

数据库应用开发技术丛书

Visual C#

数据库高级教程

李律松
陈少刚 等编著
沈 刚



清华大学出版社

数据库应用开发技术丛书

Visual C#数据库高级教程

李律松 陈少刚 沈 刚 等编著

清华大学出版社

北 京

内 容 简 介

本书以专题的形式,讲述了使用 Visual C# .NET 进行数据库应用程序开发的方法和技巧。本书共分为 13 章,主要介绍了数据库的基本知识、结构化查询语言 SQL 和数据库管理系统的使用方法,.NET 平台、C#语言基础和.NET 平台支持的数据库开发环境,ADO.NET 的基本组件和对象,使用 Visual C# .NET 的常用操作,数据库中存储过程、触发器、函数、游标以及它们在 Visual C# .NET 开发环境中的使用方法,事务的相关概念和基本原理,数据库备份和恢复的基本概念,数据复制技术,Visual C# .NET 开发环境中报表的绘制技术,.NET 对 XML 数据源所提供的强大操作,Web 服务的创建及使用 Web 服务访问数据库的技术,数据库应用程序的安全性,以及如何提高数据库应用程序的性能。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

Visual C#数据库高级教程/李律松,陈少刚,沈刚等编著. —北京:清华大学出版社,2005.6
(数据库应用开发技术丛书)

ISBN 7-302-10842-0

I.V… II.①李… ②陈… ③沈… III.C 语言—程序设计—教材 IV.TP312

中国版本图书馆 CIP 数据核字(2005)第 035506 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦
<http://www.tup.com.cn> 邮 编:100084
社 总 机:010-62770175 客户服务:010-62776969

组稿编辑:王 定

文稿编辑:杜一民

封面设计:久久度文化

版式设计:康 博

印 刷 者:北京牛山世兴印刷厂

装 订 者:北京鑫海金澳胶印有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:24.75 字数:571 千字

版 次:2005 年 6 月第 1 版 2005 年 6 月第 1 次印刷

书 号:ISBN 7-302-10842-0/TP·7207

印 数:1~4000

定 价:36.00 元

前 言

21 世纪以信息技术为代表,科学技术突飞猛进。信息技术作为当前人类社会发展最快、应用最广的技术,已经成为重要的生产力要素,广泛推动社会经济的飞速发展。因此,信息化成为当今经济和社会发展的趋势。信息化离不开信息的表示、传递、获取以及存储,而数据库技术作为信息存储的基础,在信息时代有着举足轻重的地位。目前,数据库及其应用程序的开发已经渗透到社会生活的各个方面,例如政府的办公系统、企业的日常管理、电子商务、各类信息管理系统等,成为了计算机应用领域重要的组成部分。

在数据库应用程序开发方面,各类开发语言以及开发工具的竞争日趋激烈。随着 Microsoft Visual Studio .NET 的推出,由于其功能强大,与 Microsoft 的操作系统平台和数据库系统平台能够高度融合,因此开发人员把越来越多的目光投向了该开发环境。Visual Studio .NET 以统一的开发环境把 Visual Basic .NET、Visual C# .NET、Visual C++ .NET 和 Visual J# .NET 集成在一起。其中,C#语言是目前功能最强大、最出众的语言之一,为数据库开发提供了强大的技术支持,使开发人员能够快捷、高效地开发数据库应用程序,因此普遍受到开发人员的青睐。

为了帮助开发人员全面、透彻地掌握 C#数据库编程技术,我们编写了这本《Visual C#数据库高级教程》。全书在介绍数据库基础理论和 .NET 平台数据库开发环境的基础上,以专题的形式,详细讲解使用 Visual C# .NET 进行数据库应用程序开发的方法和技巧。虽然本书介绍的是数据库编程的高级技术,但是本书提供了大量的示例代码和详细操作步骤,使得读者能够比较容易地理解和掌握这些高级技术。

本书共分 13 章。为了方便读者阅读,先简单介绍一下本书的组织结构和主要内容。

第 1 章介绍数据库的基本知识、结构化查询语言 SQL 和数据库管理系统的使用方法。掌握这些基础知识是编写数据库应用程序的必备前提。

第 2 章介绍 .NET 平台、C#语言基础和 .NET 平台支持的数据库开发环境,包括数据处理的相关技术、类库和各种工具。掌握这些知识是使用 Visual C# .NET 开发数据库的基础。

第 3 章介绍 .NET 平台中最重要的数据访问技术——ADO.NET 的基本组件和对象,并通过大量实例来阐述这些组件和对象的使用方法。

第 4 章介绍如何使用 Visual C# .NET 进行常用的数据库操作,包括连接数据库、存取数据、显示数据等。这些基本的操作在一般的数据库编程中会经常使用。

第 5 章介绍数据库中的存储过程和触发器以及它们在 Visual C# .NET 开发环境中的使用方法。

第 6 章介绍数据库中的函数和游标以及它们在 Visual C# .NET 开发环境中的使用方法。

第 7 章介绍事务的相关概念和基本原理,重点讲解事务的实现方法和如何编写使用事务的数据库应用程序。

第8章介绍数据库备份和恢复的基本概念以及如何创建备份和恢复数据库。

第9章介绍数据复制技术，它为分布式数据的存储和处理提供了有力支持。

第10章介绍 Visual C# .NET 开发环境中报表的绘制技术。

第11章介绍 .NET 对 XML 数据源所提供的强大操作能力，同时，也详细介绍了 SQL Server 2000 对 XML 的支持。

第12章介绍 Web 服务的创建及使用 Web 服务访问数据库的技术。

第13章介绍如何保证数据库应用程序的安全性，如何提高数据库应用程序的性能。

读者在阅读本书时，可以通过 <http://www.tupwk.com.cn> 下载本书相关代码或反馈意见，也可以浏览和下载《数据库应用开发技术丛书》其他图书信息。

本书由李律松、陈少刚和沈刚共同执笔编写。此外，麦聪、蓝荣香、王昊亮、喻波、马天一、魏勇、郝荣福、孙明、李大宇、武思宇、牟博超、李彬、付鹏程、高翔、朱丽云、崔凌、张巧玲、李辉、李欣、柏宇、郭强、金春范、程梅、黄霆、钟华、高海峰、王建胜、张浩、刘湘和邵蕴秋等同志在整理材料方面给予了编者很大的帮助，在此，编者对他们表示衷心的感谢。

由于编者水平有限，书中不足之处敬请读者不吝赐教和指正。

编 者

目 录

第 1 章 数据库基础知识..... 1	
1.1 数据库概述..... 1	
1.1.1 数据库系统概述..... 1	
1.1.2 数据模型和数据库系统..... 1	
1.1.3 关系数据库的主要概念..... 2	
1.1.4 主要关系数据库产品概述..... 3	
1.2 关系数据库标准语言 SQL..... 5	
1.2.1 SQL 语言概述..... 5	
1.2.2 数据定义语言 DDL..... 8	
1.2.3 数据操纵语言 DML..... 12	
1.2.4 复杂数据操纵语言..... 18	
1.2.5 数据控制语言 DCL..... 20	
1.3 SQL Server 2000 使用初步..... 22	
1.3.1 数据库的创建和删除..... 23	
1.3.2 数据库表的创建和删除..... 27	
1.3.3 数据库表的查询..... 30	
1.4 本章小结..... 33	
第 2 章 .NET 平台及其数据库开发..... 34	
环境..... 34	
2.1 Microsoft .NET..... 34	
2.1.1 .NET 平台概述..... 34	
2.1.2 .NET Framework..... 35	
2.1.3 安装.NET Framework..... 37	
2.1.4 Visual Studio .NET 开发工具..... 37	
2.2 C#语言..... 42	
2.2.1 C#语言概述..... 43	
2.2.2 C#语言的优点..... 44	
2.2.3 C#语言和其他语言的比较..... 45	
2.2.4 理解命名空间..... 47	
2.2.5 编写和编译 C#程序..... 48	
2.3 .NET 提供的数据库开发环境..... 50	
2.3.1 .NET 常用数据访问技术的介绍..... 50	
2.3.2 选择合适的.NET 数据访问技术..... 53	
2.3.3 .NET 中用于数据处理的命名空间..... 55	
2.3.4 .NET 中可视化数据库工具..... 56	
2.4 本章小结..... 63	
第 3 章 ADO.NET 的组件和对象..... 64	
3.1 ADO.NET 概述..... 64	
3.1.1 ADO.NET 的优点..... 64	
3.1.2 ADO.NET 的结构..... 65	
3.2 .NET Framework 数据提供程序..... 66	
3.2.1 .NET Framework 数据提供程序的种类..... 67	
3.2.2 组成 ADO.NET Framework 数据提供程序的核心对象..... 69	
3.3 ADO.NET DataSet 组件..... 75	
3.3.1 DataSet 的组成和结构..... 75	
3.3.2 DataSet 的分类..... 76	
3.3.3 DataSet 的创建..... 76	
3.4 DataSet 包含的 ADO.NET 对象..... 77	
3.4.1 DataColumn 对象..... 77	
3.4.2 DataRow 对象..... 77	
3.4.3 DataTable 对象..... 79	
3.4.4 Constraint 对象..... 82	
3.4.5 DataRelation 对象..... 83	
3.5 本章小结..... 84	
第 4 章 C#数据库操作的常见任务..... 85	
4.1 数据库的一般操作..... 85	

4.1.1 连接数据库	85	6.2 游标	158
4.1.2 从数据库中查询记录	86	6.2.1 游标分类	159
4.1.3 输入参数执行查询	91	6.2.2 游标函数	160
4.1.4 在数据库中插入记录	92	6.2.3 Transact-SQL 游标	161
4.1.5 更新数据库记录	95	6.2.4 数据库 API 游标	170
4.1.6 删除数据库记录	97	6.3 本章小结	172
4.2 数据的显示方法	98	第7章 事务处理和控制技术	173
4.2.1 Windows 窗体 DataGrid 控件	99	7.1 事务概述	173
4.2.2 Web 窗体的 DataGrid 控件	103	7.1.1 事务的概念	173
4.2.3 DropDownList 控件	107	7.1.2 事务的基本原理	174
4.2.4 ListBox 控件	109	7.1.3 事务的特点	175
4.3 本章小结	111	7.1.4 事务的分类	175
第5章 存储过程和触发器	112	7.2 DBMS 的事务支持	176
5.1 存储过程	112	7.2.1 数据库事务控制语句	176
5.1.1 存储过程的特点	112	7.2.2 在存储过程中使用事务的 示例	179
5.1.2 存储过程的分类	113	7.2.3 数据库事务的模式	180
5.1.3 系统存储过程的使用	114	7.3 ADO.NET 的事务支持	180
5.1.4 用户存储过程的使用	116	7.3.1 ADO.NET 的事务类	180
5.1.5 在 Visual Studio .NET 中使 用存储过程	125	7.3.2 执行 ADO.NET 事务的步骤	181
5.2 触发器	132	7.3.3 使用 SqlTransaction 类进行 事务操作详解	182
5.2.1 触发器概述	132	7.3.4 编写使用事务的数据库应用 程序	185
5.2.2 使用触发器的优点	133	7.4 事务的高级技术	194
5.2.3 创建触发器	133	7.4.1 事务的保存点	194
5.2.4 打开触发器	135	7.4.2 事务的隔离级	196
5.2.5 修改和删除触发器	136	7.4.3 事务锁	198
5.2.6 触发器的嵌套和递归	137	7.4.4 事务的嵌套	199
5.2.7 触发器的应用	138	7.4.5 不允许出现在事务中的 SQL 语句	200
5.3 本章小结	143	7.4.6 提高事务的性能	200
第6章 函数和游标	144	7.5 本章小结	200
6.1 函数	144	第8章 数据库的备份和恢复	201
6.1.1 函数分类	144	8.1 数据库备份和恢复概述	201
6.1.2 数据库内置函数	145	8.1.1 数据库备份概述	201
6.1.3 用户定义函数	150		
6.1.4 在 Visual C# .NET 中使用 函数	154		

8.1.2 数据库恢复概述	204	10.1.1 水晶报表的特点	264
8.2 数据库备份的实现	205	10.1.2 水晶报表的设计环境	265
8.2.1 用 SQL Server 企业管理器 实现数据库备份	205	10.2 报表绘制技术	271
8.2.2 使用 SQL 语句实现数据库 备份	215	10.2.1 创建数据库连接	271
8.3 数据库恢复的实现	220	10.2.2 创建报表	275
8.3.1 用 SQL Server 企业管理器 实现数据库恢复	221	10.2.3 绘制图表	283
8.3.2 使用 SQL 语句实现数据库 恢复	221	10.2.4 交叉表	286
8.4 通过 Visual Studio .NET 编程 实现数据库的备份	223	10.2.5 子报表	288
8.4.1 备份存储过程的设计	223	10.2.6 实现程序关键代码	291
8.4.2 备份应用程序的实现	225	10.3 公式编辑与报表打印	292
8.5 本章小结	226	10.3.1 公式编辑	292
第 9 章 数据库的复制	228	10.3.2 报表打印	294
9.1 复制概述	228	10.4 本章小结	295
9.1.1 复制过程中的关键实体	228	第 11 章 XML 数据源的处理技术	296
9.1.2 复制类型	229	11.1 XML 介绍	296
9.1.3 复制代理程序	230	11.1.1 XML 概述	296
9.1.4 复制的拓扑结构	231	11.1.2 XML 文档的基本结构	297
9.2 实现复制的一般过程	233	11.2 在 .NET 中处理 XML 数据	299
9.2.1 配置复制	233	11.2.1 文档对象模型处理 XML 数据	299
9.2.2 发布数据和数据库对象	241	11.2.2 流模型处理 XML 数据	307
9.2.3 确定订阅的类型	242	11.3 使用 SQL Server 的 XML 支持	310
9.2.4 生成初始并应用快照	243	11.3.1 SELECT 语句的 FOR XML 扩展	310
9.2.5 同步数据	244	11.3.2 在 IIS 中配置 SQL XML 支持	318
9.3 事务复制的实现	244	11.4 DataSet 和 XML 的相关处理 技术	324
9.3.1 创建发布	245	11.4.1 DataSet 和 XML 之间的 关系	324
9.3.2 强制订阅	252	11.4.2 DataSet 和 XML 的相互 转换	325
9.3.3 请求订阅	257	11.4.3 DataSet 的 XML 架构	325
9.3.4 查看数据	261	11.5 本章小结	326
9.4 本章小结	263		
第 10 章 水晶报表绘制技术	264		
10.1 水晶报表概述	264		

第 12 章 Web 服务和多层数据库应用程序	328
12.1 Web 服务	328
12.1.1 Web 服务简介	328
12.1.2 SOAP	329
12.1.3 WSDL	331
12.1.4 UDDI	333
12.1.5 .NET 和 Web 服务	333
12.2 创建和使用 Web 服务	334
12.2.1 创建 Web 服务	334
12.2.2 使用 Web 服务	337
12.3 Web 服务和多层数据库应用程序的开发	340
12.3.1 示例概述	340
12.3.2 创建数据层	341
12.3.3 创建业务层	343
12.3.4 创建表示层	353
12.3.5 示例小结	357
12.4 本章小结	358

第 13 章 数据库应用程序的安全和性能	359
13.1 数据库应用程序的安全	359
13.1.1 数据库的安全	359
13.1.2 代码的安全	372
13.2 数据库应用程序的性能	379
13.2.1 优化数据库设计	379
13.2.2 优化查询	380
13.2.3 优化应用程序设计	384
13.2.4 监测服务器的性能瓶颈	385
13.3 本章小结	387

第1章 数据库基础知识

掌握数据库基础知识是编写数据库应用程序的必备前提。数据库基础知识主要包括数据库的基本概念、基本原理和至关重要的结构化查询语言 SQL。

本章首先介绍数据库的基本概念和主要关系数据库产品，然后详细介绍 SQL 语言，并且列举了大量例子，最后介绍了 SQL Server 2000 数据库的基本操作方法。

1.1 数据库概述

数据库技术是计算机科学的一项重要技术。目前，数据库系统已经广泛应用于各个领域，例如管理信息系统、办公自动化系统、图书情报检索系统、票务系统、金融交易系统等等。下面将介绍数据库的基础知识和基础理论。

1.1.1 数据库系统概述

一般来说，数据库系统(DataBase System，简称 DBS)有 4 个重要组成部分：数据库(DataBase，简称 DB)、数据库管理系统(DataBase Management，简称 DBMS)、数据库管理员(DataBase Administrator，简称 DBA)和数据库应用程序。数据库是按一定的结构组织在一起的相关数据的集合；数据库管理系统是负责组织和管理数据信息的程序；数据库管理员是数据库系统不可或缺的重要角色，主要负责数据库的建立、使用和维护工作；数据库应用程序是数据库前端程序，即用于浏览、修改数据的应用程序。

迄今为止，数据管理经历了人工管理、文件系统和数据库系统 3 个阶段，与前两个阶段相比，数据库系统有以下优点：

- 数据的结构化程度高，数据冗余低，有效节省存储空间，易于管理。
- 数据的共享性高，支持大范围内的多个应用程序的同时访问。
- 数据的独立性高，数据由 DBMS 统一管理，与应用程序相对独立，可以支持不同的应用程序访问。

1.1.2 数据模型和数据库系统

数据模型是对现实生活中各种数据特征的抽象。目前，数据库领域中常用的数据模型有 4 种，它们分别是层次模型、网状模型、关系模型、面向对象模型。这 4 种数据模型分别对应 4 种数据库系统，它们分别是层次数据库系统、网状数据库系统、关系数据库系统

和面向对象数据库系统。

层次数据库系统采用层次模型作为数据的组织方式，层次模型采用树形结构来表示各类实体以及实体间的联系。现实世界中许多实体之间的联系呈现出一种自然的层次关系，如企业机构、家族关系等。其典型代表是 IBM 公司的 IMS(Information Management System) 数据库管理系统，这是 1968 年 IBM 公司推出的第一个大型的商用数据库管理系统。

网状数据库系统采用网状模型作为数据的组织方式。现实世界中实体之间的联系更多的是非层次关系的。用层次模型表示非树形结构不是很直接，而网状结构则可以克服这一弊端。

关系模型是目前最重要的一种数据模型，在数据库产品中占主导地位。关系数据库系统采用关系模型作为数据的组织方式。在关系模型中，实体以及实体间的联系都是用关系来表示。在一个给定的现实世界领域中，相应于所有实体及实体之间的联系的集合构成一个关系数据库。

面向对象数据库是数据库技术与面向对象程序设计相结合的产物，是可支持非常规应用领域的新一代数据库系统。面向对象数据库管理系统支持面向对象的数据模型。

基于网状结构和层次结构的数据库属于第一代数据库，渐渐趋于淘汰，而关系数据库和面向对象数据库则是第二代数据库，其中关系数据库的应用非常广泛。数据库系统发展的趋势是面向对象数据库和关系数据库的不断融合。

1.1.3 关系数据库的主要概念

关系数据库的主要概念包括表、字段、记录、键、关系、存储过程等，它们的含意如下。

- 表：数据库可以理解为一个或多个表对象的集合，而表是一个数据库对象，它由记录的集合而成，每一个记录又由若干字段组成。简单地说，表是一种按行与列排列相关信息的逻辑组。
- 字段：表中的每一列称作一个字段，描述了记录所具有的一个属性。字段可以包含字符、字串、数字、图像等。表的结构是由其包含的各种字段所定义的，每个字段描述了表的一个属性值。
- 记录：一条记录是表的一行，同一个数据表中任意两条记录都不能完全相同。
- 视图：视图仅仅是一个保存在数据库中的查询。因此可在查询中执行的大部分操作都可以在视图中执行。
- 键：键是表中的某个(或多个)字段，键可以是惟一的，也可以是非惟一的。惟一键可以被指定为主键，用来惟一标识表行数据。
- 关系：关系是表与表之间的联系。数据库可以由多个表组成，表与表之间可以以不同的方式相互关联。
- 存储过程：存储过程使管理数据库和显示关系数据库数据及其用户信息的工作更为容易。存储过程是 SQL 语句和可选控制流语句的集合，它们存储在一个名称下，

作为一个单元由数据库服务器处理。存储过程会被预编译,以便日后使用,所以使用存储过程能够提高 SQL 语句的执行效率。

1.1.4 主要关系数据库产品概述

现今流行的关系数据库产品有 Access 数据库、SQL Server 数据库、Oracle 数据库和 Sybase 数据库等,下面分别对这些关系数据库产品进行简单介绍。

1. Access 数据库

Access 数据库是 Microsoft Office 的一个组成部分,适用于小型商务应用。Access 数据库中有 6 种对象,即表(Table)、查询(Query)、窗体(Form)、报表(Report)、宏(Macro)和模块(Module)。

(1) 表是数据库的基本对象,是创建其他 5 种对象的基础。表由记录组成,记录由字段组成,表用来存贮数据库的数据,故又称数据表。

(2) 查询可以按索引快速查找到需要的记录,按要求筛选记录并能连接若干个表的字段组成新表。

(3) 窗体提供了一种方便的浏览、输入及更改数据的窗口,还可以创建子窗体显示相关联的表的内容。窗体也称表单。

(4) 报表的功能是将数据库中的数据分类汇总,然后打印出来,以便分析。

(5) 宏相当于 DOS 中的批处理命令,用来自动执行一系列操作。Access 列出了一些常用的操作供用户选择,使用起来十分方便。

(6) 模块的功能与宏类似,但它定义的操作比宏更加精细和复杂,用户可以根据自己的需要编写模块程序。

Access 可以方便地利用 Foxpro 数据库、Excel 电子表格的数据,还可以和 Word 混合使用,打印通用信函或信封。与下面几种产品不同,Access 不是一个 DBMS(数据库管理系统),它的性能一般,只能应付普通的桌面应用程序,无法对企业级的应用提供支持。

2. SQL Server 数据库

SQL Server 是 Microsoft 公司重要的关系数据库管理系统。它最初是由 Microsoft、Sybase 和 Ashton-Tate 3 家公司共同开发的。在 Windows NT 推出后,Microsoft 与 Sybase 在 SQL Server 的开发上就各行其道了。Microsoft 将 SQL Server 移植到 Windows NT 系统上专注于开发推广 SQL Server 的 Windows NT 版本。Sybase 则较专注于 SQL Server 在 UNIX 操作系统上的应用。

SQL Server 2000 是目前应用最多的数据库管理系统之一,具有使用方便、可伸缩性好、与相关软件集成度高等优点。SQL Server 2000 根据不同的应用需要有多多个不同的版本,常见版本有 4 个。

- 企业版(Enterprise Edition):支持 SQL Server 2000 所有的特性,可作为大型 Web 站

点、企业 OLTP 联机事务处理以及数据仓库系统等应用的产品数据库服务器。

- 标准版(Standard Edition): 用于小型的工作组或部门。
- 个人版(Personal Edition): 用于单机系统或客户机。
- 开发者版(Developer Edition): 用于程序员开发应用程序, 这些程序需要 SQL Server 2000 作为数据存储设备。

此外, SQL Server 2000 还有桌面引擎 Desktop Engine 和 Windows CE 版。用户可以根据实际情况选择所要安装的版本。

SQL Server 2000 支持各种类型的客户应用程序。用现有的任意开发平台编写的应用程序几乎都可访问 SQL Server。这就是说不管用什么语言编写的程序都可以用于 SQL Server。对于 SQL Server 来说, 最适合的软件包是已发布的 Microsoft 产品 Visual Studio, 其中包括几种工具可供用户方便、迅速地开发新的 SQL Server 应用程序。通过 ODBC, 几种第三方的软件包也可以用来开发在 SQL Server 上运行的应用程序。

在数据库领域, SQL Server 过去常被认为离企业级数据库服务器还有一定差距。随着 SQL Server 2000 的发布, SQL Server 真正步入企业级数据库领域。SQL Server 2000 支持分布式事务处理, 并且在处理大量数据方面已做了很多改进。SQL Server 2000 还将 OLAP(在线分析处理工具)服务内建于服务器中。该服务叫作 Microsoft Decision Support Services(Microsoft 决策支持服务)。与市场上的其他数据库系统不同, 用户不用再购买第三方的通常很昂贵的应用程序, 从而降低了花费在 SQL Server 上的总费用。总之, SQL Server 在硬件、软件、客户许可证、管理费用、开发所需费用方面均比市场上其他 RDBMS 要低, 是一个性价比很高的 RDBMS 产品。

3. Oracle 数据库

Oracle 公司成立于 1977 年, 是一家专门开发数据库的公司。1984 年, Oracle 公司首先将关系数据库移植到了桌面计算机上。在随后的几个版本中, Oracle 不断推出分布式数据库、客户/服务器结构、锁定模式以及对称多处理计算机的支持等技术。Oracle 9i 增加了面向对象技术, 成为关系-对象数据库系统。目前, Oracle 公司在数据库领域处于领先地位, 其产品覆盖了大、中、小型机等几十种机型, Oracle 数据库成为世界上使用最广泛的关系数据系统之一。

Oracle 数据库产品具有以下优良特性。

(1) 可移植性: Oracle 的产品可运行于范围很广的硬件与操作系统平台上。可以安装在 70 种以上不同的大、中、小型机上; 可在 VMS、DOS、UNIX、WINDOWS 等多种操作系统下工作。

(2) 可联结性: 能与多种通讯网络相连, 支持各种协议。

(3) 高效性: 提供了多种开发工具, 能极大地方便用户进行进一步的开发。

4. Sybase 数据库

Sybase 是 Sybase 公司开发的数据库系统, 是一个面向联机事务处理, 具有高性能、高

可靠性的、功能强大的关系型数据库管理系统(RDBMS)。Sybase 数据库服务器自 SQL Server 11.5 起, 改名为 Adaptive Server Enterprise(ASE), 它集成了原有的服务器系列, 如 SQL Server, SQL Anywhere, Sybase IQ, Sybase MPP 等。

Sybase 数据库服务器支持企业内部各种数据库应用需求, 如数据仓库、联机事务处理、决策支持系统和小平台应用等。Sybase 数据库的多库、多设备、多用户、多线索等特点极大地丰富和增强了数据库功能。因为 Sybase 数据库系统是一个复杂的, 多功能的系统, 所以对 Sybase 数据库系统的管理就变得十分重要, 管理的好坏与数据库系统的性能息息相关。

Sybase 为用户提供了良好的开发工具和开发环境, 支持组件创建和快速应用开发, 包括 Powerbuilder、Power Designer、Power J 等。

1.2 关系数据库标准语言 SQL

结构化查询语言(Structured Query Language, 简称 SQL)最早于 1974 年由 Boyce 和 Chamberlain 提出, 1986 年被 ANSI(美国国家标准协会)定为关系型数据库的标准语言, 后来被 ISO 采纳为国际标准。最新的版本是 SQL-99(目前对最新标准的有效性还存在一些争论, 一些组织和厂商并不支持 SQL-99)。

绝大多数关系型数据库管理系统(RDBMS)都至少对 SQL-92 提供了完整的支持, 许多关系型数据库管理系统还对 SQL 语言进行了扩充, 在其产品中包含了自己特有的语法集合。例如, Transact-SQL(T-SQL)是 Microsoft SQL Server 在 SQL 语言基础上的扩充。对于需要支持多种数据库系统或者有可能升级到其他数据库系统的应用程序来说, 最好不要使用某种数据库系统对 SQL 语言的扩充部分(即非标准 SQL), 因为它们通常是不被其他数据库系统所兼容的。

1.2.1 SQL 语言概述

SQL 是一个非过程化的语言, 它允许用户在高层的数据结构上工作, 而不对单个记录进行操作。SQL 可操作记录集, 所有 SQL 语句接受记录集作为输入, 返回记录集作为输出。SQL 的集合特性允许一条 SQL 语句的结果作为另一条 SQL 语句的输入。

SQL 不要求用户指定对数据的存储方法, 这种特性使用户更易集中精力于要得到的结果; 所有 SQL 语句使用查询优化器, 它是 RDBMS 的一部分, 由它决定存取指定数据的最快方法。查询优化器知道存在什么索引, 在哪儿使用索引合适, 而用户则从不需要知道表是否有索引、有什么类型的索引。

SQL 语句为各种各样的任务提供了命令, 其中包括:

- 查询数据记录。
- 在表中插入、更新和删除记录。

- 建立、修改和删除数据对象。
- 控制对数据和数据对象的存取。
- 保证数据库一致性和完整性。

以前的数据库管理系统为上述各类操作提供单独的语言,而 SQL 将全部任务统一在一种语言中。

SQL 是所有关系数据库的公共语言。由于所有主要的关系数据库管理系统都支持 SQL 语言,用户可将使用 SQL 的技能应用于多个 RDBMS,所有用 SQL 编写的程序都是可以移植的。

1. SQL 语言支持的数据类型

SQL 语言支持多种标准数据类型,包括字符型、整型、货币类型、二进制数据类型、空类型等,下面对 SQL 语言支持的数据类型进行简单的介绍。

- Char(N)、Varchar(N)、Text: 字符型数据,其中 N 表示其长度,范围是 0~8000; Text 类型数据的最大长度为 $2^{31} - 1$,即 2GB。
- Nchar(N)、Nvarchar(N)、Ntext: 宽字符型数据,其中 N 表示其长度,范围是 0~4000; Ntext 类型数据的最大长度为 $2^{30} - 1$,即 1GB。
- Bigint、Int、Smallint、Tinyint: 整型数据,其中 Int 也可以写为 Integer,占用字节数分别为 8、4、2、1,数值表示范围分别为 $-2^{63} \sim 2^{63} - 1$ 、 $-2^{31} \sim 2^{31} - 1$ 、 $-2^{15} \sim 2^{15} - 1$ 、0~255。
- Bit: 位数据,占用一个二进制位,表示 0 或者 1。
- Float、Real: 近似数值型数据,数值表示范围分别为 $-1.79E+308 \sim 1.79E+308$ 、 $-3.40E+38 \sim -3.40E+38$ 。
- Decimal、Numeric: 精确数值型数据,数值表示范围是 $-1.00E38+1 \sim 1.00E38 - 1$,两者功能完全相同。
- Money、Smallmoney: 货币型数据,可以精确的表示货币数据到货币单位的千分之十,数值表示范围分别为 $-2^{63} \sim 2^{63} - 1$ 、 $-214\,748.3648 \sim 214\,748.3647$ 。
- Datetime、Smalldatetime: 日期时间类型数据。
- Binary(N)、Varbinary(N)、Image: 二进制数据类型,以二进制的形式存储数据,其中 N 表示其长度,范围是 0~8000; Image 类型数据的最大长度为 $2^{31} - 1$,即 2GB。
- Null: 空类型,指没有任何数据内容。

2. SQL 语言的表达式

表达式是字符与运算符的组合。一个常量、变量、列或者标量函数称为简单表达式。运用运算符将两个或多个简单表达式连接起来就构成复杂表达式。

SQL 语言通过各种运算符可以完成多种复杂的运算,它支持的运算符主要有以下类型。

- 算术运算符: +、-、*、/、%, 分别表示加、减、乘、除、取余。

- 位运算符：~、&、|、^，分别表示非、与、或、异或。
- 比较运算符：=、<>、<、>、<=、>=，分别表示等于、不等于、小于、大于、小于等于、大于等于。
- 逻辑运算符：Not、And、Or、In、Between And、Like、All、Any、Exists，含义分别为：非、与、或、在集合内、在范围内、模式匹配、满足所有条件、满足其中至少一个条件、存在。

由运算符组合起来的两个表达式必须具有该运算符支持的数据类型，并且满足以下条件之一：

- 两个表达式有相同的数据类型。
- 优先级较低的数据类型可以隐式转换为优先级较高的数据类型。

3. SQL 语言的注释方法

注释是程序代码中不参与执行的文本字符串，用于说明代码或暂时禁用正在进行诊断的部分，使用注释对代码进行说明，可增加代码的可读性，并能够使代码更易于维护。

(1) 单行注释

单行注释使用双连字符(--)标识，从双连字符开始到行尾均为注释部分。该类型的注释语句可与要执行的代码处在同一行，也可另起一行。

例：使用单行注释。

```
-- 对名为 Mydb 的数据库进行操作
Use Mydb
-- 显示 Mydb 库中 Table1 表中的全部内容
Select * From Table1
Go
```

其中，Use 语句用于将当前数据库切换到指定的数据库，用户在登录到 SQL Server 时，通常被自动连接到 Master 数据库，除非为每个用户的登录 ID 设置了各自的默认数据库。如果要操作 Master 以外的数据库，每个用户都必须执行 Use 语句从 Master 切换到另一个数据库。在切换数据库时，用户必须拥有该数据库的安全账户，该账号由数据库所有者提供。

Go 命令不是 SQL 语句，而是可以被 SQL 查询分析器等工具识别的命令。Go 命令标志批处理的结束，当 SQL 查询分析器在某代码行的前两个字节中读到字符 Go(不区分大小写)，则把从上一个 Go 命令开始的所有代码作为一个批处理发送给服务器进行处理。

(2) 多行注释

多行注释使用/* ... */ (正斜杠—星号对)标识，从开始注释对(*/)到结束注释对(*/)之间的全部内容均视为注释部分。该类型的注释可与要执行的代码处在同一行，也可另起一行，甚至可以在可执行代码内使用。对于多行注释，必须使用开始注释字符对(*/)开始注释，使用结束注释字符对(*/)结束注释。

例：使用多行注释。

```
/* 对 Mydb 数据库进行操作
   显示 Mydb 数据库中 Table1 表中的全部内容
   其中，GO 命令标志批处理的结束 */
Use Mydb
Select * From Table1
Go
```

多行注释不能跨越批处理指令，整个注释必须包含在一个批处理代码块内。例如，在 SQL 查询分析器中，如果 Go 出现在/*和*/分隔符之间的一行行首，则会将注释语句分成两部分，发送不匹配的注释分隔符，从而导致语法错误。

例：SQL 语句中出现的错误注释。

```
Use Northwind
Go
Select * From Orders
/* 错误的多行注释
Go
这里 Go 命令将注释分成了两部分 */
Select * From Orderdetails
Go
```

4. SQL 语言的分类

SQL 语言可以分为 3 个部分，分别是数据定义语言、数据操纵语言和数据控制语言，下面分别对这 3 种语言进行简单介绍。

- 数据定义语言(Data Definition Language, 简称 DDL), 如 Create Database、Drop Database、Create Table、Drop Table、Alter Table、Create View、Drop View、Create Index、Drop Index、Create Procedure、Drop Procedure、Create Trigger、Drop Trigger 等，主要完成对数据库以及数据库中的表、视图、索引、存储过程、触发器等的创建和删除工作。
- 数据操纵语言(Data Manipulation Language, 简称 DML), 如 Select、Insert、Update、Delete，主要完成数据的查询和存取等数据处理操作。
- 数据控制语言(Data Control Language, 简称 DCL), 如 Begin Transaction、Commit Transaction、Rollback Transaction、Grant、Revoke、Deny，主要完成事务管理、数据的完整性、存取权限等控制操作。

1.2.2 数据定义语言 DDL

数据定义语言完成对数据库对象的定义操作，主要包括 Create Database 语句、Drop