

实用计算机编程技术丛书
Visual Basic 6.0 数据库编程指南
中国石化出版社

实用计算机编程技术丛书

Visual Basic 6.0 数据库编程指南

姜晓铭 主编



SHIYONG JISUANJI BIANCHENG JISHU CONGSHU

12
M/5

中国石化出版社

7-31-
JXM/5

实用计算机编程技术丛书

Visual Basic 6.0 数据库编程指南

姜晓铭 主编

中国石化出版社

157704

内 容 提 要

本书主要讲解了如何使用 Visual Basic 6.0 进行数据库应用程序的设计和开发。其中第一章介绍了数据库的基本概念和如何设计一个数据库；第二章介绍了 Visual Basic 6.0 的数据存取工具以及存取方法；第三章介绍了 Visual Basic 6.0 的一个功能强大的数据库设计工具——可视化数据管理器；第四章讲的是如何利用数据控件和数据绑定控件编写数据库应用程序；第五章简单介绍了结构化查询语言；第六、七、八章分别重点讲解了 Visual Basic 6.0 的三种主要的数据存取方法——DAO、RDO 和 ADO；第九章介绍了开放式数据库连接(ODBC)API 以及 OLE DB API；第十章以 DAO 为基础介绍了多用户数据库的处理；第十一章介绍了使用 Visual Basic 6.0 的数据报表设计器设计数据报表的方法。

本书适用于对 Visual Basic 有一定基础的计算机爱好者，对于 Visual Basic 的专业开发人员也有很高的参考价值。

图书在版编目 (CIP) 数据

Visual Basic 6.0 数据库编程指南/姜晓铭主编.—北京：
中国石化出版社，2000

(实用计算机编程技术丛书)

ISBN 7-80043-941-0

I. V… II.姜… III. Basic 语言-程序设计
IV.TP312

中国版本图书馆 CIP 数据核字 (2000) 第 05026 号

JS396/13

中国石化出版社出版发行

地址：北京市东城区安定门外大街 58 号

邮编：100011 电话：(010)84271859

<http://press.sinopec.com.cn>

东远先行彩色图文中心排版

海丰印刷厂印刷

新华书店北京发行所经销

*

787×1092 毫米 16 开本 20.5 印张 516 千字 印 1—3000

2000 年 5 月第 1 版 2000 年 5 月第 1 次印刷

定价：34.50 元

前 言

计算机技术的飞速发展是 20 世纪科学技术发展的最重要的体现。计算机网络的出现更使得计算机逐渐成为人们生活中不可缺少的一部分。Internet 把每一台计算机连接起来,使我们彼此之间的距离一下子拉到了眼前。世界变得如此狭小,许多工作不出门就可以轻松完成。我们同时也发现,在这里有着如此之多的信息,多得让我们不知该如何去运用它们。数据库技术在这里得到了广大的发展空间,并已经成为 Internet 上必不可少的基础设施。

Microsoft 公司于 1991 年推出的 Visual Basic,是一种通用的程序设计语言。由于其界面友好,简单易用,功能强大,一出世就得到了广大用户的喜爱。Visual Basic 在许多领域得到了广泛的应用,特别是数据库应用程序的设计,更是受到了人们的重视。

Visual Basic 6.0 是 Visual Basic 的最新版本,它在以前版本的基础之上,更加强了其数据库功能。对于那些希望快速开发和使用基于客户机服务器的数据库应用程序的开发人员来说,Visual Basic 已经成为他们的最佳选择。

本书是针对有一定 Visual Basic 编程经验的读者编写的,用于提高读者开发数据库应用程序的水平。如果您只是一个初学者,建议您先看一下有关 Visual Basic 基础知识的书籍,当对 Visual Basic 有一定了解之后再来翻阅本书效果更好。

本书在编写过程中,力求语言简单明了,并附有大量的例程供读者借鉴,较为全面地介绍了 Visual Basic 的数据库功能,是一本程序开发人员必备的内容齐全的参考书。

在编写本书的过程中,使用的是 Visual Basic 6.0 的中文企业版,操作系统是中文版 Windows95。书中的例程都通过测试,可以运行。Visual Basic 全部缩写成 VB,以便于读者阅读。

本书由姜晓铭主编,朱曙光、吕伟、杨合庆、刘斌、韩志学、魏宏刚执笔。由于我们水平有限,编写时间仓促,疏漏之处在所难免,希望读者指正。

编者

2000 年 1 月

目 录

第 1 章 数据库基础.....	1
1.1 数据库有关概念	1
1.1.1 信息、数据和数据处理	1
1.1.2 数据库	1
1.1.3 数据库管理系统	2
1.1.4 关于客户机/服务器结构	3
1.2 设计一个数据库	5
1.2.1 原则及步骤	5
1.2.2 数据组织	6
1.2.3 索引的使用	12
1.2.4 使用查询	14
1.2.5 修改数据库	14
1.3 数据库设计方法及工具	15
1.3.1 可视化数据管理器	15
1.3.2 Microsoft Access	15
1.3.3 第三方数据库设计工具	16
1.4 数据库完整性	16
1.4.1 维持表间关系	16
1.4.2 保持数据完好	17
1.5 数据库安全性	18
1.5.1 保护数据库	18
1.5.2 数据库恢复	19
1.6 小结	19
第 2 章 VISUAL BASIC 6.0 与数据库.....	20
2.1 Visual Basic 6.0 的数据库	20
2.1.1 Visual Basic 数据库的数据类型	21
2.1.2 用 Visual Basic 6.0 作为数据库开发平台	21
2.1.3 Visual Basic 6.0 支持的数据库类型	22
2.1.4 Visual Basic 6.0 的数据库——关系型数据库	22
2.1.5 Visual Basic 6.0 操作数据库	22
2.2 Visual Basic 6.0 中新的数据存取工具	23
2.2.1 关于 OLE DB	23
2.2.2 支持 ADO(Active 数据对象)	25
2.2.3 数据绑定控件方法	25
2.3 Visual Basic 6.0 的数据存取方法概述	27

2.3.1 DAO 简介	27
2.3.2 RDO 简介	28
2.3.3 ADO、ODBC 及 OLE DB 简介	29
2.3.4 方法选择	30
2.4 小结	30
第 3 章 可视化数据管理器	31
3.1 简介	31
3.1.1 数据库窗口	31
3.1.2 编辑窗口	32
3.1.3 SQL 语句窗口	32
3.2 创建数据库	33
3.2.1 添加一个新表	34
3.2.2 为表增加一个索引	36
3.2.3 修改表中的字段	36
3.2.4 查看或修改表的结构、属性	36
3.2.5 重命名或删除一个表	37
3.2.6 复制一个表	37
3.3 打开已有数据库	38
3.4 修改表中数据	38
3.4.1 添加	39
3.4.2 删除	39
3.4.3 编辑	39
3.4.4 排序	39
3.4.5 移动	40
3.4.6 查找	40
3.4.7 过滤器	40
3.5 数据窗体设计器	41
3.6 小结	41
第 4 章 使用数据控件和数据绑定控件	42
4.1 数据控件简述	42
4.1.1 什么是数据控件	42
4.1.2 数据控件的两个基本属性	43
4.1.3 在窗体中加入一个数据控件	44
4.2 数据控件属性、方法和事件	46
4.2.1 属性	46
4.2.2 方法	47
4.2.3 事件	47
4.3 数据绑定控件	50

4.3.1 数据绑定控件	50
4.3.2 使用 DBGrid 控件浏览数据库	51
4.4 自动创建窗体	52
4.4.1 使用数据窗体向导	53
4.4.2 连接数据库	53
4.4.3 选择窗体布局及数据绑定类型	54
4.4.4 选择记录源和可用字段	55
4.4.5 控件的选择	56
4.5 创建应用程序	57
4.5.1 窗体设计过程	58
4.5.2 程序代码的编写	61
4.6 小结	67
第 5 章 结构化查询语言	68
5.1 SQL 概述	68
5.1.1 ANSI SQL 的执行方式	68
5.1.2 SQL 的特点	69
5.1.3 SQL 在 Visual Basic 中的功能	69
5.1.4 SQL 语言基础	70
5.2 DDL 的使用	72
5.2.1 建立表	72
5.2.2 CONSTRAINT 子句方法建立索引	73
5.2.3 CREATE INDEX 子句方法建立索引	75
5.2.4 ALTER TABLE 方法修改表结构和建立、删除索引	75
5.3 DML 中 SELECT 语句的使用	76
5.3.1 SELECT 语句的使用	76
5.3.2 WHERE 子句的使用	77
5.3.3 ORDER BY 子句的使用	79
5.3.4 利用统计函数与分组函数进行查询	80
5.3.5 多表查询	84
5.4 表的连接	87
5.4.1 内部连接	87
5.4.2 外部连接	89
5.5 DML 语句的动态查询	91
5.5.1 用 INSERT 语句添加记录	92
5.5.2 使用 DELETE 及 UPDATE 语句	94
5.5.3 用 SELECT...INTO 语句在查询中建表	95
5.5.4 子查询的应用	96
5.6 小结	98

第 6 章 DAO 对象库	99
6.1 DAO 应用体系	99
6.2 DAO 对象模型	100
6.3 生成 ODBC 数据源	102
6.4 DBEngine 对象	106
6.4.1 DBEngine 属性	106
6.4.2 DBEngine 方法	107
6.5 Workspace 对象	108
6.5.1 Workspace 属性	108
6.5.2 Workspace 方法	109
6.6 Database 对象	109
6.6.1 Database 属性	109
6.6.2 Database 方法	110
6.7 连接到数据库	110
6.7.1 用 Jet Database Engine 进行连接	111
6.7.2 用 ODBCDirect 进行连接	113
6.7.3 创建动态数据库	113
6.8 TableDef 和 TableDefs 对象	115
6.8.1 创建一个 TableDef 对象	115
6.8.2 TableDef 属性	116
6.8.3 TableDef 方法	117
6.8.4 TableDefs 对象的属性与方法	117
6.9 RecordSet 对象	118
6.9.1 RecordSet 对象的基本类型	118
6.9.2 建立 RecordSet 对象	119
6.9.3 Recordset 对象的属性	120
6.9.4 Recordset 对象的方法	121
6.10 Field 和 Fields 对象	123
6.10.1 添加一个 Field 对象	123
6.10.2 Field 对象的属性	124
6.10.3 Fields 对象的属性和方法	126
6.11 Index 对象	126
6.11.1 添加 Index 对象	126
6.11.2 Index 对象的属性	127
6.11.3 Index 对象的方法	128
6.11.4 删除索引	128
6.12 Relation 对象	128

6.12.1 Relation 对象的属性	128
6.12.2 Relation 对象方法.....	129
6.13 QueryDef 对象	129
6.13.1 建立 QueryDef 对象.....	129
6.13.2 QueryDef 对象的属性.....	129
6.13.3 QueryDef 对象的方法.....	130
6.14 事务处理过程	131
6.14.1 使用事务	131
6.15 错误处理	133
6.15.1 Error 对象.....	133
6.15.2 捕获错误	134
6.16 小结.....	137
第 7 章 RDO 对象库	138
7.1 RDO 简介	138
7.1.1 RDO 与 DAO 的比较.....	138
7.1.2 RDO 的功能.....	139
7.1.3 RDO 对象模型.....	141
7.2 rdoEngine 对象.....	143
7.2.1 rdoEngine 对象的属性	143
7.2.2 初始化 rdoEngine 对象的缺省属性	144
7.3 rdoEnvironment 对象.....	144
7.3.1 创建新的 rdoEnvironment 对象	144
7.3.2 rdoEnvironment 对象的属性.....	145
7.3.3 rdoEnvironment 对象的方法.....	145
7.3.4 使用 rdoEnvironment 对象管理事务	146
7.4 rdoConnection 对象	147
7.4.1 创建 rdoConnection 对象	148
7.4.2 rdoConnection 对象的属性	149
7.4.3 rdoConnection 对象的方法	150
7.5 RDO 游标.....	151
7.5.1 选择一个 RDO 游标库	152
7.5.2 选择 RDO 游标类型.....	154
7.6 rdoResultset 对象	157
7.6.1 创建 rdoResultset 对象.....	157
7.6.2 rdoResultset 对象的属性与方法.....	158
7.6.3 处理 RDO 结果集.....	161
7.6.4 定位当前行指针	162
7.6.5 更新 RDO 数据.....	163

7.6.6 处理多 RDO 结果集.....	166
7.6.7 使用 GetClipString 方法	168
7.7 异步执行 RDO.....	169
7.7.1 处理异步查询	169
7.7.2 编写 RDO 事件处理程序	171
7.7.3 使用 RDO 事件.....	172
7.8 远程数据控件	173
7.8.1 RDC 的属性	173
7.8.2 RDC 的事件	175
7.8.3 RDC 的方法	177
7.8.4 使用 RemoteData 控件.....	178
7.9 RdoQuery 对象.....	185
7.9.1 生成 rdoQuery 项目	188
7.9.2 执行存储过程	193
7.10 在 RDO 中使用事务处理	197
7.10.1 使用事务处理	198
7.10.2 事务举例	198
7.11 RDO 错误处理.....	200
7.11.1 rdoError 和 rdoErrors 对象.....	200
7.11.2 管理 RDO 错误和信息	201
7.12 小结.....	202
第 8 章 ACTIVEX 数据存取对象.....	203
8.1 ADO 和 RDO、DAO 的比较.....	203
8.1.1 在 Visual Basic 中引用 ADO 2.0.....	204
8.1.2 ADO 2.0 对象模型	204
8.2 一般数据访问方案的比较 (RDO 和 ADO)	205
8.2.1 建立到某个数据库的连接.....	205
8.2.2 运行一个基本查询	208
8.2.3 在一个 MSHFlexGrid 控件中显示结果集	209
8.3 高级数据访问方案的比较(RDO 和 ADO)	212
8.3.1 执行参数查询	212
8.3.2 执行参数驱动的存储过程.....	214
8.3.3 运行返回多个结果集的存储过程.....	216
8.3.4 执行动作查询	217
8.3.5 运行开放式批处理查询.....	218
8.4 ADO 错误处理.....	221
8.4.1 Errors 集合	221
8.4.2 Error 对象.....	222

8.4.3 ADO Error 参考	222
8.5 小结	225
第9章 ODBC 与 OLE DB API	226
9.1 ODBC 简介	226
9.2 ODBC 3.x 函数	227
9.2.1 与数据源连接	227
9.2.2 获取驱动程序和数据源信息	227
9.2.3 设置或者获取驱动程序属性	228
9.2.4 设置或者获取描述符字段	228
9.2.5 预备 SQL 语句	228
9.2.6 提交要求	228
9.2.7 检索结果集及其相关信息	228
9.2.8 获取关于数据源系统表的信息	229
9.2.9 终止语句执行	229
9.2.10 中断连接	230
9.3 连接数据源	230
9.3.1 环境和连接句柄	230
9.3.2 语句句柄	232
9.3.3 数据源名称	233
9.4 OLE DB	235
9.4.1 OLE DB 访问数据源	235
9.4.2 部件对象模块(COM)	236
9.4.3 OLE DB 对象	237
9.5 小结	238
第10章 多用户数据库	239
10.1 用户访问控制	239
10.1.1 数据库级锁定	239
10.1.2 表级锁定	240
10.1.3 页级锁定	241
10.2 拷贝数据库	244
10.2.1 对象、属性和方法	244
10.2.2 数据库转换为可复制	247
10.2.3 保持本地数据库	248
10.2.4 创建副本拷贝	249
10.2.5 副本同步	249
10.2.6 同步冲突	250
10.3 数据库的安全性	251
10.3.1 分组管理	251

10.3.2 权限的设置与修改	252
10.3.3 确保安全地打开一个 Access 数据库	254
10.3.4 打开 ODBC 数据库	254
10.4 共享数据库	254
10.5 关系数据库的完整性	255
10.5.1 创建关系	255
10.5.2 连锁式关系的改变	257
10.6 小结	258
第 11 章 生成报表	259
11.1 Data Report Designer	259
11.1.1 数据报表设计器特性	259
11.1.2 数据报表的结构	260
11.1.3 数据报表设计器的组成	260
11.1.4 数据报表控件	260
11.2 简单数据报表的开发	261
11.2.1 创建层次结构游标	261
11.2.2 创建数据报表	263
11.2.3 数据报表预览	264
11.2.4 数据报表的功能扩充	265
11.2.5 数据报表中的计算字段	267
11.2.6 向数据报表添加控件	269
11.2.7 数据报表对数据进行分组	272
11.2.8 数据报表中的合计字段	274
11.2.9 在数据报表中强制分页	275
11.2.10 向数据报表添加日期、时间、页号和标题	275
11.3 使用 ExportReport 方法导出数据报表	277
11.3.1 ExportFormat 对象	277
11.3.2 模板代码	279
11.4 数据报表的输出	279
11.4.1 选择显示“打印”对话框	280
11.4.2 不显示对话框打印	280
11.5 数据报表设计器上控件的设置	280
11.5.1 Command 对象	281
11.5.2 控件的放置管理	282
11.5.3 Function 控件的例外情况	282
11.5.4 Command 对象的例外情况	282
11.6 Data Report 事件	283
11.6.1 超时和异步调用	283

11.6.2 Error 事件——对于异步函数	284
11.7 小结	285
附录 A 可能的 DAO 错误信息	286
附录 B 可能的 RDO 错误信息	307

第1章 数据库基础

计算机技术的蓬勃发展是 20 世纪科技发展史上最重要的事件之一。特别是近些年来 Internet 的发展, 我们突然发现信息居然如此之多, 以至使人全然不知该如何来获得对自己有用的信息。数据库技术的出现及其发展, 使这个问题随之迎刃而解。由于数据库中所储存的数据具有结构化、最小的冗余度、较高的程序与数据之间的独立性的特点, 尤其是关系数据库概念简单、使用方便, 并建立在一定的理论基础上, 所以这项技术很受人们欢迎。人们越来越多的利用数据库来搜索信息和管理事务。各行各业的信息系统、包括 Internet 上的信息系统, 也大多离不开数据库的支持。在本章中, 将简要介绍数据库及其设计的一些基本概念, 使读者对于数据库有一个初步的了解, 并更好地理解以后各章的内容。

1.1 数据库有关概念

数据库问题是在 60 年代初提出的, 从 70 年代至今, 数据库的理论、技术和方法已日趋完善, 并得到广泛的应用。在信息化社会的今天, 它正渗透到社会活动的各个领域, 发挥着重要的作用。

1.1.1 信息、数据和数据处理

信息

信息是人们用以对客观世界直接进行描述的、可以在人们之间进行传递的一些知识。物质的存在伴随着信息的变化, 物质的变化会引起信息的变化。

数据

信息产生后需要被处理和加工, 需要被交流和使用。为了记载信息, 人们使用各种各样的物理符号和它们的组合来表示信息, 这些符号及其组合就是数据。数据是信息的具体表现形式, 信息是数据的有意义的表现。

数据处理

有了数据就产生了数据处理的问题, 人们收集到的各种数据需要经过加工处理。所谓数据处理包括对数据的收集、记载、分类、排序、存储、计算或加工、传输、制表、递交等工作, 使有效的信息资源得到合理和充分的使用。

1.1.2 数据库

所谓数据库是指在计算机的存储设备上合理地存放相关数据的集合。它是通过文件或类似文件的数据单位而组织起来的。

两种数据库类型

用于计算机的数据库有两种类型：平面文件数据库和关系数据库。

平面文件数据库的每一单个记录为每一人口存储了所有信息，并且单个表中包含了数据库中所有记录，这种文件管理方式为许多计算机的数据库系统所使用。以电话簿为例，它的每一行包含姓名、地址和电话号码，这相当于一个记录，众多这样的记录的集合就构成一个平面文件数据库。

相比之下，关系数据库在许多表中存储信息，这些表之间由关联相连接。与前一种数据库类型相比，关系型数据库可以使信息存储更有效率。这正是后面将提到的设计关系数据库的目的之一——消除冗余数据。由于 Visual Basic 6.0 采用的数据库是关系型数据库，故以后的内容如无特别说明，都是指关系型数据库。

结构

数据库中的实际数据将存放成表格(table)，类似于随机访问文件。表格中的数据是由行(row)元素和列(column)元素构成的。各行中包含结构相同的信息块，称为记录。列元素则表示同一类信息，称为字段。记录中的一个特定数据项目称为域，如下表中的“性别”。

表 1-1 教学系统数据库的职员表

编号	姓 名	性别	出生日期	职称	工资
1001	李 芳	女	15/08/68	讲师	1200
1002	林正彬	男	23/10/56	教授	1500
1003	范 宏	男	19/01/69	讲师	1200
1004	刘玉伦	女	12/05/65	副教授	1300
1005	曹红霞	女	22/04/72	助教	1000
1006	徐 静	女	04/06/50	教授	1500
1007	石贵来	男	08/12/74	助教	1000

域

行/记录

列/字段

1.1.3 数据库管理系统

数据库管理系统——DBMS(DataBase Management System)，即对数据进行管理的软件系统，是一个数据库系统的核心。数据库系统的一切操作，包括创建各种数据库对象，如表、视图、存储过程等，以及应用程序对这些对象的操作，例如插入数据到表中，对表中原有数据的检索、修改、删除等操作，都是通过数据库管理系统进行的。

功能

大多应用程序都必须具备检索、分析及存储数据的功能。以前的开发者必须既要控制用户接口和程序计算部分，又要开发存储和检索顺序文件、二进制文件、随机访问文件中数据

的方法,同时还必须编写程序代码来控制程序所需要进行的搜索,或是生成必要的索引来控制记录的顺序,DBMS使得这一过程变得容易多了。创建DBMS仅需对所需存储的数据类型进行初始化,需要检索或存储时,程序向DBMS发出请求,DBMS会自动地控制数据管理。同时,DBMS也能按用户的指令搜索和创建索引。

DBMS的这些优点使得:

- 由于不必编写搜索算法程序或是读写语句来处理当前记录,故程序的初始化设计更方便了。
- 可以方便地改变数据格式。以前必须为把数据文件移植到新格式编写例程,还得修改相应的输入输出语句,而现在只需改变存储在数据库中的数据的定义。
- 容易创建附加程序来访问数据库的数据。要达到这一点,不必知道数据格式,只需知道数据域名及其包含的数据类型,而这些信息大多包含在DBMS系统中。

组成

DBMS主要由两个部分组成:编程接口和数据库引擎。

编程接口由用户接口和数据检索请求组成,主要提供命令,让程序指挥数据库引擎的操作。它通常包括下列元素:

- (1)数据定义语言(DDL),它告知数据库引擎数据对象(表、记录域和索引)的格式,DDL也定义数据库的数据有效性验证和数据完整性规则。
- (2)数据操作语言(DML),它告知数据库引擎对数据执行的功能(检索、修改、删除和存储)。
- (3)数据控制语言(DCL),它告知数据库引擎对于不同的用户各自所允许访问的数据类型。

数据库引擎能提供数据的物理存储结构、数据检索及更新、数据搜索和索引的功能,也能提供保证数据有效性验证、数据完整性和数据安全性的方法。同时,数据库引擎的设计决定着数据库支持的数据操作功能。

比较重要的Microsoft Jet引擎为Visual Basic与数据库的接口提供了基本的方法,这种方法也为Microsoft Access和其它Microsoft产品所使用。Jet引擎基本功能包括:数据操作、数据定义、数据库安全、数据库修复、数据库压缩和重新索引等。

1.1.4 关于客户机/服务器结构

Visual Basic 6.0是开发基于客户机/服务器数据存取应用程序的关键组件,由于本书是讲述Visual Basic 6.0的数据库编程,因此有必要对客户机/服务器加以介绍。

特征

客户机/服务器计算是模块化编程的逻辑扩展。基于模块的应用程序设计把大的软件分解成较小的模块,从而方便开发和维护。而客户机/服务器计算则比模块化编程更进一步:它允许模块可以不在同一存储空间运行。在这种结构中,客户机是调用模块,而服务器则是被调用模块。

这种体系的基本特征如下:

- (1)它由一个和用户交互的前台客户机及一个和共享资源交互的后台服务器构成。
- (2)前台任务和后台任务对计算资源的需求有本质区别。
- (3)这个环境是一个多机种、多供应商的环境。
- (4)可伸缩性。水平放缩指添加或删除一些客户工作站对系统性能只有轻微影响。垂直放缩指可以迁移到更大更快的服务器或多个服务器上。

客户进程与服务器进程

客户进程指客户发送信息请求服务器完成某项任务的进程。它通常管理整个系统的用户接口,检查用户输入数据的有效性并向服务器进程发送请求,有时还执行一些商业逻辑运算。客户进程是用户可以看到并与之交互的应用程序前端。客户进程提供了特定问题的解答逻辑,并且提供了用户与其他应用系统的接口。当然,它还管理与用户交互的本地资源。图形用户界面是客户工作站的一个关键要素,它检测用户的动作,管理显示窗口并负责在窗口内显示数据。

服务器进程指完成用户请求任务的进程。它接收客户的请求,执行数据库查询和更新操作,集中地管理数据,并响应客户请求,有时还执行商业逻辑运算。服务器进程可能运行于网络中的另一台机器上,服务器为宿主机操作系统或网络文件服务器,可提供文件系统服务和应用程序服务。另外应用程序服务也可由另一台台式机提供。服务器进程作为软件引擎工作,管理着共享资源,完成同类应用程序中常用的那些后台服务。

至于两者的联系,客户和服务器的硬件和操作系统平台可以不同而且通常都是不同的。客户进程和服务器进程通过一组严格定义的应用程序编程接口(API)相通信。

两种类型及各自特点

客户机/服务器计算有两种类型:基于两层和基于三层。

两层客户机/服务器体系结构用一个数据库服务器代替网络中的文件服务器,这样,网络就可以响应用户在关系数据库上的查询处理。这样做可以显著地减少网络流量,而且在数据库服务器的管理之下可以很容易地实现多用户更新。在这种结构中,远程数据调用和 SQL 命令一般用于客户和服务器之间的通信。服务器同时还支持存储过程和触发器,存储过程可用于实现一些适于运行在服务器上的商业运算规则,这可以大大地提高整个系统的效率。

两层体系结构在交互人数较少时,体现了很好的性能。但对于规模较大的企业环境中,两层体系便力不从心了。这就是三层体系产生的原因。

三层体系在客户与服务器之间添加了一个中间层,这个中间层能完成排队、程序执行和数据库分级等功能。中间层的引入,使两层方法中的同步处理变成异步处理,即相当于把中间层当作一个虚拟服务器,它接受客户请求并对其排队,按一定规则确定优先级,到合适的时候将用户请求送至服务器。这将解决两层体系中用户更多所导致的问题。

实现三层客户机/服务器体系的中间层功能当今一般采取四种方法:使用事务处理监视器,使用消息服务器,使用应用程序服务器,使用分布式组件。

两层体系和三层体系的对比简图如下所示: