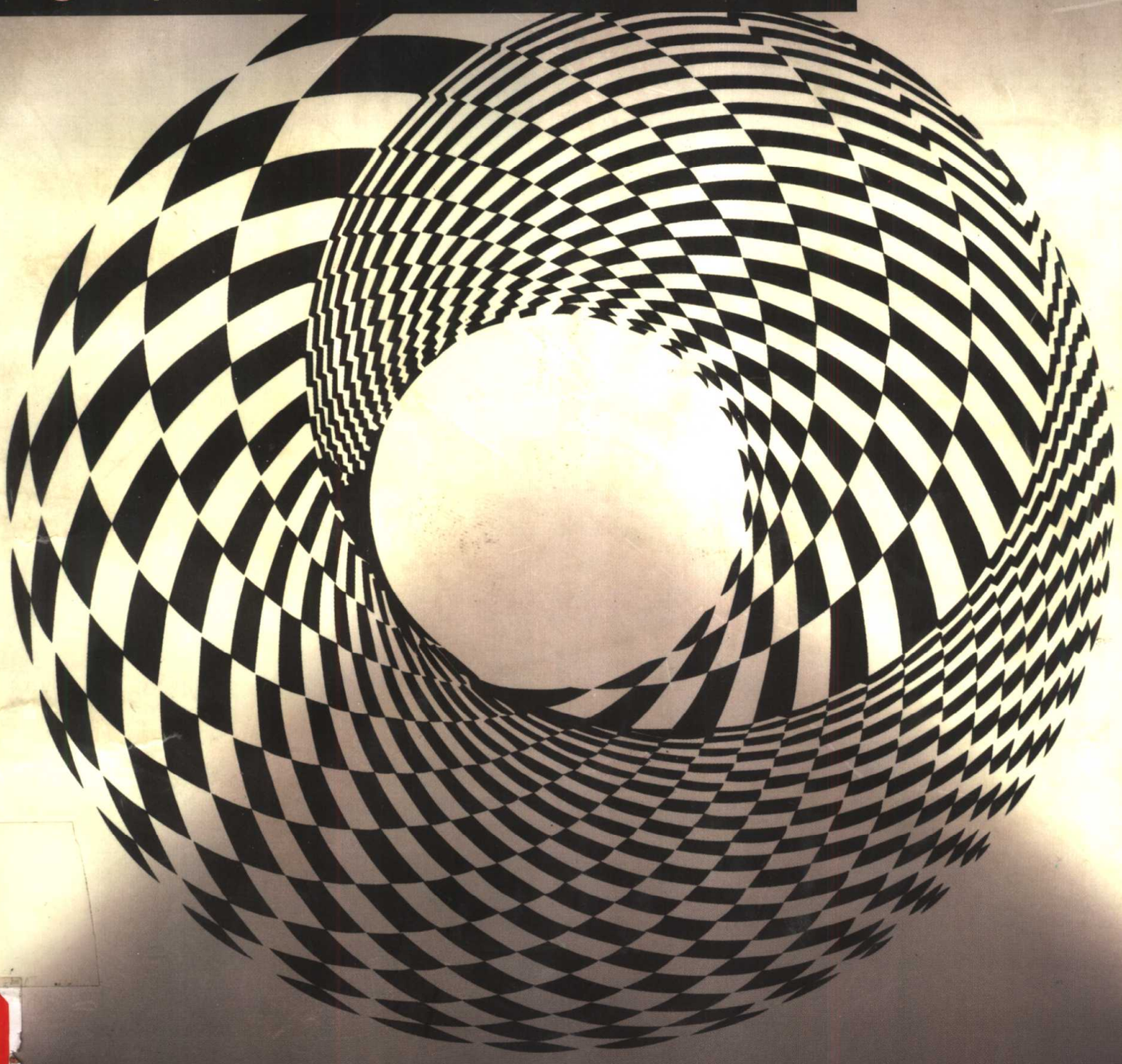
# Client/Server Database Application Development

# 客户/服务器数据库应用开发

朱扬勇　凌　力/编著



复旦大学出版社

## 内 容 提 要

本书对客户/服务器数据库应用系统开发的各个方面都进行了介绍。内容包括客户/服务器数据库的基本概念、数据库应用系统开发方法、数据库概念设计的ER方法、关系数据库规范化原理和方法、数据库完整性安全性设计、ODBC和存储过程技术的使用、系统开发中人的因素、计算机网络原理和网络集成技术等。

本书的读者对象主要是大学本科高年级学生、研究生、信息系统开发的高级设计人员和项目管理人员等，需要读者具有计算机及相关专业的基础知识，尤其需要具备数据库和软件工程知识以及一定的程序设计经验。本书可作为大学本科高年级学生和研究生教材或教学参考书，以及信息产业高级技术管理人员的参考书。

# 客户/服务器数据库应用开发

作　　者　朱扬勇　凌　力
责任编辑　孙未未
出　　版　复旦大学出版社
　　（上海国权路579号　邮政编码　200433）
发　　行　新华书店上海发行所
印　　刷　复旦大学印刷厂
开　　本　787×1092　1/16
印　　张　15.25
字　　数　380 000
版　　次　1997年12月第1版　1997年12月第1次印刷
印　　数　1—3 000
书　　号　ISBN 7-309-01975-X/T・196
定　　价　22.00元

# 前　言

客户/服务器技术是90年代新兴的计算机技术，其中，尤其以客户/服务器数据库技术的迅猛发展和广泛应用而备受欢迎。与此同时，连接计算机网络中各软件（尤其是连接客户和服务器）的中间件技术也得到了快速发展。ODBC是广泛用于客户/服务器数据库之中的一个中间件，它使数据库应用程序能跨越多个数据库平台，实现了数据库之间的可互操作性。如果说先前的数据库仅仅实现数据共享的话，那么数据库服务器上的存储过程在一定程度上实现了程序的共享。这种用扩充的SQL语句编写的并事先存放于数据库服务器上的程序，在提高应用系统的性能、保护数据库完整性和安全性方面发挥着巨大作用。相应地，软件开发方法也在发展，面向对象的方法正被越来越多的人所接受，ER方法也由于扩充了泛化层次等面向对象的特征而使其建模能力大大加强。在编程方面，可视化的开发工具使我们能够开发出令用户满意的界面，而工作量却在减少。当然，数据库本身的设计是客户/服务器数据库应用开发的核心内容。传统的数据分析、ER建模和集成、关系模式规范化以及数据库保护等技术仍然是必须的并且也是有效的，这些技术在新的环境下会有新的内容。

新技术带来了新观念。客户/服务器系统的灵活性、可扩充性，以及开发用户界面的方便性使得我们有可能面对易变的用户需求，这就给客户/服务器数据库应用系统的开发带来了活力，“以用户为中心”的开发观念将取代“以技术为中心”的开发观念，即让技术适应用户而不是让用户适应技术。

本书力求较为全面地介绍客户/服务器数据库应用系统开发的各个方面。主要内容包括客户/服务器数据库的基本概念、数据库应用系统开发方法、数据库概念设计的ER方法、关系数据库规范化原理和方法、数据库完整性和安全性设计、ODBC和存储过程技术的使用、系统开发中人的因素、计算机网络原理和网络集成技术等。系统开发方法研究是一项实践性很强的工作，因此本书在介绍各种原理时，更强调原理的实际应用。

本书不太协调地在一本介绍理工技术的书籍里讨论了项目开发中人的因素。我们从开发经验中感觉到，很多客户/服务器数据库应用项目的失败不是技术上的原因，而完全是因为人和人之间的交流出了问题，因此，我们认为有必要对这一问题进行讨论。

客户/服务器数据库的应用对象可能是企业、学校、机构、部门等等，所以为了叙述上的方便，我们将它们统称为“应用体”，权且作为enterprise一词的翻译。

本书的内容大致安排如下：

第1章介绍了一些基本知识，主要有关系数据库、客户/服务器计算机网络、客户/服务器数据库开发环境和一些较新的概念，如中间件、ODBC和存储过程等。

第2章概要介绍了已有的数据库应用开发的方法，包括软件系统生命周期、结构化方法、原型法和面向对象方法等，讨论了为什么要进行数据库设计以及数据库设计的主要工

作、原理和方法等问题。

第3章系统地介绍了数据库设计的ER方法，主要有基本ER构造、高级ER构造、建立ER模型的方法（包括划分实体和属性、找出泛化和汇集层次、视图集成等技术）、将ER模式转换到关系模式等。

第4章介绍了关系数据库的规范化原理和方法，包括范式、规范化框架和规范化算法等内容。

第5章介绍了数据库保护，包括数据库完整性、安全性、事务、数据库备份与恢复等。

第6章介绍了客户/服务器数据库环境下新的编程技术，主要是ODBC和存储过程的使用。

第7章介绍了客户/服务器数据库应用开发的一个新方法。

第8章讨论了客户/服务器数据库应用开发中人的因素，给出了系统开发中各个角色及其职责，较为详细地介绍了需求收集技术，本章还介绍了项目管理方面的知识（包括项目的技术管理、行政管理和索赔技术）。我们把需求收集作为系统开发中人的因素，是因为我们认为需求收集技术更多的是人际交流技术。

第9章介绍了计算机网络原理。

第10章讨论了客户/服务器计算机网络的集成，包括网络规划、分析、设计和评价等。

本书的许多方面都是我们多年从事数据库应用开发的经验。其中，我们认为最重要的一条是：数据库应用项目一般都是用户投资的，因此，满足用户需要就是这类系统开发的唯一目的。这就要求整个开发思想应该是“以用户为中心”，这一思想将贯穿系统生命周期的每个阶段。以用户为中心的一个具体体现就是，用户改变需求是有道理的，无论什么理由都不能阻止用户改变需求。以用户为中心的一个具体方法是让用户参与开发，开发人员和用户良好地进行交流。“以用户为中心”这一开发思想在本书的许多地方都能看到。

本书第1章至第8章由朱扬勇编著，第9章和第10章由凌力编著。

复旦大学施伯乐教授一直在关心和支持本书的写作，他还在百忙之中审阅了全书并提出了许多宝贵的建议。在本书的写作过程中，与顾宁、郭德培和王宇君三位博士多次讨论了有关系统开发方法方面的问题并得到许多有益的启示，他们还对本书的初稿提出了许多好的建议。复旦大学出版社的范仁梅、陆盛强和孙未未等同志一直关心本书的出版并做了大量工作。在此谨向他们，向所有关心本书写作的朋友表示诚挚的谢意。

应该说编写这本书的时间不是很紧张，本书中的部分内容也已在复旦大学研究生课程“数据库与知识库”和“计算机网络工程”中讲授过，书中的不妥之处全属作者水平有限所致，如蒙读者指正，不胜感谢。

编　者

1997年10月于复旦

# 目　录

# 第1章 导 论

客户/服务器数据库一出现就得到广泛欢迎，并迅速发展。存储过程的使用提高了应用程序的性能；ODBC 技术则提高了应用程序的开发效率，并在一定程度上实现了数据库的可互操作性；可视化开发工具则使我们开发出非常友好的用户界面，而工作量却在减少。客户/服务器数据库应用已经成为当今数据库应用的潮流。当前的客户/服务器数据库都是关系型数据库，运行在客户/服务器计算机网络上，用一个数据库服务器管理数据库。

本章主要介绍客户/服务器数据库应用系统的开发所涉及的一些基本概念，包括：关系数据库、计算机网络、客户/服务器数据库环境和数据库应用系统等几个方面的内容，作为后面章节的基础知识。

## 1.1 关系数据库

关系数据库是由著名科学家 E.F.Codd 于 70 年代初提出的，被认为是第二代数据库，是当前数据库的主流，几乎所有运行在客户/服务器计算机网络上的数据库都是关系数据库。

### 1.1.1 数 据 库

**一、数据库系统**

在程序设计中，早先我们用数组、字符串和数据语句（如 BASIC 中的 DATA 语句）等来处理少量的数据，随后用数据文件（.DAT 或.TXT 文件）来处理较大量的数据，这就是所谓的文件系统。但随着计算机广泛进入数据处理领域，文件系统处理的数据量越来越大，这时，文件系统就显得力不从心了。这主要表现在数据不能快速访问、数据冗余量大、数据的一致性维护困难，等等。同时人们也注意到，对数据的操作类型通常是简单的，主要有查询、插入、删除和修改等，但良好地实现这些操作需要大量的存储技术、索引技术和查询优化技术等，而这些技术是一般开发人员难以掌握的。因此就有必要并且也能够（因为操作类型少）将这些操作交给系统软件来完成，这样就产生了数据库管理系统（DBMS）这一系统软件。DBMS 提供了数据定义语言（DDL）用于定义数据库，提供数据操纵语言（DML）来实现对数据库的任何操作，用户必须通过 DBMS 才能使用数据库进行数据处理操作。这样，数据从应用程序中分离出来，统一存入数据库之中，因而数据库又可以被多个应用程序所共享，同时数据的冗余度也就减小了。数据共享要求 DBMS 提供数据安全性机制和并发控制机制。另外，还必须保证数据库中的数据是正确的和有效的，这就是

数据库的完整性约束。总之，相对于文件系统，数据库系统具有快速准确地访问数据库中任意部分数据、数据可共享且冗余度较小、数据和程序的独立性较强、统一数据控制（安全性控制、完整性控制和并发控制）等特征。其中，数据库系统提供了对数据库中任意部分数据的快速准确地访问是数据库系统和文件系统的根本区别。

数据库系统是提供对数据进行存储、管理、处理和维护功能的计算机软件系统。它由计算机软件、数据库和有关人员组成。其软件主要包括数据库管理系统、宿主语言、开发工具和应用程序，其中，数据库管理系统是用于建立、使用、维护数据库的软件系统，宿主语言是可以嵌入数据库语言的程序设计语言；数据库是长期存储在计算机内的有组织的、大量的、可共享的相关数据的集合，它独立于具体的应用程序，为多个应用程序所共享；有关人员包括负责数据库的设计、修改和日常维护的数据库管理员（DBA）、应用系统开发人员和用户。

出现 DBMS 后，编写数据处理软件就不能再由单个高级程序设计语言完成了，而是通过“宿主语言 +DML ”来实现，其中 DML 用于数据库操作。这种宿主语言是一种通用的高级程序设计语言（如：C 、COBOL 等），与 DBMS 一起提供。DML 语言的命令通过两种途径包含在宿主语言中：

- 宿主语言对 DML 语句的引用是通过 DBMS 提供的过程调用实现的。
- 将 DML 作为宿主语言的一部分，提供一个预处理器来处理 DML 语句，并将它们转换为 DBMS 提供的过程调用。

显然，第二种形式本质上是用预处理器转换成第一种形式。但在大多数情况下我们比较喜欢用第二种形式，这是因为把 DML 作为宿主语言的一部分使得编写程序比较自然。

## 二、数据库系统的体系结构

下面我们来了解一下数据库系统的体系结构，它是一个三级抽象的结构（见图 1-1 ）。

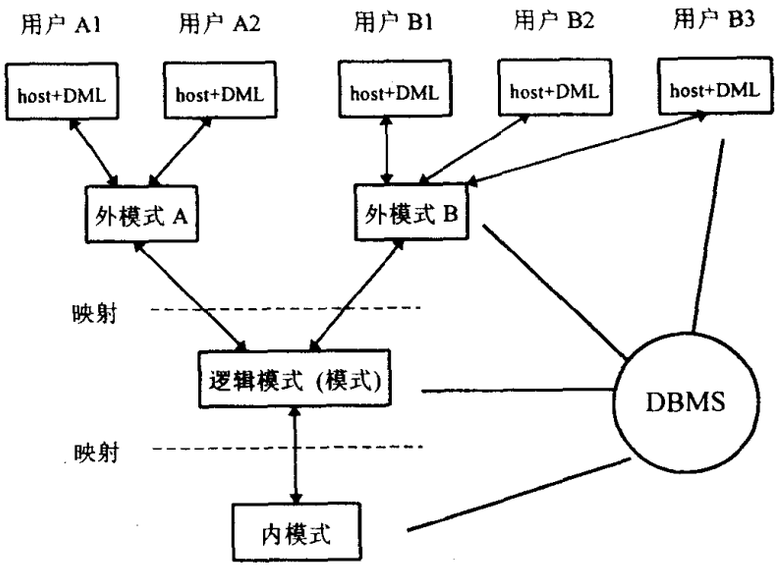



**图 1-1　数据库的三级抽象**

- **内模式**（又称存储模式）　是文件、索引和其他一些存储结构的汇集。它定

义了所有内部记录类型、索引、数据在存储介质上的安排等。从这一级看到的数据库称为物理级的数据库。

- **模式**（常称逻辑模式） 是对现实世界全局逻辑的抽象。它描述了全部数据的逻辑结构，包括数据之间的联系、数据的约束和安全性要求等。从这一级看到的数据库称为逻辑级的数据库。
- **外模式**（又称子模式） 是由模式导出的局部数据逻辑，是单个用户所要处理的数据集合。应用程序只能接触到外模式。模式与外模式之间的映射由VIEW语句实现。

通过三级抽象，数据库就有了物理独立性和逻辑独立性。物理独立性是指：如果内模式改变，则可以通过修改映射，而使逻辑模式以上的内容不变；逻辑独立性是指：如果模式改变，则可以通过修改映射，而使外模式不变，也不用修改程序。

三、数据模型

一个数据库一般是某个应用体（enterprise，指企业、机构或部门等）所涉及数据的综合，但现实世界中的数据是相互联系和制约的，因此，要用一种方法来抽象、表示和处理现实世界中的数据及其数据之间的联系。在数据库中是用数据模型来完成这一工作的。

数据模型是数据库系统中用于表示现实世界数据及数据之间联系的形式构架，由数据结构、数据操作和数据约束三要素组成。常见的数据模型有层次数据模型、网状数据模型、关系数据模型、逻辑数据模型和对象数据模型等。每个DBMS至少支持一种数据模型，到目前为止大多数DBMS也只支持一种数据模型，因此又常常根据DBMS所支持的数据模型来称呼数据库，如：层次数据库、网状数据库、关系数据库、逻辑数据库（演绎数据库）和对象数据库等。其中，层次数据库和网状数据库被认为是第一代数据库，关系数据库被认为是第二代数据库，而第三代数据库目前仍然处于研究与开发之中，演绎数据库和对象数据库是第三代数据库的代表，当前最流行的数据库是关系数据库。

## 1.1.2 关系数据库

所谓关系数据库是指数据按照关系模型来存放数据的数据库。那么，关系模型是什么样的呢？简单地说，关系模型就是形如表1-1的二维表。表1-1的特点是每一行（或者说每一个人）都有且只有姓名、单位、通信地址、邮编、电话等信息。虽然有些人可能没有电话，但通信录上也给他留出了电话这一栏目。我们说表1-1每一行的数据格式是一样的。

**表1-1 日常中的通信录**

| 姓 名 | 单 位 | 通 信 地 址 | 邮 编 | 电 话 |
|---|---|---|---|---|
| 朱博雅 | 复旦大学计算机系 | 上海邯郸路220号 | 200433 | 65487787 |
| 王 红 | 西安交通学院 | 西安黄河路278号 | 150021 | 3245798 |
| 李建民 | 新疆大学计算中心 | 乌鲁木齐胜利路14号 | 830046 | 8677685 |

表的一行称为一个元组（或称一条记录）；一个元组中的一项称为一个属性（或称为

字段）；相同的属性构成表的一列，每个属性列都有一个名称，称为属性名；全部属性名构成表的表头，整个表头又称为关系模式；一张表内的全部数据称为一个关系。图 1-2 展示了这些概念。

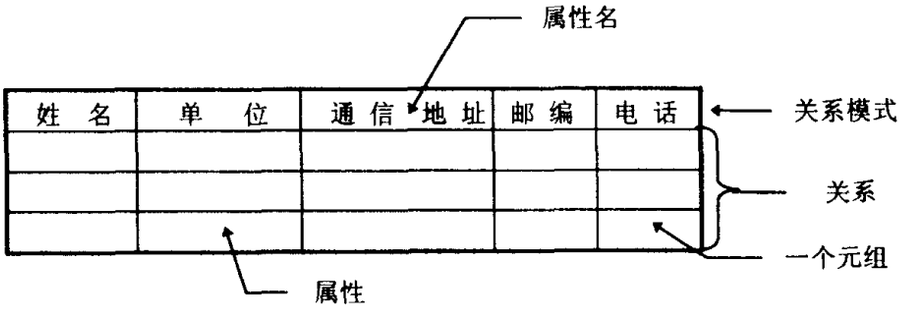



图 1-2 关系的概念图示

另外，关系模型对表有一个要求：属性必须是最小数据单位，不可再细分。正式地说，满足下列五个条件的表格称为关系表。

- 每个属性都是不可分解的。
- 表格中所有的行都具有相同的形式（称为关系模式），一行称为一个元组。
- 关系模式中的属性个数是固定的，每个属性都要命名，并且在同一个关系模式中，任何两个属性名都不相同。
- 任何两个元组都不相同。
- 属性的先后次序和元组的先后次序是无关紧要的。

由于要求任何两个元组都不相同，所以必然可以用某些属性的值（至少可以用所有属性的值）来确定出一个元组。比如，给定一个身份证号，就能找出一个人，而决不会是两个人。这些属性就称为候选键（常常简称为键），一个候选键可以由一个或多个属性组成，甚至可由全部属性组成。在一个关系模式中，候选键可以有一个或多个，我们要选一个候选键定为主键，这是因为我们要用主键来约束表、对表操作和建立表之间的联系。

### 1.1.3 SQL

SQL(Structured Query Language)语言是关系数据库的查询语言，包含 DDL 和 DML 。该语言的初始版本是在 70 年代中期的 SEQUEL （ Structure English QUEry Language ），IBM 把 SEQUEL 作为它的关系数据库产品 System R 上的查询语言。到 70 年代末，SEQUEL 这个名字被缩写为 SQL，不过出于习惯，“ SQL ”的发音仍与“ SEQUEL ”保持一致。 1986 年， ANSI （美国国家标准协会）将其批准为关系数据库语言的美国标准，随后 ISO（国际标准化组织）也将其定为标准。 SQL 成为标准后，各数据库厂商推出的关系数据库产品都尽可能地支持 SQL 。 SQL 是一种查询语言，它既可以在界面上做数据库的交互式查询（此时称为交互式 SQL 或动态 SQL ），同时也可在程序设计语言编写的应用程序中使用（此时被称为嵌入式 SQL ）。

SQL 语言中最主要、最核心的部分是它的查询功能（所谓查询就是从数据库中提取满足条件的数据）。查询是由 SELECT 命令实现的，其命令的格式为：

SELECT [<属性名列表>]
FROM <表名列表>
[WHERE <条件表达式>]
[GROUP BY <分组属性名> [HAVING <筛选条件>] ]
[ORDER BY <排序属性名> [ASC | DESC] [, <排序属性名>[ASC | DESC] ...]]

SELECT 命令中有三个基本子句： SELECT 子句、 FROM 子句和 WHERE 子句。SELECT 子句用于说明“查找什么数据（哪些属性）”， FROM 子句用于说明“数据从哪里（哪些表）查找”， WHERE 子句用于说明“查找的条件是什么”。

SELECT 子句的<属性名列表>指出要查找的数据。“属性名”可以是要包含在查询结果中的属性、常量和表达式。对于属性，必须是 FROM 子句所包含的表中的属性名。对于常量，查询结果中每一行都出现这个常量值。对于表达式，可以是用户自定义函数名。默认情况下，查询结果中包含所有行，即使用 ALL 。

GROUP BY <分组属性名>子句将结果按“分组属性名”分组，每个组产生查询结果中的一条记录，分组的附加条件用 HAVING 子句给出，只有满足条件的组才允许输出到查询结果表。

ORDER BY 子句将查询结果表按“排序属性名”升序（ ASC ）或降序（ DESC ）排序。

## 1.2 客户/服务器数据库

### 1.2.1 客户/服务器计算机网络

人类不断进化，社会分工越来越细，每个人所掌握的知识只是知识海洋中很小的一部分，因此每个人只能处理很少的问题，而大部分情况是靠分工合作来完成的。其中最常见的一种分工合作的方式是向别人请教。向别人请教的工作模式为：请教者向被请教者提出（用电话、信件或口头）一个请教问题，被请教者运用他的知识将问题解决并返回结果。在这一模式中，请教者、被请教者和媒介（如电话和信件等）为主体。如果被请教者为专门的咨询机构、服务机构，将同时为多个请教者提供服务。这时，请教者就称为一般意义上的客户，被请教者就称为一般意义上的服务机构，而媒介将是一般意义上的通信网络。咨询热线电话是这一模式的最为形象的表现。事实上，这一模式在人类社会是普遍的，我们每时每刻都在求助于别人，让别人为自己提供服务；同时我们也接受别人的请求，向别人提供服务。这就是我们日常中所看到的客户/服务器计算。

现在计算机碰到了和我们人类相同的问题：需要分工合作。为实现多台计算机的分工合作，就要将这些计算机连接起来构成一个计算机网络，网络中各计算机之间能够相互传送数据信息。

一个计算机网络有如下特征：

- 必定包含多于一台计算机。
- 有确定的连接方式。
- 以联系为目的。

在一个计算机网络中，如果一些计算机扮演客户（称为客户机），另一些计算机扮演服务者（称为服务器），客户机通过计算机网络向服务器提出计算请求，服务器计算机经过计算，将结果返回给客户机，这样的计算机网络就是客户/服务器计算机网络。常用的客户/服务器计算机网络为带有文件服务器和带有数据库服务器的客户/服务器计算机网络两类。

### 1.2.2 客户/服务器数据库

简单地说，客户/服务器数据库是在客户/服务器计算机网络上运行的数据库系统，有一个数据库服务器来管理数据库，应用程序则运行在客户机上，当需要对数据库进行操作时，就向数据库服务器发一个请求，数据库服务器收到请求后执行相应的数据库操作，并将结果返回给客户机上的应用程序。具体地说，客户/服务器数据库将数据库处理任务划分给两部分：客户机运行数据库应用程序，数据库服务器运行全部或部分 DBMS 。客户机上的数据库应用程序被称为“前端系统”，它负责所有屏幕和用户输入输出的处理；数据库服务器上的“后端系统”则负责数据处理和磁盘访问。

数据库服务器型的客户/服务器系统的结构如图 1-3 所示。

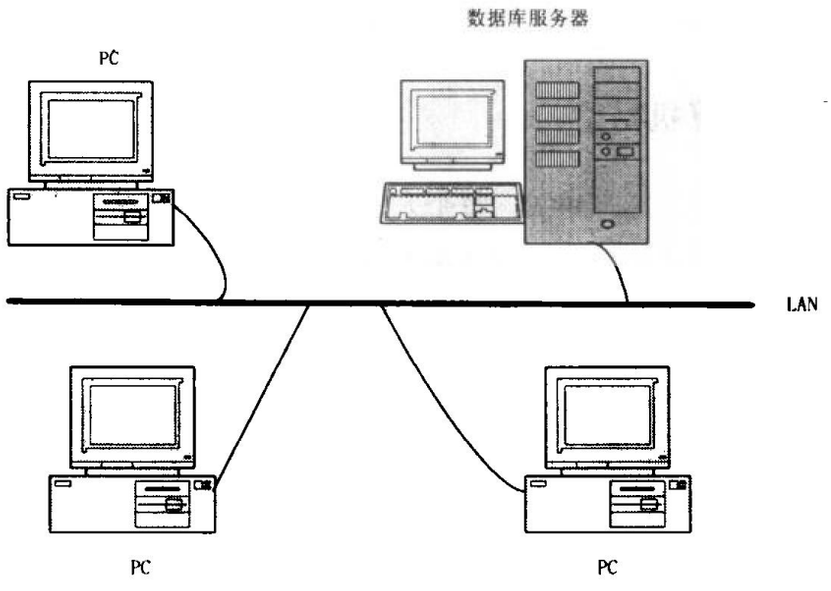



**图 1-3　数据库服务器型的客户/服务器系统的结构**

另外，在客户/服务器数据库计算机网络中，局域网文件服务器仍然提供了共享资源（如提供应用程序使用的磁盘空间和打印机），而数据库服务器既可以在与文件服务器相同的