SQL Server 2000 学习教程

操晓春　编著

北 京 大 学 出 版 社

北　京

需要全本请访问 www.ertongbook.com

内 容 简 介

本书以简洁、通俗的语言，全面而系统地介绍了 Microsoft SQL Server 2000 中文版的功能和操作、使用方法。

Microsoft SQL Server 2000 是微软最新一代数据库产品，它建立在先进的、可扩展的 SQL Server 7.0 的基础之上，秉承了 SQL Server 7.0 的高性能，并在创建可伸缩电子商务、在线商务和数据仓储解决方案等方面进行了显著的改进和提高，成为一种真正意义上的关系型数据库管理与分析系统。Microsoft SQL Server 2000 代表着下一代 Microsoft .NET Enterprise Servers（企业服务器）数据库的发展趋势。

本书图文并茂、通俗易懂、深入浅出，既适合初学电脑的用户阅读，也适合公司、企业、机关、部队、科研院所、大专院校的电脑操作人员以及广大家用电脑用户使用，还可以作为 Microsoft SQL Server 2000 的自学教材和各类培训班的培训教材。

图书在版编目（CIP）数据

SQL Server 2000 学习教程/操晓春编著．—北京：北京大学出版社，2001.11
ISBN 7-301-05305-3

Ⅰ．S… Ⅱ．操… Ⅲ．关系数据库—数据库管理系统，SQL Server 2000—教材 Ⅳ．TP311.138

中国版本图书馆 CIP 数据核字（2001）第 076918 号

书　　名：SQL Server 2000 学习教程
著作责任者：操晓春
责 任 编 辑：段志刚　黄庆生
标 准 书 号：ISBN 7-301-05305-3/TP·0626
出　版　者：北京大学出版社
地　　址：北京市海淀区中关村北京大学校内　100871
网　　址：http://cbs.pku.edu.cn
电 子 信 箱：xxjs@pup.pku.edu.cn
印　刷　者：
发　行　者：北京大学出版社
经　销　者：新华书店
787 毫米×1092 毫米　16 开本　14.75 印张　378 千字
2001 年 11 月第 1 版　2001 年 11 月第 1 次印刷
定　　价：23.00 元

前　言

Microsoft SQL Server 2000 是微软最新一代数据库产品，早在 SQL Server 2000 正式投放市场以前，SQL Server 7.0 就已经是在 Web 上存储数据的最流行的数据库了。自 SQL Server 7.0 发布以来，由于其优良的性能、可伸缩性、可管理性、可编程性等优点，它已成为众多客户关系管理（CRM）、商业智能（BI）、企业资源规划（ERP）以及其他商业应用程序供应商和客户的首选数据库。此外，SQL Server 7.0 作为 Internet 数据库已取得巨大成功。

作为继 SQL Server 7.0 之后的又一个重要版本，Microsoft SQL Server 2000 建立在先进的、可扩展的 SQL Server 7.0 的基础之上。并对性能、可靠性、质量以及易用性进行了进一步扩展。SQL Server 2000 中包含许多新特性，这些特性使其成为针对电子商务、数据仓库和在线商务解决方案的卓越的数据库平台。其增强的特性包括对丰富的扩展标记语言（XML）的支持、综合分析服务以及便捷的数据库管理。

特别值得一提的是，Microsoft SQL Server 2000 在最重要的速度方面也有了质的飞跃。微软公司在 2001 年 4 月份宣布 Microsoft SQL Server 2000 再创数据库速度记录："微软 SQL Server 2000 企业版和 Windows 2000 Datacenter Server 在新近的包括 TPC-C 在内的系列工业测试中显示了其强大的可扩展性，再度有力地对传统昂贵的企业系统供应商提出挑战。在新近的 TPC-C 测试中以每分钟处理 688220 笔交易打破 SQL 2000 先前创造的 505302 TPC-C 的世界纪录。这样在前 6 名 TPC-C 记录中 SQL 2000 占据了 5 名。值得注意的是，前 6 名纪录都运行在 Windows 2000 平台之上。同时 SQL 2000 还在 SAP 公司的 mySAP SD 基准测试中创造了支持 14400 个用户的世界纪录。"[1]

总之，建立在先进的、可扩展的 SQL Server 7.0 的基础之上的 SQL Server 2000 代表着下一代 Microsoft .NET Enterprise Servers（企业服务器）数据库的发展趋势，是创建可伸缩电子商务、在线商务和数据仓储解决方案的最佳选择。

本书是一本系统介绍 Microsoft SQL Server 2000 中文版使用的教材，本书面向应用，所以书中尽量不解释 SQL Server 2000 的原理性知识，而主要介绍其操作及各项管理。同时，本书面向初学者，所以文中运用大量图片和实例，便于读者理解。而对于某些实践中很少用到并且非常复杂的功能，本书只简单提及，以免读者浪费过多时间而无所获。这样即使对 SQL Server 完全陌生的读者在阅读完本书之后也可以掌握 SQL Server 2000 中文版的使用，并且利用它进行基本的开发工作。

全书共由 13 章和 3 个附录部分组成，具体内容包括：Transact-SQL 语言基础；Microsoft SQL Server 2000 的新增功能特性，以及它的各个中文版本的功能；SQL Server 的安装；介绍 SQL Server 的一些最基本的操作；数据库管理；表管理；索引管理；视图管理；存储过程管理；触发器管理；数据转换服务；SQL Server 2000 和 ASP 的结合；对 XML 和 Internet 的支

---

[1] 参见微软公司官方网站 http://www.microsoft.com/china/press/2001/0402b.asp。

持；附录部分主要列出了数据库的基础知识、Microsoft 为 SQL Server 以后版本预保留的关键字以及 SQL Server 2000 中的系统表及其结构信息。

本书以简洁、通俗的语言，全面而系统地介绍了 Microsoft SQL Server 2000 中文版的功能和操作、使用方法；本书图文并茂、通俗易懂、深入浅出，既适合初学电脑的用户阅读，也适合公司、企业、机关、部队、科研院所、大专院校的电脑操作人员以及广大家用电脑用户使用，还可以作为 Microsoft SQL Server 2000 的自学教材和各类培训班的培训教材。

本书由操晓春主编，另外，韩永鹏、魏宝晶、朱明峰、袁怡翔、肖依永、邱亚希、苗泽纯、孟鹃、喻湘宁、张晟、盖宇、黄涛、朱凯、刘梓超、翟丽君、裴浴、叶萱、白致铭、黎君雪、谢宏威、杨继运等也参加了本书的编写工作。.

由于时间仓促，而且编者经验和水平有限，疏漏之处在所难免，恳请广大读者批评指正，我们将不胜感激。

编　　者

2001 年 11 月

目　录

第 1 章　Transact-SQL 语言基础……1

1.1　数据控制语言……1
1.1.1　GRANT 语句……2
1.1.2　REVOKE 语句……3
1.1.3　DENY 语句……3
1.2　数据定义语言……4
1.3　数据操作语言……5
1.3.1　SELECT 语句……5
1.3.2　INSERT 语句……6
1.3.3　UPDATE 语句……7
1.3.4　DELETE 语句……9
1.4　Transact-SQL 的其他元素……11
1.4.1　函数……11
1.4.2　注释……17
1.4.3　保留关键字……18
1.4.4　控制流语言……19
1.4.5　运算符……19

第 2 章　揭开 SQL Server 2000 的面纱……20

2.1　SQL Server 2000 版本介绍……20
2.2　SQL Server 2000 新特性列表……21
2.3　SQL Server 2000 特性介绍……23
2.3.1　Internet 集成……24
2.3.2　高度可伸缩性和可靠性……24
2.3.3　易于安装、部署和使用……25
2.3.4　企业级数据库功能……25

第 3 章　SQL Server 2000 的安装……27

3.1　准备安装 SQL Server 2000……27
3.1.1　安装 SQL Server 2000 的硬件和软件要求……28
3.1.2　升级现有的 SQL Server 安装……29
3.1.3　创建 SQL Server 账户……30
3.1.4　排序规则选项……30

需要全本请访问 www.ertongbook.com

3.1.5 网络协议……30
3.2 开始安装……31
第 4 章 开始运行 SQL Server 2000……37
4.1 SQL Server 服务管理器……37
4.2 连接到 SQL Server 服务器……38
4.3 SQL Server 2000 服务器配置……40
4.3.1 常规（General）选项卡……40
4.3.2 内存（Memory）选项卡……42
4.3.3 处理器（Processor）选项卡……43
4.3.4 安全性（Security）选项卡……43
4.3.5 连接（Connections）选项卡……44
4.3.6 服务器设置（Server Settings）选项卡……44
4.3.7 数据库设置（Database Settings）选项卡……45
4.3.8 复制（Copy）选项卡……45
4.3.9 Active Directory 选项卡……45
4.4 企业管理器简介……46
4.5 查询分析器简介……47
4.6 SQL Server 2000 向导工具介绍……48
第 5 章 数据库管理……51
5.1 数据库理论基础……51
5.1.1 数据库对象介绍……52
5.1.2 数据库存储结构……52
5.2 数据库的创建……53
5.2.1 使用创建数据库向导创建数据库……54
5.2.2 使用 SQL Server 企业管理器创建数据库……56
5.2.3 使用 Transact-SQL 语句创建数据库……58
5.3 数据库的选项设置……61
5.4 数据库文件的管理……62
5.4.1 数据库文件大小的增长……62
5.4.2 压缩数据库文件大小……65
5.5 数据库的删除……67
5.5.1 在 SQL Server 企业管理器中删除数据库……67
5.5.2 使用 Transact-SQL 语句删除数据库……68
5.6 数据库的备份……69
5.6.1 使用创建数据库备份向导备份数据库……69
5.6.2 使用 SQL Server 企业管理器备份数据库……72
5.6.3 使用 Transact-SQL 语句备份数据库……73

5.7 还原数据库备份 ……74
5.8 其他数据库管理操作 ……75
5.8.1 数据库维护计划 ……76
5.8.2 生成 SQL 脚本 ……77
5.8.3 分离和附加数据库 ……79
第 5 章 数据库管理 ……51
5.1 数据库理论基础 ……51
5.1.1 数据库对象介绍 ……52
5.1.2 数据库存储结构 ……52
5.2 数据库的创建 ……53
5.2.1 使用创建数据库向导创建数据库 ……54
5.2.2 使用 SQL Server 企业管理器创建数据库 ……56
5.2.3 使用 Transact-SQL 语句创建数据库 ……58
5.3 数据库的选项设置 ……61
5.4 数据库文件的管理 ……62
5.4.1 数据库文件大小的增长 ……62
5.4.2 压缩数据库文件大小 ……65
5.5 数据库的删除 ……67
5.5.1 在 SQL Server 企业管理器中删除数据库 ……67
5.5.2 使用 Transact-SQL 语句删除数据库 ……68
5.6 数据库的备份 ……69
5.6.1 使用创建数据库备份向导备份数据库 ……69
5.6.2 使用 SQL Server 企业管理器备份数据库 ……72
5.6.3 使用 Transact-SQL 语句备份数据库 ……73
5.7 还原数据库备份 ……74
5.8 其他数据库管理操作 ……75
5.8.1 数据库维护计划 ……76
5.8.2 生成 SQL 脚本 ……77
5.8.3 分离和附加数据库 ……79
第 6 章 表管理 ……82
6.1 创建表 ……82
6.1.1 设计表 ……82
6.1.2 使用 SQL Server 企业管理器创建表 ……83
6.1.3 使用 Transact-SQL 语句创建表 ……87
6.2 修改表 ……92
6.2.1 注意事项 ……92
6.2.2 使用 SQL Server 企业管理器修改表 ……93
6.2.3 使用 Transact-SQL 语句修改表 ……94

6.3 数据检索 —— SELECT 语句 ····· 98
6.3.1 基本检索 ····· 99
6.3.2 WHERE 子句 ····· 101
6.3.3 ORDER BY 子句 ····· 105
6.3.4 GROUP BY 子句和 HAVING 子句 ····· 106
6.3.5 COMPUTE 子句 ····· 108
6.3.6 联接查询 ····· 109
6.3.7 内联接 —— INNER JOIN 运算符 ····· 111
6.3.8 外联接 —— OUTER JOIN ····· 113
6.3.9 交叉联接 —— CROSS JOIN ····· 115
6.3.10 子查询 ····· 116
6.3.11 UNION 运算符 ····· 119
6.4 数据插入 —— INSERT 语句 ····· 121
6.4.1 INSERT 语句的基本用法 ····· 121
6.4.2 使用 INSERT...SELECT 插入数据 ····· 121
6.4.3 DEFAULT 选项 ····· 122
6.4.4 使用 SELECT INTO 插入数据 ····· 122
6.5 数据修改 —— UPDATE 语句 ····· 122
6.6 数据删除 —— DELETE 语句 ····· 123
6.7 在 SQL Server 企业管理器中处理数据 ····· 124
6.7.1 查询窗口简介 ····· 124
6.7.2 关系图窗格 ····· 126
6.7.3 网格窗格 ····· 126
6.7.4 SQL 窗格 ····· 127
6.7.5 结果窗格 ····· 127
6.7.6 设计查询 ····· 128
6.8 删除表 ····· 129
6.8.1 使用 SQL Server 企业管理器删除表 ····· 129
6.8.2 使用 Transact-SQL 语句删除表 ····· 130

第 7 章 索引管理 ····· 131

7.1 概述 ····· 131
7.2 创建索引 ····· 132
7.2.1 使用创建索引向导创建索引 ····· 133
7.2.2 使用 Transact-SQL 语句创建索引 ····· 135
7.3 索引的维护及删除 ····· 137
7.3.1 DBCC SHOWCONTIG 语句 ····· 137
7.3.2 DBCC INDEXDEFRAG 语句 ····· 137
7.3.3 DROP INDEX 语句 ····· 138
7.4 索引服务在 IIS 中的应用 ····· 138

第 8 章 视图管理 140
8.1 概述 140
8.2 创建视图 141
8.2.1 使用创建视图向导创建视图 141
8.2.2 使用 SQL Server 企业管理器创建视图 143
8.2.3 使用 Transact-SQL 语句创建视图 144
8.3 修改视图 145
8.3.1 使用 SQL Server 企业管理器修改视图 145
8.3.2 使用 Transact-SQL 语句修改视图 146
8.4 删除视图 147
8.5 在视图中修改数据 148
第 9 章 存储过程管理 149
9.1 概述 149
9.2 创建存储过程 150
9.2.1 使用创建存储过程向导创建存储过程 150
9.2.2 使用 SQL Server 企业管理器创建存储过程 152
9.2.3 使用 Transact-SQL 语句创建存储过程 152
9.3 执行存储过程 154
9.4 修改存储过程 155
9.4.1 使用 SQL Server 企业管理器修改存储过程 155
9.4.2 使用 Transact-SQL 语句修改存储过程 156
9.5 删除存储过程 156
9.5.1 使用 SQL Server 企业管理器删除存储过程 156
9.5.2 使用 Transact-SQL 语句删除存储过程 157
第 10 章 触发器管理 158
10.1 概述 158
10.2 创建触发器 159
10.2.1 使用 SQL Server 企业管理器创建触发器 159
10.2.2 使用 Transact-SQL 语句创建触发器 160
10.3 修改触发器 162
10.4 删除触发器 162
第 11 章 数据转换服务 164
11.1 概述 164
11.2 数据转换服务导入/导出向导 165
第 12 章 SQL Server 2000 和 ASP 的结合 168
12.1 ASP 介绍 168
12.1.1 概述 168

12.1.2 ASP 的运行 169
12.2 ASP 的创建 170
12.2.1 .asp 文件 170
12.2.2 ASP 脚本 171
12.2.3 其他指令 172
12.2.4 脚本中的空白字符 173
12.3 ASP 的设计原则 173
12.3.1 采用标准方法 173
12.3.2 封装业务逻辑 175
12.3.3 尽晚获取资源，尽早释放资源 175
12.3.4 使用数据库维护复杂的状态 176
12.3.5 使用 Server.CreateObject 创建对象 176
12.3.6 提供丰富的疑难解答信息 176
12.3.7 测试性能、可伸缩性和可靠性 177
12.3.8 增加隔离性 177
12.4 SQL Server 2000 提供的数据库开发接口 178
12.4.1 DB-Library 178
12.4.2 ODBC（开放式数据库连接） 179
12.4.3 OLE DB 179
12.4.4 DAO（数据访问对象） 180
12.4.5 RDO（远程数据对象） 180
12.4.6 ADO（活动数据对象） 180
12.5 在 ASP 中使用 ADO 对象 181
12.5.1 ADO 对象模型 181
12.5.2 打开连接 182
12.5.3 访问数据 183
12.5.4 一个实际的例子 185
第 13 章 对 XML 和 Internet 的支持 187
13.1 XML 简介 187
13.1.1 XML 的起源 187
13.1.2 XML 的优点 188
13.1.3 XML 的应用 189
13.1.4 XML 受到的支持 190
13.2 使用 HTTP 访问 SQL Server 191
13.2.1 建虚拟目录 192
13.2.2 SQL Server 支持的 HTTP 访问方式 196
13.2.3 直接在 URL 中指定 SQL 语句 197
13.2.4 使用模板进行访问 197
13.3 检索和编写 XML 数据 198

13.3.1 FOR XML 子句 ····· 198
13.3.2 使用 OPENXML 编写 XML ····· 199
13.3.3 使用 XPath 进行检索 ····· 201
13.4 SQL Server OLE DB 提供程序对 XML 的支持 ····· 201
附录 A 数据库知识简介 ····· 203
A.1 数据库系统模型和结构 ····· 203
A.1.1 层次模型 ····· 203
A.1.2 网状模型 ····· 204
A.1.3 关系模型 ····· 204
A.2 数据库管理系统（DBMS） ····· 205
A.3 关系数据库理论基础 ····· 205
A.3.1 主键 ····· 206
A.3.2 数据库设计中可能的问题 ····· 206
A.3.3 数据依赖 ····· 206
A.3.4 范式 ····· 207
附录 B 为将来而保留的关键字 ····· 208
附录 C SQL Server 2000 的系统表 ····· 210
C.1 系统数据库 master 中的系统表 ····· 210
C.2 系统数据库 model 中的系统表 ····· 217

需要全本请访问 www.ertongbook.com

第 1 章　Transact-SQL 语言基础

众所周知，结构化查询语言（SQL）在关系数据库中占有极其重要的地位。作为一种关系数据库，Microsoft SQL Server 以 SQL 语言作为基础，其名称也由此而来。Transact-SQL 语言则是 Microsoft SQL Server 中使用的扩展了的 SQL 语言。

SQL 源自二十世纪七十年代 Boyce 和 Chamberlin 提出的 Structured English QUEry Language（结构化英语式查询语言），简称 SEQUEL 语言，后来经改进并更名为 Structured Query Language（结构化查询语言），即现在的 SQL 语言。1982 年 ANSI 采纳 SQL 为数据库系统的工业标准并进行推广。Microsoft 对最新的 ANSI SQL-92 进行了扩展，制订了专用于 SQL Server 的 Transact-SQL 语言。

Transact-SQL 语言由以下四个部分组成：

- λ　数据控制语言（DCL）：进行安全性管理。包括 GRANT、DENY 和 REVOKE 等。
- λ　数据定义语言（DDL）：执行数据库任务，创建数据库及其对象。包括 CREATE、ALTER 和 DROP 等。
- λ　数据操作语言（DML）：操作数据库中各对象。包括 SELECT、INSERT、UPDATE 和 DELETE 等。
- λ　附加的语言元素：Transact-SQL 语句的附加语言元素。包括变量、运算符、函数、流程控制语句和注释等。这部分不是 ANSI SQL-92 的内容。

其中数据操作语言是 Transact-SQL 的主要内容。SELECT 语句（检索）、UPDATE 语句（更改）、INSERT 语句（添加）和 DELETE 语句（删除）等四种数据操作语言构成了 Transact-SQL 语言的核心。

本章主要内容有：

数据控制语言

数据定义语言

数据操作语言

Transact-SQL 的其他元素

1.1　数据控制语言

数据控制语言用于管理数据库的安全性，包括用户权限的授予、拒绝和删除。数据控制语言的语句有 GRANT 语句、DENY 语句和 REVOKE 语句。

1.1.1 GRANT 语句

GRANT 语句用来向数据库中用户或角色授予执行特定的 Transact-SQL 语句或处理数据库中特定对象的权限。

授予语句权限的 GRANT 语句语法形式为：

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

授予对象权限的 GRANT 语句语法形式为：

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

其中主要参数及其意义如下：

- λ ALL：表示授予所有可用的权限。
- λ statement：是被授予权限的语句。
- λ security_account：是权限将应用的安全账户。安全账户可以是：Microsoft SQL Server 用户、SQL Server 角色、Microsoft Windows NT 用户、Windows NT 组。
- λ permission：是当前授予的对象权限。

例如下面的例子授予 public 角色 SELECT 权限；授予用户 Mary、John 和 Tom INSERT、UPDATE 和 DELETE 权限。

```
USE pubs[1]
GO
GRANT SELECT
ON authors
TO public
GO
GRANT INSERT, UPDATE, DELETE
ON authors
TO Mary, John, Tom
```

[1] USE 语句用来指定数据库。USE pubs 表示下面的 Transact-SQL 语句在数据库 pubs 中执行，直到下一个 USE 语句出现。

```
GO
```

1.1.2　REVOKE 语句

REVOKE 语句与 GRANT 语句恰恰相反，它删除授予用户或角色的权限。与 GRANT 语句相对应，REVOKE 语句也有语句权限和对象权限两种操作。

注意：REVOKE 语句只删除授予该用户或角色的权限，但不影响该用户或角色作为其他角色成员而继承的权限。

删除语句权限的 REVOKE **语句语法形式为：**

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

删除对象权限的 REVOKE **语句语法形式为：**

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

其中各参数意义与 GRANT **语句中相同。**

例如下面的例子删除前一小节利用 GRANT **语句授予的权限。**

```
USE pubs
GO
REVOKE SELECT
ON authors
FROM public
GO
REVOKE INSERT, UPDATE, DELETE
ON authors
FROM Mary, John, Tom
GO
```

1.1.3　DENY 语句

DENY 语句在 REVOKE 语句的基础上更进一步，它不仅删除已授予的权限，而且禁止

该用户或角色从其他角色中继承相应的权限。

拒绝语句权限的 DENY 语句语法形式为：

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

拒绝对象权限的 DENY 语句语法形式为：

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

其中各参数意义与 GRANT 语句中相同。

例如下面的例子首先将 SELECT 权限授予 public 角色。然后，拒绝用户 Mary、John 和 Tom 包括 SELECT、INSERT、UPDATE 和 DELETE 在内的所有权限。最后，用户 Mary、John 和 Tom 将没有任何操作 authors 表的权限。

```
USE pubs
GO
GRANT SELECT
ON authors
TO public
GO
DENY SELECT, INSERT, UPDATE, DELETE
ON authors
TO Mary, John, Tom
```

1.2 数据定义语言

数据定义语言用来定义数据库中的各个对象，包括各种数据库对象的创建、修改和删除。数据定义语言的语句有 CREATE 语句、ALTER 语句和 DROP 语句。

Microsoft SQL Server 中的各种数据库对象，如数据库、表、视图、触发器、存储过程、规则和缺省等，都是用 CREATE 语句创建，用 ALTER 语句修改，用 DROP 语句删除。但随具体对象的不同，这些语句差别较大，所以不在此统一介绍，而是在后面章节中当介绍各种具体对象的管理时分别说明。

下面给出一个创建和删除数据库的例子供读者参考。

使用 CREATE DATABASE **语句创建数据库：**

```
CREATE DATABASE Students
ON
    PRIMARY (NAME = Student_Data,
    FILENAME = 'C:\MSSQL\DATA\Student.mdf',
    SIZE = 10MB,
    MAXSIZE = UNLIMITED,
    FILEGROWTH = 10%)
LOG ON
    (NAME = Student_Log,
    FILENAME = 'C:\MSSQL\DATA\Student.ldf',
    SIZE = 3MB,
    MAXSIZE = 15MB,
    FILEGROWTH = 1MB)
```

使用 DROP TABLE **语句删除数据库：**

```
DROP TABLE students.dbo.Students
```

1.3　数据操作语言

数据操作语言用来操作数据库中的数据，包括数据的检索、插入、修改和删除。具体语句有 SELECT **语句、**INSERT **语句、**UPDATE **语句和** DELETE **语句。**

这四个语句是 SQL **语言的核心内容。掌握这四个语句的使用是学习** SQL **的重点。**

1.3.1　SELECT 语句

SELECT **语句从数据库中检索数据，它是使用最广泛的** SQL **语句。**SELECT **语句的语法形式如下：**

```
SELECT statement ::=
    < query_expression >
    [ ORDER BY { order_by_expression | column_position [ ASC | DESC ] } [ ,...n ]       ]
    [ COMPUTE
        { { AVG | COUNT | MAX | MIN | SUM } ( expression ) } [ ,...n ]
        [ BY expression [ ,...n ] ]
    ]
    [ FOR { BROWSE | XML { RAW | AUTO | EXPLICIT }
                [ , XMLDATA ]
                [ , ELEMENTS ]
                [ , BINARY base64 ]
```

```
            }
    ]
        [ OPTION ( < query_hint > [ ,...n ]) ]

    < query expression > ::=
        { < query specification > | ( < query expression > ) }
        [ UNION [ ALL ] < query specification | ( < query expression > ) [...n ] ]

    < query specification > ::=
        SELECT [ ALL | DISTINCT ]
            [ { TOP integer | TOP integer PERCENT } [ WITH TIES ] ]
            < select_list >
        [ INTO new_table ]
        [ FROM { < table_source > } [ ,...n ] ]
        [ WHERE < search_condition > ]
        [ GROUP BY [ ALL ] group_by_expression [ ,...n ]
            [ WITH { CUBE | ROLLUP } ]
        ]
        [HAVING <search_condition>]
```

下面是一个简单的数据检索的例子，它从表 authors **中检索出所有加州的作者。**SELECT **语句的详细介绍及实例请参考第** 6 **章。**

```
SELECT AU_LNAME,CITY
FROM authors
WHERE STAT E = ’C A ’
```

1.3.2 INSERT 语句

INSERT **语句向表或者视图中添加数据。**INSERT **语句的语法形式为：**

```
INSERT [ INTO]
    { table_name WITH ( < table_hint_limited > [ ...n ] )
        | view_name
        | rowset_function_limited
    }
    {     [ ( column_list ) ]
        { VALUES
            ( { DEFAULT | NULL | expression } [ ,...n] )
            | derived_table
            | execute_statement
        }
    }
```

需要全本请访问 www.ertongbook.com