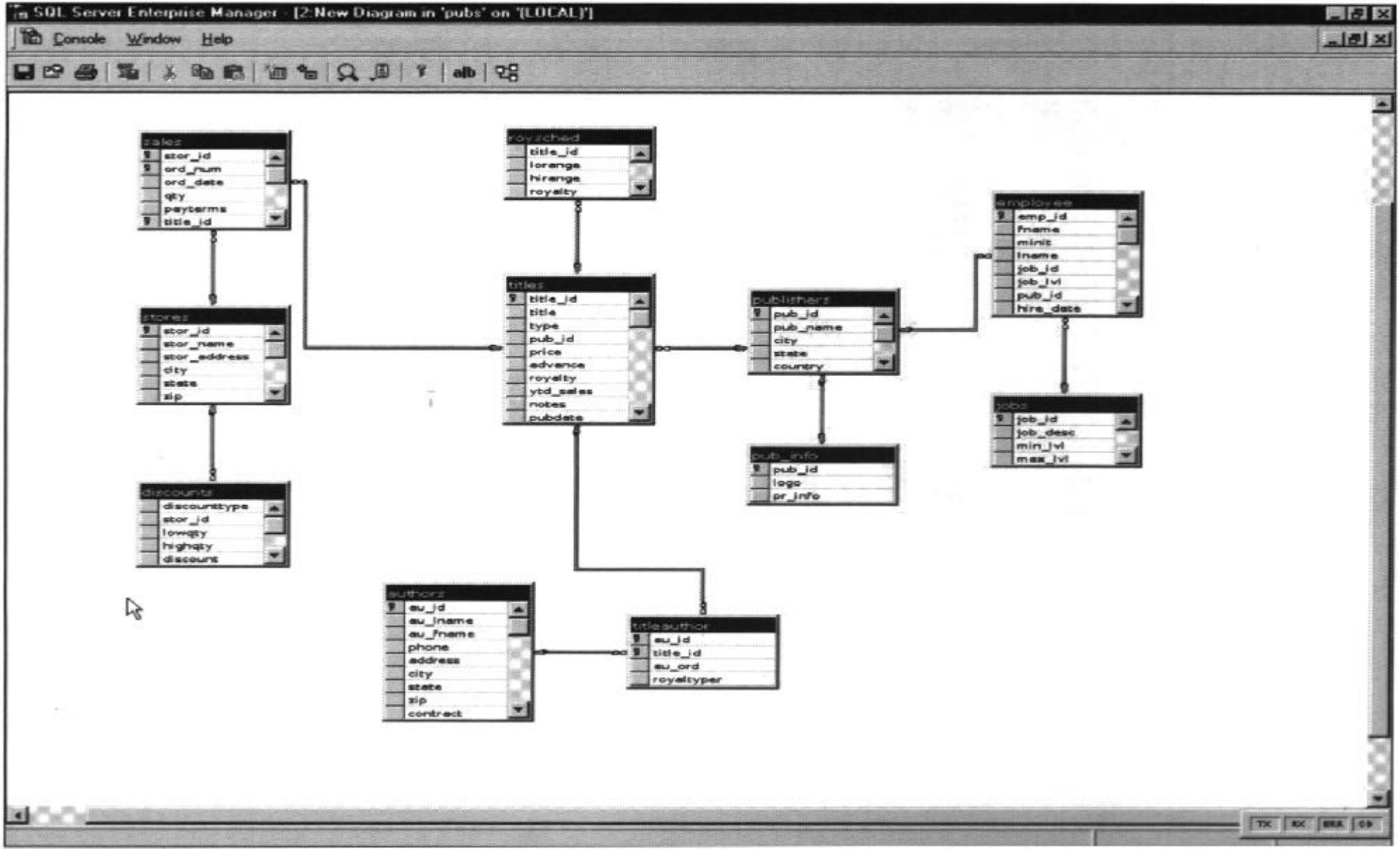


图 7.13 通过 Enterprise Manager 中的 Create Database Diagram Wizard 创建的

Visual Database Tools 也可以用于 Microsoft Access。该工具有下列功能：

- 创建数据库图解图表
- 创建表格
- 建立查询
- 连接到任意 ODBC 兼容的数据库
- 创建复杂的查询
- 增加、更新和删除数据库中的数据
- 创建 SQL 对象，如数据库(从图表创建)、表格、存储过程，并触发 Microsoft SQL Server 和 Oracle 数据库 Microsoft Visual Database Tools 的四个部分分别称作：
  - Data View(数据视图)
  - Database Designer(数据库设计器)
  - Query Designer(查询设计工具)
  - Source Code Editor(源代码编辑器)

7.6.1 数据视图(Data View)

Data View 可显示以下内容：

- Live connections to the databases(与数据库的现场连接)：连接到 SQL Server 的 SQL Server Login ID
- Database diagrams(数据库图表)：显示数据库中表格之间关系的描述

- Tables(表格)：组成用于存储数据的数据库对象的行和列
- Triggers(触发程序)：在对表格中的行进行 UPDATE、INSERT 或 DELETE 时自动运行的存储过程
- Views(视图)：由数据库服务器上存储和维护的查询组成的虚拟表格
- Stored procedures(存储过程)：用 Transact-SQL 编写的程序，可包含程序流、逻辑和对数据库的查询

7.6.2 数据库设计器(Database Designer)

Database Designer 可创建和修改以下内容：

- Microsoft SQL Server 数据库的结构
- 数据库对象，如表格、关系、索引和约束
- Data Definition Language 命令
- 用于改变数据库图表的 Transact-SQL 代码的脚本

7.6.3 查询设计工具(Query Designer)

Query Designer 可建立和执行对任意 ODBC 兼容的数据库进行的查询，用它能够进行以下工作：

- 使用可视查询图表

- 使用标准网格指定搜索条件、分类顺序、输出列
- 生成 SQL
- 编辑表格中的数据

7.6.4Source Code Editor(源代码编辑器)

Source Code Editor 是一个工具，用它可以：

- 编辑存储过程和触发程序
- 执行存储过程

7.7Replication Monitor（复制监视工具）

Enterprise Manager 的 Replication Monitor 特性允许查看 Replication Agents 的状态。为了查看 Enterprise Manager 中的 Replication Monitor，必须通过执行下列步骤，使服务器能够成为 Distributor：

1. 用 SA 特权注册
2. 使用 SQL Server Enterprise Manager
3. 展开服务器
4. 选择 Tools | Replication
5. 使用 Configure Publishing and Distribution Wizard

注意:要调用 Configure Publishing and Distribution Wizard，可展开 Enterprise Manager 中的服务器，选择 Tools 菜单，并选择 Replication。

用户既可以把 SQL Server 配置成 publisher and distributor，也可以把它配置成 dedicated distributor。在 Configure Publishing and Distribution Wizard 中，可使用 Configure with Defaults 屏幕，用缺省值进行配置，或定制分布式数据库名、分布式数据库文件的位置和分布式数据库事务日志文件。

如果选择定制发布和分布式配置，可选择定制设置的选项，此时看到的屏幕上支持发布程序、供发布的数据库和订阅者。在使 SQL Server 成为分发程序后，可以使用 Replication Monitor 查看 Publishers、Publications 和 Subscribers 的列表；查看安排好的 Replication Agents；监视每个代理如何运作；查看 Replication Agents 的历史等。甚至还可以设置复制事件的警告。在本书后面关于 Replication(复制)一章将详细讨论复制。

在下一章，将介绍 SQL Server 7 实用程序的变化。与本章讨论的 GUI(Graphical User Interface)工具相反，实用程序是命令行工具。

第 8 章实用程序的变化

Microsoft SQL Server 实用程序的主要目的是增强功能、性能和可靠性。随着对太字节(1000,000,000,000 字节)数据库的支持，数据库实用程序还必须能够处理大量的数据，Microsoft SQL Server 实用程序已在这些方面进行了改进，特别是提高执行速度的问题。

本章介绍了下列实用程序，主要论述每个实用程序的新功能：

- Bulk Copy Program(BCP)(成批拷贝程序)
- Backup(备份)
- Restore(恢复)
- Database Consistency Checker(DBCC)(数据库一致性检查程序)
- OSQL 和 ISQL
- SQLMaint

8.1BCP

Bulk Copy Program(BCP)实用程序可在 DOS 命令提示符下执行。该实用程

序可将数据从数据库拷贝到操作系统文件，也能从操作系统文件中取出数据，并把它放到 Microsoft SQL Server 数据库上的表格中。在版本 7 中，BCP 实用程序已完全改写，从设计和体系结构的角度来看，它是新版本中改进最大、最有趣的内容之一。

BCP 实用程序的体系结构已经进行了更改，它不再使用 DB_Library 成批拷贝 API；而是使用 ODBC 成批拷贝 API。以前，成批拷贝将数据放入预格式化的行中，这样就形成了无休止的错误源。现在，成批拷贝实用程序使用 Tabular Data Stream(TDS)结果集。这就意味着使用 SQL Server 的主插入路径，所进行的任何优化都将包容数据插入路径的所有方面。在成批装入部分接收到 TDS 结果集后，OLE DB 就接管一切，并把行集移到 Relational Engine 上。然后，OLE DB 把数据从 Relational Engine 移到 Storage Engine，同时把数据格式化成页，并把数据存储在数据库中。

令人欣喜的是，Microsoft SQL Server 一直致力于数据传递的工艺，且已有多种方法解决此问题。在 Microsoft SQL Server 7 中，已经进行了尝试：通过 Data Transformation Services 工具的 Import／Export 功能，“其他”经销商的数据库可以从 SQL Server 输入和输出数据。Data Transformation Services 工具在第 7 章已经作过介绍。

BCP 实用程序的完整格式和插入路径已经发生变化，添加了一些新特性，所以 BCP 在将作业中的数据作为命令文件进行移动时是相当便利的。用户还可以选择是否记录 BCP 事件，以及是加强约束还是激活触发程序。如果不希望记录 BCP 事务，则可设置数据库选项 SELECT INTO ／ BULKCOPY。为此，可

在 Enterprise Manager 中展开数据库、选择某个数据库，右击它，从快捷菜单中选择 Properties，选择 Options 面板，然后单击复选框 Select Into ／ Bulk Copy，如图 8.1 所示。如果希望加强约束，可使用下表中介绍的 BCP 的 CHECK_CONSTARINTS 自变量。

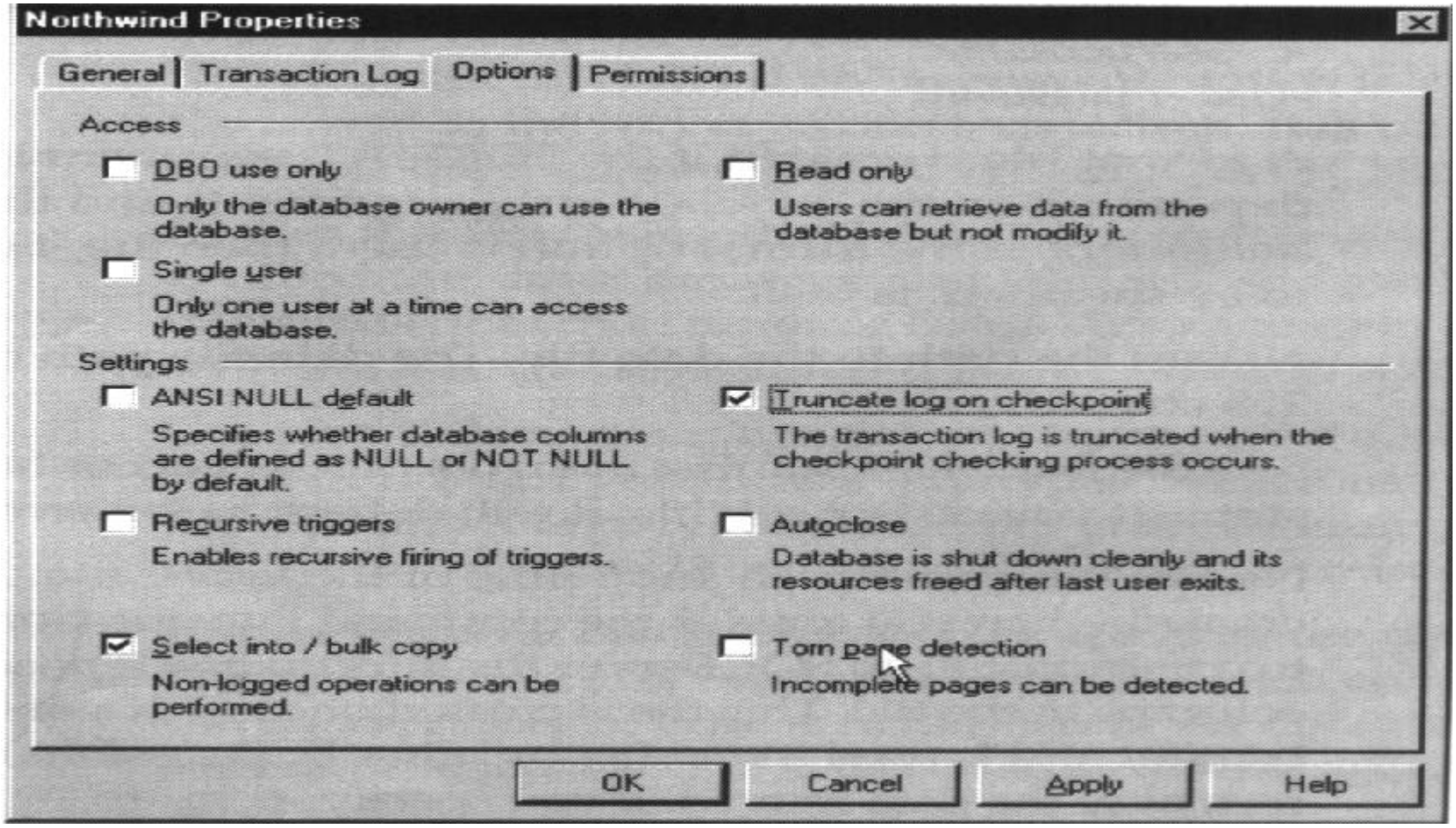


图 8.1Select Into ／ Bulk Copy 数据库选项 BCP 实用程序可用于：

· 从数据库外获取数据，有时可在恢复状态下获得

· 把数据放入数据库，数据有时来自另一个经销商的数据库，有时来自随产品附送的应用程序的非关系型文件格式

· 把大量的数据移入和移出数据库，形成普通的文件格式

· 把数据从某个查询复制到数据文件中（新功能）。请参见表 8.1 中 bcp 的 QUERYOUT 参数

· 在批处理作业和作业步骤中将数据移入和移出服务器

· 将数据从数据库移到电子表格或另一种类型的数据库中，如 Microsoft Excel 或 Microsoft Access

另外，某些应用程序通过使用 DB Library、ODBC 或 OLE DB 程序中的成批拷贝 API，可把 BCP 写到应用程序中。

使用 BCP 实用程序的语法如下：

```
bcp [[databasename.] tableowner.] tablename { in | out | format} datafile
 [-m] [-f] [-e] [-F] [-L] [-b] [-n] [-c] [-w] [-6]
[-E] [-t] [-r] [-i] [-o]
 [-U] [-P] [-S] [-v] [-a] [-T] [-q] [-k] [-h]
```

表 8.1 描述了 BCP 实用程序的自变量。自变量旁的星号(*)表示该自变量很重要。这并不是说，其他的自变量就不需要或不重要了，但如果还没有使用带星号的自变量，就应确定是否需要使用它们。

表 8.1BCP 实用程序的自变量

BCP 自变量	描述
DatabaseName	所调用的数据库名
TableOwner	表格拥有者的名称
TableName	显示数据库表或视图
IN OUT	表明是把数据拷入或拷出该数据库，还是创建格式文件
QUERYOUTQUERYOUT	的一个实例为：bcp "select * from Sysobjects" QUERYOUT objects.txt-c -Sservername-Usa –Ppassword
FORMAT	如果根据-n、-c、-w 或-6 自变量和表格分隔符来使用 FORMAT 自变量，就可创建格式文件。如果使用格式选项，还需要指定-f 选项
DataFile	给出数据文件的路径。数据文件的路径最多可有 225 个字符
-m *	设置执行中止前发生的错误的最大数。如果没有设置此参数，其缺省值为 10
-f	给出格式文件的完整路径和名称。格式文件是在回答了命令行问题后创建的，用于为 SQL Server 提供它所需的文件中的列信息。该文件用缺省名 bcp.fmt 创建一次，此后由其他 BCP 会话使用，作为移动数据的固定格式。如果指定了-n、-c、-w 或-6 自变量，但没有指定-f，则不能进入格式命令行调查表。如果还要避免显示该对话框，可使用-c 自变量，该自变量将使用列或域的字符数据类型。如果使用了-c，BCP 对每列都不作提示，缺省存储类型是字符，制表符是缺省的列分割符，新列是缺省的行终结符

续表

-e *	给出错误文件的完整路径和名称。这是所有“坏”行都要进入的位置。数据库将不接受这些行。如果没有使用-e，就不会创建错误文件
-F	确定要标记为开始拷贝的第一行的行。如果不使用-F，就将第一行标记为开始拷贝的第一行
-L	确定要标记为结束拷贝的最后一行的行。如果不使用-L，就将最后一行标记为结束拷贝的最后一行
-b *	设置要拷贝的每个批处理的行数。每个批处理就成为一个事务。如果不使用-b，就将所有行放到一个事务中该自变量不适于与-h(提示)“ROWS_PER_BATCH=bb”自变量一起使用，-h 自变量是数据文件作为一个事务发送到服务器的位置。服务器负责用 ROWS_PER_BATCH bb 优化装载
-n *	将数据库的数据类型用作缺省的数据类型复制数据，该数据类型称为原(native)数据类型，BCP 对每一列或域的数据类型都不做提示。该自变量与和 SQL Server 交互操作的格式文件命令行相关联
-c *	将 char 数据类型用作缺省的数据类型复制数据，BCP 对每一列或域的数据类型都不作提示。缺省存储类型是字符型(char)，制表符(tab)为缺省的列分隔符，新行是缺省的行结束符。

续表

-w *	将 Unicode 数据类型用作缺省的数据类型复制数据，BCP 对每一列或域的数据类型都不作提示。缺省存储类型是 nchar，制表符(tab)是缺省的列分隔符，新行是缺省的行结束符
-6 *	当使用-c 自变量或-n 自变量时，可将 6.x 数据类型用作缺省数据类型，BCP 对每列或行的数据类型都不作提示。当需要使用 SQL Server 6.x 数据库中的 BCP 文件时，这是很有用的。该自变量与和 SQL Server 交互操作的格式文件命令行相关联
-E *	在包含的表格上定义等同列，且等同值已存在于数据文件中的情况下使用。保存该文件中已有的等同值，如果等同列为主键，并且已有的其他表格用外来键连接到该列上，这是特别有用的
-t *	允许在该自变量后面指定缺省的行结束符。如果要用制表符(tab)，可使用＼t
-r *	允许指定缺省的行结束符。如果要用新行，可使用＼n。
-I	指定重定向输入到 BCP 的数据文件名
-o	命名 BCP 的重定向输出数据文件
-U	指明拥有运行 BCP 许可的的用户注册 ID
-P	命名注册 ID 的密码
-S	给出所使用的，并试图连接的 Microsoft SQL Server。
-v	说明当前的 BCP 版本。当在网络中使用了 Microsoft SQL Server 的多个版本时，或让用户使用过时的版本或 BCP 可执行程序的 beta 拷贝时，这一点就非常重要

续表

-a	对于通过网络发送的每个网络软件包的字节数，允许忽略服务器配置选项(sp_configure)。Microsoft SQL Server 的缺省值为 4096 字节，通常最快的 BCP 的字节范围是 4096-8192 字节。用户可以看到 BCP 输出到屏幕或输出文件时所使用的软件包的尺寸
-T *	需要 Microsoft SQL Server 的受托连接
-q	指定引用标识符 BCP 选项，该选项对应于为每个 SQL Server 连接设置的 ANSI 引用标识符约定。如果标识符名中包含通常无效的字符，如表格名，则可使用此自变量。双重引用限定使用这类标识符
-k *	当数据文件中的域为 NULL 时，数据库列保持 NULL 值，而不允许在已满的数据库列上进行任何缺省操作
-C	如果数据中包含的 char、varchar 或 text 列的长度大于 127 或小于 32，就需要给出代码页，此处 ACP=ANSI／Microsoft Windows(ISO 1252)、OE=-缺省代码页，RAW=代码页不发生转换，或可提供代码页数

续表

-h	当用 BCP 把数据移到表格中时，可传递提示(hint)以备使用。该自变量只用于版本 7 的数据库。可使用下列提示：ORDER (column_list) where column_list = {column [ASC | DESC] [, ...] }使用存储的数据文件，通过把表中的数据按已有的成簇索引顺序进行排列，可提高 BCP 的速度。缺省情况下，BCP 是不排序的。ROWS_PER_BATCH = bb 利用该提示可将数据文件作为一个事务发送到服务器上；服务器负责用 ROWS_PER_BATCH 值优化装载 KILOBYTES_PER_BATCH = nn 使用 KILOBYTES_PER_BATCH 提示可以指明每个批处理的千字节数。TABLOCK 该优化程序提示可为整个 BCP 操作提供独占的表格块。缺省设置是无表格块，这可能意味着对表格的争用。并行 BCP 数据装载需要该提示。CHECK_CONSTRAINTS 该提示意味着接收 Microsoft SQL Server 表格的表格约束得到重视。如果不使用该自变量，约束就得不到重视

8.2 备份(Backup)

在以前版本中，原来称作“转储”数据库或事务记录的操作现在称为 BACKUP。这一变化对应于 Microsoft SQL Server 7 包含的新数据结构，Microsoft SQL Server 7 可提供下列功能：

- 备份整个数据库
- 备份文件和文件组
- 备份事务记录

部分 BACKUP 程序有时包括转储设备的创建，该设备是一个物理文件的逻辑名。数据库、文件、文件组或记录都可卸除到这个与物理文件相关联的逻辑设备名上。

在创建了转储设备之后，就可以备份和恢复数据库了。BACKUP 和 RESTORE 语句都有许多不同的变体。在第 21 章中将介绍这些不同变体的实例和自变量。

8.3 恢复 (Restore)

在以前版本中原来称作装载(loading)数据库事务记录的操作现在称为 RESTORE。这一变化对应于 Microsoft SQL Server 7 包含的新数据结构，Microsoft SQL Server 7 可提供下列功能：

- 恢复整个数据库
- 恢复文件和文件组
- 恢复事务记录

RESTORE 实用程序除了以前版本中已有的其他选项外，新增了两个选项：RESTART 和 REPLACE。如果使用 REPLACE，SQL Server 将删除原数据库，并根据备份集中的规范重新创建数据库。在下列条件下，如果不使用 REPLACE

选项，将不恢复原数据库：

- 数据库已存在于服务器
- 数据库名不同于备份集中的数据库名
- 数据库中的文件集不同于备份集中包含的数据库文件集

在恢复过程中，Microsoft SQL Server 不识别文件尺寸的差别，恢复时不检查文件尺寸。

RESTART 是发生中断时重新启动恢复的选项。为此，可重复使用 RESTORE 命令并指定 RESTORE 选项。然而，该选项有一个问题必须说明：它只能用于磁带介质中的恢复，并写在多卷磁带上。

另一个重要的新特性是 RESTORE VERIFYONLY。该特性可检查备份集是否完整，数据是否能读。它不检查数据库是否损坏。

RESTORE 语句还有一些很有用的变体：

- RESTORE FILELISTONLY 返回的结果集带有数据库列表和备份集中包含的记录文件
- RESTORE VERIFYONLY 检查备份集是否完整，数据是否能读。它不检查数据库是否损坏
- RESTORE HEADERONLY 显示备份文件中所有备份的所有头文件信息

8.4 数据库一致性检查程序(Database Consistency Checker)

Database Consistency Checker(DBCC)可用于扫描数据库的不同部分，以检查其不一致性。该检查程序可查找错误，并以 DBCC 输出报告的形式报告其查找结果。随着使用该检查程序的次数不断增多，用户将根据编号熟悉 DBCC 错误，了解其内容、严重程度，并根据特定的环境采取相应的措施。使用 DBCC 有许多不同的方法，每个环境都会根据应用程序所使用的各种数据库功能出现不同的 DBCC 错误。影响如何使用 DBCC 的因素有：数据库的尺寸，该数据库上的用户数,以及数据的重要性等。在版本 7 中,CHECKDB 是运行 CHECKCATALOG 和 CHECKALLOG 的一次作业站。由于版本 7 的安全体系，它无须执行很多步骤。如果数据库不在单用户模式下，并且有许多选项用来修复 CHECKDB、CHECKALLOG 和 CHECKTABLE 中的数据库，就不会再发生虚假错误。

8.4.1DBCC 方案

下述三个方案论证了如何根据用法和尺寸在不同的环境下运行 DBCC。

8.4.1.1DBCC SCENARIO #1 重要数据，中小型数据库，上班时间使用

如果有了数据库，如帐目数据库，内含不在多兆字节范围但日常使用最频繁的重要数据，就可以在下班时间以下列方式使用 DBCC：

- 运行 DBCC CHECKDB
- 检查 DBCC 输出错误
- 如果发现错误，要确定错误的严重程度(有些错误是由于空间分配造成的，不能认为是严重错误)
- 检查 Knowledge Base 以确定如何处理错误
- 按照 Knowledge Base 程序更正错误
- 转储数据库

上面涉及到的 Knowledge Base 是一个 Microsoft 工具，它记录已作过注释的各种错误。Knowledge Base 可以给出每个错误的替代方案或正确操作。在带有许多服务器和数据库的环境中，Knowledge Base 是非常重要的工具，因为它能快速分析最近发生的错误的各个方面，所以笔者一直在使用该工具。对 Knowledge Base 的访问可以通过 Internet 进行，也可通过与 Microsoft Technical Support 联系以确定如何访问它。如果不能访问 Knowledge Base，在发现严重错误时可与 Microsoft Technical Support 电话联系。

Microsoft SQL Server 能通知 DBA 需要的正确操作，所以下班时运行 DBA 的整个过程可以自动进行。要充分利用这一功能，应按优先顺序对数据库进行分类，当这些数据库开始工作时，就应确定是立即寻呼 DBA，还是一早向 DBA 发送 e-mail。在美国和欧洲频繁使用的数据库可能每天 24 小时都在使用，并可能需要每天 24 小时／每周 7 天的无线寻呼支持，或在发生严重的 DBCC 错误时，需要某些其他类型的即时通知。在这种情况下，DBCC 可能没有在该服务器上运行，而是运行在其他服务器上。这就是下一个 DBCC 方案要讨论的问题。

8.4.1.2DBCC SCENARIO #2 重要数据、巨型数据库(VLDB)、连续不停的使用

如果包含重要数据的数据库是巨型多兆字节的数据库，且用户连续使用它，就可以考虑下列策略：

- 获得备份服务器
- 在备份服务器上以单用户模式运行 DBCC CHECKDB

这对管理员来说是较难以掌握的一种方案，因为必须采取额外的措施才能确保数据的完整性和可恢复性。此时最好是得到另一个服务器(为了避免由于硬件异常或某些与硬件相关的其他问题，而不得不通宵达旦地把应用程序移到另一个服务器上，就应拥有备份服务器)。使用该备份服务器可在该数据库上运行 DBCC 检查，备份服务器也可以用作报告服务器(请参见下一节)。

对于这样重要的数据库采用备份服务器具有各种充分、坚实的理由。对于多兆字节的数据库，DBCC 必须在页面层次上考察它。对大型数据库而言可能花费较长时间，因此就不可能为客户维护一个良好的计算环境，但是 DBA 在把数据库恢复到备份服务器上之后，就可以充分利用备份服务器，以单用户模式在数据库上运行 DBCC。

8.4.1.3DBCC SCENARIO #3 报告服务器、复制从另一个数据库中恢复的数据库、只读数据库、连续使用

如果报告服务器的数据库实际上是从另一个服务器上恢复的，就可以考虑下列策略：

- 不必在原服务器上运行 DBCC
- 确保 DBCC CHECKDB 运行在原数据库或报告服务器上

如果 DBCC 未运行在原数据库中，而使 DBCC 运行在该报告服务器上会更有意义，就应考虑获得另一个报告服务器的可能性。这将允许维护窗口以数据库的预防性维护为中心进行升级和操作，并允许用户随时访问数据库。通过监视环境来确定用户数量和使用次数何时增长。随着用户数量和使用次数的增加，要设法满足硬件的最大需求并增加服务器。坚持位于该曲线的前面可节省大量时间。

大多数 DBCC 错误都出现在运行 DBCC 前的错误记录中。错误记录是每个 Microsoft SQL Server 写入的内容，在有许多服务器的环境下，当错误记录中出现错误时，使用 SQL Server 的自动创建警告功能向 DBA 发送 e-mail 或寻呼 DBA 是很重要的。这是对出现的故障发出的第一个警告。这些警告可按错误的严重程度排列，用户可指定是通报具有某种严重程度的错误，还是根本不通报某些错误。现在为 SQL Server 错误设置自动警告是很容易的，第 4 章 SQL Server Agent 已介绍了设置自动警告的基本原理。

8.4.2DBCC 的功能

在数据库上运行的 Database Consistency Checker 有不同的类型，每种类型可完成不同的事务。下面几节将介绍最常用的几种类型。

8.4.2.1DBCC CHECKCATALOG

DBCC CHECKCATALOG 可检查各系统表是否一致。每次安排 DBCC 在该数据库上运行时都应该运行 DBCC CHECKCATALOG。这就可用 systypes 检查 syscolumns 的完整性，用 syscolumns 检查 sysobjects 的完整性。syscolumns 中的每个数据类型在 systypes 中要有一行，每个表格和视图在 syscolumns 中最少要有一列。

8.4.2.2DBCC CHECKDB

DBCC CHECKDB 可检查数据库中每个对象的完整性，应修复所发现的任何错误。如果运行了 DBCC CHECKDB，可不必再运行 DBCC CHECKALLOG 或 DBCC CHECKTABLE。应检查每个表格的索引和数据页，看看其中是否有正确的页面连接和一致的指针。进一步检查各索引，以确保各项已正确排序，且每个页面上的偏移和数据都是合理的。要检查每个表格的文本、ntext 和图像页面，以及所有页面上的页面分配。

8.4.2.3 其他 DBCC STATEMENTS

许多其他的 DBCC 语句是 DBCC CHECKDB 功能的子集，或可执行某些其他实用程序类型的功能。在第 24 章“Transact-SQL Programming Language Reference”中，将介绍其语法、说明和自变量。

· DBCC CHECKALLOG 可检查数据库中数据页的分配。NOINDEX 表示不检查用户表格中的非成簇索引。NO_INFOMSGS 通知 SQL Server 不显示信息类消息，而只显示错误消息

· DBCC CHECKIDENT 返回已命名表格的当前等同值，或更正该等同值

· DBCC SHRINKDATABASE 可压缩数据库中数据文件的尺寸

· DBCC SHRINKFILE 可压缩数据文件的尺寸。目标尺寸是一个整数，且是想要的文件尺寸(以兆字节为单位),如果还指定了 TRUNCATEONLY,则 SQL Server 不使用该尺寸。如果不使用目标尺寸，SQL Server 将尽可能压缩文件的尺寸。NOTRUNCATE 表示在数据库文件中保留空间。TRUNCATEONLY 表示不重新为各行定位，并将未用空间添加到目录中的自由空间上

· DBCC CHECKFILEGROUP 可查看文件组中所有表格的分配情况和完整性。NOINDEX 表示不检查用户表格上的非成簇索引。NO_INFOMSGS 通知 SQL Server 不显示信息性消息，而只显示错误消息

· DBCC UPDATEUSAGE 可更新 sysindexes 表，以指明表格中尺寸数据(行数、已使用的行、保留行和 dpages)的最新值。NO_INFOMSGS 通知 SQL Server 不显示信息性消息，而只显示错误消息

· DBCC UPDATEUSAGE WITH COUNT_ROWS 可为 indid 是 0 或 1 的 sysindexes 行用表格中的当前行数来更新 sysindexes 系统表。COUNT_ROWS 处理大型表格的时间较长

· DBCC DBREINDEX 可重新建立表格的一个或所有索引。如果卸除或重建索引，这对于保护唯一性和可能丢失的主键约束是非常有用的。填充因子是每个索引页上有多少数据的百分比。如果使填充因子为 0，就使用创建索引时所用的原始填充因子。NO_INFOMSGS 通知 SQL Server 不显示信息性消息，而只显示错误消息

· DBCC OPENTRAN 可显示最旧的活动事务和已复制(分布式和非分布式)的事务。WITH TABLERESULTS 以可载入表格的格式给出结果。NO_ INFOMSGS 通知 SQL Server 不显示信息性消息，而只显示错误消息

· DBCC PINTABLE 表示 SQL Server 不从内存中清除该表格。该表格在内存中有它自己的位置

· DBCC UNPINTABLE 表示 SQL Server 可以从内存中清除表格

· DBCC CHECKTABLE 可检查数据库上的特定表格(数据、索引、文本、ntext、图像页面)的完整性。NOINDEX 表示不检查用户表格上的非成簇索引。NO_INFOMSGS 通知 SQL Server 不显示信息性消息，而只显示错误消息

· DBCC SHOWCONTIG 说明已命名的表格和已命名的索引中是否包含碎片

· DBCC TEXTALLOG 可检查表格的文本或图像列的分配情况。FULL 通知 SQL Server 生成分配报告

· DBCC MEMUSAGE 可显示 SQL Server 的内存使用情况，以及以 8K 页面表达的 SQL Server 的过程高速缓存

· DBCC PERFMON 可显示三个统计 IOSTATS、LRUSTATS 和 NETSTATS 的性能信息

· DBCC DBREPAIR 可卸除不能恢复的数据库。该命令只包含向后的兼容性，可用 Drop Database 代替它

· DBCC TEXTALL 可选择数据库中包含文本、ntext 或图像的表格，并在表格上运行 TEXTALLOG。FULL 通知 SQL Server 生成分配报告

· DBCC USEROPTIONS 可显示当前对话的 SET 选项

· DBCC dllname(FREE)从内存中取出已命名的扩展存储过程

· DBCC SQLPERF(LOGSPACE)可显示 Microsoft SQL Server 上数据库中的事务记录空间

· DBCC TRACEON 可打开已命名的跟踪标志

· DBCC TRACEOFF 可关闭已命名的跟踪标志

· DBCC TRACESTATUS 说明服务器上使用的跟踪标志

· DBCC PROCCACHE 可显示过程高速缓存的信息

DBCC 是可执行许多功能的强有力工具。用户越熟悉 DBCC 的，就越能管理好服务器，并使数据库保持良好的状态。

8.5 OSQL 和 ISQL

OSQL 实用程序除了使用 ODBC，而不使用 DB-Library 连接和与 Microsoft SQL Server 7 通讯之外，它的功能类似于 ISQL。在 OSQL 和 ISQL 中都可以运行 Transact-SQL 语句、系统过程和脚本文件。这两个实用程序都可直接从操作系统上启动，都包含区分大小写的自变量。用户可以使用这两个实用程序与发送给 STDOUT 的结果交互运行，也可通过发送给命名文件的结果来传递自变量，以批处理模式在命令文件中运行 SQL。

不仅 Transact-SQL 语句可以在 ISQL 和 OSQL 内部运行，这两个实用程序还有专用命令：

- GO 运行语句
- RESET 通过清除已输入的语句来重置对话
- ED 调用编辑器
- !!执行操作系统命令
- QUIT 或 EXIT()从 ISQL 和 OSQL 中退出
- CTRL＋C 中断查询，但不退出

在两个实用程序中，都可以先输入两个惊叹号(!!)，再输入命令，来执行操作系统命令。另外，还可以在实用程序内部的提示下键入下列语法，交互地输入每个实用程序，把一个文件从操作系统文件结构中读入 ISQL 或 OSQL 缓存区：

: r filename

要在 Transact-SQL 语句（该语句是从 ISQL 或 OSQL 中递交给 SQL Server 的）中包含注释，可在每行的开始使用两个连字符(--)，或使用普通的 C 编程语法(／*...*／)。其语法、说明和自变量将在第 21 章中介绍。

8.6SQLMAINT

Sqlmaint 实用程序可以用来创建备份数据库的作业，并在命令提示行上运行。在版本 7 中，如果每次运行 Database Maintenance Plan 时，对把数据库备份成新的备份文件名的方法不满意，可用该实用程序代替 Database Maintenance Plan(Database Maintenance Plan 可能会使服务器上的空间出现问题)。要运行 sqlmaint，SQL-DMO 必须在 SQL Server 上注册。

Slqmaint 实用程序对数据库进行维护操作，并且非常方便，因为它可在命令提示下或在 SQL Server Agent 中运行。该实用程序可用于备份数据库、检查数据库的一致性、运行 UPDATE STATITICS、校验备份、只有清除了数据库一致性检查才备份数据库，并能把结果放到 HTML 文件中，或用电子邮件将结果发送给操作员。

所用的语法如下：

[-?] |

[

```
 [-S server]
 [-U "login_ID"  [-P "password"]]
{
 [  -D database_name | -PlanName name | -PlanID guid]
 [-Rpt text_file  [-DelTxtRpt <time_period>]  ]
 [-To operator_name]
 [-HtmlRpt html_file  [-DelHtmlRpt <time_period>]  ]
 [-RmUnusedSpace threshold_percent free_percent]
 [-CkDB | -CkDBNoIdx]
 [-CkAl | -CkAlNoIdx]
 [-CkTxtAl]
 [-CkCat]
 [-UpdSts]
 [-UpdOptiStats sample_percent]
 [-RebldIdx free_space]
 [-WriteHistory]
 [
{-BkUpDB  [backup_path]  | -BkUpLog  [backup_path]  }
{-BkUpMedia
{DISK  [  [-DelBkUps <time_period>]
 [-CrBkSubDir ]  [ -UseDefDir ]
```

]
| TAPE
}
}
[-BkUpOnlyIfClean]
[-VrfyBackup]
]
}
]
<time_period> ::=
number [minutes | hours | days | weeks | months]

变量如表 8.2 所示。

表 8.2sqlmaint 实用程序变量

sqlmaint 变量	描述
-?	显示 sqlmaint 的语法
-S server	服务器是目标服务器
-U "login ID"	Login ID 是连接 SQL Sever 时对 se 的许可
-P "password"	密码是注册 ID 的密码
-D database_name	数据库是执行维护的数据库名
-PlanName name	名称是数据库维护计划的名称

续表

-PlanID guid	Guid 是数据库维护计划的全球唯一标识符(GUID)
-Rpt text_file	Text_file 是文件的合法路径和名称
-DelTxtRpt<time_period>	Time_period 可用于删除名称与 text_file*相匹配，且比该时限更旧的任何文件
-To operator_name	Operator_name 是通过 SQLMail 发送报告的操作员
-HtmlRpt html_file	Html_file 是放置 HTML 报告的文件的合法路径和名称
-DelHtmlRpt<time_period>	Time_period 可用于删除名称与 html_file*相匹配，且比该时限更旧的任何文件
-RmUnusedSpace threshod_percent free_percent	RmUnusedSpace 指示 SQL Server 从数据库中删除未用的空间，以用于自动扩展的数据库
	Threshod_percent 是数据库在删除未用空间前所应有的兆字节数
	Free_percent 是可以在数据库中保留的未用空间量。这是数据库尺寸的百分比
-CkDB|CkDBNoIdx	-CkDB|CkDBNoldx 指示 SQL Server 用或不用 NOINDEX 选项运行 DBCC CHECKDB

续表

-CkAl | CkAlNoIdx	-CkAl | CkAlNoIdx 指示 SQL Server 用或不用 NOINDEX 选项运行 DBCC NEWALLOC
-CkTxtAl	-CkTxtAl 指示 SQL Server 运行 DBCC TEXTALL
-CkCat	-CkCat 指示 SQL Server 运行 DBCC CHECKCATALOG
-UpdSts	-UpdSts 指示 SQL Server 在所有数据库表格上运行更新统计
-UpdOptiStats sample_percent	-UpdOptiStats sample_percent 指示 SQL Server 使用 UPDATE STATISTICS 的 WITH SAMPLE 选项在所有用户创建的数据库表格上运行更新统计
-RebldIdx free_space	-RebldIdx free_space 指示 SQL Server 通过将 free_space 百分值用作填充因子的反数，来重建数据库中表格的索引。例如，如果 free_space 百分比为 10，填充因子就是 90。如果 free_space 等于 100，就使用原始填充因子
-WriteHistory	-WriteHistory 把行写入 msdb 系统数据库的 sysdbmaintplan_history 表中

续表

-BkUpDB［backup_path］ |-BkUpLog［backup_path］	-BkUpDB［backup_path］ |-BkUpLog［backup_path］可定义要执行的备份操作-BkUpDb 备份整个数据库。-BkUpLog 备份事务记录。［backup_path］是备份的目录名。如果提供了自变量-UseDefDir，就不需要［backup_path］，如果二者都提供，将不使用［backup_path］。数据库备份的文件名由系统创建，该文件名可以是：dbname_db_yyyyMMddhhmm.BAK，或 dbname_log_yyyymmddhhmm。用户还必须提供-BkUpMedia 参数
-BkUpMedia DISK|TAPE	-BkUpMedia DISK|TAPE 是备份的介质类型
-DelBkUps<time_period>	-DelBkUps<time_period>指示系统如果目录中的备份文件比 time_period 还旧，就删除它们 number［minutes	hours	days	weeks	months］是一个整数，代表时间。例如，12 周、3 个月或 15 天。如果只提供一个整数，缺省时代表周-CrBkSubDir-CrBkSubDir 将在［backup_path］目录下创建系统命名的子目录，它是一种不需改变［backup_path］自变量，而把每个数据库的备份都放到各自的目录下的方法

续表

-UseDefDir	-UseDefDir 指示 SQL Server 把备份文件放到缺省的备份目录下，不考虑[backup_path]自变量的内容。对于香草 SQL Server 安装，这是 C:\Mssql7\Backup 目录
-BkUpOnlyIfClean	如果-Ck 检查发现在数据中存在 DBCC 错误，-BkUpOnlyIfClean 禁止进行备份。然而，维护操作是定位的，并以自变量出现的顺序运行。如果使用这个自变量，就要把任何-Ck 检查都放在-BkUpDB／-BkUpLog 自变量之前，否则将得不到预期的结果
-VrfyBackup	-VrfyBackup 指示 SQL Server 在备份结束后对备份运行 RESTORE VERIFYONLY

所有的 SQL Server 实用程序在版本 7 中都进行了改进，对过去出现的性能问题都给予了应有的重视。Microsoft 知道，这些实用程序都必须能在 T 字节的数据库上运行，它们经过大幅度的修改后可达到这个目标。在下一章中，将回顾 SQL Server 7 增强的复制性能。

第 9 章 SQL Server 7 的复制增强

复制用于把数据从一个 SQL Server 上发布到其他位置的 SQL Server 上，复制操作实现可以通过建立该数据的原始拷贝，然后再设置 SQL Server 同步数据，再使用 Publisher 和 Subscriber 使此后的数据从一个服务器转移到另一个服务器上。虽然只有一个 Publisher，但一个 Publisher 却有许多个 Subscribers。在 SQL Server 7 中，Subscriber 可以更新 Publisher 中的数据，这与传统模式相反，在传统模式中，数据只能在 Publisher 中才能改变。其实，Subscriber 也可以是 Publishers。

通过把两个新的 Agents 加入 SQL Server 7，提高了它的复制能力。目前连同 Log Reader 和 Distribution Agents 一道还有 Snapshot 和 Merge Agents，而且在我们想要复制数据时，还有以下三个主要选择：

- 合并复制(Merge Replication)
- 快照复制(Snapshot Replication)
- 事务复制(Transaction Replication)

本章主要介绍的复制增强如下：

- 多点更新复制(Multi-site Update Replication)
- 合并复制(Merge Replication)

- 快照和数据转换服务(Snapshot and Data Transformation Services)
- 分区事务复制(Partitioned Transactional Replication)
- 更新订阅器(Updating Subscriber)
- 推、拉订阅(Push and Pull Subscriptions)
- 匿名拉回订阅(Anonymous Pull Subscriptions)
- 因特网复制增强(Internet Replication Enhancements)
- 数据加密(Data Encryption)
- 多机环境复制服务(Heterogeneous Data-Replication Services)
- 创建复制脚本(Creating Replication Scripts)

9.1 多点更新复制(Multi-site Update Replication)

与复制一道产生的问题是，被更新的数据是何时和怎样跨越复制场景有关的所有的服务器的。多个站点如何同时更新数据，而且不用有一个站点是复制场景的拥有者?如果同时有更新发生会怎样呢?谁首先更新和谁必须等到解除锁定?复制会涉及很多问题，例如进行二阶段更新时的数据丢失和有用性、不确定性。一台机器在等待更新所有的远程服务器时，也许变得不可用。在这些领域，应用程序具有不同的需求，取决于你的用户怎样观察和使用数据。

为了解决允许多站点更新复制数据的问题，微软 SQL Server 已拥有许多方法，以根据应用程序的需要实施多站点更新复制。

应用需求	多位更新选择
自主最重要	Merge Replication
延迟时间相容性	Snapshot and Data Transformation Services
为了避免冲突使用分区应用程序，时间仅是时间问题	Partitioned Transactional Replication
订阅器必须是相容的	Updating Subscribers

一个复制工具在试图满足客户对复制拷贝需要的同时，还必须遵循物理和网络的法则。

9.2 合并复制(Merge Replication)

合并复制是微软 SQL Server 7 中的一种多站点更新复制选择，它用于在 SQL Server 的自主性是最重要的条件下。在这个进程中，当调度的合并发生时，数据仅在成双地调解中移动。这种合并进程与 Server 和 Reconclier 二者都相连以便在合并数据的同时解决出现的任何冲突。到达的数据与现有的数值合并，同时又按照调解规则解决了冲突，该调解规则通过 COM／OLE 能够扩展和定形。另一方面，根据先前指定的优先次序解决各种冲突。数据从改变的 SQL Server 移到需要同步的一个 SQL Server 上。

就合并复制来说，调解数据并不使用日志读和事务，而是用实际数据。可以多次改变一个列，但只有最后一行图像用来调解。可以使用 Configure Publishing and Distribution Wizard 配置 Merge Replication。为了在 Enterprise

Manager中访问该向导，必须扩展一个服务器并选择 Tools／Wizatds Replication／Configure Publishing and Distribution Wizatd。该向导将使 Publishing 数据库用于 Merge Replication。

首先，必须选择一个分布器。这个进程将产生一个分布(distribution)数据库。可以在将该 distribution 数据库建立在 Publishing Server 上，或用一个不同的服务器配置成分布器之后存放 distribution 数据库。图 9.1 给出了 Enable Publisher 窗口。如果选择的话，可以注册其他的服务器作为此屏的 Publishers。

可以单击列表框中的...按键，来看 Publisher 的选项，其中包含工作文件夹信息以及 Replication Agents 是怎样存入 Publishing Server 的。下一个荧屏给出复选框，它能使 Publishing 数据库用于 Merger Replication。

接下来一个窗口给出允许 Subscriber的程序。在这种情况下，Publisher,Distribution 以及 Subscriber都在相同的 SQL Server 上。接下来一个窗口给出 Subscriber 的选项，其中包括 Replication Agents 如何存入 Subscriber。下一个窗口显示的是 Publishing 和 Distrbution Wizard 成功完成配置 distribution 数据库。

如果决定从 distribution 数据库的建立和配置中退出，可以用 Disable Publishing and Distribution Wizatd 来删除 distribution 数据库和刚刚建立的 Publisher和 Subscribers。要访问 Enterprise Manager 的向导，扩展该服务器，查阅 Tools 菜单并选择 Replication Configure Publishing and Distrbution Wizatd／Disable the Publishing and Distribution Wizard。

就象为了合并复制 Merge Replication 而配置一个 Publisher,并删除 Distributor 和 Publisher 那样，为了合并复制也能配置出版物。为了这样做，首先应通过扩

展该服务器和选择 Tools／Wizards 访问 Enterprise Manager 中的 Publish and Push Subscription 窗口(图 9.2)。

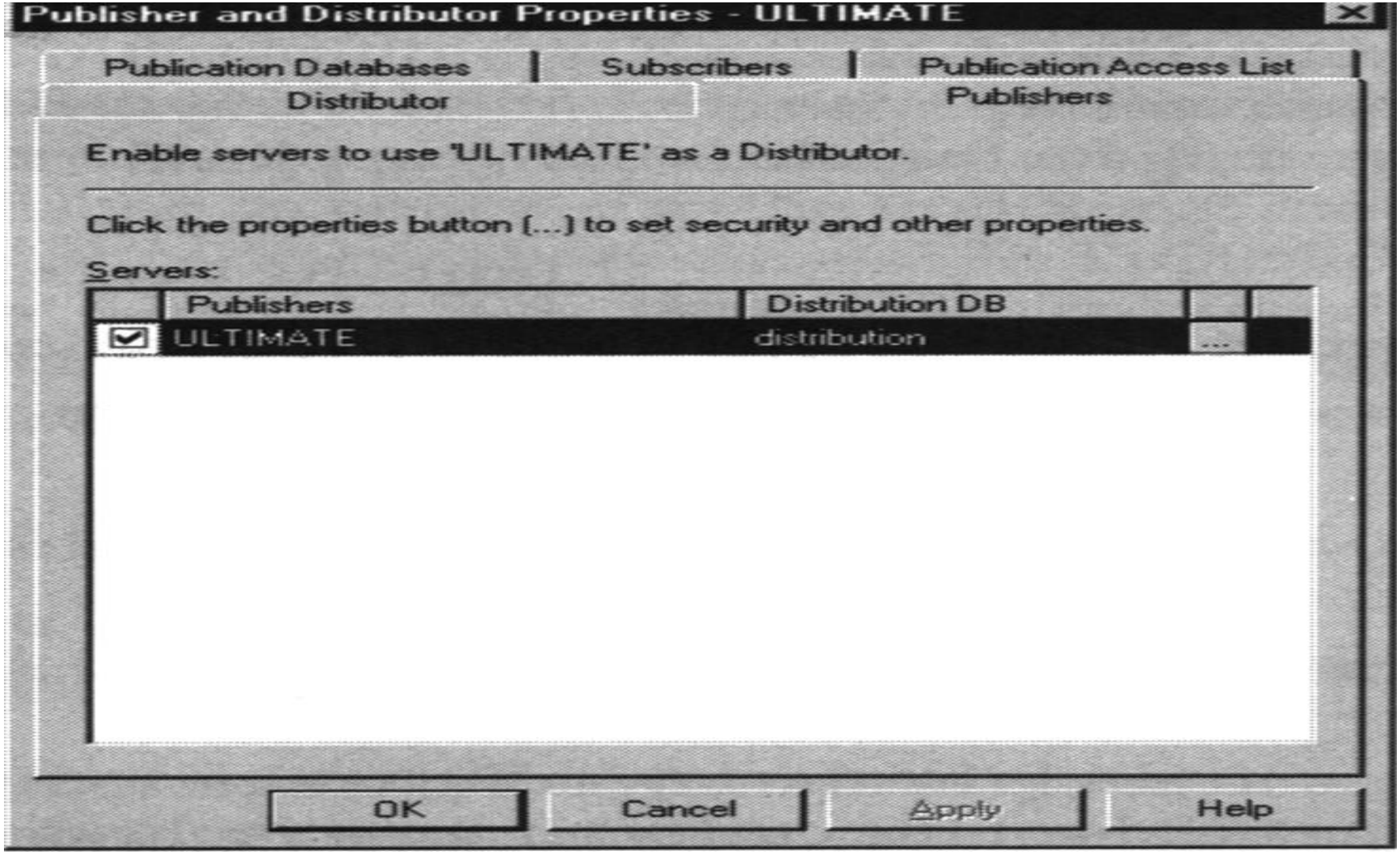


图 9.1Configure Publishing and Distribution Wizard 的 Enable Publishers 窗口

图 9.2Create Publication Wizard

接下来，选择 Wizards，**双击** Replication **并选择** Create Publication Wizard。

该 Wizard 将让你选择出版物的类型：快照、事物或合并。The Create Publication Wizard 请求确定出版物的名称和说明。再下一个窗口就显示成功完成 Create Publication Wizard。在该 Create Publication Wizard 窗口关闭之后，新的出版物就出现在 Enterprise Manager 中的 Replication Monitor 层次树下，如图 9.3 所示。

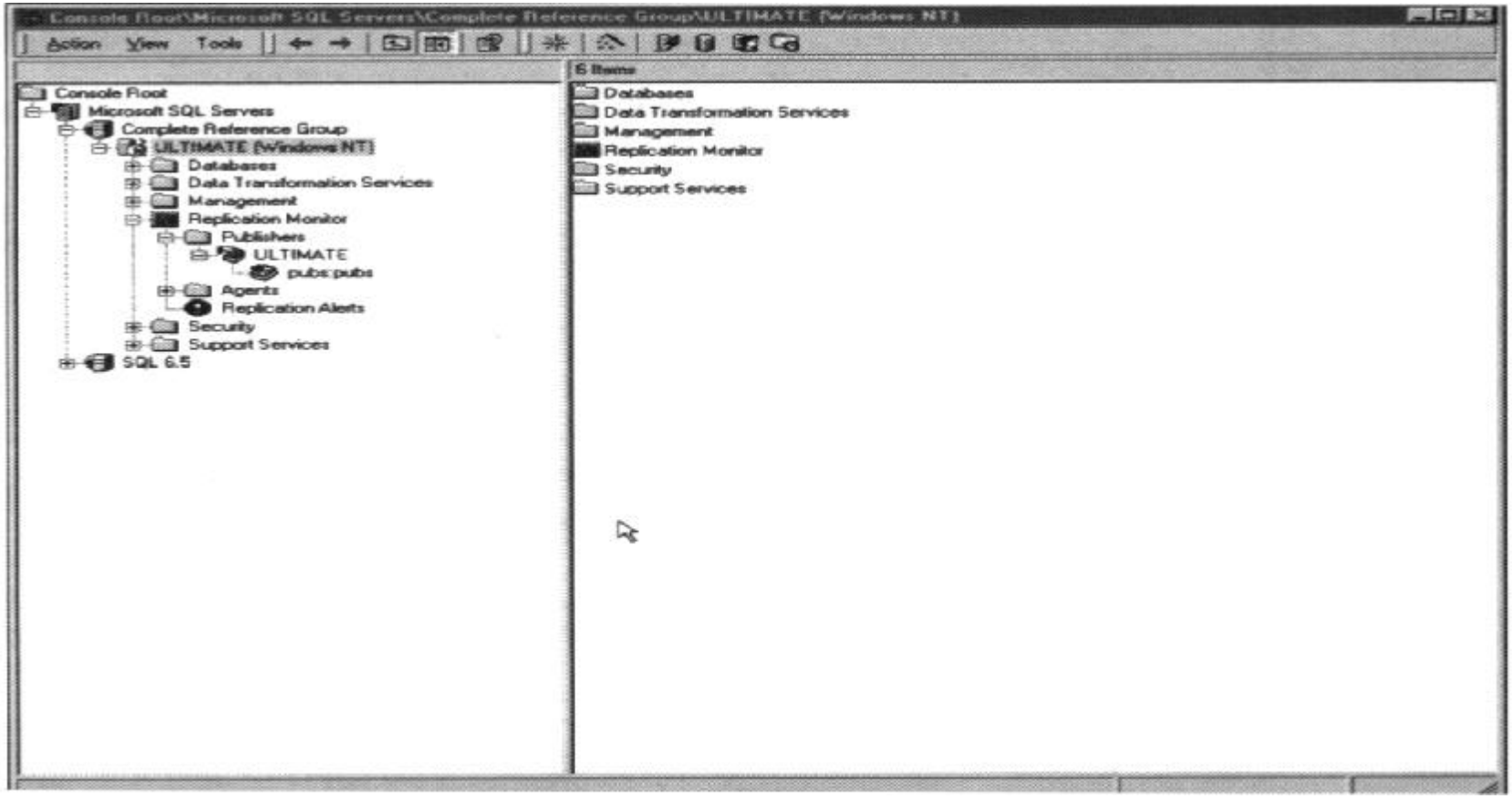


图 9.3 新的出版物出现在 Replication Monitor 之下

现在必须建立 Subscription 作为一个 Merge Subscription。对这个例子来说，

我们将使用 Pull Subscription Wizard 从该订阅器中拉回出版物，与从出版器那里推出版物相反。选择 Pull New Subscription 按钮，则 Pull Subscription Wizard 在荧屏上出现。

必须为订阅器选择一个出版物。从 Choose Publication 窗口中选择 Publication。下一个窗口给出是现在初始化订阅物还是等待调度代理时进行合并，再下一个窗口允许创建 Merge Agent 调度。

在 Set Merge Schedule 荧屏上单击 Change 按键，调度窗口，现在叫做 Edit Recurring Tob 窗口出现。这与先前微软 SQL Server 提供的调度窗口是相同的，并且与老的 Task Schedule 荧屏非常相似。

接下来出现的窗口如图 9.4 让你设定订阅优先级，多位更新就是这样解决更新冲突。如果两个订阅器同时更新一行数据中相同的列，该列将显示谁拥有更优先的权力。

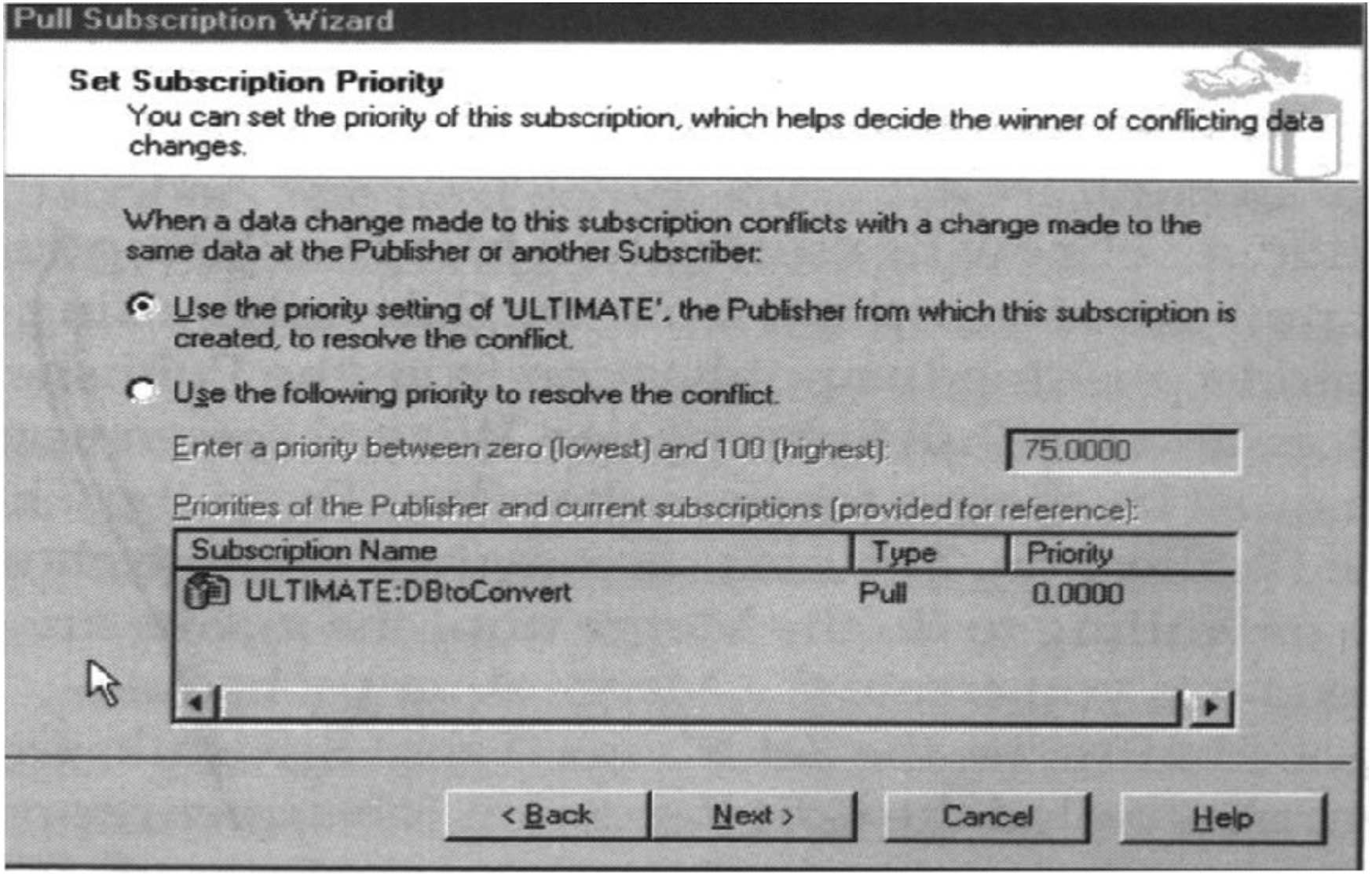


图 9.4 设置订阅优先级

最后一个窗口完成建立一个 Merge Replication Pull Subscription 的进程。

9.3 快照和数据转换服务

快照多站点更新复制选择在指定的时间拍摄一张该数据库的结构和数据。这就是第一步的同步基础。BCP 实用程序是用来移动该结构和数据的。这之后更新的数据施加于预定的目标服务器上。在调度复制配置的同时，一个新的快照被拍摄并按照一定的周期施加在服务器上。

这个选择在处理日志读和事务日志时有利于减少管理费用。有一个时间延迟包含在其中，但数据保留在连续状态，对管理和配置来说，这是最容易的复制方案。

注意：Create Publication Wizard 是通过扩展 Enterprise Manager 选择 Tools|Wizards 来实现的。

The Choose Publication Type 窗口在 Create Publication Wizard 中有代表了三种主要复制类型的三种选择即：

- 快照复制
- 事务复制
- 合并复制

The Choose Publication Type 屏幕简要地描述了每一种复制类型的情况，如图 9.5 所示。

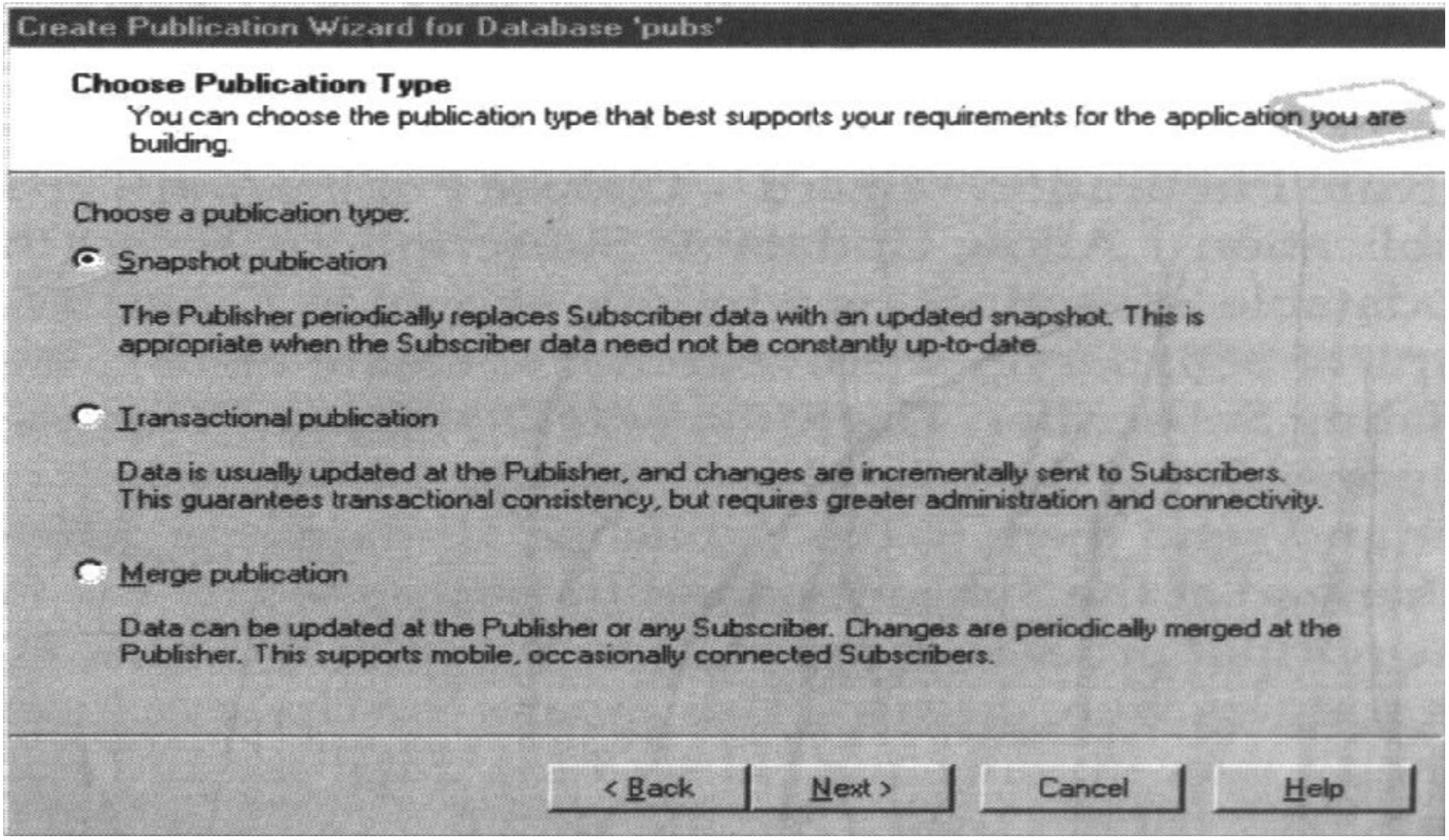


图 9.5 选择出版物类型屏幕

9.4 分区事务复制

当应用程序需要在出版器和订阅器之间在一个事务的片刻之内具有相容性时使用分区事务复制是最好的。在分区应用时更好的办法是避免更新冲突，而不是在他们发生时试图解决它们。冲突的消除必定有一个胜者和一个失败者，而且任何时候都有一个失败者，总有一个事务不能得以应用。通过分区数据使用事务复制可以避免这种情况。

事务复制有一个出版器和一个或多个订阅器。一个出版器将数据以 Articles 的形式送到该订阅器。该事务日志用于存入对数据的改变，然后将这些改变放在一个 distribution 数据库上，在那里他们被送到用户，与出版器订制的次序相同的。这个进程需要在出版器和订阅器之间有一个连络并且复制只需数秒的延迟时间。

The Snapshot Agent，Log Reader Agent 以及 Distribution Agent 共享事务复制。你可以与可更新的订阅器和拉回阅或匿名的拉回订阅来配置事务复制。

9.5 更新订阅器

更新 Subscriber 是在该 Subscriber 上更新数据的同时，对该出版器执行更新的一种复制 Subscriber，这是借助于两阶段提交和 Distributed Transaction

Coordinator 来完成的。这个叫做同步事务的特征是一个新的复制特征——首次在 SQL Server 7 中可以获得，它对事务复制是可以利用的，对快照复制也是可以利用的。两者都能在 Create Publication Wizard 中通过检查复选框设定。为了能更新 Subscribers，必须在 Enterprise Manager 中扩展该服务器并选择 Tools／Replication／Publish and Push Subscriptions／Create Publication／Create Publication Wizard／Choose Publication Type，Snapshot(或 Transaction)Publication／Allow Updatable Subscription。在 Allow Updatable Subscriptions 窗口中选择 Yes 单选按钮，如图 9.6 所示。

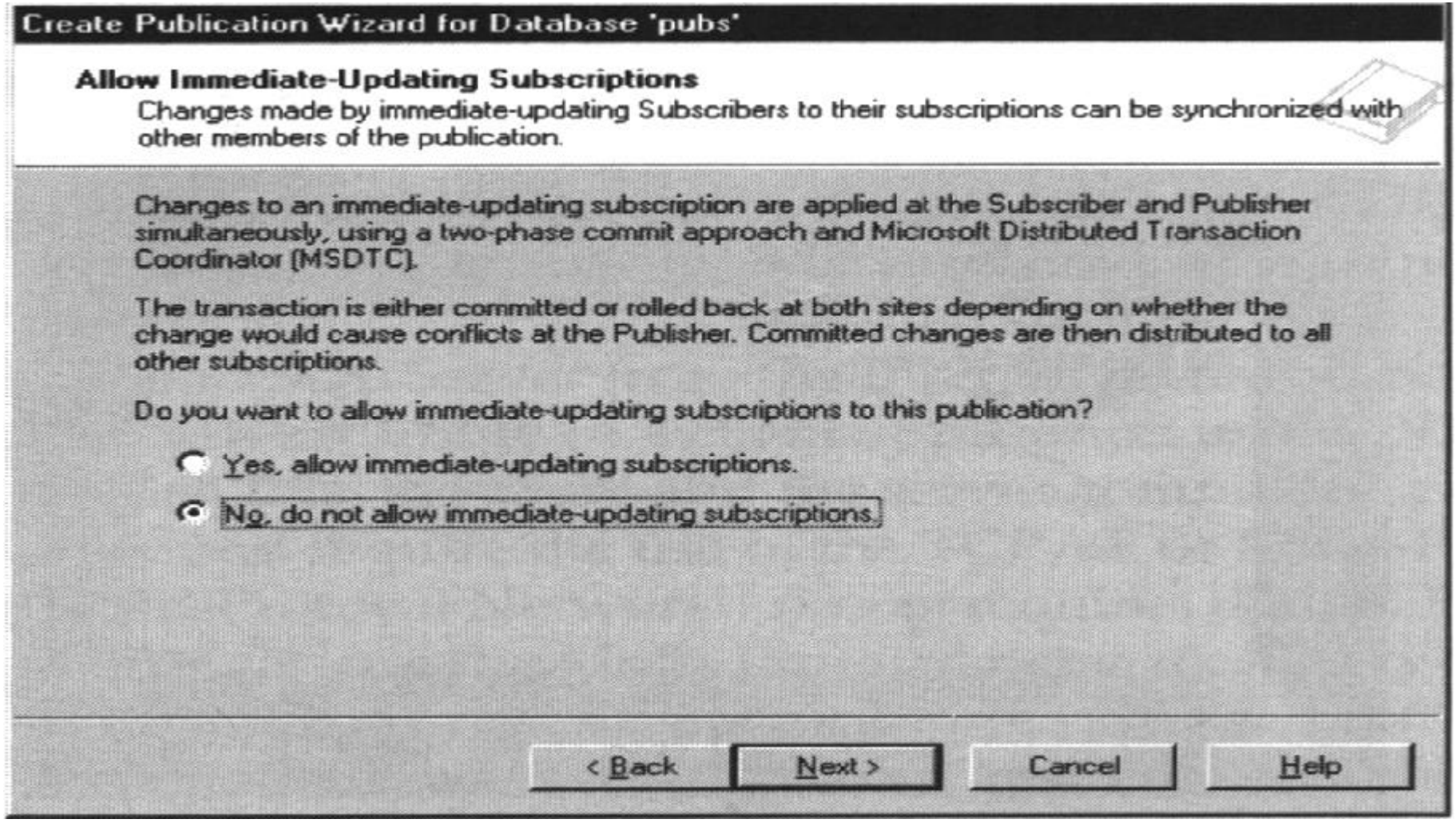


图 9.6 允许更新订户选 Yes

SQL Server 的各个不同部分为创建“Immediate Updating Subscriber 而协同工作。Micriosoft Distributed Transaction Coordinator 负责两阶段提交，而触发器在该 Subscriber 捕获更新的资料并把它们送到该 Publisher。更新资料是通过使

用 NOT FOR REPLICATION 子句在触发器生成时，将该 Subscriber 处那些实际地变化和那些在正常的复制程序中用 Publisher 施加到该 Subscriber 的资料区分开来。这意味 Update,Insert 或 Delete 触发器仅在用户更新 Subscribing 数据库时调用，而在复制事务发生时不调用，该触发器用一个远程存储过程，叫做该 Publisher 上的一个存储过程，如果它成功返回，该事务被提交。

在事务提交之前，它必须通过一系列的检测来看 Publisher 上的那个数据会发生什么。在 Publisher 上如果在来自 Subscriber 的事务被应用之前数据被更新，那么该事务退回重来。SQL Server 查看在文章上是否有时间戳。如果有，它比较时间戳，如果相同的数据在 Publisher 上被改变过就不应用事务。如果没有时间戳，一行比较便发生。在 Subscriber 上数据被改变之前，用数据观察，如果在 Publisher 和 Subscriber 各行是相同的，那么该事务在 Subscriber 试图更新数据之前，Publisher 曾经被更新过，该 Subscriber 的更新资料将不被应用。这种情况下 Publisher 是胜利者。只有当文章有时间戳的情况下，在 Publisher 复制期间，一个叫做 Loopback detection 的进程保存从复制返回到 Subscriber 的 Subscriber 更新资料。

9.6 推、拉订阅

将数据移到一个 Subscriber 有两种选择：一种是 Publisher 将数据推到 Subscriber，另一种是 Subscriber 从 Publisher 拉回数据。一个出版物可用这两种

方法移动数据。

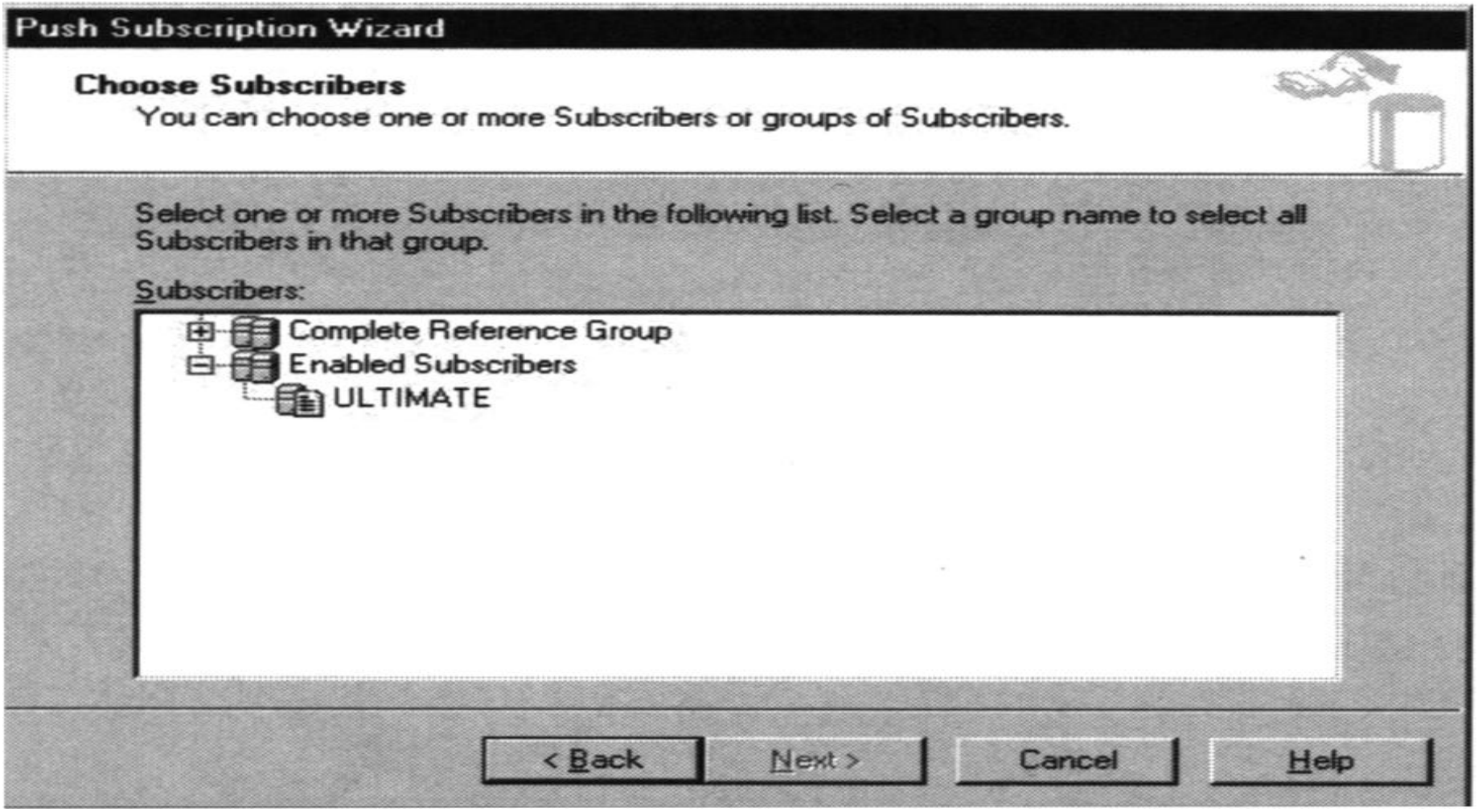


图 9.7Push Subscriber Wizard

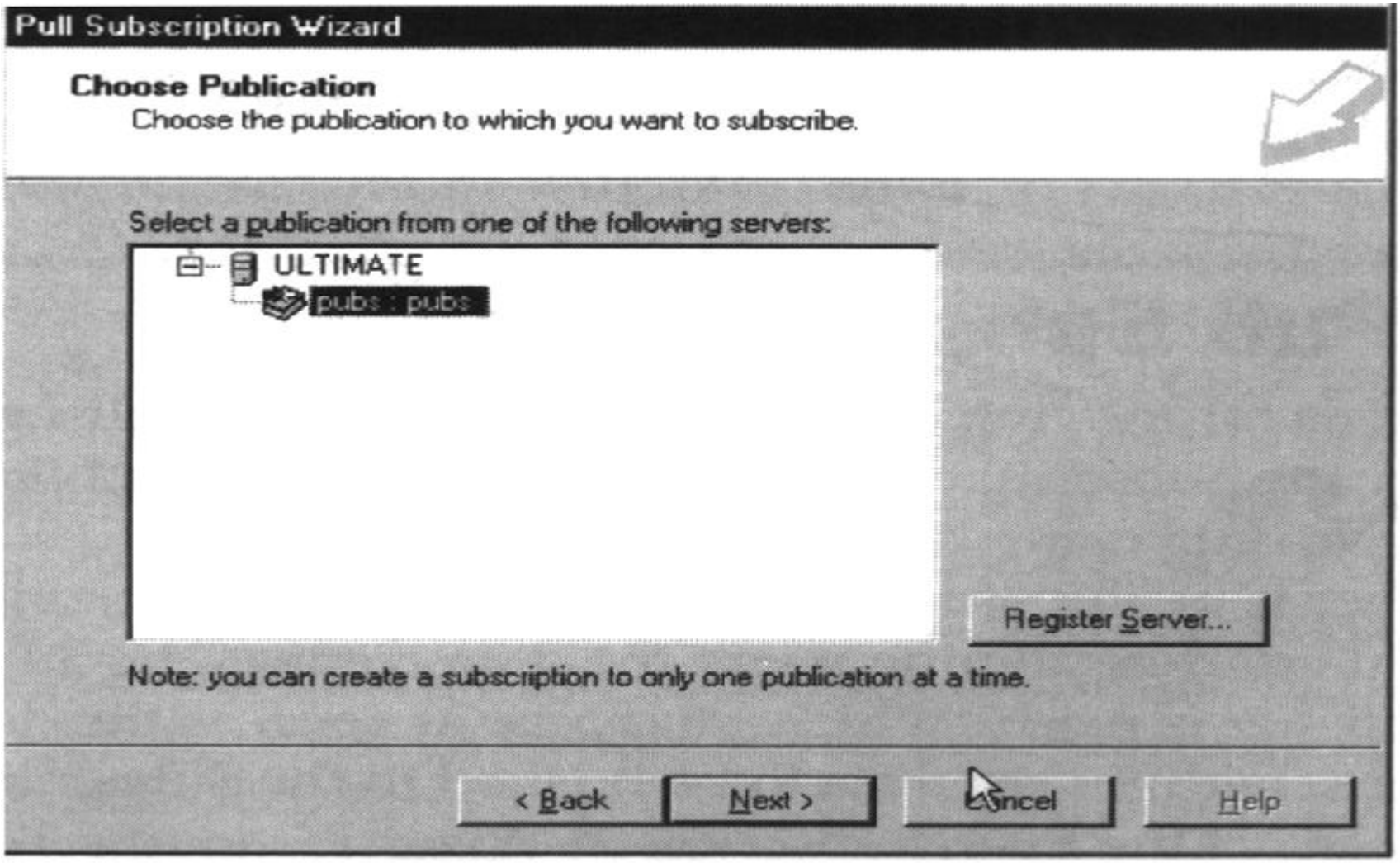


图 9.8Pull Subscription Wizard

当 Publisher **推数据到** Subscriber **时，当数据到来时，** Subscriber **与找数据或确定数据无关。当** Publisher **准备好时，** Subscriber **便获得数据，只要该数据被**

改变或 Publisher 调度将要推到 Subscriber 的数据，这一过程随时会发生，为了实现这种复制,可从 Enterprise Manager 中 Tools／Wizards 中选择 Push Subscriber Wizard。Push Subscriber Wizard 如图 9.7 所示。推式订阅最好用在变化必须在 Subscriber 上能看见的情况下，只要它们发生在 Publisher 上，并且应用要求安全。在一个调度的基础上，变化也能被推到 Subscribers。

在 Internet 网上有许多 Subscribers 时，Subscriber 从 Publisher 上拉回数据是最好的方法。使用这种方法,可在 Enterprise Manager 上从 Tools 菜单中选择 PULL Subscription Wizard，如图 9.8 所示。Subscriber 能在方便的时候，在由 Subscriber 设置的调度上从 Publisher 处接收变化。这种方法没有推式方法安全。

9.7 因特网复制增强

Microsoft SQL Server 使用因特网已使从 Publisher 到 Subscriber 获得数据变得容易了。有几个应用规则。如果有防火墙，Publisher 和 Distributor 必须在防火墙的同一端。如果没有防火壁，Publisher 和 Distributor 必须有一个直接的网络连接。

从因特网连接到 Microsoft SQL Server 是容易的。它能从 SQL Server Query Analyzer 实用程序中完成或从使用 ODBC 或 DB Library 委托的应用中完成。为了完成这个任务，该委托和该服务器计算机必须与 Internet 相连并且需要使用 TCP／IP 或 Mulit-Protocol Network Libraries。使用 Multi-Protocol Network Library

需要 TCP／IP 支持。管理者能用 Multi-Protocol 网络库与一个或所有用户的 SQL Server 建立一个加密连接。TCP／IP Windows Sockets,NWLink IPX／SPX 以及 Named Pipes 在 Multi-Protocol Net-Library 之下是受到支持的。在一个或多个 Interprocess Communication 之间的通讯受到 Windows NT 的支持。Multi-Protocol Net-Library 利用 Windows NT 的 Remote Procedure Call，提供了完整的安全性。对 Multi-Protocol 网络库来说，简单地使用服务器的名字来确定服务器的地址。

9.8 匿名拉回订阅

匿名拉回订阅就是在出版器和分配器上没有储存关于每个订阅器的信息的一次订购。如果服务器对出版器不是一个订阅器，并且该出版器被设置为无记名订购，则拉购向导将为订户创造一次无记名的订购，该订购需要订户发出订购并从出版器处拉回数据。这能由 Internet 的订户使用。

用复制分配代理程序已经使 Internet 出版物的复制增强了，并且便于从源服务器(出版者)到目标服务器(订户)数据拷贝的分配。这是开始设置在一个或更多的服务器上，然后在调度或实时或者要求对远程服务器分配数据。这个特性的优点是订户在一个远程区域内能被物理定位，任何数据的改变都能象在源服务器上所做的一样到达订户。为了让许多 Internet 数据定阅器，一个匿名订购的想法提出了。这是一种新建的功能性的特殊的拉回订购。过去，数据是从源服务器上推出；现在它能从订户服务器拉回。匿名订户的信息是不存储在出版者处

的，订户为了保持无记名订购同步，成为有责任的一方。然而对大量的订户存储信息的开销是不希望的。

用于创造一次匿名订购的步骤与用于创造一次拉式订购的步骤一样。另外，因为匿名的订购和服务器不能作为订户在出版者处登记，出版物必是激活的。使用 Replication Distribution Aqent Utility 增加一次匿名订购。

注意:复制分配代理实用程序是一个向导，意味着它是帮助用户执行以前需要编程的任务的一个用户界面。该向导建立并且开始将保留在数据库事务和快照作业表格中移动到订购服务器的复制任务。

9.9 数 据 加 密

多协议网络库有能力从 SQL Server 服务器中加密数据并送到 SQL Server，这在 Internet 环境中显然有用。加密能为一个或所有用户设置。这是一个使用 Microsoft SQL Server 客户网络实用程序所需的进程。

注意:使用客户网络实用程序和服务器网络实用程序(定位在 SQL 服务器 7 程序组)。

在计算机上运行客户(运行客户网络实用程序)和 SQL Server(运行服务器网络实用程序)，在客户和在 SQL 服务器上运行，有以下一些操作步骤：

1. 在 Microsoft SQL Server 程序组中，启动 SQL 客户网络实用程序(或者如果你在 SQL Server 上，启动 Server Network Utility)。

2. 单击 Genral 标签。

3. 在 Genral 标签中按下 Add 按钮，选择 Multi-Protocol，在服务器别名中输入其名字，如图 9.9 所示。

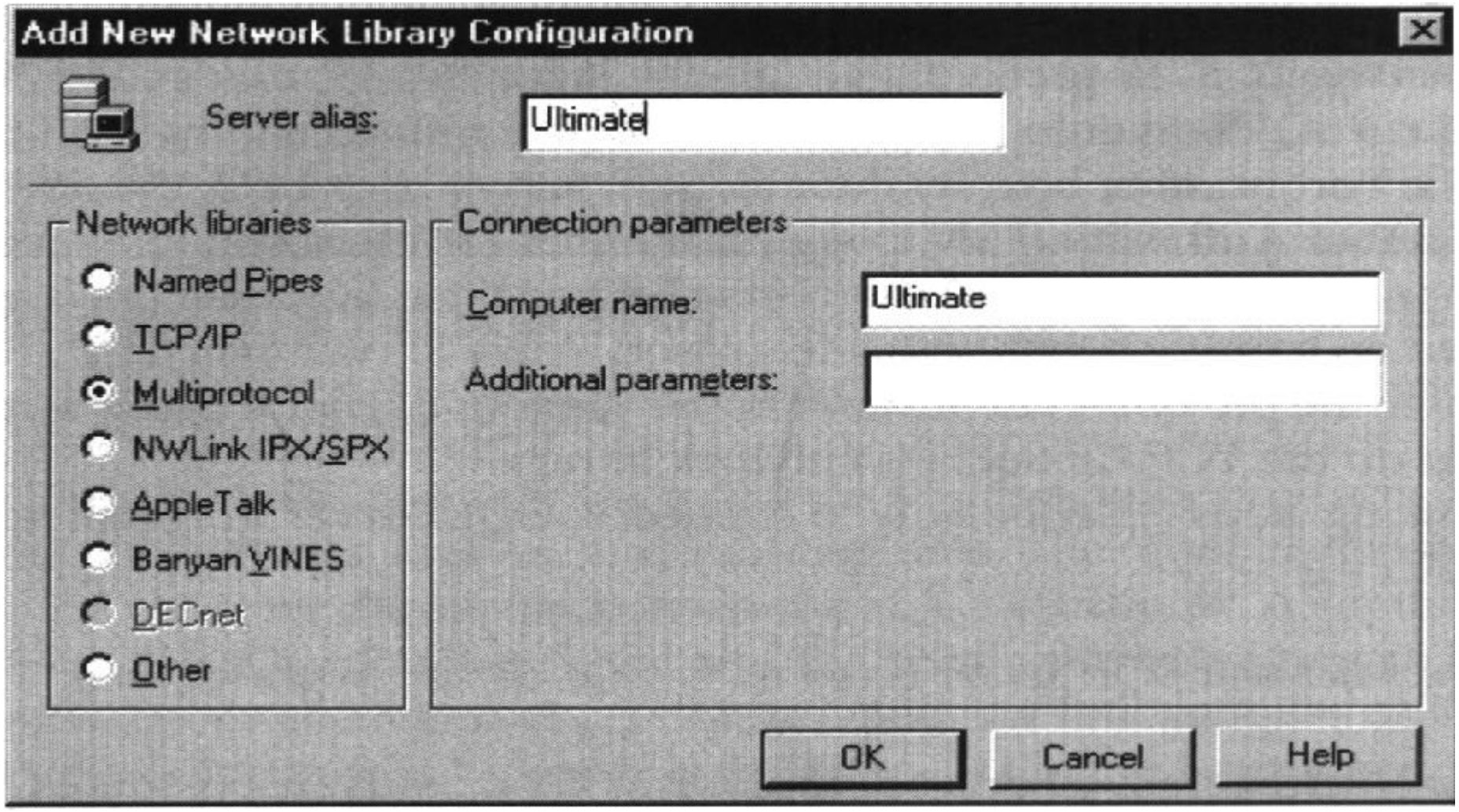


图 9.9Client Network Utility | Add New Network Protocol

4. 选择确定按钮。

5. 在 Server Network Utility 中，在 SQL Server Network Utility 上有一个附加的步骤。编辑 Multi-Protocol 选项并选择 Enable Multi-Protocol 加密复选框。

9.10 在因特网上的保密：微软代理服务器

为在因特网上确保秘密，Microsoft SQL Server 是通过微软代理服务器连接的，该代理服务器与用户确认的 Windows NT 服务器紧密结合。代理服务器是一种程序，提供在 Microsoft SQL Server 数据库上存储的数据的安全的访问。代理服务器是由控制监听端口提供安全的访问的独立程序，可通过锁定访问 IP 地址、域或用户的范围，让不需要的用户不能访问该端口。TCP／IP Sockets 也能与 Microsoft 代理服务器一起组合。这允许 SQL Server 使用标准的 Windows Sockets 作为 IPC 方法通过 TCP／IP 协议通讯。Windows 95 默认使用 TCP／IP Sockets。如果设置 SQL Server 在 TCP／IP 上监听，你必须在端口数字框中输入 TCP／IP 端口数。当从 TCP／IP Sockets 委托方接受连接时，在默认情况下，SQL Server 使用 1433，官方的因特网分配数字协会(IANA)插口数字为 Microsoft SQL Server，此时接受来自 TCP／IP 的插口客户。当 SQL Server 被设置安装到代理服务器上监听，在 TCP/IP 插口之上使用 Microsoft 代理服务器，当你设置 TCP／IP Sockets 联网库时，你必须在 Remote Winsock 代理地址方框中输入代理服务器地址。下列步骤描述如何使用 Microsoft Proxy Server 连接 SQL Server 与 TCP

／IP：

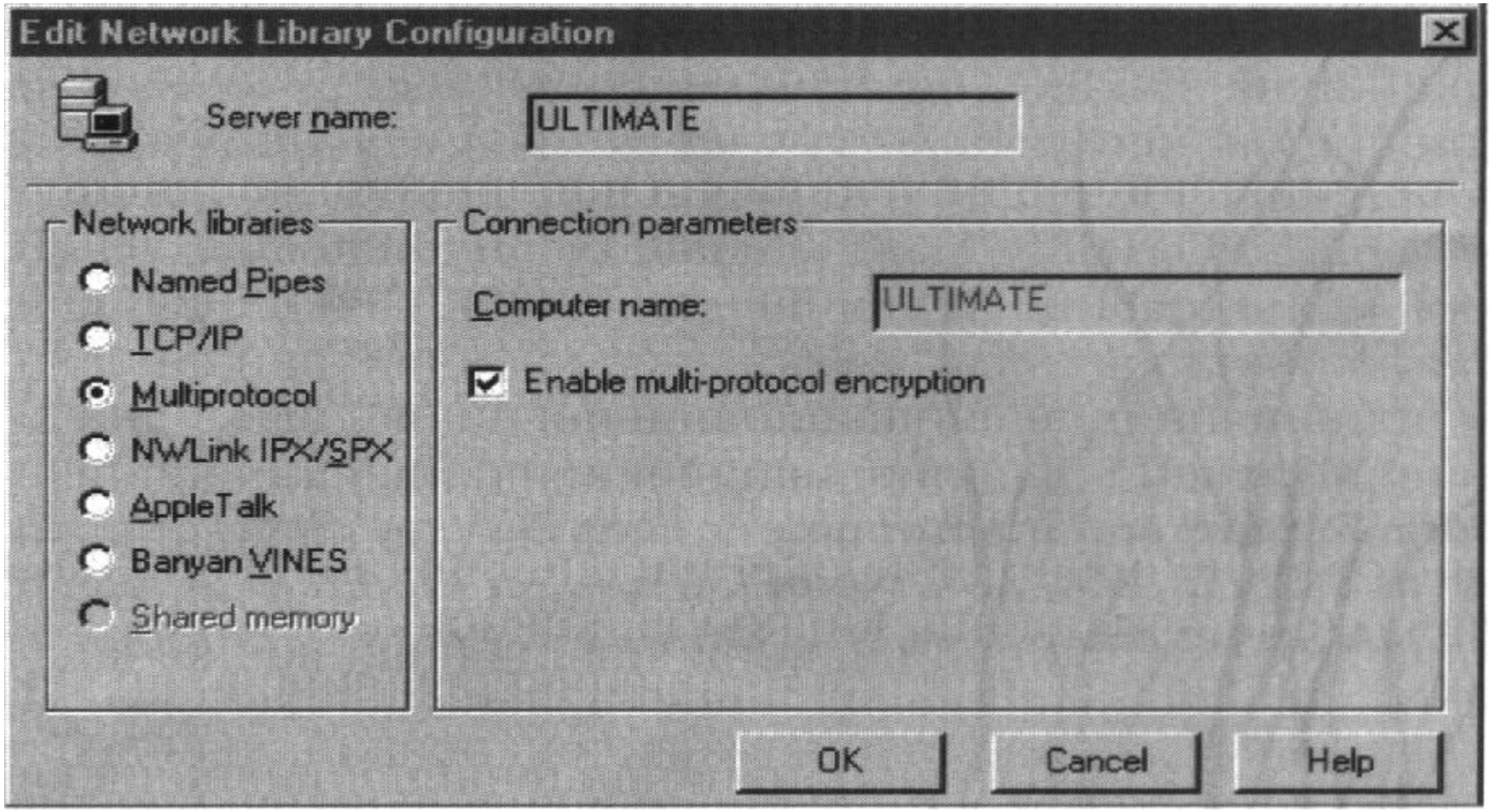


图 9.10Server Network Utility|Edit

注意:在 SQL Server **程序组中使用微软** SQL Server Setup。

1. **在** Microsoft SQL Server Setup **中，用现有的** SQL Server **选择** Work，**然**

后选择 Next。

2. 选择 Change Network Support 选项并选择 Next 按钮。

3. 单击 TCP／IP Sockets 方框。

4. 在 Port Number 方框中，输入端口数(SQL Server 默认值是 1433)。

5. 在 Remote Winsock 代理地址方框中，输入节点名或代理服务器的 IP 地址。

6. 选 Finish，然后选 Install Now。

7. 停止和启动 SQL Server。

我知道人们在使用了代理服务器和 TCP／IP 网络库的远程 Winsock 代理客户部分后，对结果是满意的，说性能与慢速 WAN 相似，他们在网上处于从新西兰到华盛顿。通过 Remote Winsock 远程监听，为代理服务器定义局部地址表格(LAT)，以便听的节点地址在 LAT 输入项的范围外面。

个人也能使用 Microsoft SAN 服务器，与 PC 系统和 Microsoft SQL Server 一起组装 IBM 主机和 SA／400 计算机和数据。Microsoft SNA Server 支持 Internet 和 Internet 访问。每台 PC 使用一个局域网(LAN)协议，例如 TCP／IP，IPX／SPX，NetBEUI,Banyan VINES 或 Apple Talk 连接到 Microsoft SNA Server 计算机，然后用 IMB SNA 协议连接到主机。

SQL Server Data Transformation Service 提供引入引出功能和利用 OLE DB-based 体系结构，从多种不同种类源中转换数据。OLE DB 是 Internet 图像的一个重要的部分，并且为在因特网上转换和发表 HTML 数据提供支持。

9.11 多机种环境数据复制服务

非 Microsoft SQL Server 第三方数据库和数据源能成为出版者。SQL-DMO,Visual Basic,C 和 C＋＋程序设计能用来使 Microsoft SQL Server 复制服务对第三方产品可用。SQL-DMO 是用来创建出版物、文章和订购的。另外的程序语言被用来在 SQL 分配器上存储事务处理。在这完成以后，SQL 分配代理程序、企业管理器能被用来提交事务处理和监视复制。你能够在你的 Microsoft SQL Server 上找到 Visual Basic SQL-DMO 的样本，在＼samples＼ sqlrepl＼ samppub ＼sqldmo 目录中，用 Samples＼Sqdred＼Samppub＼repdist 目录中一个样本 C 程序安装 CD。复制分配界面是允许 SQL-DMO 和编程语言与 Microsoft SQL Server 复制结构交互的界面。复制分配器界面在分配数据库中存储该复制的事务。复制分配器界面使用 OLE DB 连接和对象。使用 Visual C＋＋和 OLE 库可以访问这些对象。

9.12 创建复制脚本

一旦在你选择 Enterprise Manager 中的 Tools|Replication|Generate Replication Scripts 完成复制方案后，如果不得不从头重建复制方案，你可创建一个 Transact-SQL 脚本以便使用(参阅图 9.11)。从 Enterprise Manager 简单运行重建

复制方案后创建的脚本。

有了 Microsoft SQL Server 7，使用因特网复制数据是从未有过的容易了。从加密技术提供复制脚本的每一步骤都简化了。

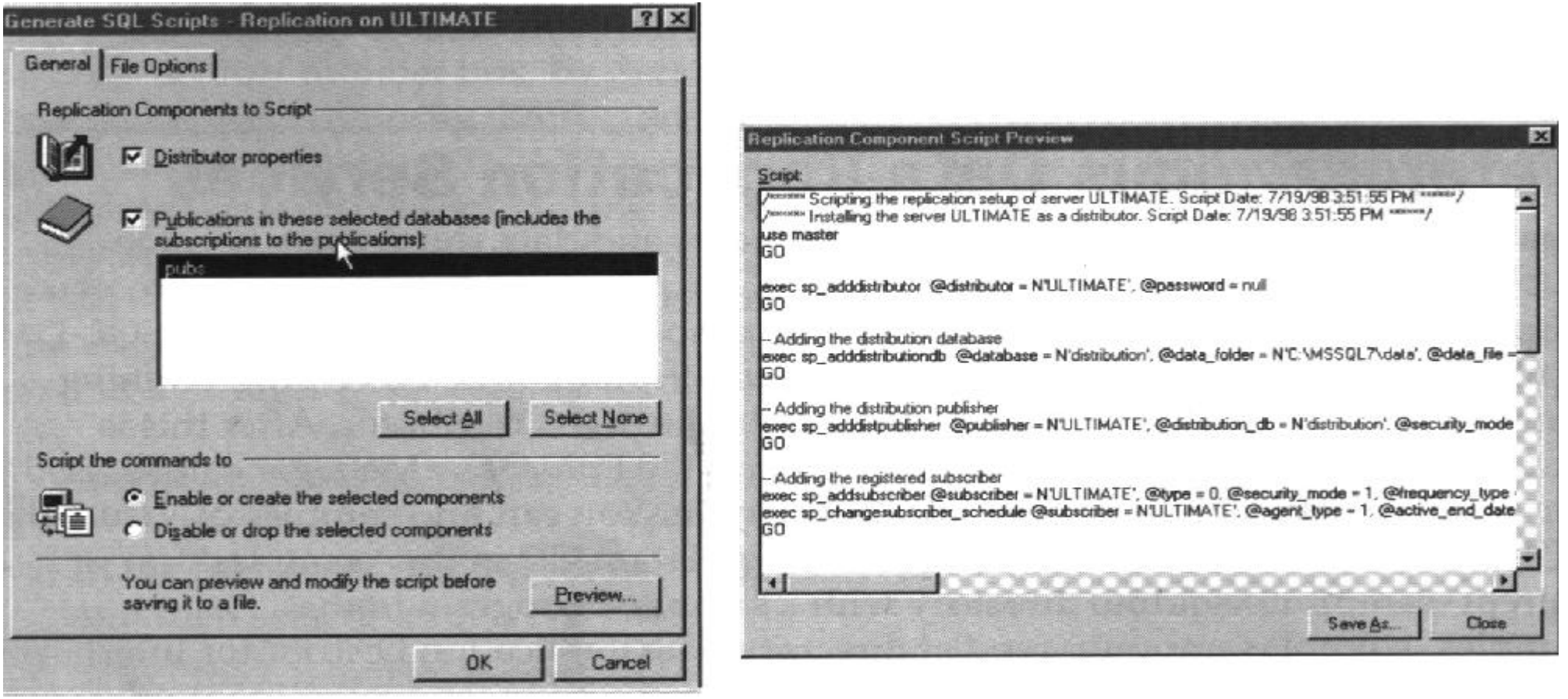


图 9.11 为重建复制方案创建脚本

第 10 章 简化数据库管理员的任务

Microsoft 的目标之一是降低包括 Microsoft SQL Server 的 Microsoft 产品线所有者的全部费用。他们完成这个目标的方法之一是用企业管理器简化与数据库维护有关的任务，使其数据库管理直接并且容易。不放弃工作以外的任何数据库，然而却能够使它们高速和高效地执行数据库维护任务。这一章介绍简化 Microsoft SQL Server 数据库管理员角色的一些增强功能。

- 自动和动态配置(Automatic and Dynamic Configuration)
- 内存(Memory)
- 锁定(Locks)
- 数据库文件自动扩展(Database Files Expand Automatically)
- 自动增大 TempDB 和启动重置(Autogrow TempDB and Startup Reset)
- 当恢复时数据库自动创建和改变(Automatic Database Creation and Alteration When Restoring)
- Transact SQL 版本兼容性级别(Transact SQL Version Compatibility Levels)
- 配置选项的简化(Simplification of Configuration Options)
- 动态结构管理(Dynamic Schema Management)
- 删除列(Drop a Column)

· 修改视图、触发器和存储过程(Alter a View,Trigger,Procedure)

· UNIQUE IDENTIFIER 数据类型和全局唯一标识符(UNIQUE IDENTIFIER Datatype and the Globally Unique Identifier(GUID)

10.1 自动和动态配置

在 SQL Server 7 中有一些自动和动态设置的服务配置选项，其行为叫自我调节。以前这些选项不得不手工地设置，它们保持一样的值直到数据库管理员手工地再次改变配置。用 SQL Server 7 服务器时经常地自动地改变。

10.1.1 内存

Microsoft SQL Server 现在能动态地使用内存。内存自动地设置、增强并且收缩取决于在 NT 服务器上使用什么。如果内存服务器配置选项设置为零(0)，内存是自动地和动态地分配到 SQL Server；如果该内存选项是设置为一非零设置，SQL Server 得到的内存不超过该数量。设置内存使用，在企业管理器中展开服务器组，选择一服务器，在它上面正确按选择属性，选择 Memory 标签(参阅图 10.1，SQL 服务器性能对话框的 Memory 标签)。

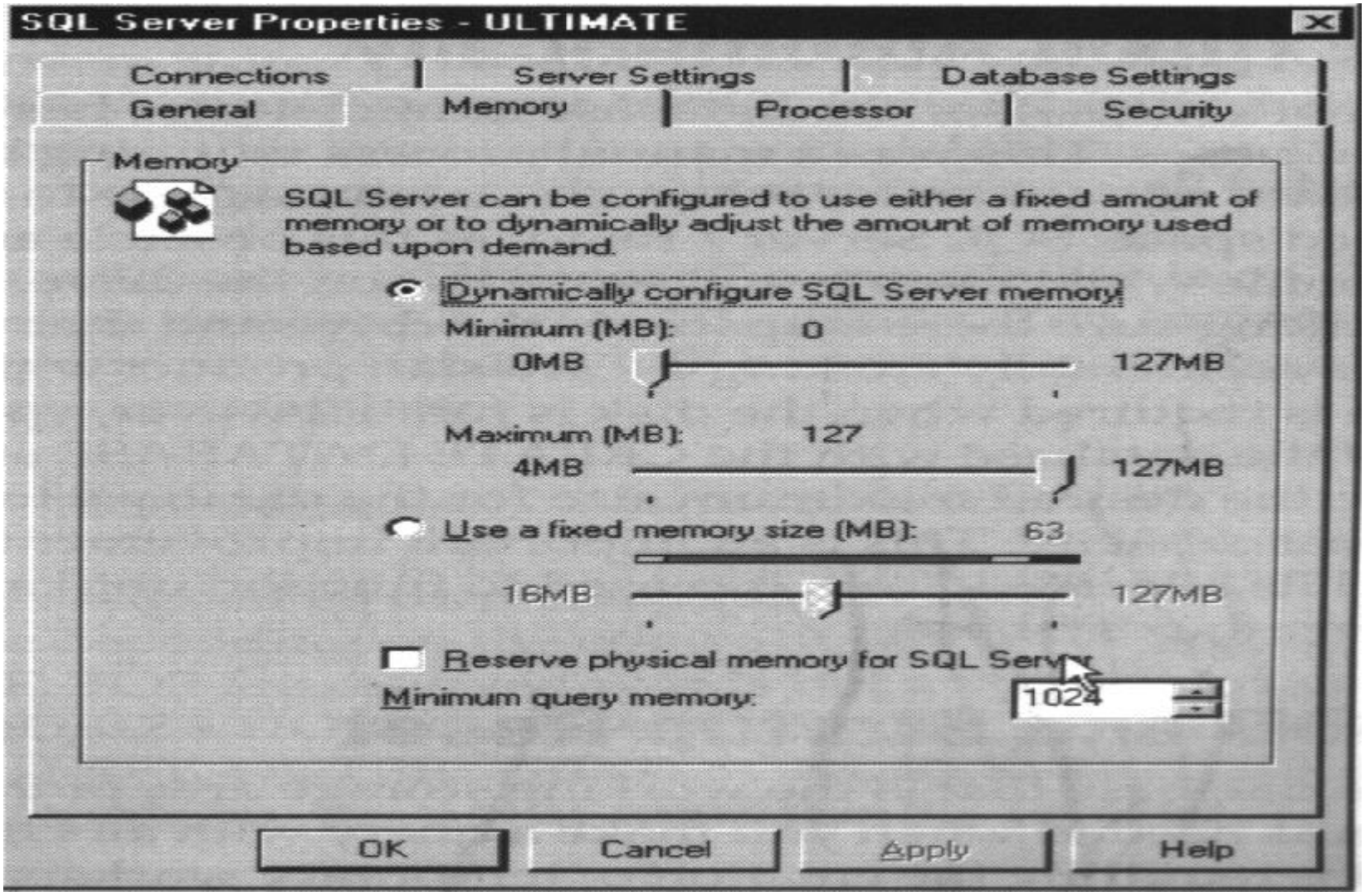


图 10.1SQL Server **属性对话框的内存面板**

懒惰的作者扫描该系统来决定物理内存有多少是空的。目的是保持 NT **页面调度但是将内存给出** SQL Server**。懒惰的作者的程序始终保持在** 5MB **中的** 200K

以内是空的。如果在那里没有 5MB 是空的，它将减少用于 SQL Server 内存的数量，把它给 NT。当更多的内存变得可用时，懒惰作者将再次增加 SQL Server 缓冲区高速缓冲器并且使 SQL Server 具有 5MB(加或减 200K)可用物理内存。

10.1.2 锁定(Locks)

可用于锁定配置的最大值是默认值为零。在该设置中，SQL Server 自动并且动态地将该锁定内存分配成每锁 96 字节。

在 SQL Server 高速缓冲器中,不到 40%的内存将被用于锁定,包括 SQL Server 内存分配的 2%初始分配。

如果希望自己设定锁定选项，必须首先使用 SP 设置系统存储过程来设定另一个服务器配置选项，Show Advanced Options 设为 1(真)。可以在查询分析器窗口里执行这个任务，或在命令行使用 OSQL 实用程序。

```
EXEC sp_configure ' show advanced options' ,1
GO
RECONFIGURE WITH OVERRIDE
GO
EXEC sp_configure ' locks' ,yourintegervalue
GO
```

注意:查询分析器能够通过选择在分层结构树中服务器，然后在企业管理器工具菜单中被调用。

10.1.3 数据库文件自动扩展

传统地说，数据库管理员的一个重要任务是管理数据库的使用空间。当数据库发展到能使用全部分配空间时，这时需要一个有见识的数据库管理员的干预。当驱动器上留有空间的时候，SQL Server 7 删除这个障碍并自动地扩展数据库文件。现在唯一的问题成为确定在驱动器上有空间，并且在偶然生成一笛卡儿产品以及插入查询以后你不用插入百万行。当磁盘已满时，需要人工干预。然而，可以用 CREATE DATABASE 语句变量控制该数据库的最大增长和最大规模。为使驱动器避免充满过去的容量，将该数据库设置到理想的最大规模。这是一个重要的改进；过去，当数据库达到它的分配极限时，它停止工作，直到一个 DBA 通过扩展数据库来解决问题。

10.1.4 自动增长 TempDB 和启动重置

tempdb 是一个系统数据库，该系统数据库是数据库的工作存储区域。现在，当通过 ORDER BY 句子做一次查询或其它复杂查询时，需要创建临时工作区表格时，该数据库在需要时可以增长。然而对 tempdb 的分配空间不停止分配，但是，当 SQL Server 停止和启动时，该数据库返回到它原有的规模。

10.1.5 当恢复时数据库自动创建和改变

当发出一个 RESTORE 命令时，有一个新的变量：REPLACE(替换)。这个变量意味着 SQL Server 将毁掉该数据库并根据备份集合中的详细说明再创建它。如果不使用 REPLACE(替换)选项，该数据库在下列条件下将不还原：

- 数据库已经在服务器存在
- 数据库名称与备份集合数据库名称不同
- 在数据库中文件的集合与包含在备份集合中的数据库文件集合不同

在恢复进程中，用微软 SQL Server 是识别不出文件的大小不同的,而且当恢复发生时，它们是不被检查的。

10.1.6Transact SQL 版本兼容性级别

存储过程的 sp_db cmptlevel 兼容性级别系统是设置兼容姓级别的关键。在已将数据库升级到 SQL Server 7 以后(参阅 13 章“升级 SQL Server”)，用户数据库的兼容性级别能被设置到升级之前的位置，应用程序将起到升级前的作用。

系统数据库将有一个 SQL Server 7 的兼容性级别。对 model 系统数据库，兼容性级别初始设置为 70。现在你可将它改为 60 或 65，其后在该 model 系统数据库中，所有的新数据库都将以你设置的兼容性级别创建。

sp_dbcmptlevel [databasename,compatibilitylevel]

每个数据库关键字的改变取决于兼容性级别。对 70 来说，BACKUP，PERCENT，RESTORE，TOP 是保留的词，但不在低一级的兼容性级别中。对 65 来说，AUTHORIZATION，CASCADE，CROSS，DISTRIBUTED，ESCAPE，FULL，INNER，JOIN，LEFT，OUTER，PRIVILEGES，RESTRICT，RIGHT，SCHEMA，WORK 都是保留的词，但不在兼容级别为 60 之中。

10.1.7 配置选项的简化

在 Sybase Legacy 中，许多服务器配置选项不能使用，并且关于这些选项所做工作的文本也是缺少的。下面一些配置选项已经从 SQL Server 7 中删除：

- 备份缓冲区规模
- 过程缓存
- 备份线程
- RA 缓存碰撞极限
- 释放缓冲区
- RA 缓存错过限制
- 噪声干扰存储桶
- RA 延迟
- LE 最大阈值
- RA 预先读取
- LE 最小阈值
- 每线程的 RA 槽
- LE 阈值百分比
- RA 工作器线程
- 日志记录暂停
- 恢复标记位
- 最大懒惰存入 IO
- 暂停远程控制
- 打开数据库
- SMP 并行性

以下列出的各项服务器配置选项在 SQL Server 7 中是有效的：

- affinity 掩码
- 网络包大小
- 允许更新
- 打开对象
- 对应的阈值成本
- 优先升级
- 游标阈值
- 查询调速器费用限制
- 默认语言
- 查询等待
- 默认排列顺序 id
- 恢复时间间隔
- 填充度(%)
- 远程访问
- 索引创建内存(KB)
- 远程登录超时选项
- 缓冲器中的语言
- 远程处理转换
- 全文本中间语言
- 远程超时查询
- 轻便合并
- 资源超时
- 锁定
- 扫描启动程序
- IO 最大同步
- 设定工作集大小
- 最大平行
- 显示高级选项
- 最大服务器内存
- 旋转计数器
- 最大文本记录规模
- 时间限制
- 最大工作线程
- Unicode 比较样式
- 媒质保留
- Unicode 地点 id
- 查询最小内存
- 用户连接
- 最小服务器内存
- 用户选项
- 嵌套触发器

在 SQL Server 上更新的服务器配置选项提供一个更清晰的调整机会的列表。储存程序的 sp_configure 系统能在 master 系统数据库中执行，更改配置选项的值。

sp_configure [configurationoption,value]

有些服务器配置选项在 Enterprise Manager 中，以一系列的屏幕提供，这些屏幕，你可以通过选择 SQL Server，从 Enterprise Manager 分层结构树中右击并选择 Properties 来访问。General 标签显示服务器信息。

Memory 标签决定 SQL Server 是否动态地和自动地分配内存或允许用户设定内存。这与内存服务器配置选项相对应。Processor 标签允许你设置最大的工作线程、推前优先级、处理器数以及考虑平行执行查询的最小查询平面阈值标记。Security 标签决定认证方式和 SQL Server 启动说明以及非管理员在使用 XP-Cmdshell 执行命令时使用的 SQL Agent CmdExec 程序的说明。

Connections 标签设定服务器最大同时连接用户配置选项，默认连接选项，远程超时查询，以及分布事务的增量(MTS)。Setting 标签包含有嵌套触发器开关，允许更新，以及默认语言。SQL Mail 设置也在该屏幕上。有些选项被认为是 Advanced Option。这些不在上述屏幕的选项必须在使用 sp-Configure 系统存储过程之后设定，该系统储存的程序可设置该服务器配置选项。

```
sp_configure ' Show Advanced Options ',1
GO
RECONFIGURE WITH OVERRIDE
GO
```

使用 RECONEIGURE WITH OVERRIDE 选项，能够立刻更改被认为的动态的服务器选项的配置，不用关闭和启动 SQL Server。否则，该 SQL Server 必须停机和重新启动来更改运行时的值。

10.2 动态结构管理

能容易地改变一个数据库结构的能力对提高数据库管理效率，使数据库管理更直观起重要的作用。删除列可以从其它相关联的产品中获得的能力多年来已经从 SQL Server 中消失。现在这些能力完全包含在 SQL Server 7 中同时，这种能力还可以改变一个视图，触发器或存储过程。

10.2.1 删除列

ALTER TABLE Transact-SQL 命令现在已经有能力删除列。DROP COLUMN 增加到 ALTER TABLE 中后使之成为可能。

ALTER TABLE tablename DROP COLUMN columnname

10.2.2 修改视图、触发器和存储过程

对 SQL Server 来说,ALTER VIEW，ALTER TRIGGER 以及 ALTER PROCEDURE 是新的 Transact-SQL 指令。这对数据库管理员来说是有用的，因为，在这些指令执行的时候，允许依赖存储过程并且触发器不受影响。在以前的版本中，视图、触发器和存储过程必须被丢掉并且重新建立。然后，许可必须重新施加并且依赖存储过程和触发器必须重新编译。

10.2.3 唯一标识符数据类型和全局唯一标识符

唯一标识符数据类型可以解决这样一个问题，即在处理多于一个表格的所有表格所有行中，有一列需要是唯一性的标识。现在两表格的联合仍然产生具有全局唯一性的标识符的行。在先前的版本中 Transact-SQL 程序设计是要求产生该全局性唯一的 ID 的。对唯一标识符数据类型允许的唯一操作是平等的关系比较运算(=,<,和 >)以及 IS NOT NULL 和 IS NULL 运算符。有一称为 NEWID() 的新建函数，能被用来初始化一个唯一标识符列或变量。用全局唯一标识符(GUID)初始化一变量：

```
DECLARE @var_1 uniqueidentifier
SET @var_1=NEWID()
```

对全局唯一标识符初化一列：

```
UPDATE tablename
SET columnname=NEWID()
WHERE columnname=columnvalue
```

总的来说，所有这些新的特征，贡献出来的作用使数据库管理员的管理更容易、更有效、更令人愉快。