

现代远程教育 及校园网建设全书

(下 卷)

中国工人出版社

第七篇

校园网数据库建设指南

第一章 电子资料库基本概念

第一节 数据管理及基本概念

在这一节中将简单介绍数据管理技术的发展情况，从中可知数据库技术是数据管理中的一门新技术；另外还将介绍数据管理中所涉及的如实体、实体间的联系等基本概念。

一、数据管理技术的发展

数据处理是指对数据进行收集、组织、加工、储存、抽取、传播等工作。而数据管理是指对数据的组织、存储、检索和维护等工作，所以数据管理是数据处理的中心。

数据处理工作由来已久，早在 1880 年美国进行人口统计，就采用了穿孔卡片存储信息，称为机器卡片文件 (Machine Card File)。这样用机械方法进行数据处理，称之为机械的数据处理系统。后来由于电子计算机的出现，在数据处理中采用了计算机，数据处理就进入电子数据处理系统时代。随着计算技术的发展，由于采用了磁芯、磁带、磁鼓、半导体存储器和磁盘等存储技术，数据管理也得到了迅速发展。它主要围绕着提高数据独立性、降低数据的冗余度、数据共享、提高数据的安全性和完整性等方面来进行改进，让使用者能方便地运用这些数据资源和管理这些资源。

数据在物理存储设备上的组织称之为数据的物理组织 (Physical Organization)，一般以文件形式组织。根据其组织结构的特点可分为顺序文件、索引文件和随机文件等。数据在使用者面前所呈现的组织方式，称之为数据的逻辑组织 (Logical Organization)。对于一种数据的逻辑组织，可以用不同的物理组织来实现。物理组织的好坏影响系统的性能和效率，所以一种数据的逻辑组织，在运行阶段中，由于性能要求或存储数据的设备更新，就引起数据的物理组织的改变，这种改变称之为数据的再组织。用户 (USER) 在编制应用程序时，是根据数据的逻辑组织对数据进行操作。应用程序对数据的物理组织的依赖程度称为物理数据的独立性 (Physical data independence)，依赖程度越低，则物理数据的独立性越高。如果在数据处理环境中，数据具有物理独立性，则在数据再组织时不会影响原来应用程序的执行。数据管理方法可根据数据的独立性、数据的冗余度、数据间相互联系及数据的安全性、完整性等特点划分成三个不同阶段：人工管理阶段，文件系统阶段和数据库系统阶段。

(一) 人工管理阶段 (至 60 年代早期)

在该阶段，数据的逻辑组织和它的物理组织是相同的，计算机系统仅提供基本的输入输出操作，应用程序员亲自设计物理组织。当数据的物理组织或存储设备改变时，其

应用程序必须重新编制。由于数据的物理组织是由应用程序员根据应用的要求设计的，很难实现多个应用程序共享数据资源，造成数据大量重复。此阶段数据的逻辑组织与物理组织之间的关系可用图 7.1-1 表示。

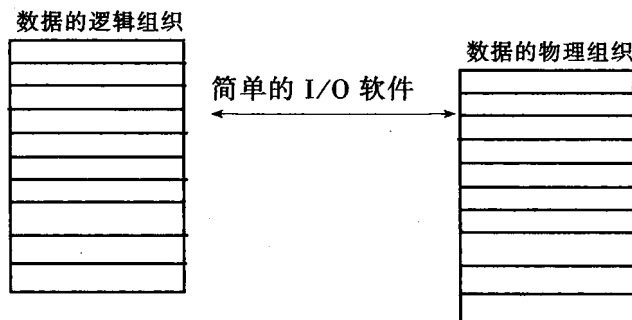


图 7.1-1 人工管理阶段的特征

(二) 文件系统阶段 (至 60 年代后期)

人工管理阶段的数据管理有许多缺点：数据独立性差，应用程序依赖于物理组织；由于数据的组织是根据用户的要求设计，不同用户之间有许多共同的数据，而分别保存在各个文件中，造成很高的数据冗余度，给数据的维护带来许多困难。而在文件系统中，对上述问题有较大的改进。系统设置专门的软件——文件管理系统，负责对数据进行管理，使物理数据具有较高的独立性。在数据的逻辑组织和物理组织之间由存取方法 (Access Method) 实现转换，以便数据的逻辑组织和物理组织之间可以有所区别，当物理组织改变时可不影响逻辑组织，从而提高了数据的物理独立性。在文件系统中，还提供了多种文件组织形式：如顺序文件组织、索引文件组织和直接存取文件组织。在这一阶段中，数据的逻辑组织和物理组织之间的关系可用图 7.1-2 表示。

在这一阶段，实现了以文件为单位的数据共享，但未能实现以记录或数据项为单位的数据共享，数据的逻辑组织还是根据应用要求设计的，所以数据还存在大量的冗余。

(三) 数据库系统阶段 (至目前阶段)

由于计算机工业的迅速发展，提供了大容量的直接存取设备，计算机广泛地应用于企业管理，对数据管理提出了更高的要求：要求具有更高的数据共享；要求数据具有更高的独立性，从而降低应用程序的研制、维护等的费用。但文件系统的管理数据方式还不能适应上述这些要求，因而导致了数据库管理技术的发展。数据库管理技术为用户

提供了更广泛地数据共享，为应用程序提供了更高的程序独立性，进一步减少数据的冗余度，并为用户提供了方便的用户接口等优点。下面简单介绍数据库系统阶段的特点：

1. 面向数据组织数据，提高了共享程度，减少了数据冗余度。

在文件系统中，文件一般定义为等长同格式的记录的集合。一个文件的记录与另一个文件的记录之间是无联系的。但在实际管理中，一些数据和另一些数据之间往往是有联系的。例如在学校管理中涉及到系及其下属教研组的信息管理，其中有关系的信息是系名、系主任、地点等等，其下属教研组包含教研组名、教研组主任等信息。这些数据

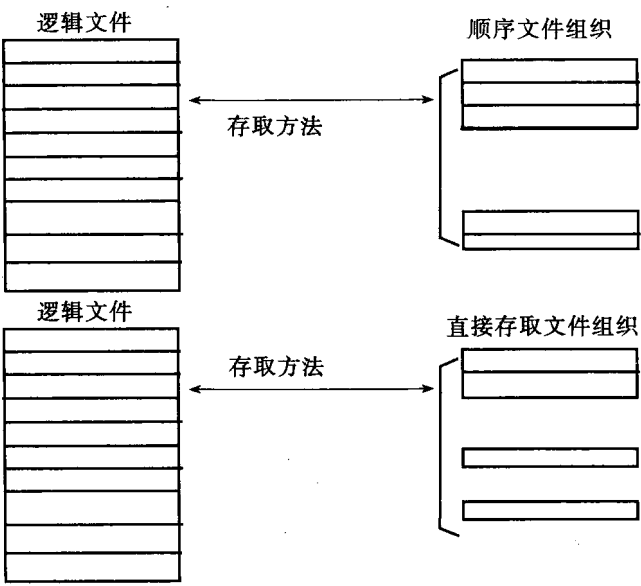


图 7.1-2 文件系统的数据组织

如何组织呢？根据文件系统所提供的功能可把它设计成一个文件，该文件的记录由一个

系 名	系主任	地 点	教 研 组		...	教 研 组	
			教研组名	教研组主任		教研组名	教研组主任

图 7.1-3 系记录的组成

系的信息及其下属全部教研组的信息组成，如图 1.3 所示。这是一个变长记录格式，记录的长度是可变的，这给存储管理带来困难。如果采用定长记录格式表示，则必须对一个系的下属教研组的最大数目加以限制，规定系的下属教研组的最大数目。这样的文件组织存在着明显的缺点：由于每个系下属教研组数目不一，但都要以最大的数目分配存储，造成了存储空间的浪费。如何解决上述问题呢？将上述记录划分成几种不同格式的记录，如可划分成两种类型的记录：系记录和教研组记录。这两种记录之间存在着联系，这种联系表示对应的对象之间的下属关系或组成关系等等。上述结构可用图 1.4 表示。在数据库阶段中常采用上述方法处理，使简单的记录结构变成由记录和联系构成复杂的结构数据。

在数据库系统阶段，组织数据是从整体角度出发的，而不是仅考虑个别应用。下面通过例子说明。

例如在学校管理中，要求所管理的信息能供领导了解各系的情况，可供教务部门了解各专业开设课程的情况，可供财务部门了解教师工资情况，可供人事部门了解师资队伍情况等。在这种情况下数据如何组织？我们应综合考虑各种要求，进行平衡，尽量减少数据的冗余度等原则来设计数据结构。根据上述要求，将数据按图 7.1-5 所示的数据结构组织。

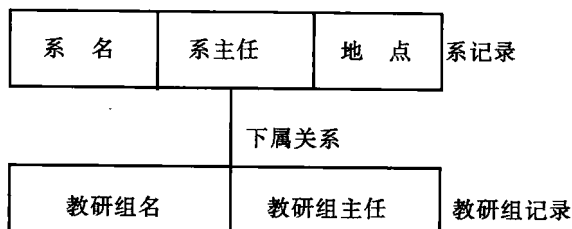


图 7.1-4 结构型的记录组织

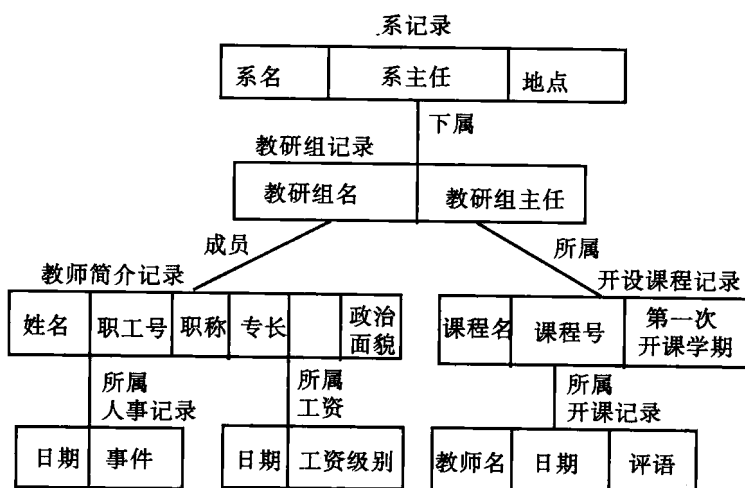


图 7.1-5 学校管理的全局逻辑结构

图 7.1-5 所设计的数据结构可满足各种不同的应用需要，如利用系记录、教研组记录和教师简介记录供各级领导了解各系情况；利用教师简介记录和人事记录供人事部门了解教师的人事情况；利用教师简介记录和工资记录供财务部门了解工资情况，利用教研组记录、开设课程记录和专业课开设记录供教务部门了解各专业开设课程的情况等等。这样复杂的数据结构可供多个应用部门使用，实现了数据共享。由于数据结构的设计是面向数据本身，所以可大大地减少数据的冗余度。数据库系统与文件系统的最大差别在于：在文件系统中各种文件的记录之间是无联系的，而在数据库系统中不同类型的记录之间允许有联系，联系反映了自然界对象之间的相互关系。联系的实现是数据库系统所需解决的问题。

反映整个数据库数据之间的逻辑关系的数据结构，称之为数据库数据的全局逻辑结构。

数据库数据可以供多个用户（或应用程序）使用，每个用户仅涉及数据库的部分数据。例如学校管理部门在使用时仅涉及系记录、教研组记录及教师简介等记录，其结构如图 7.1-6 所示。

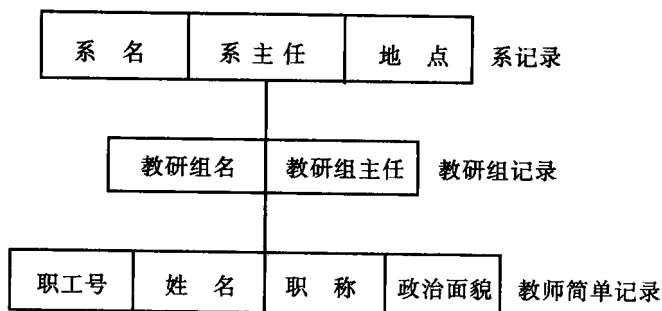


图 7.1-6 局部逻辑结构

只表示个别用户所涉及的数据结构，称为数据库数据的局部逻辑结构。在用户（或数据库应用程序）、局部逻辑结构、数据库的全局逻辑结构和数据库的物理结构之间具有图 7.1-7 所示的联系。

用户所涉及的数据库的数据文件，称为逻辑文件。它与数据库的物理组织之间有由图 7.1-8 所示的对应关系。

2. 数据库管理软件（称作数据库管理系统 DBMS）为应用程序和数据库数据之间提供了数据的物理独立性和数据的逻辑独立性。数据的物理独立性指当数据库的物理布局 and 物理组织形式改变时，不影响数据库的全局逻辑结构的性质。由于数据的物理组织不影响全局逻辑结构，当然不会影响应用程序。数据的逻辑独立性是指当数据库的全局逻辑结构改变时，不影响某些局部逻辑结构的性质。

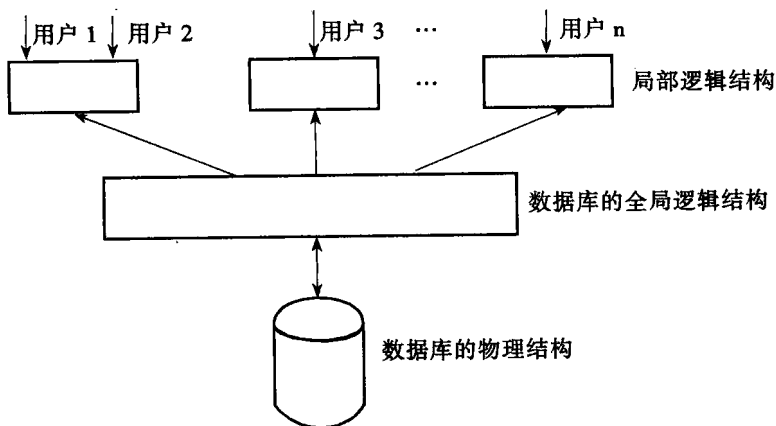


图 7.1-7 数据库的用户、局部逻辑结构、全局逻辑结构和物理结构之间的关系

辑结构改变时（如因某种要求，需要扩充一些记录的数据项或扩充一些新的记录类型），

不影响某些局部逻辑结构的性质。由于应用程序只与局部逻辑结构相联系，使用那些未发生变化的局部逻辑结构的应用程序不必修改。在数据库的局部逻辑结构与全局逻辑结构之间由数据库管理系统进行转换，这种转换称之为映象（MAPPING）。在全局逻辑结构与物理结构之间同样也存在映象。如图 7.1-8 所示。

3. 数据库管理系统为用户提供了方便的用户接口，用户可使用查询语言或简单的

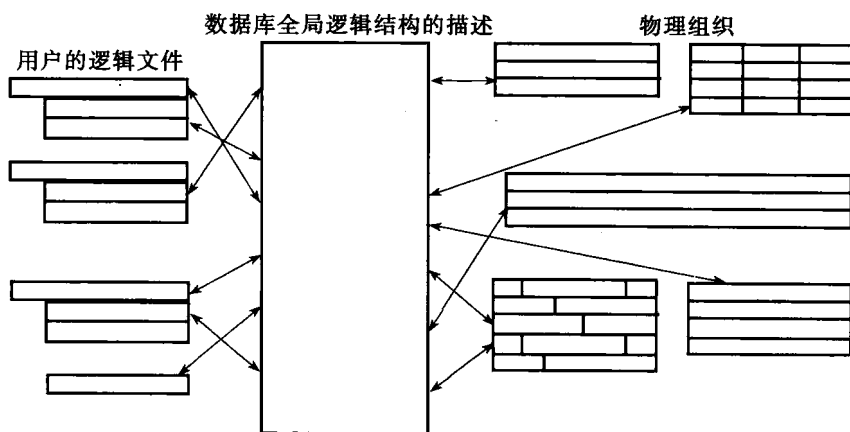


图 7.1-8 用户的逻辑文件与数据库的物理组织之间的对应关系

终端命令操作数据库，也可以用程序方式（用高级语言如 COBOL、FORTRAN 语言和数据库操纵语言编制的程序）操作数据库。

4. 提供了数据保护和并发控制的功能。由于数据库为多个用户共享，对数据库数据进行完整性、安全性控制是必要的，尤其在并行操作条件下，保证数据库数据的一致性是非常重要的问题。

数据库的安全性 (Security) 是指为了保护数据库数据而采取的措施，防止未经允许的用户去存取数据库数据，避免那些对数据库有意的或无意的破坏。一般采用口令 (Password)、密码锁和密码等方法实现数据库的保护。

完整性 (Integrity) 是指保证数据库数据的正确性、有效性的问题。可采取检验措施以控制数据在一定范围内，防止无效的修改，同时还要保证一部分数据与另一部分数据之间满足一定关系等等。一般采用完整性约束的方法来实现完整性的控制。

并发控制是为了防止由于多个用户并行地操作数据库时，他们之间的相互干扰引起数据库数据发生不一致性的问题，为此对并行操作要采取控制措施。最常用的方法是封锁技术，在分布数据库系统中还采用时间戳 (Timestamp) 方法进行控制。

根据上述分析，数据库为相关数据的集合，以综合的方法进行组织，具有最小的数据冗余度、可供多个用户共享；数据的存储不依赖于应用，数据的组织具有较高的独立性；允许并发地使用数据库，能有效地及时地处理数据。

二、数据管理在不同范围中使用的术语

(一) 三种范围

在数据处理中将涉及不同的范围，我们将通过例子说明。例如仓库管理，在仓库管理中首先涉及的是货物的管理，货物的存放，货物的进出等等，这种管理称为现实世界管理。在现实世界管理中被管理的对象称之为实体 (Entity)。实体可为实际存在的而且可区分的客观事物，如机器零件、学生、教师等等，也可为事件和抽象的概念。一个实体具有一定的特征 (性质)，如零件有颜色、重量、名称、规格等等。具有相同特征的一类实体的集合称之为实体集。例如在学校中，全体学生可组成学生实体集，所有的教

员可组成教师实体集，所有开设的课程可组成课程实体集等等。

在进行现实世界管理时，这些客观事物必然在人们头脑中产生反映，称之为信息。对这些信息进行记录、整理和归类，成为格式化的信息。如在仓库管理中可使用帐本进行管理，这种管理称为信息管理，所以信息是现实世界状态的反映，信息管理是现实世界管理的反映。要对现实世界管理好，必须借助信息管理。在这个范围内用实体记录表示实体，用实体记录集表示实体集，用属性表示实体集的特性，一个实体的特性用属性值表示，实体记录是由实体的属性值组成。

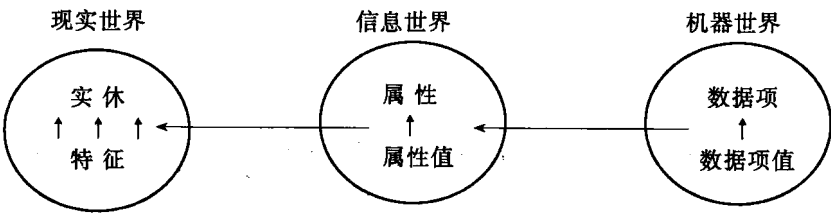


图 7.1-9 三种范围之间的对应关系

当数据管理进入计算机管理时代，这一范围称之为机器世界。由于计算机只能处理数据化的信息（即只能用字母、数字或符号表示），所以对信息世界中的信息必须进行数据化，数据化后的信息称之为数据（Data），所以数据为信息的符号表示。例如当前许多计算机系统尚未能很好处理汉字信息，所以汉字信息尚未能广泛作为数据。在本书中为了表示直观起见，有时数据用汉字表示。在这一范围内用数据项表示信息世界中的属性，实体记录的属性值用数据项值表示，实体记录用记录值或片段值表示，实体记录集用数据集表示。

数据管理所涉及的三种范围之间的联系可用图 7.1-9 简单表示。

在三种不同范围内使用不同的术语，在表 7.1-1 中作简单说明。

表 7.1-1 在三种范围中使用的术语对照

现实世界	信息世界	机器世界
系统所涉及企业单位中所有的对象（如学校管理、其对象为教职工、课程和学生等等）	条理化的信息（如学生名单册、教职工卡片、课程记载等等）	数据库
实体集（如学生实体集、教职工实体集等等）	实体记录集（Entity Record Set）	数据集（Data Set）
实体（具体的对象，如学生王方）	实体记录（Entity Record）	记录值或片段值
特征（实体集的性质，如学生的姓名、学号等等）	属性（Attribute）	数据项（Data Item）或字段（Field）
具体事物的特征（如学生王方其学号为 780178）	属性值	数据项值



现实世界	信息世界	机器世界
区分对象的特征	标识属性 (Identity Attribute)	标识码

(二) 常用术语的说明

属性 (Attribute) 为实体特征的抽象描述。如学生实体集在信息世界中如何描述呢? 一般通过实体集所具有的特征或性质来描述, 如学生的“姓名”、“学号”、“性别”、“年龄”及“政治面貌”等特征来描述, 称“姓名”、“学号”、“性别”、“年龄”和“政治面貌”为属性。

再如学生高考成绩, 可通过“姓名”、“考生报名号”、“语文成绩”、“数学成绩”、……“外语成绩”等属性描述。

属性值为属性的具体取值。例学生王方, 其“姓名”为王方, “学号”为 780178, “性别”为男, “年龄”为 23, “政治面貌”为团员, 这些具体值称为属性值。

域 (Domain) 为属性的取值范围, 称之为属性的域。例如大学生的“年龄”在 (14, 30) 范围内, 其域为 (14, 30), “性别”的域为 (男, 女)。

实体记录是由若干个属性的属性值组成的集合, 用来表征一个实体。例如 (王方、780178, 男、23、团员) 表示学生王方。

实体记录集为同类实体的实体记录的集合。

实体类型 (Entity Type) 是由描述实体的属性组成的集合。在机器世界中与此相应的术语称为记录类型或片段类型。

标识码 (Identification Key) 为唯一标识实体记录的属性或属性组。例如在学生记录集中, 学号可以唯一标识每个学生记录, 所以学号为标识码。在有些实体记录集中, 可以有多个标识码, 例如学生实体集, 假设学生姓名没有重名, 那末属性“姓名”也可以为标识码。我们经常选定其中一个, 被选定的那一个标识码称为主码 (Primary Key), 其它的标识码叫后选码 (Candidate Key)。

次码 (Secondary Key), 在实体记录集中经常选择一些不能唯一标识实体记录的属性来标识实体记录, 这样的属性称之为次码。

第二节 客户机/服务器基本概念

什么是客户机/服务器? 这是近年来众说纷纭的话题。客户机/服务器是一个描述非常通用的技术而又有些模糊不清的术语: 一个给定的计算任务以某种方式分布在多个执行点上。如果前面的描述有些模糊的话, 我表示歉意。问题在于客户机/服务器是那么深奥, 它的任何一种特性都不能轻易地漏掉。

让我们把以上定义进一步解释, 或许能够表达出它的意思。首先, 计算任务是什么? 计算任务是需要处理的特定的函数或操作。例如, 它可以是为银行所有 ATM (银行自动出纳员机) 卡客户建立列表, 或者也可以是在屏幕上画一个图框。

进一步考虑上面的定义, “分布式 (distributed)”意味着什么呢? 一个分布式的任务



是在多个点上运行的任务，任务的不同部分由几个处理过程或执行引擎来分担。

最大规模的执行点（execution point）可以是一台计算机或者是分担任务的并行计算机群。而最小的处理过程或执行点可以小到单一 CPU 的 PC 机上的单线程。

通过以上分析，就可以看出要为客户机/服务器下一个完美的定义确实有些困难。因此，我不准备进一步从文字上去讨论客户机/服务器，而是通过列出一系列客户机/服务器的实现方式，以便阐明我的意思：

- 客户机/服务器经常用于描述处理大量来自客户机的查询的中心数据库服务器（如 Sybase System XI）方案，而典型的客户机是个人计算机（PC）。处理过程的分布基于：数据库服务器将为客户机分析和取得适当的数据，而客户机负责向用户提供数据。

- Microsoft 用客户机/服务器来描述 Windows NT 的各种操作系统要素与 OS（操作系统）内核之间的关系。从这种意义上讲，Microsoft 认为：服务器是内核，它处理来自客户机诸如图形设备接口（GDI）之类的请求。事实上，正是因为使用了客户机/服务器技术，Windows NT 才较稳固，内核不易崩溃，这是因为所有其他事务都是内核的客户，即使其他任务的崩溃也不会影响到内核。

- 现在，客户机/服务器是在 LAN（局域网）上作为分发电子邮件的方式。在客户机/服务器基础上，Microsoft 的 Exchanges Server、HP 的 OpenMail、Novell 的 GroupWise 都获得了成功。

由于具有分布式任务，客户机/服务器可以十分顺利地发展为开放式系统。客户机/服务器本身就是一项技术，然而，由于采用了各种标准，客户机/服务器已经成了不同技术之间行通信的通用方法。例如，Sybase 通过 Open Server 产品实现了客户机/服务器与 IBM 主机上的老系统的通信。这些服务器起到了网关的作用，提供了主机与主机所在 LAN 和 WAN 上该主机正在为之服务的客户机之间的通信传输层。

到 Java 和 Internet 逐渐流行之前，客户机/服务器一直都是 20 世纪 90 年代人们讨论最多的技术话题。你应该自己去寻找更多的有关客户机/服务器方面的材料。附录 C “推荐的读物” 为你提供了关于客户机/服务器技术方面的更多的背景材料。

一、客户机/服务器的发展过程

自从 Oracle 在 20 世纪 80 年代后期获得了成功以来，客户机/服务器计算机经历了大约八年的飞速发展。在此期间，发生了很大变化。首先是基本技术和服务器的定义及功能的变化。例如，最初的服务器是小型计算机，而客户机则是具有字符模式界面和相对低速的 CPU 的 PC 机，尽管如此，大多数公司的 MIS（管理信息系统）工作室仍在进行客户机/服务器的设计，这些公司保留着对技术和对公司内部的可访问性的强硬的中心控制。与 IMS 数据库和主机文件系统相比，这没有给用户带来多少好处。但是，它开始了一个在组织内部进行分布式计算的新时代。

随着时间的推移，客户机/服务器中的服务器变得更能为一般的用户、小的部门和事务组所访问。MIS 组的高层向公司里员工和事务小组的技术下放必然要求简化其安装、维护及常规运行的方式。Centura Software 公司以一个简单可调度的数据库 SQLBase 成了工作组数据库的开拓者。

其他公司也认识到了这个市场，并在 Centura 之后也转入了这项工作：Microsoft 脱离 Sybase，产生了它的 BackOffice 数据库服务器和 Microsoft SQL Server；Oracle 舍弃了它

的一些技术，转向更易访问、更易安装的产品 Oracle Workgroup 2000；Sybase 通过它已有的 Powersoft（最近被 Watcom 收购），于 1995 年年底发布了它的工作组数据库：SQL Anywhere 5.0。

这些工作组数据库的成功和普及要归功于应用程序服务、网络化操作系统的成功，这些操作系统有 Windows NT Advanced Server，UNIX 和 Novell 的 NetWare 等。没有这些新的操作系统和方便的 OS 管理能力，客户机/服务器是决不会这样快地成为中心角色和获得如此的普及。

随着服务器所依托的操作系统能力的提高，数据库服务器自身的能力也在提高。Oracle，IBM，Sybase 和 Informix 早期的处理相对简单的查询的数据库服务器都依从所谓的 SQL89 标准。该 ANSI 标准描述了 DML 和 DDL 结构。它使客户机程序（例如 PC 机上的 Windows）能够在没有任何物理存储知识的情况下向服务器发送请求，同时返回与指定格式相匹配的数据。

但是，很快就感觉到 SQL89 的基本查询不能充分满足服务器上数据集的高级处理。于是各厂家服务标准转向开发自己的存储过程形式。Oracle 的 PL/SQL 作为 Sybase 的 Transact-SQL 的扩展获得了极大的成功。对 SQL 的这些扩展，使得开发者能够更加灵活地处理数据。处理工作可以根据需要，由客户机和服务器均匀分担。

二、胖客户机

第一代客户机/服务器应用程序被称之为“胖客户机（fat clients）”，原因是大量的处理工作在客户机上进行。数据库服务器起到了中心数据仓库的作用，但是基于服务器的数据处理所能使用的存储过程类型语言是十分贫乏的。这就意味着对任意一个需要执行的重复操作，客户机要对大部分工作作出反应。这是非常低效的。

例如，假如在银行应用系统中有 10,000 个帐户需要进行收支调整并为它们支付利息。一个胖客户机会向服务器申请一张账号表，然后服务器通过网络向客户机传输该表。每一行传到客户机后，其他 SQL 语句会对其进行处理并传回给服务器。在这种方式的数据处理中所花费的网络费用是巨大的。

胖客户机目前仍用在那些试图维护数据库厂商独立性的应用系统中。这些应用系统的目标是与任何专门的数据库厂家的 SQL 工具或存储过程扩展没有任何联系。相比较而言，用类似于 PowerBuilder 这样的语言编写一个应用程序和对服务器使用通常的 SQL 语句要简易得多。而先有一个简单的应用系统，然后再把存储过程移植到诸如 Sybase，Informix，Oracle 这样的数据上去的做法要复杂一些。一般情况下，胖客户机要求快速 PC 机来完成应答处理，原因是在 PC 机上的分布式处理份额很大。

在数据集和连续的数据处理请求一下子增大时，基于胖客户机的应用系统要出现可匀展性问题。

三、胖服务器

与胖客户机能上能相对立的是“胖服务器（fat server）”。所有的处理都在服务器上进行，不与客户机分担。客户机只作为一个显示工具。IBM 主机上的 3270 终端是胖服务器的典型的例子，服务器（此时它是主机）执行了除数据展示之外的所有（或几乎是所有）的功能。

胖服务器的优点是可以在一个地方强化所有的处理能力，从而简化技术上的管理。

然而，这也是它最大的缺点，当需要增加系统并发用户数时，胖服务器存在可匀展性问题。当用户数量增加时，对服务器的要求也增加，需要更大规模的服务器。

四、“两层半”和存储过程

胖客户机和胖服务器是两层（Two-tier）客户机/服务器应用模式。这是因为处理要么在服务器上进行，要么在客户机上进行，平衡工作量的能力极小。

由于存储过程的出现，在客户机与服务器或服务器群之间更加均衡地分配处理工作成为可能。客户 PC 机请求执行的存储过程在服务器上全速运行，这样就可以获得性能上的优势。同时，若需要进行特定的分析（有些事情是胖客户机所擅长的），则仍旧可以在 PC 机上进行处理。

存储过程能够将事务的逻辑处理放在服务器上，在智能客户机应用方面具有巨大的意义。这种分布处理的方法就是众所周知的“两层半”客户机/服务器。半层是指事务规则已被封装成服务器上的存储过程。

五、在应用程序服务器使用三层客户机/服务器

如果对事务要采用逻辑的存储过程模式并进一步发展，则存储过程本身就成了一个独立的服务器：一个应用程序服务器。这就是三层客户机/服务器。

第一层是客户机。它负责基本的可在客户机上执行的规则验证、数据描述和显示以及查询生成。

第二层是应用程序服务器。它存储着应用程序的所有事务规则。客户机向应用程序服务器发出处理请求，然后应用程序服务器负责与数据库服务器打交道。

第三层是数据库服务器。对于所有特定应用程序数据和以存储过程形式表现事务规则的某些特定厂商的实用工具，它起到了中心数据仓库的作用。

三层计算模式的主要优势是处理分布进一步均衡。用于特定的事务处理的大型计算机或专业计算机可用于完成第三层的处理业务，让数据库管理数据而不是使用存储过程。另外，第三层可以在无需客户机知道的情况下处理来自多个数据源的数据。从客户机的角度看，它只请求了一个在应用程序服务器上执行的事务，而应用程序服务器则对它所有的各种资源的管理作出反应，完成事务处理并向客户机返回适当的信息。

三层计算模式之所以流行，主要原因是为了使原有系统与客户机/服务器应用程序能够实现有效的结合。各家公司使用三层结构解决了数据访问问题，从而无需在客户机/服务器网络上向相应的数据库传递大量的数据。第三层从数据和事务规则中抽象化了客户机，如果考虑到数据格式或存储变化，或内部使用数据的方法发生变化，则这一点是一个巨大的进步。

客户机/服务器技术发展的最后一步是对等层到对等层（peer-to peer）处理模式。在这种模式中，网络上的每台计算机的作用都是相同的：所有的计算机既是客户机也是服务器。

在被专业计算机组织接受为通用方法之前，对等层到对等层模式有了一些发展，但是，随着 Internet 的迅速发展和网上计算资源的多样化，它就成了必然要到来的东西。

第三节 数据库历史简介

具备一些数据库发展的背景知识和数据库理论有助于你理解 SQL 是如何工作的。数据库系统存储着各行各业的信息。从大型数据库，如飞机订票系统，到孩子们收集棒球卡片的系统，数据库系统存储并分布着我们所需要的各种数据。直到前几年，大型数据库系统还仅能在大型计算机上运行。通常，这些大型计算机系统的设计、购买与维护费用是十分昂贵的。然而，今天的功能强大、价格低廉的工作站的产生使得程序员能够设计出易于维护、分布数据迅速、低价的软件。

数据库模型的十二条准则

关系数据库是最通用的一种数据存储模型，它是由 E. F. Codd 博士于 1970 年在一篇名为“一种存储大型共享数据的关系模型”的富有创意的论文中提出的。SQL 语言采用了 Codd 博士为关系数据库模型定义的 13 条原则（很奇怪，人们通常称为 Codd 十二条准则）。

0. 一个关系型 DBMS 必须能完全通过它的关系能力来管理数据库。

1. 信息准则——关系型数据库（包括表和列名）的所有信息都被清楚地表示成表中的数值。

2. 保证访问准则——保证关系数据库中的每一个数值都可用表名、主键和列名的组合来访问。

3. 支持系统空值——DBMS 对空值（未知或不可使用的数据）应提供系统支持。空值不同于缺省值，它独立于任何域。

4. 活动的、联机的、关系型的数据字典——在逻辑上，数据库的描述及其内容都被表示为表的形式，并能用数据库语言进行查询。

5. 统一的数据子语言——至少有一种支持语言，该语言应具有严格、统一的语法格式。它必须能够支持数据定义、数据操作、完整性规则、授权和事务处理。

6. 视图更新准则——所有理论上可更新的视图也可以被系统更新。

7. 集合级的插入、更新和删除——DBMS 不仅支持集合级上的检索，还应支持集合级上插入、更新与删除。

8. 物理数据的独立性——当数据的存储结构或数据的物理存取方法改变时，应用程序和终端活动程序在逻辑上应不受影响。

9. 逻辑数据独立性——当改变表的结构时，应用程序和终端活动程序在逻辑上应尽可能地不受影响。

10. 数据完整性的独立性——数据库语言必须能够定义完整性规则。这些规则必须存储在联机数据字典中，不能被忽略。

11. 分布独立性——当首次引入分布式数据或数据重新分布时，应用程序和终端活动请求在逻辑上应不受影响。

12. 无损害准则——决不能用一种低级的语言绕过用数据库语言定义的完整性规则。



大多数数据库具有一种“父/子”关系，即父结点有一个指向子结点的指针（如图 1.10 所示）。

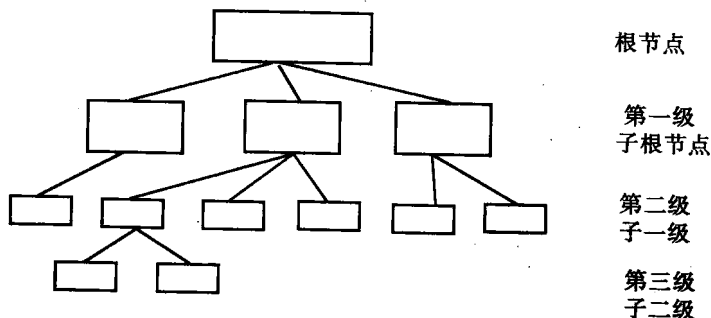


图 7.1-10 Codd 博士的关系数据库管理系统

这种方法既有优点也有缺点。有利之处表现为数据在磁盘上的物理结构显得并不重要，程序员只须简单地将指针指向下一个位置，即可访问所需数据。数据的插和删除都很容易。然而不同组的信息不能方便地同新的信息相连接。数据库创建之后，数据在磁盘上的格式不能被任意改变。若要改变，则只能通过创建一个新的数据库结构来实现。

Codd 为 RDBMS 提供的思想是使用关系代数的数学概念将数据划分成集合和相关的公共子集。

由于信息可以自然地分成不同的集合，Codd 用集合来组织数据库系统。在关系模型下，数据被分成类似于表结构的集合。这个表结构由不同的数据成分（称作列或字段）组成。一组字段的单一集合被称做记录或行。例如，为建立一个由雇员组成的关系数据库，需要先创建一个名为 Employee 的表，该表包括 Name、Age、Occupation 等信息。这三类数据构成了表 Employee 的三个字段（如表 7.1-2 所示）。

表 7.1-2 EMPLOYEE 表

Name	Age	Occupation
Will Williams	25	Electrical engineer
Dave Davidson	34	Museum curator
Jan Janis	42	Chef
Bill Jackson	19	Student
Don DeMarco	32	Game programmer
Becky Boudreaux	25	Model

表 7.1-1 中有 6 条记录。要从此表中检索特定的记录，例如检索 Dave Davidson，用户要指示数据库管理系统去检索 Name 字段的值等于 Dave Davidson 的记录。如果 DBMS 被指示检索此记录中的所有字段，则雇员的姓名、年龄和职业都将提供给用户。SQL 是一种告诉数据库去检索数据的语言。下面是一个用 SQL 语句进行这个查询的例子：