

面向对象的新一代数据库系统

MIANXIANG DUIXIANG DE XINYIDAI SHUJUKU XITONG

雷光复 编著

国防工业出版社



1.132.4

面向对象的新一代数据库系统

雷光复 编著

国防工业出版社

·北京·

图书在版编目(CIP)数据

面向对象的新一代数据库系统/雷光复编著. —北京:
国防工业出版社, 2000. 1
ISBN 7-118-2196-2

I. 面… II. 雷… III. 数据库系统 IV. TP311.13

中国版本图书馆 CIP 数据核字(1999)第 46443 号

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京怀柔新华印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 16¼ 372 千字

2000 年 1 月第 1 版 2000 年 1 月北京第 1 次印刷

印数: 1—3000 册 定价: 22.00 元

(本书如有印装错误, 我社负责调换)

前 言

到目前为止,数据库仍然是数据管理最有效的方法之一,其中的关系型数据库管理系统已成为最实用且普及程度最广的数据库。在其持续了近 30 年的发展历程中,数据库追求的性能指标始终是数据类型的多样化和查询功能的灵活性、智能化。从目前的情况看,无论是市场的需求,还是技术条件的成熟,基于 Web 的对非传统的复杂数据类型的存储、处理和查询已成定局,并将成为 21 世纪数据库应用的主流技术。这使得数据库技术与面向对象技术结合形成的新一代数据库系统成为发展的必然趋势。从对新技术应用的角度看,这类新型数据库系统是对目前的三大热点技术——Web、超媒体及面向对象技术综合应用的典范。

目前出版的同类书籍要么是着重于理论,属于导论性质的;要么是产品介绍,属于手册性质的。本书采用综合的方法,围绕面向对象技术在数据库中的应用,既有基本概念和基础理论的原理性,又有使用一个典型产品的实用性;同时,融会了三大热门技术的综合应用,体现了理论联系实际的原则。另外,通过对数据库发展历程的回顾以及对未来的前瞻,使本书站在一个新的高度客观地审视数据库的发展趋势。

概括本书的特点有以下几个方面。

(1)先进性 以新一代的数据库系统为核心,论述本学科及其相关应用领域中最先进的观点、方法、理论和技术。

(2)系统性 对数据库技术的论述从发展的历程到市场的需求,从基本技术到实际应用,从基础理论到实现机制,从产品介绍到具体使用,力求体现方法与结论相结合,历史与逻辑相结合,理论与实际相结合的原则。

(3)通俗性 用浅显通俗的语言阐述复杂技术的原理,概念清晰易懂,论述深入浅出。

(4)实用性 对每一种新型数据库都附有一个典型产品,以使读者阅后能达到实际应用一个数据库管理系统的程度。

本书在编著过程中得到王启智教授及系统工程师丁卫东先生和孟成女士的大力支持;王秀彩完成了全书的录入排版。在此一并致谢!

由于作者本身水平有限,书中难免存在一些缺点和错误,殷切希望广大读者批评指正。

第1章 数据管理的发展历程

1.1 数据结构的演变

计算机对数据处理的依据,或对数据存储和管理的基础是数据之间的内在联系,即数据的逻辑结构,简称数据结构。数据结构的基本内容按照对数据的组织有以下三步:第一步,分析、确定数据的逻辑结构。第二步,把从逻辑上组织好的数据存入计算机,即物理组织;数据结构在计算机中的表示称数据的物理结构或数据的存储结构,它包括对数据的存储和数据之间关系的存储。第三步,对存储在计算机中的数据结构进行运算或操作,故要寻求一组数据操作的算法;这些算法是定义在数据的逻辑结构上的,而算法的实现是在数据的存储结构上进行的。

1.1.1 文件系统

50年代后期到60年代中期,计算机对数据的管理是通过文件系统实现的。程序员必须通过文件系统来管理应用程序中的数据。1954年,美国通用电器公司(GEC)研制出第一台商业数据处理的电子计算机 UNIVAC I,它标志着计算机开始进行事务处理并存储信息文件,由此产生了电子信息系统。

1. 文件

对文件的最初定义是:以某种目的收集起来的数据的有组织的集合。在后来的使用中,人们习惯地将存储在外存中、并有一定标识的数据的集合都叫做文件;这里的数据除了指那些被收集的数据外,还包括处理这些数据的程序,这可以看作是广义的文件。

对文件中的数据记录可以进行读取、更新、追加/插入和删除,这也就是文件的处理方式。

根据数据存储结构(逻辑记录的安排形式)的设计,产生了对文件的不同的组织方法。记录的逻辑顺序与它存放在存储设备上的物理顺序一致的文件叫顺序文件,特点是宜于顺序存取和成批处理;记录的逻辑顺序与物理顺序不一致,而是通过在每一个记录中保存另一记录的地址,从而将记录串联起来的文件称链接文件,在实现对数据记录的增、删方面,它比顺序文件要灵活得多;带有记录索引的文件称索引文件,它与上面两种文件的不同之处是能够随机地查找记录,从而大大提高了查找记录的速度;通过对记录的关键字进行某种计算,直接产生存取该记录的存储地址的文件称直接存取文件。

2. 文件系统

将用户的数据和程序按一定的物理结构形式组织成相互独立的数据文件,利用“按文件名访问,按记录进行存取”的管理技术,为用户提供对文件的存取、修改、插入和删除等命令的一种数据管理软件即文件系统。

3. 文件系统的特点

(1) 使用简便

用户只需通过文件命令实现按文件名的数据存取,而无须了解具体的存储设备的特性,也不用编制烦琐的输入/输出指令。

(2) 数据可靠

文件系统可以采用定期转储、复制副本等方法保存文件,以便在发生各种故障时,及时恢复。

(3) 既可以实现共享又能保证安全

文件系统可以给出一个文件的使用权限,通过对用户使用权限的检查,保证授权的用户在规定权限内共同使用有关文件。

文件系统的局限性在于只实现了记录内的结构性,而没有作为一个整体的文件系统的结构性,即不能反映和处理文件之间在内容方面的联系,只有文件的共享,而无数据的共享,即数据面向应用,独立性差,冗余度大。也就是说,要根据具体应用建立数据文件,并编写处理这些文件的应用程序;一旦应用有了变化,就要重新建立数据文件和编写新的应用程序。因此,文件系统管理和维护的开销很大。由于文件系统只具备通过打开/关闭文件和从文件中读/写一个记录提供的对简单数据的存取功能,在数据量小、数据变化不大时尚能维持,反之就显得力不从心了。尽管如此,文件系统毕竟为数据库系统的发展奠定了基础。

1.1.2 数据库系统

从客观上看,数据是经常改变的,且数据之间存在着这样那样的联系。随着应用的发展,要求存储和管理有结构的数据,即不但管理数据本身,还要管理数据间的联系,以便实现按数据间的联系进行的导航式的查询,使大量的查询应用不必通过应用程序而由系统直接满足。于是,60年代中期产生了数据库系统。与文件系统的区别是数据库系统管理的数据是有结构的,从而提供了强有力的数据查询功能。

1. 数据库和数据库系统

数据库(Data Base,简称 DB)是相互关联的数据(Data)的集合。利用数据库管理数据的方法可概括为:把某系统要用到的各种数据综合整理为一组相互关联的文件而实现数据共享的方法;汇集许多应用程序共同使用的数据,形成数据库;使用一组特定的软件——数据库管理系统(Data Base Management System,简称 DBMS)对数据库进行统一管理,以处理所有用户对数据库的存取请求,加上用户在数据库管理系统上开发的应用程序。这三者组成一个数据库系统(Data Base System,简称 DBS)。所以,数据库系统是一个用来存储、管理和恢复集中或分布存储的数据的软件系统。一种更完整的说法还应该包括数据库管理人员和用户。

2. 数据库系统的特点

(1) 数据共享性

数据的共享性体现在数据库中的同一数据可被不同用户不同程序地使用。它的意义就在于多种应用、多种语言、多种用户可以相互覆盖地使用数据集合。

(2) 数据独立性

这是指用户的应用程序与存储在磁盘上的数据库中的数据是相互独立的,即用户不需要了解数据在数据库中的物理存储,这由数据库管理系统去管理,应用程序要处理的只是数据的逻辑结构。这就使得数据的物理存储,甚至逻辑结构改变时,应用程序可以不变。

(3)数据最小冗余度

这是指尽可能地减少数据的重复。数据库系统从整体角度看待和描述数据,这意味着数据不再面向某个应用,而是面向整个系统,从而大大减少数据冗余,节约了存储空间,避免了数据之间的不一致性。

3. 数据库管理系统的功能

(1)数据库定义功能

数据库管理系统为数据库的建立提供了数据定义语言(DDL)。用户使用 DDL 可以定义数据、数据子模式和数据存储结构,以及保密定义和信息格式定义等。

(2)数据库操作功能

数据库管理系统为数据库的使用和维护提供了数据操纵语言(DML)。DML 一般是嵌入像 COBOL 这样的主语言的,即在主语言中增加了 DML 语句。数据操纵语言的语句大致可分为四类:控制语句、修改语句、存储语句和检查语句。

(3)数据库维护功能

数据库管理系统提供了对数据库的建立、更新、重编、结构维护、恢复及性能监测等功能。

(4)通信功能

数据库管理系统提供数据库与操作系统的联机处理接口,以及与远端作业录入等处理的相应接口。

4. 数据结构

数据结构就是研究数据的逻辑结构和物理结构以及它们之间的相互关系,对这种结构定义相应的运算和设计相应的算法,而且确保经过这些运算后所得到的新结构仍然是原来的结构类型。数据结构是组成数据模型的核心部分,可分为层次型、网状型和关系型。

层次数据库产生于 60 年代中期,是数据库系统的先驱,最典型的代表是 IBM 研制的,在美国诞生的第一个商品化的数据库系统——IMS(Information Management System)。网状数据库的典型代表是 DBTG 系统,也称 CODASYL 系统,这是 70 年代数据系统语言协会(Conference On Data System Language,简称 CODASYL)下属的数据库任务组(Data Base Task Group,简称 DBTG)提出的一个系统方案,是数据库概念、方法、技术的奠基。它们是数据库技术中研究最早的两种数据库。当时,数据库系统大都采用层次数据模型或网状数据模型。

从文件系统到数据库系统,标志着数据管理技术质的飞跃。数据库系统的出现使信息系统的研制中心从加工数据的程序转向共享的数据库。这样,既便于数据的集中管理,又利于应用程序的研制和维护,从而提高了数据的利用率、相容性和决策的可靠性。

但是,层次和网状数据库都是非关系型的,主要的缺点首先是结构比较复杂,而且随着应用环境的扩大,数据库的结构会变得更加复杂,不利于最终用户掌握;其次是操作比

较复杂,编写应用程序时,用户必须选取适当的存取路径,因此必须了解系统结构的细节,从而加重了编写应用程序的负担。还有一点不能忽略的是,非关系型的数据库系统只能使用过程化的查询语言,实现简单查询。

1.1.3 关系数据库系统

70年代初,提出了关系数据理论,建立了关系数据方法。1970年IBM公司的研究员E.F.Codd发表题为《大型共享系统的关系数据库的关系模型》的论文中首次提出关系模型,继而诞生了关系数据库管理系统(Relational Data Base Management System,简称RDBMS),主要的研究工作是由IBM实施的。关系模型是数据库模型中最重要的模型。与以往的模型不同,它是建立在严格的数学集合论基础上的,数据结构简单、清晰。

1. 关系的概念

关系模型把一个研究对象用一个二维表来表示,也就是将数据以二维表的形式进行存储,这就是关系模型的逻辑结构。二维表的一行称作一个记录或一个元组,表示一个描述的对象;二维表的一列称作一个字段,表示对象的一个属性。这样的二维表或元组的集合称为“关系”。表之间的数据联系通过一个表的码与另一个表的外码的连接来体现。

2. 关系数据语言

关系数据库系统提供了一种十分方便、易用的非过程化语言——结构化查询语言(Structured Query Language,简称SQL),它是一种在关系数据库中定义和操纵数据的标准语言。它将各种独立功能,如查询、数据的定义、修改和控制集成在单一语言环境中,从而使数据库的查询功能大大增强。这是关系数据库系统通用性的体现,也是与其他数据库系统的主要区别之一。关系语言的特点是高度非过程化,只要说明“做什么”,而不必说明“怎么做”,所以用户不必请求数据库管理员建立特殊的存取路径,而是由数据库管理系统来完成存取路径的选择。事实上,SQL强大的查询能力是关系型数据库系统在企业中得到普遍采用的强大推动力。

3. 关系数据库系统的特性

关系数据库系统提供了并发控制、错误恢复、对数据伸缩性的适应和强大的查询能力。可以说,关系数据库系统最重要的特性不是存储数据的能力,而是它们检索数据的能力。正是由于这种灵活的检索能力,使得它有别于第一代的数据库系统和文件系统。作为应用程序共享企业信息的核心,关系数据库提供了一种强有力的高级查询语言SQL,并提供了访问控制手段和机制——存储过程、完整性控制、触发器等功能。

自80年代以来,关系数据库以良好的可操作性和优越的性能价格比逐渐取代了50年代和60年代的文件数据库、层次数据库、网状数据库等其他种类的数据库,成为数据库的主流技术。计算机厂商新推出的数据库管理系统的产品几乎都是支持关系模型的。著名的商品化关系数据库管理系统有:DB2、ORACLE、INGRES、SYBASE、INFORMIX等。关系数据库系统的出现使数据库的应用达到了空前的普及。80年代后,不仅在大型机上,在多数微机上也配置了数据库管理系统,使数据库技术成为实现和优化信息系统的基本技术。

数据库技术发展到了关系数据库系统,标志着它的成熟阶段。概括数据库技术的主要成就是数据模型的建立和数据库的数学理论基础。建立在牢固的关系代数和数学计算上的数据库关系模型和语言给用户提供了建模、查询和操作数据库的命令和语言。尽管如

此,关系数据库终究是比较简单、单纯的关系数据库管理系统,还只能管理简单对象及对数据进行简单操作,不能满足日益复杂的应用需求。于是,数据库研究者开始寻求新的数据模型——语义数据模型和复杂对象模型。

1.1.4 面向对象的数据库系统

在计算机辅助设计(CAD)、计算机辅助制造(CAM)、计算机辅助软件工程(CASE)、图像处理、地理信息系统(GIS)等新的应用领域,以及管理信息系统(MIS)、办公自动化(OA)等传统应用领域中,面临的共同问题是大型工程复杂数据的管理问题。单纯依靠传统的数据库系统是难以胜任的。因此,把先进的面向对象机制和方法与数据库技术相结合而形成新一代的面向对象的数据库管理系统是面向对象技术和数据库管理系统发展的必然产物。由于新一代应用系统的需求,计算环境性能的提高,面向对象程序设计语言和面向对象方法学的普遍使用,80年代中后期产生了将面向对象技术与数据库技术结合的面向对象数据库管理系统(Object-Oriented Data Base Management System,简称OODBMS)。它以一种面向对象的语言为基础,增加数据库的功能,用以支持持久对象和实现数据共享。由于建模和处理能力大大提高,面向对象的数据库技术已经可以处理企业内复杂和变化的事务对象。

OODBMS是具有变革性的、新一代的数据库系统,尤其是在复杂系统的模拟、表达和处理能力上可算是独树一帜,故被称为数据库史上的一场革命。

1. 新一代的应用系统对数据库提出的要求

(1) 支持复杂的数据类型

新一代的应用领域的数据往往是几十种乃至上百种元素组成的对象,具有许多传统数据库中没有的数据类型,如变长数据、非结构化字符串、图形、图像、声音、动画信息等。而这些信息在传统的数据库中一般是难以描述、操纵和检索的。

(2) 支持复杂的数据结构

工程设计往往是采用分层的设计方法(通常有自顶向下和自底向上两种),分层设计不仅要求有层次的结构,而且对于不同的设计对象,层次的数目也不一样。例如超大规模集成(VLSI)设计,一个单元设计就可以分为三个层次:功能描述(Functional Description),逻辑描述(Logic Description),布局描述(Layout Description)。而传统数据库无法提供揭示数据层次的物理存储结构和方便地处理这种层次的手段。

(3) 对象实例和类的演进

工程设计是一个渐进的过程,特别是在大规模的应用中,类(相当于关系数据库中的关系)是随着环境、需求、研究人员研究的深入而发生变化,这些变化称为模式演进。持久性对象也同样存在版本的管理。

(4) 支持工程事务管理

工程事务的特点表现为长时间(几小时甚至几天)、协作性(多个设计人员协同完成)和试探性(不是一次成功)。而传统数据库的并发控制策略无法满足这类事务的要求。

(5) 版本和配置管理

工程设计是一个试探性的过程,具有反复性、尝试性和发展性,需要有对象的历史记录 and 同一对象的不同设计方案,这就是所谓的版本。一个复杂的对象由许多子对象构成,

而这些子对象又有多个版本,这就需要用版本配置来组合成一个合理的复杂对象。

(6) 导航式查询和关联查询访问

复杂应用中的对象关系是大型的图状结构,需要系统提供在对象图中的高效导航能力,特别是多媒体对象实时性强,要求更高的交互式性能。另一方面对于大型应用,通过关联查询来访问对象也是必须的,例如在软件工程的应用中查询调用某函数的所有模块。

(7) 支持分布式系统中独立于平台的大型对象的存储

新一代的应用领域表现为数据密集,例如一幅分辨率为 1024×768 , 256 色的图像,就需要占用 768KB 的空间,而类似的多媒体对象在数据库中可能是成千上万,所以系统提供有效的存储和控制手段是十分必要的。新一代的应用系统多运行于分布式环境中,要求对象的存储独立于产生它的平台,可供不同的硬件和操作系统平台共享。

(8) 对约束、触发、规则等方面的支持

新一代应用系统的特点是对象的结构和操作非常复杂,有着丰富的语义信息。为了保证数据库的完整性和一致性,引入约束机制是非常必要的。

2. 面向对象系统的机制

面向对象数据库系统之所以能够满足新一代应用系统的要求,而且在性能上大大优于传统数据库,在于它的几个关键性机制。

(1) 内容管理

面向对象系统中与复杂对象有关的数据被当作一个整体访问和存取,而无须众多的二级存储和重新构造数据的关系节点,即在存储和检索对象时无需对象的拆卸与装配,从传统数据库的数据管理转向面向对象系统的内容管理。

(2) 面向对象的识别技术

对象的虚拟和物理地址索引图为数据的恢复和操作提供了有效的方法。

(3) 程序对象

在面向对象的数据库中,任何被开发的应用都成为目标库的一部分,被用户和开发者共享,即程序/方法与数据一样被当作对象由数据库管理系统统一管理,使数据库中的程序和数据可以真正地共享,从而缩小了数据库与应用程序之间的差距,降低了应用系统的开发费用,同时也减小了系统出现问题的可能性。

3. 面向对象数据库系统的概念

一个面向对象的数据库系统应该满足两个准则:首先它是一个数据库系统,具备数据库系统的基本功能,例如查询语言、散列或成组存取方法、事务管理、一致性控制及恢复等;其次是一个面向对象系统,是针对面向对象的程序设计语言的永久性对象存储管理而设计的,充分支持完整的面向对象概念和机制,例如用户自定义数据类型、自定义函数、对象封装等都是必不可少的。所以,可以将一个面向对象的数据库系统表达为:面向对象系统+数据库能力。实际上,它是一个将面向对象的程序设计语言中所建立的对象自动保存在磁盘上的文件系统。一旦程序中止后,它可以自动按另一程序的要求取出已存入的对象。所以,面向对象数据库是一种系统数据库,它的用户主要是应用软件和系统软件的开发人员,即专业程序员,而不是最终用户。这类系统的好处是可与面向对象程序设计语言一体化,使用者不需要学习新的数据库语言。

4. 面向对象数据库系统存在的问题和局限性

(1)没有统一可行的标准

目前,面向对象数据库还缺乏坚实的形式化理论的支持,没有一个切实可行的标准,缺乏数据库的许多功能,以致产品之间的兼容性和可移植性比较差,这已成为它广泛应用的主要障碍。此外,面向对象数据库涉及程序设计语言和数据库系统集成这一复杂问题,更需要用标准来加以约束。所以,为了使面向对象数据库的产业健康发展,增强用户的信心,制定标准已刻不容缓。

(2)与关系数据库系统不兼容

面向对象数据库管理系统可用于复杂数据的应用程序,所以对 CAD 这样一些特定应用领域,能较好地满足其应用的需求。但是,它不能与关系数据库系统兼容,没有提供使关系数据库和面向对象数据库相互转换的“通道”或“桥”,以便用户可以根据需求,发挥这两种数据库各自的优势。同时,它也缺乏关系数据库系统成功使用的一些功能特点,如非过程化查询语言、视图、授权等,在需要利用传统的关系数据库系统功能完成面向对象的工作时,它就无能为力了。另外,它需要与 SQL 不同的应用程序接口(API),故不支持 SQL,因此失去了通用性方面的优势,使其应用领域受到很大的局限。

为了区别于对象-关系数据库系统,将面向对象数据库系统称作纯粹的面向对象数据库系统,或称第一代面向对象数据库系统。典型的商品化的面向对象数据库系统有: ONTOLOGIC 公司的 ONTOS DB,另外还有一些新兴的规模不大的公司推出的若干个 OODB 产品,例如 Gemstone、Objectivity、ObjectStore、O2、Poet、Versant 等。

1.1.5 对象-关系型数据库系统

关系数据库管理系统具有优异的企业级数据库管理及异种数据的访问功能,可以在多个站点中分发数据,同时保持数据的完整性,并适用于管理数据库环境的集成化工具,又擅长于快速高效的查询;但是,缺乏对大型数据类型应用程序的支持。面向对象的数据数据库系统在描述这些复杂数据类型方面得心应手,却不能利用关系数据库系统的成功之处。目前还有许多应用是关系数据库管理系统和面向对象数据库管理系统各自能力无法满足的。比如网络管理应用、模拟复杂金融手段的贸易系统、大规模事务的面向对象应用以及使用多媒体数据的复杂文件管理系统等。面对这种形势,用户需要的是透过单一的数据库平台高效地实现多种需求,其中包括所有应用程序的良好性能、集成化访问传统和复杂数据以及支持特殊业务需求的可扩展性。所以,在面向对象技术与数据库技术相结合的过程中,同时又形成另一发展途径。这就是从传统的关系数据库加以扩展,增加面向对象的特性,把面向对象技术与关系数据库相结合,建立对象-关系型数据库管理系统(Object Relational Data Base Management System,简称 ORDBMS)。两者的结合将可以较好地实现对复杂数据类型的快速高效查询。

1. 对象-关系型数据库系统产生的背景

从关系数据库系统到面向对象的数据库系统,数据管理领域正经历着一种阶段性转移,其实质是寻求将对象与关系于一炉的系统支持平台。驱动并加速这一转移的背景如下。

(1)数据类型及使用方式的变化

随着数据日益成为企业的重要资产,企业越来越需要管理大量的非传统类型的数据对象,包括文档、图像、声音、视频、时序性和空间性数据;并且需要把这些数据对象与传统

的数据类型相关联并进行有效的访问。为了做到这一点,数据库需要提供综合性数据检索的能力,要求 SQL 查询既适用于传统的数据类型,又适用于这些新的复杂数据对象。面向对象数据库技术中所使用的描述对象属性的方法非常适合于关系数据库模型。它能精确地处理现实世界中复杂的目标对象,将它们定义为抽象的数据类型,而且在系统运行时对它们的内含进行定义和检查。

(2)管理动态内容

最初的商业数据类型是稳定的,但电子贸易出现后,要求处理动态改变的数据,或者结构难以预测的数据。在关系数据库管理系统的优先目标中,不曾考虑过对动态内容的管理。面对这类数据,通常是将其视作二进制大对象(BLOB)处理。遗憾的是,这类 BLOB 没有内部结构,因而不可搜索,不能加索引,也不能比较。

(3)非数字化数据处理

今天,一些新的数据库应用,例如金融服务业的决策辅助、产品广告、电影制作、儿童动画、医学影像处理、卫星遥感资料处理等,正以前所未有的速度进入主流数据库应用领域。据国外的市场调查,在全部等待计算机处理的数据中,数字化的数据只占 15% 左右,其余 85% 左右的数据都是以原始形式存在。现在几乎是社会各界都迫切需要一个能全方位支持数据处理的系统平台。

2. 对象-关系型数据库系统的概念

对象-关系型数据库系统是支持 ANSI SQL 的第二代面向对象的数据数据库系统。它的目标是既支持复杂对象和复杂对象的复杂行为,使关系型数据库管理程序能够处理新的数据类型,允许在数据库中添加更多的业务和应用程序逻辑;同时又不会破坏关系数据库管理系统的优势——SQL 的强大功能、数据的完整性、联机事务处理(OLTP)性能、支持大数据量、并行控制、备份及恢复应用程序、可用性及对业界标准的兼容。

(1)第三代数据库系统的三条原则

由 Stonebraker 等组成的 DBMS 功能委员会在 1990 年发表的《第三代数据库系统宣言》中提出了第三代数据库系统的三条原则:支持更加丰富的对象结构和原则;包含第二代(关系的)DBMS;对其他子系统开放。根据这三条原则,对象-关系型数据库系统将关系模型与面向对象程序设计语言中的核心概念加以综合。这些核心概念包括将数据和程序封装到对象中、对象标识、多重继承、任意数据类型、嵌套对象等。对象-关系型数据库系统分别将传统关系型语言 ANSI SQL 和调用级界面扩展为面向对象的 SQL 和相应的调用级界面。应用开发人员可在任何宿主语言中使用此面向对象的 SQL 及调用界面。它的好处是将面向对象技术引入到数据管理的非面向对象编程语言中来,并且将面向对象程序设计语言界面层与对象-关系型数据库系统结合起来,实现了强大的功能。此外,尽管这类数据库系统并不依赖某种特定的面向对象的程序设计语言,但利用后者在此数据库中开发一个界面层并非难事。这个界面层和位于其下的对象-关系型数据库系统相结合,就形成了一个带有全部数据库特性的永久存储管理器。

(2)新的关系模型

为了满足新一代应用的需求,关系数据库为其关系模型引入三项新技术。

①二进制大对象。这是关系模型的非结构化的成分,用于存储变长字符串和二进制数据。这项技术是 DEC 公司在 1981 年引入的。以后 Oracle 和 Informix 又用它来处理多

媒体对象;实际上它仍然只能处理较小容量的数据。

②分布式数据库。这项技术由 Oracle 和 Ingres 在 80 年代中期引入,它允许数据和数据的处理分布在不同的机器上。

③用户定义类型和抽象数据类型。从 80 年代末期开始,许多关系数据库,如 DBaseIV、Ingres 等开始允许用户用第四代语言来定义自己的函数和抽象数据类型,并且可以嵌入到数据库系统中去;但是可定义函数和抽象数据类型的种类和数量有一定的限制。

3. 对象-关系型数据库管理系统的特点

概括地说,对象-关系型数据库管理系统应具有以下主要特点。

- ①可扩充的数据类型和函数。
- ②扩充的数据访问方法。
- ③对数据的查找和 SQL 访问。
- ④按内容查询。
- ⑤对数据库有多种界面。
- ⑥健全的数据库管理功能。
- ⑦异构数据访问。

典型的对象-关系型数据库管理系统产品有:Informix 公司的 Informix Universal Server、Oracle 的 Oracle 8、Sybase 公司的 Adaptive Server、IBM 公司的 DB2 UDB、微软公司的 Microsoft SQL Server 等。

对象-关系型数据库管理系统既适应了传统应用领域深化的趋势,又满足了新应用领域发展的需求,因而获得了快速的发展。

综观数据库管理系统的发展,其追求的主要性能指标是数据类型的复杂化和可扩充性,以及查询功能的强大。

1.2 数据库系统应用环境的演变

人们常把信息设施比做“路”和“车”,把信息资源比做“货”。在信息系统中,“路”由各种类型的网络构成,“货”则来自网络上大小数据库中的数据。这就注定了数据库与网络之间的依存关系。数据库的运行离不开网络这个环境,网络的应用依靠数据库系统的支撑。数据库技术的发展和网络技术的发展相互渗透、相互促进,成为当今计算机学科中发展迅速、应用广泛的两大领域。硬件技术和网络技术的进展为数据库提供了越来越好的运行环境。

1. 网络应用环境的演变

网络应用环境的演变是全方位、多线索的。沿着计算机体系结构发展的过程是:集中式(Centralized)主机-终端体系结构→分散式(Decentralized)计算机体系结构(分布计算机网络)→客户机/服务器体系结构;沿着数据通信或常规网发展的过程是:局域网(LAN)→城域网(MAN)→广域网(WAN);沿着计算模式/方法发展的过程是:集中式计算环境→分布式(Distributed)计算环境→客户机/服务器(C/S)计算模式→Internet/Intranet 的网络计算环境;沿着数据库技术与网络技术结合发展的过程是:初步的松散结合(集中式)→高级的紧密结合(分布式)。这就是说,数据库系统从最初的集中式发展为分

布式,从主机-终端体系结构发展到客户机/服务器系统结构。今天,数据库系统不仅可以通过局域网、广域网,而且可以通过 Internet 相互连接,使数据资源的共享范围从一个部门、一个行业扩大到全国乃至全世界的各行各业。

2. 数据处理模式层次结构的演变

数据库系统应用环境是服从于数据处理模式的。从数据处理模式的层次结构上看,经历着单层结构、两层结构和多层结构的发展。早先的观念是由应用程序控制关系型数据库。这种数据处理的模式可称为单层结构(1-Tier)。由于这种结构的数据库程序占用计算机资源较多,在 80 年代中期,数据库应用开始转向客户机/服务器结构,即所谓的两层结构(2-Tier)。这种结构在近十年内不但得到了广泛的应用,而且相当成功。然而,两层客户机/服务器结构在构架上的缺陷却逐渐暴露出来,其中最明显的问题是应用程序在伸缩性和维护性方面的弱点。比如,一个跨国企业如何把数据库管理系统及其应用程序分散到十分缓慢的网络上,如何控制数据的统一性和完整性;因为一旦应用程序有任何改动,维护人员就必须修改每一个客户端上的应用。新一代数据库管理系统在传统的客户机/服务器结构中增加了应用程序服务器,这就是所谓的多层结构(n-Tier 或 Multi-Tier)。应用程序服务器负责统一界面、制定业务规则和数据处理逻辑的规定等。多层应用服务技术允许分割应用程序,在本地计算机上无须安装一整套数据库工具就可以在另一台机器上存取数据;同时允许对业务规则和进程进行集中管理,并在整个网络上分发和实现进程负载的动态调节。

总之,无论是从哪种观点发生的演变,总的趋势始终是集中式向分布式的转变。

1.2.1 集中式主机-终端体系结构

1. 集中式网络(Centralized Network)结构

集中式网络结构是以一个中央控制器或计算机为中央节点,连接并控制网络上其他节点的数据处理设备,向它们提供计算能力和其他服务,以便直接进行数据交换。通常,中央节点的计算机称为主机(Host),在网络中向终端用户提供诸如计算(运行用户程序)、数据库服务以及执行网络控制功能。其他节点的数据处理设备称为终端(Terminal),作为主机的人机接口设备,负责数据的输入输出。这是早期的面向终端的计算机网络,是 70 年代典型的应用模式。与其说它是计算机网,不如说是终端网更确切。图 1.1 是集中式网络的计算环境示意图。

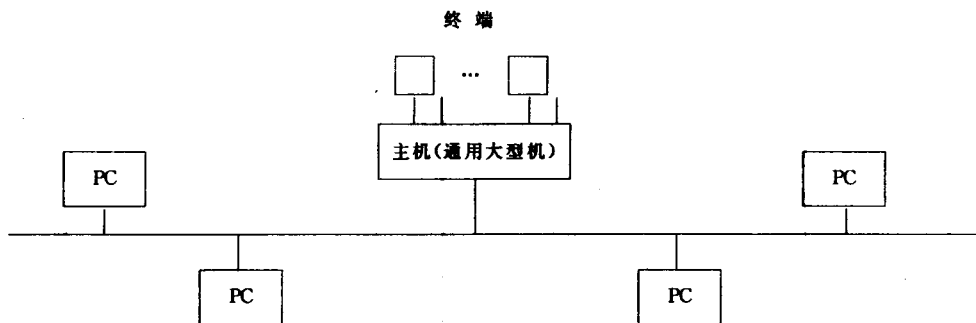


图 1.1 集中式网络的计算环境

由于当时的组网技术缺乏系统性和长远性,导致众多的网络结构和通信协议互不兼容的混乱局面,使终端网的继续扩大和发展面临困境。直到 1974 年 IBM 发表了面向集中式的终端网——“系统网络体系结构”(SNA),终端网的发展进入体系结构化的高级阶段。

终端网分为两类:为特定用户开展专门服务的专用网和为任意用户普遍利用的分时服务网。例如,美国通用电器公司(GEC)经营的商业数据处理网曾经是世界上最大的分时网。它在世界的几十个大城市都有本地数据链路,并通过卫星和洋底电缆连接到一些国家的首都和大城市。该网为树形多级结构,设有两套线路交换器以提高工作的可靠性和灵活性。它除了提供远程批处理服务外,还提供集中的数据库服务。例如,它允许用户访问货币兑换数据库,可提供 46 种货币每日的开盘与收盘兑换率。

2. 分时系统

分时服务网的基础是分时系统(Time Sharing System)。分时系统是以实现人-机交互作用为主要目标,通过终端设备实现人-机联系的一种方式。这种方式既能充分利用处理器,又能方便用户使用。分时系统有下列几个特性。

- ①同时性,不同的人能在相同时间内各自使用同一台计算机。
- ②独立性,不同作业在计算机内可以彼此独立、互不干扰地进行工作。
- ③即时性,计算机能在较短时间内响应每个终端用户的请求。
- ④交互性,用户可以较方便地与正在机内运行的作业进行会话,及时检验和修改机内的程序。

⑤空间无限性,分时用户不仅可以在机房或机房附近使用计算机,还可以通过数据通信网在远离机房的地方用终端处理作业。

将各分时系统中的计算机用通信网络相连,就构成一个具有分时功能的计算机网络。这种系统的终端用户可以共享更多的资源,特别是网络中丰富的软件资源。随着计算机网络的发展,单独的分时系统发展成为一种简单的计算机网络,即面向终端的计算机网络。

3. 集中式数据库系统的联网

集中式网络中由主机控制的、向各终端用户提供的数据库系统即是集中式数据库系统。把原有的多个集中式数据库系统用网络连接起来属于初步的松散结合,通常称为数据库联网。这些集中式数据库系统是相互独立的。例如,图 1.2 中有三个集中式数据库管理系统 DBMS1、DBMS2 和 DBMS3 用网络相连。不同数据库系统上的用户可以通过网络远程登录到另一台机器的数据库系统上,存取该机上的数据库资源。每个数据库系统中的数据库由各自的数据库管理系统集中管理,在 DBMS 这一层次上是相互独立、没有任何联系的。因此,这种联网仅仅是数据库技术与网络技术的简单结合。网上用户虽然可以存取异地机上的数据库,但不能执行全局(分布)应用;也就是说,当需要对网络上两个以上数据库系统中的数据进行综合处理时,必须从一台机器退出再登录到另一台机器上。典型的例子是银行转账,它包括两个子事务,需要同时更新两个场地上的数据库。

总之,主机-终端体系结构的计算模式是以大型机为中心的集中式计算。这在计算机的发展历史中是第一个重大的计算阶段。这种计算模式的最大的特点就是集中:一个很大的中央处理机外接成百上千个哑终端,一切处理均依赖主机,集中的数据、集中的应用软件、集中的系统管理和网络管理。这种模式对最终用户而言很不理想,然而这种模式所需的维护管理成本很低。

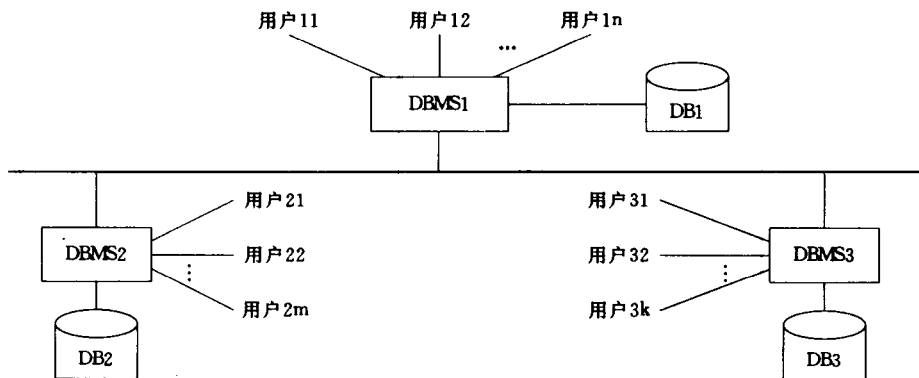


图 1.2 多个集中式数据库联网示意图

1.2.2 分布式系统

分散化与分布处理是信息社会的自然格局。按性质,大多数数据处理都具有分散性。由于常规的集中式数据处理系统具有“一点牵动全局”的特性,有可能会致大的经济损失。常规网虽然可以实现资源分享,并提供通信能力,但不能满足用户要求系统提供的高度的可用性。可用性是指:系统中若有一个甚至几个计算机或通路发生故障时,其余部分仍可重构系统;新的系统可以工作,甚至可以继续其失效设备的部分或全部工作。如果给每个部门或单位都提供一台小计算机,局部的任务局部处理,并由最熟悉业务的人控制,且对结果全权负责,将会是十分有益的。于是人们开始在局域网的基础上研究独立的计算机系统之间的互联,使得在任何一对系统之间都可以实现交互式的资源(软件、数据和硬件)共享,而不依赖于某一主机。这就是分散(Decentralization)的概念:将计算(或控制)功能分布在网络上的多个节点上的一种分布式计算机体系结构——分散式网络(Decentralized Network),可理解为系统硬件设备的物理分布——物理上松散结合的多机系统,是实现分布的第一个阶段,主要解决系统的可用性问题。

在解决了系统可用性的同时,又提出多机之间的合作性问题,即由若干个互联的计算机共同分担完成系统指定的某个计算任务。这样的计算机系统即分布式系统(Distributed System)。它实现了系统中的软件在逻辑上的分布,使系统不仅是一个物理上的松散结合,而且是一个逻辑上的紧密结合。美国卡内基-梅隆大学和俄亥俄州立大学的环形结构多机系统均为早期分布式系统的代表。80年代个人计算机的兴起使我们进入了第二个重大的计算阶段,由集中式计算模型转向分布式计算模型。

1. 分布式系统的特征

一个分布式系统就是若干计算机的集合。这些计算机不分享存储器或 CPU 的一个时钟,每个计算机均有它自己的局部存储器和外部设备。它们既可独立工作(自治性),也可协同工作(合作性),彼此间的通信是经由通信链路的消息交换完成的。多机(并行)合作和可用性是分布式系统的两大本质特征。

分布式系统的特征可概括为以下几点。

- ①有任意数目的系统和用户进程。
- ②结构是模块化的,处理部件的数目是可变的。

- ③通过分享通信结构(存储器除外)的信息交换完成通信。
- ④通过系统范围内的控制,实现动态进程间的合作和运行时间管理。
- ⑤进程间的信息传输总有延迟,且延迟时间的长短是可变的。

2. 分布式计算环境

(1) 分布式计算环境的概念

分布式计算机网络(Decentralized Computer Network)是针对集中式而言的,故也称分散式网络,强调的是系统的可用性;分布式计算环境(Distributed Computing Environment,简称 DCE)是一种计算模式,强调的是系统的合作性。

分布式计算环境与网络是相互关联但并不相同的两个概念。企业、学校、研究单位都建有自己的局域网,并大多数与 Internet 相连,构成全球范围的广域网,但这并不代表它们已经拥有了分布式计算环境。为了实现企事业单位内部几个企业之间的合作,分布式计算环境是非常必要的。分布式计算环境即以计算机网络作为一个整体,支持两台以上的计算机通过网络通信能够协同工作的环境。这个环境的计算机可以是个人电脑、小型机、工作站,甚至巨型机。这些计算机可以分布在一座办公大楼里,也可以分布在全球的任何一个角落。它们通过分布式计算环境共享资源,协同工作。从用户角度看,分布式计算环境实际上就是一种能提供透明网络服务的网络环境。网络是分布式计算的基础,但有了网络并不一定就实现了分布计算。

(2) 分布式计算环境的组成部分

- ①DCE 线程。提供在单个进程内多线程的创建、管理和同步的手段。
- ②DCE 远程过程调用(RPC)。包括一个开发工具和一个运行时间服务。开发工具提供用于开发分布应用程序的语言(Interface Definition Language,简称 IDL)及编译工具。
- ③DCE 目录服务。维护关于各种分布资源(如用户、机器、基于 DCE RPC 的分布应用程序等)的信息,包括资源的名字和位置。
- ④DCE 分布时间服务。提供分布式计算环境中计算机的同步时间。
- ⑤DCE 安全服务。提供分布式计算环境下安全通信及资源访问控制。

除以上五个核心服务外,DCE 还提供一个附加服务,即 DCE 分布文件系统 DFS。它实际上是构造在 DCE 上的一个基础分布应用。DFS 使用户访问或共享位于网络上任一文件服务器上的文件时,不需知道文件的物理地址。DFS 包括缓存管理器、文件分布器、令牌管理器、DCE 本地文件系统(LFS)、复制服务器等。

(3) 分布式计算环境的优点

- ①提供了支持分布应用程序的服务与工具。利用这些服务工具开发的分布应用程序能够在不同计算机、不同操作系统的网络上运行。
- ②各项 DCE 核心服务的高效集成。其中包括分布应用程序开发手段、不同系统上运行进程间的通信手段和解决时间同步问题的分布时间服务等专门服务。
- ③提供了异构环境下互操作和移植手段。DCE 能架构在不同操作系统和硬件平台上。DCE 环境下,一台机器上的进程可以和另一台不同操作系统的机器上的另一进程进行通信,因而网络覆盖面很广。
- ④提供了与其他外部世界接轨的手段。通过 ISO 标准 X.500 及 Internet 流行标准 DNS,DCE 可以与非 DCE 环境很好地交流。