

HOPE

适用于大、中、小和微型计算机

关系数据库系统的深入分析和比较

孔祥清 胡 杉 译
王 珙 谢 坤

北京希望电脑公司



适用于大、中、小、微型计算机

关系数据库系统的深入分析和比较

孔祥清 胡 杉 编译
王 珙 谢 冲

北 京 希 望 电 脑 公 司

一九九一年六月

前 言

从1970年开始,关系数据库模型对于数据库研究者、设计者和最终用户变得越来越普遍了,这主要是因为关系概念具有简单和功能强的特点。与层次和网络模型不同,关系模型为数据库管理提供了理论基础,这就推动了数据库研究群体的出现。关系数据库模型研究的主要成果就是开发了高级数据操纵语言及为使概念数据库设计过程自动化提供了坚实的基础。

经过十年的研究之后,第一代关系数据库管理系统(RDBMSs)出现在市场上。虽然从可用性和最终用户的生产率看,这些系统具有明显的优点,但它们仅限于决策支持应用,并且由于性能较差而使应用受到限制。自从第一代RDBMSs出现之后,关系数据技术在可用性及性能方面都得到了很大的改进。今天大多数RDBMSs都支持一个第四代编程语言工具的完整集合以提高最终用户的生产率。有些RDBMSs为在线处理应用提供了高性能。进而,不断增加的RDBMSs为分布式数据库管理提供了广泛的支持。功能强的关系模型促进了第四代工具的开发和分布式数据库的管理能力。

本书考察了现行的能最好实现关系数据库概念的关系数据库管理系统。介绍和分析了主要为大中型和小型机设计的十种RDBMS,四种面向微机应用的RDBMS,以及主要的关系数据库机器。分析准则包括RDBMS的功能、能力和工具选择。比较结果概括在表中。本书所举的数据库管理系统虽然不全面,但都是最好产品的样板。这里提供的信息都是根据每个系统最近的版本而编写的。

本书是面向广大读者的,包括计算机科学方面的大学生(包括本科生和研究生)、计算机科学家以及打算更多了解RDBMS的用户。为了理解本书,需要具备关系数据库管理方面的基础知识。同时,也要求读者比较熟悉现代操作系统,本书也可用来为教师对传统数据库技术的知识进行更新。本书力求使读者认识到在集中式数据库管理中关系模型具有实际优越性。本书还比较和评述了主要的关系数据库产品,因此,它能帮助读者选择最合适的RDBMS。比较准则也可应用到本书未提到的关系系统中去,以了解和分析它们的能力。我们希望读者对本书所考察的所有系统感兴趣,并认为这些系统是很有意义的。

目 录

第一章 引言

- 1.1 数据库系统的进展..... (1)
- 1.2 什么是关系数据库管理系统..... (1)
- 1.3 关于关系模型的误解..... (2)
- 1.4 组织..... (3)
- 1.5 定义..... (3)

第二章 十种RDBMS介绍

- 2.1 引言..... (4)
- 2.2 RDBMS的描述..... (5)
- 2.3 样板数据库..... (14)
- 2.4 十个关系系统综述..... (16)
- 2.5 比较表..... (18)
- 2.6 结论..... (29)

第三章 系统R, SQL/DS和DB2

- 3.1 引言..... (31)
- 3.2 系统R..... (32)
- 3.3 用例子进行查询 (QBE) (45)
- 3.4 SQL/DS 产品..... (48)
- 3.5 DB2产品..... (49)
- 3.6 R*..... (50)
- 3.7 结论..... (52)

第四章 INGRES

- 4.1 引言..... (55)
- 4.2 INGRES原型..... (56)
- 4.3 分布式INGRES..... (73)
- 4.4 INGRES产品..... (74)
- 4.5 INGRES/STAR..... (78)
- 4.6 结论..... (79)

第五章 SABRINA

- 5.1 引言..... (82)
- 5.2 用户界面..... (84)
- 5.3 语义数据控制..... (92)
- 5.4 存贮模型和查询处理..... (96)
- 5.5 事务管理..... (101)

5.6	系统的体系结构.....	(106)
5.7	SABRINA 产品.....	(108)
5.8	结论.....	(109)
第六章 SUPRA		
6.1	引言.....	(112)
6.2	数据定义.....	(113)
6.3	数据操纵.....	(120)
6.4	关系数据库的管理.....	(125)
6.5	SUPRA结构	(129)
6.6	结论.....	(129)
第七章 六种其它的关系系统		
7.1	引言.....	(131)
7.2	DATACOM/DB.....	(131)
7.3	FOCUS	(135)
7.4	NOMAD.....	(139)
7.5	ORACLE.....	(143)
7.6	SYBASE	(146)
7.7	UNIFY.....	(150)
7.8	结论.....	(153)
第八章 用于微机的RDBMS		
8.1	引言.....	(153)
8.2	dBASE III PLUS.....	(154)
8.3	KNOWLEDGEMAN/2.....	(161)
8.4	R: BASE 5000.....	(168)
8.5	EXCEL	(170)
8.6	结论.....	(173)
第九章 数据库机		
9.1	引言.....	(176)
9.2	数据库机的体系结构.....	(178)
9.3	国际计算机有限公司的CAFS—ISP	(180)
9.4	Teradata出产的DBC/1012	(182)
9.5	Copernique生产的DORSAL 32.....	(185)
9.6	Britton Lee 的 IDM 500	(189)
9.7	数据库机的研究.....	(193)
9.8	结论.....	(203)
附录 系统设计者.....		
		(208)

第一章 引言

1.1 数据库系统的进展

自从60年代初期出现第一代数据库管理系统 (DBMS) 以来, 它们在质量 (功能、性能、易用性等方面) 和数量 (不同产品的数目) 方面都取得了较大的进展。质量进展由两个互补的趋势所驱动: 数据库理论和技术方面的显著进步和数据库用户不断提出的复杂要求。DBMS的数量进展主要是因为数据库应用数量和种类增加以及计算资源的不断分散。

一个DBMS主要由其支持的数据模型所刻画。第一代以层次或网络模型为基础的DBMS现在仍然被大量应用。它们可以被视为文件系统的扩展, 在文件系统中文件间的连接通过指针完成。这些系统的数据操纵语言是导航式的, 即: 编程员必须在层次或网络中通过导航来确定存取数据的路径。

在80年代初期, 第一个基于关系模型的系统出现在市场上, 与前期的产品相比, 它具有一些明显的优点。今天, 大量的关系产品可用于大、中型计算机、小型机、微机及专用机 (数据库机) 上, 其市场正在迅速扩展。关系模型在研究者, 设计者和用户中的成功主要是由于其概念简单且功能强。

1.2 什么是关系数据库管理系统

由E.F.Codd [Codd 70] 发明的关系模型的优点已由许多数据库研究者作了全面的介绍。它的一个主要优点是提供了逻辑数据描述 (以概念的形式) 和物理数据描述 (以文件的形式) 之间的完全独立性。作为物理独立性的结果, 它可以支持高级数据操纵语言。某些语言可将编程员从物理细节中解放出来。因此, 允许由系统而不是用户进行查询优化。数据库语言标准化也促进了关系模式的发展, 标准化产生了结构式查询语言 (SQL) [ANSI86]。SQL为所有类型的用户在数据定义、控制和操纵方面提供了一个唯一的接口。

关系模型可由三个特点来刻画 [Codd 82]:

1. 数据结构简单。它们是称为关系 (或表) 的两维表, 其元素是数据项。一个关系可以视为一个文件, 一个关系的一行, 称为元组 (或行), 可以视为一个记录。一个关系的一列, 称为属性 (或列), 可以视为一个数据项。连接两个关系的联系由两个关系中共有的属性确定。例如: EMPLOYEE关系和DEPARTMENT关系之间的联系可由存贮在两个关系中的dept_name属性确定。

2. 一个称为关系代数的八运算符的集合 (并、交、差、笛卡尔积、选择、投影和联接等) 使得数据定义, 数据查询和数据更新很容易完成。每个关系运算符以一个或两个关系作为输入并产生一个关系。

3. 一个完整性限制的集合定义数据库的一致状态。

一个关系DBMS (RDBMS) 是一个支持关系模型的软件程序。关系数据库系统的定义由关系任务组 (Relational Task Group) 提出, 其结果得到E.F.Codd的认可,

发表在 [Schmidt 83] 上。某些定义对于表征那些称为关系的系统是有用的。如果一个系统满足下面三个条件，它就被称为最小关系系统：

(1) 数据库的所有信息被表示成表中的值。

(2) 表间的指针对用户是不可见的。

(3) 系统必须至少支持下面几种关系代数运算：选择、投影和自然连接，这些运算符必须不受内部约束所限制。在某些系统中存在内部约束的例子是在完成连接的连接属性上存在一个索引。这些约束限制了数据库语言的功能。

如果一个系统除了满足上面三个条件外还满足下面两个条件，则称为是完全关系的：

(4) 它支持所有关系代数运算而不会因为执行程序而受任何限制。

(5) 必须支持两个基本的完整性限制。第一个是每个关系有一个或多个构成唯一键的属性。例如：社会安全号是 PERSON 关系的一个唯一键。第二个限制，常常被称为引用 (referential) 完整性限制，确定一个关系中的一个属性值已经在另一个关系中存在。例如：关系 EMPLOYEE 中的 dept-name 必须在关系 DEPARTMENT 中存在。

Codd [Codd 85] 已经将这些条件包括在一个更大的具有十二个准则的集合中以表征一个完全的关系 DBMS。然而，由于我们的目的不是决定一个 DBMS 是否为完全关系的，我们在分析 RDBMS 中将不用他的准则集合。本书是为了分析和比较存在许多关系特点的 DBMS。本书讨论的所有系统居于最小和完全关系之间。

大多数 RDBMSs 支持标准的 SQL，此外，它们一般提供一个第四代工具的完整集合，例如：数据字典、屏幕布置、应用发生器和模式辅助设计，这个集合改进了应用开发和提高了数据库程序员的生产率。某些 RDBMS 也提供强有力的分布式数据库管理能力。这些能力由于关系模型的物理数据独立性及其提供的高级功能而变得很容易实现。

1.3 关于关系模型的误解

象任何其它新观点一样，也存在关于关系模型和在市场上引进 RDBMS 的误解。一般，这些误解有如下五种：

1. 为了理解关系模型必须有一个理论背景。的确，关系模型的最初表示是理论性的 (采用集合论)，且是面向特定的对象的。然而，由于关系模型概念简单，没有任何理论背景也能理解关系模型。某些关系系统的售主甚至已经禁止使用与关系模型相关的最初词汇，并重新使用传统术语。例如：文件和表。

2. 标准的概念模式设计是一个很强的约束。在关系框架下建立的标准化理论不简单，但在数据库设计过程中产生许多优点。然而，标准化对设计一个概念模式不是一个约束。多数 RDBMS 可以处理非标准关系。此外，在关系框架下，有大量的工具可用来帮助数据库设计过程。

3. 关系模型太差。这句话意味着模型的数据结构太简单以致不能精确地建立真实世界的模型。我们已经看到具有丰富语义的模型的概念 (例如实体/关系模型) 能容易地映射到关系概念。关系模型的简单性使得这个转换变得容易。进而，关系模型借助于完整性限制而更丰富，这个完整性限制使得应用语义能被抓住。

4. RDBMS 的数据类型有限。大多数RDBMS 提供很少的数据类型,例如:整数、浮点数和字符串。主要原因是它们被设计用于传统的数据处理(商业)应用,而这些应用不需要丰富的数据类型。然而,关系模型领域里的基本概念不限于特殊的类型。正如在[Gardarin 88]中所示,关系系统可以被扩展成具有丰富的数据类型以支持新的数据库应用,例如办公室自动化和知识库系统。

5. RDBMS的性能差。这是反对与层次和网状模型相对立的关系模型的主要论点之一。的确,第一代RDBMS性能不好,然而,由于关系模型在概念级上是非常有用的,一个好的关系系统必须完成有效的查询优化技术和对数据的快速存取。有些现行的RDBMS使用由层次或网状系统所用过的功能强的工具技术。正如我们将在本书将看到的那样,现在的关系系统,特别是关系数据库机器具有很高的性能。

总之,随着关系模型及其相应技术的成熟,第一代关系系统的缺点正在消失,技术的成熟正通过关系产品的不断成功而显示出来。许多研究者为了支持非传统数据库的应用正在提出向关系模型扩展的建议[Gardarin 84]。最显著的扩展包括在[Gardarin 88]中所介绍的推理能力的支持和面向目标处理的能力。

1.4 组织

本书全面地介绍了[Gardarin 88]中提出的概念的应用。第二至第七章介绍主要设计用于大中型机和小型机的RDBMS。第二章介绍了概况及十种RDBMSs的比较,在后面几章对每种RDBMS作了详细介绍。第三章到第七章每章分别介绍一个RDBMS,第三章和第四章考察了两个先行的关系数据库样机,现在它们已成为成功的产品。第三章介绍由IBM开发的SYSTEM R(系统R)样机,以及在系统R基础上研制的IBM SQL/DS和DB2关系产品。第四章介绍INGRES系统,其样机是由California 大学Berkeley分校研制的,产品由RTI销售。第五章介绍了在法国INRIA设计的SABRINA系统,其产品由三个法国公司联合销售。第六章介绍了SUPRA,它是开发用于工业环境的关系数据库系统。SUPRA是由通用数据库系统TOTAL的开发者Cincom 设计的。第七章对其它六种感兴趣的成功的系统作了简要介绍。这些系统是:DATACOM/DB, FOCUS, NOMAD, ORACLE, SYBASE和UNIFY。第八章介绍了具有关系特点的仅能用于微机上的数据库系统。根据相同的计划,对四个通用的DBMSs—DBASE III Plus, KNOWLEDGE—MAN/2, R. BASE 5000和EXCEL作了介绍。第九章介绍关系数据库机器。对四种商品—ICL 的 CAFS/ISP, Teradata 的 DBC/1012, Copernique的DORSAL32和Britton-Lee的IDM 5000作了介绍。同时对研究者提出的大量数据库机器作了介绍和分类。

1.5 定义

我们在介绍DBMS语言的句法中将经常用到下面的记号:

- {a} 表示a是可选择的;
- {b} ... 表示元素b可重复n次,这里 $n \geq 0$;
- {a | b} 表示选择a或者b。
- <p> 表示p是一个必须限定到一特定值的参数。

参考文献

- [ANSI 86] American National Standard for Information Systems,
Database Language SQL, ANSI X3.135-1986, October 1986
- [Codd 70] E.F.Codd, "A Relational Model for Large Shared DataBanks,"
Comm of ACM, Vol.13, No. 6, June 1970.
- [Codd 82] E.F.Codd, "Relational Databases: A Practical Foundation for
Productivity, " Comm.of ACM, Vol.25, No.2, February 1982.
- [Codd 85] E.F.Codd, "An Evaluation Scheme for Database Management
Systems That are Claimed to be Relational, " Int.Conf.on Data
Engineering, Los Angeles, February 1985.
- [Date 83] C.J.Date, An Introduction to Database Systems, Vol. 2,
Addison-Wesley Publishing Company, 1983.
- [Date 86] C.J.Date, An Introduction to Database Systems, Vol.2, 4th
ed, Addison-Wesley Publishing Company, 1986.
- [Delobel 85] C.Delobel, M.Adiba, Relational Database Systems, North
Holland, 1985.
- [Gardarin 84] G.Gardarin, E.Gelenbe, New Applications of Databases,
Academic Press, 1984.
- [Gardarin 88] G.Gardarin, P.Valduriez, Relational Databases and
Knowledge Bases, Addison-Wesley Publishing Company,
1988.
- [Schmidt 83] J.W.Schmidt, M.L.Brodie, Relational Database Systems
Analysis and Comparison, Springer-Verlag, 1983.
- [Ullman 82] J.D.Ullman, Principles of Database Systems, 2d ed.
Computer Science Press, 1982.

第二章 十种RDBMS介绍

2.1 引言

目前,市场上有大量关系数据库系统(RDBMS)。这些系统一般都支持几种关系概念,但由于其应用目的及工具选择不同而在几个重要方面有差异。

本章介绍用于大型机和小型机的十种RDBMS,这十种系统是:DB2, INGRES, SABRINA, SUPRA, DATACOM/DB, FOCUS, NOMAD, ORACLE, SYBASE 和 UNIFY。当然,市场不仅仅有这些系统,选择它们的目的是因为它们有大量的关系特点,都具有文本,从市场的观点看它们都很成功。因此,本书所介绍的这些系统在RDBMS范围内具有一定的代表性。在后面的几章中对每一种系统都作了详细的介绍。三种系统(DB2, INGRES 和 SABRINA)是主要研究项目的成果,且具有很好的文

本, 因此, 每一种系统都用一章(第3至5章)来介绍。SUPRA是工业系统的一种典型例子, 也用一章(第6章)来介绍, 其它六种以稍微粗略的形式在第七章介绍。

本章中系统之间所做出的比较是基于它们的语言、它们的功能及它们的工具方面。用于这些比较而不可缺少的相关准则在2.2节中定义。为了说明这些系统的语言, 在2.3节中描述了一个样板数据库和一组查询, 通过这个数据库及查询可以说明系统的大量功能。在第2.4节中大致介绍了十种RDBMS, 这些系统的特点概括在第2.5节的比较表中。

2.2 RDBMS的描述

RDBMS的最显著的特点也许它们是面向应用的, 这就决定了大多数设计和工具选择。大多数RDBMS被设计用于传统的数据处理(商业)应用, 这可以大致分为两类: 第一类由决策支持应用组成, 比如项目计划, 它们是面向复杂的查询的。这种数据库是典型的小型或中型库。有少量的(也许十个)并行用户存取, 这些库的基本应用要求是易用、响应快。第二类由在线处理应用组成, 例如飞机订票系统, 它们是面向事务的、且修改频繁。这种数据库是典型的大型数据库, 由大量的并行用户存取。这些应用需要高的用户吞吐量及广泛的数据控制和可用性。虽然大多数RDBMS可用于这两种系统, 但它们中的一些更适合用于在线处理。在介绍每个RDBMS时, 我们将指出其对在线处理应用的特别支持。

RDBMS的描述由下面几部分组成: 数据定义, 数据操纵, 语义数据控制, 事务管理、算法、结构、附加工具和分布式数据库管理。本书将介绍每一部分并提供将用于本书的基本术语。对于要更详细了解有关情况的读者, 我们建议读者去查阅有关关系数据库系统原理和方法的书籍(比如: Gardarin 88, Date 86, Ullman 82)。

2.2.1 数据定义

本小节介绍数据定义语言(DDL)及系统管理数据定义的方式。我们仅考虑概念和物理(也称内部)模式级上的数据处理软件。外模式级(视图)定义包括在关于语义数据控制一节中。

概念和物理模式的定义语言可以不同, 也可以用同一种语言。概念数据定义有DBMS支持的数据模型来指导, 虽然本章分析的大多数RDBMS支持一个纯关系模型, 但也有支持混合模型的, 在混合模型中关系可能被组织成层次的。然而, 一个RDBMS的混合模型可以转化为一个纯关系模型。因此, 概念数据可由下面的实体定义, 这些实体是: 数据库, 关系层次, 关系, 元组, 域和属性。元组定义一般由关系定义包含了, 关系定义要求一系列构成元组的属性。概念DDL的主要能力是数据库, 关系层次(当可以应用时), 关系和属性的增加和删除。关系中属性的增加和删除是复杂的, 因为它要求重新组织数据库。

物理数据定义规定了一种方式, 用这种方式, 概念目标应能在磁盘上有效地完成。一般说来, 这包括根据存储结构(如杂凑或B树)指定关系(或关系层次)的聚集, 辅助索引及有时对频繁进行连接的关系进行连接。必须定义关系聚集以使选择/投影操作的性能最好。辅助索引借助于提供快速的间接存取关系而优化选择操作。与CODASYL系统中的使用类似, 一个联接(Link)对连接(Join)操作提供了有效的支持, 物理DDL的重要能力是关系聚集的定义和重定义以及索引和联接的增加和删除。关系聚集

的重定义是复杂的，因为它要求数据库的重新组织。

DDL 语句的编译产生概念和物理数据描述，这些描述一般存贮在一个称为数据字典的中心数据结构里。数据字典是一个重要的工具，它包括关于数据、应用和系统的所有种类的信息。这样的信息的例子是语义数据控制信息、被编译的查询以及性能信息。当DDL 语句产生自动数据库重新组织时，数据字典是动态的。数据字典的变化对于并发用户是立即可见的。数据字典可作为一个称为元库的关系数据库来管理。在这种情况下，数据字典可用相同的数据操纵语言授权的用户查询。由于数据字典是为编译用户查询而被系统地存取的，所以对它必须进行有效地管理。数据字典可以通过允许对各种设计表示的存贮和管理来提供对数据库设计的直接支持。在这种情况下，数据字典一般与数据库辅助设计工具接口，这个工具可能通过一个图形接口与设计者相连接，最终生成概念和物理模式。由某些RDBMSs 提供的辅助设计能力在使DBA (Database Administrator) 任务自动化时具有重要的意义。

2.2.2 数据操纵

通常称为查询语言的数据操纵语言使得一个用户或编程员能够从数据库查询数据和修改数据库。我们将查询语言 and 用户接口区分开来。查询语言是单个用户进入数据库系统的入口点，在查询语言的顶部可提供各种用户接口，每个用户接口可完成一类应用，由最好的RDBMSs提供的用户接口是高度可见的，并将表格、菜单、图象和窗口的应用结合起来。本章也包括对它们的介绍。

关系模型的价值是给高级查询语言一个牢固的基础。第一代 DBMSs (例如层次和 CODASYL 系统) 的查询语言基本上是过程的。用这种的语言，在层次或网状中，用户 (实际上是编程员) 通过一个“一次一条记录”接口导航，并确定数据应该被存取的方式 (过程)。以集合论为基础，关系模型允许非过程的面向集合查询语言的开发，用这种语言，用户确定需要的结果而不必描述得到它的方式。

关系查询语言有一个牢固的理论基础：关系代数或关系演算。以集合论为基础的关系代数 [Codd 72] 提供集合运算 (笛卡尔积，并，差，选择，投影，连接及其它运算) 以处理关系。复杂的查询可表示成关系代数运算的序列。进而，对完成每一个运算存在有效的算法。以一阶逻辑为基础，基于关系演算的语言 [Codd 72] 使得查询表达成一个逻辑公式。因此，它们与其说是以关系代数为基础的语言，不如说是说明型语言。说明型语言的最大优点是减少了确定一个查询所需的信息的数量。这就减少了用户出错的危险，提高了用户的生产率，提供了更多的系统优化机会。关系演算分成元组关系演算和域关系演算，元组关系演算将元组与一个逻辑公式中的每个变量关联起来。SQL (见第三章) 是基于元组关系演算的，域关系演算将域与一个逻辑公式中的每个变量关联起来。关系演算的一个重要特点是它容易被映射到关系代数。因此，大多数支持关系演算为基础的查询语言的RDBMS都能有效地完成关系代数运算。

查询语言是特地为要求强烈的数据操纵设计的。此外，它们为结果表示提供了基本支持，例如存贮和分组；然而，其数据类型支持和计算能力一般都相当有限。由现行RDBMSs 支持的域包括一些基本的数据类型，例如：整数，实数，字符串，日期和时间。计算能力限于算术和聚集功能。进而，复杂应用要求具有嵌套几个查询语言命令和控制这个嵌套的几种形式的能力。

解决上述问题的一般方法是将查询语言嵌入通用编程语言。编程语言是第三代语言,例如C, COBOL或Pascal(第一代,第二代语言分别为机器语言和汇编语言)。嵌入查询语言,将数据库的查询操纵能力与编程语言的一般编程能力结合起来。用这种方法的一个明显问题是应用程序员必须处理两种语言,一种是过程的,一种是非过程的,它们不能很好地匹配。

每种主语言都要求一种不同的嵌入查询语言的预处理器。

另一个解决办法是提供将查询语言能力与基本的编程语言指令集成起来的一种语言。产生的语言叫第四代语言(4GL),与第三代语言(例如COBOL)相比,4GL是非过程的。第四代语言无精确定义。然而,一个典型的4GL包括查询语言命令,控制能力(IF, GOTO等),迭代(WHILE DO, FOR),变量处理和错误处理。查询语言命令可以以关系代数或关系演算为基础。4GL通常集成一个统一的功能强的工具,例如报表生成器或应用生成器,这些都提高了程序员的生产率。虽然比起嵌入查询语言方法来通用性差一些,但是4GLs是迈向数据库管理语言的一步。在数据库管理语言中,数据库操纵和编程能力都是被完全集成的。

2.2.3 语义数据控制

本节介绍控制数据操纵可能采用的方式,通过这种控制,使得被授权的用户完成正确的对数据库的操作。语义数据控制一般包括视图管理,安全性控制和语义完整性控制。这些功能对数据库的完整性具有贡献。在关系框架中,语义数据控制可以以一个统一的方式取得。视图、安全性限制和语义完整性限制可以被定义为系统自动强制的规则。一个用户事务或一组数据库操作(见2.2.4节)违反某些规则通常意味着对该事务有效性的抑制。

控制数据操纵的规则的定义是数据库管理的一部分,而数据库管理一般是由数据库管理员(DBA)完成的。DBA还负责施加组织政策。语义数据控制一般指定使用DDL。与数据定义类似,语义数据控制的规则在数据字典中仍然有效。

视图使得用户组对数据库有它们特别的理解。在一个关系系统中,一个视图是一个虚拟的关系,它被定义为基于库关系查询的结果,但不象库关系那样存贮在数据库中。一个视图在某种意义上是一个动态窗口,它反映对数据库的所有修改。一个外模式可以定义成一组视图和(或)库关系。除用于外模式外,视图对于用一种简单方式来保证数据库安全性也是非常有用的。借助于选择一个数据库的子集,视图隐藏了一些数据。如果用户仅能通过视图存取数据,那么他们不能看到和操纵被隐含的数据,这些数据是保密的。

数据安全性—数据库系统的一个重要功能保证了数据不被非法存取。数据安全性拥有数据保护和权力控制。为了防止数据的物理内容被非法用户了解,就要求数据库保护。这个功能一般由文件系统提供。保护磁盘上数据的最终办法是数据加密。被加密(编码)的数据仅可由知道该码的被授权的用户解密(或译码)。授权控制必须保证被授权的用户仅仅能完成允许他们对数据库的操作。许多不同的用户可能在单个系统的控制下存取大量的数据。因此,DBMS必须能够将对数据库子集的存取局限于对用户子集的存取。授权控制长期被操作系统提供作为文件系统的服务、在这个范围内,提供了一个集中控制。一个实体(object)的建造是集中控制器,该控制器允许特殊的用户对这个实体完成特殊的操作(读、写、执行)。实体是由它们外部名所辨识的。数据库系统中的

授权控制在几个方面与传统的文件系统的授权控制不同。授权可被加工以使不同用户对相同数据库实体有不同的权力。这个要求意味着比名字更精确地确定实体子集的能力及能对一组用户进行区分。此外,授权的分散控制在分布情况下特别重要。在关系系统中,权力可由DBA用高级指令统一控制。例如,被控实体可以与查询条件相同的方式由谓词确定。

语义数据控制对数据库的一致性的维护具有贡献。如果一个数据库满足一组称为语义完整性限制的条件,则该数据库的状态称为一致的。维护一个完整数据库要求各种进程,例如,并发控制,可靠性,保护和语义完整性控制。语义完整性控制保证数据库的一致性,这是由控制那些导致数据库的不一致的更新事务或对数据库状态的特定活动进行激励来完成的,这里的数据库状态对更新事务的结果进行补偿。被更新的数据库必须满足一组完整性限制。由于得不到DBMS的支持,语义完整性控制的传统解决方法是将控制嵌入到应用程序中,说明方法已与关系模型溶和在一起以缓和程序/数据依赖,编码冗余和传统方法性能差的问题。一种想法是用谓词演算的断言来表达完整性限制。因此,一组语义完整性断言就定义了数据库的一致性。这个方法允许人们容易地说明和修改复杂的完整性断言。然而,支持自动的语义完整性控制需要有效的算法,因为检查断言的费用是很高的。因此,虽然一些最近推出的RDBMSs有很好的语义完整性控制能力,但是,大多数RDBMSs在对语义完整性控制的支持方面是有限的。

2.2.4 事务管理

本节介绍当系统与介质失败时,在多用户(并发)环境下事务管理的方法。数据库的一致性,可由更新操作改变,为了保证数据库的一致性,更新操作一般分成由数据库系统控制的称为事务的一致性单元。一个事务是一个执行单元,对于一个一致性数据库,这个单元产生一个一致的但可能不同的数据库。一个数据库是原子的,不论它被执行而完成(提交)或根本不执行它(放弃)。并发控制和恢复是事务管理的两个方面,RDBMSs中的并发控制和恢复与非关系系统是一样的。可以在[Bernstein 87]中找到事务管理的综合处理。

并发控制保证多事务的分解执行产生相同的结果,就象它们被顺序地执行一样。因此,并发控制使得多用户数据共享对每个用户是透明的。并发控制的一个重要目标是取得高度的并发以使得信息的吞吐量最大化。以锁为基础,对并发控制的最通用的解决办法是从操作系统用于为任务分配资源的方法中引申出来。数据的各个部分可以视为资源,这些资源可以分配给(锁存)或重新分配(取消锁存)给事务。一个影响性能的重要参数是数据区组,这个区组可以被锁存或释放。并发控制粒度是逻辑(事务级)和物理(系统级)的。一个逻辑区组可以是数据库、关系、域或者元组。一个物理区组可以是段、文件或页。一个逻辑区组与一个或多个物理区组对应。一个小的区组明显地使并发度最大变化了,但是与大的区组相比,它将招致更高的控制开销(特别是锁存表更大)。一个小的区组(例如元组)对于与几个元组打交道的短的事务是最好的,一个大的区组对于长的事务(例如批事物)是最好的。锁存算法最大的问题是可能产生死锁。以检测死锁为基础的主要方法可能异常地结束事务。

事务的可靠性保证事务以可靠的方式执行。可靠性是通过容错技术取得的,容错技术从许多种失败中恢复。事务可靠性所涉及到的最重要类型的失败是事务、系统和介质的失败,可靠性的目标是在最小化用户交互作用期间产生失败之后恢复一致性的数据库

状态。理想的情况是：恢复应该是对用户透明的。总之，可靠性是一个复杂的功能，该功能与其它重要功能（如并发控制，缓冲存贮管理和文件管理）不是独立的。所有可靠性算法所基于的一个原理就是数据的复制：不可靠的或丢失的数据应该总能从冗余的拷贝中恢复。管理这个复制是费用较高的，因为这要增加用于处理更新的磁盘存取数。用于复制的最通用的数据结构是运行记录和备份拷贝。一个运行记录记载与被更新事务的状态和被修改的页相关的信息。记录在运行记录上的主要信息类型是前映象文件和后映象文件。运行记录对于启动或重新启动系统是有用的。在正常结束或失败后启动系统所执行的进程称为恢复进程。恢复进程是以运行记录和磁带档案为基础的。恢复进程的类型是：在正常结束之后恢复，在事务失败（异常结束）后恢复，在系统失败后恢复和在介质失败后恢复。在介质失败恢复是最复杂的进程，因为它要求联合使用运行记录和备份拷贝。

有两种基本技术用于完成与一种可靠性算法相结合的磁盘页更新：定位更新（update-in-place）和阴影法。定位更新试图将被更新的目标放在它们原来的位置。阴影法决不被更新的目标放在它们原来的位置。事务的更新不是在原来的地方进行的，而是在一个新页进行的。其旧内容称为阴影页。每种技术对于不同的条件有用。

2.2.5 查询处理和存贮模型

查询处理策略和数据存贮模型对于数据库查询的有效处理是关键的因素。大多数RDBMSs提供非过程的高级语言（例如SQL）和使用随时设置的运算符。当查询一个关系数据库时，用户指定希望的结果而不必提供存取数据的路径。因此，当系统应用一个查询处理算法来负责确定如何存取数据，这个算法对一个给定的查询产生一个存取规则。查询处理是一个困难而又重要的任务，在过去十年中得到了相当的重视（Jarke 84, Kim 85）。其原因是查询处理的效率决定了RDBMS在完成高级查询的性能，因此也就决定了它的可用性。性能问题对于关系数据库是很关键的，这是因为查询被用于完成各种任务，例如：检索和更新查询，模式管理和语义数据控制。第一代RDBMSs由于其性能差而受到强烈的批评。从那以后，所设计的查询处理方法无论在样机还是在后来的商业产品中都证明是有效的。

查询处理的目的是以使费用最小化的方式来执行查询的。最好存取规则的选择，称为查询优化，对于一般查询在计算上是很难处理的。因此，在优化费用函数时需要使用试探法。这个费用函数一般参考机器资源，例如磁盘存取时间和CPU时间。查询优化的最常用方法是全部解空间的穷举查询，在这个空间中，对每一个可能存取规划的费用进行估计。这样的优化是以统计为基础的，这个统计考虑数据库（关系基数，属性选择等等），可以存取数据的路径（例如聚集和二次索引）和完成关系代数运算的算法。

物理数据库设计是与查询优化有关的重要问题。事实上，非关系数据库系统，如IMS和SYSTEM 2000，为组织物理数据库结构提供了非常复杂的方式。物理数据库设计允许人们使得所用的数据结构适合于给定的数据存取模型，这个模型是从数据库应用的知识中推导出来的。因此，存贮模型的能力和物理数据库设计的准确性对于查询处理的效率是至关重要的。存贮模型规定了存贮结构和由DBMS提供的算法。

2.2.6 附加的工具

通常完全与RDBMS集成在一起的附加工具是设计用来改进数据表示，应用开发和

与其它系统接口的。它们增加了数据库的可用性和应用开发的生产率。

查询语言（例如SQL）的有限的数据表示能力可由报表生成器和图形接口增强。报表生成器允许用户很快地用字符（表头、页头和页脚注，带有题头和总计的组和子组）和从数据库选择来的数据定义复杂的报表。报表定义存贮在数据字典中。报表可以通过给一个报表名和给一个选择希望数据的准则来实现。一个图形接口允许用户以图形的形式（如直方图、图表或者曲线，这些图形可能是彩色的）显示搜索查询的结果。与报表相同，图形被定义并被存贮在数据字典中。

复杂的数据处理应用的开发可以通过屏幕管理器和应用生成器这样的工具来显著地加快。一个屏幕管理器允许为数据编辑进行定制屏幕的定义，也可能允许将报表生成器或图形接口用于数据表示。屏幕定义存贮在数据字典中。应用生成器完成更复杂的任务，它将前面的工具结合和同步起来以更快地定义复杂的应用。RDBMSs一般提供与其它系统的接口以使数据库中的非关系数据集集成在一起。一般，数据库与文本编辑，电子表格程序，邮件系统或文件系统接口。

2.2.7 体系结构

本节介绍RDBMS的功能和进程结构。功能结构是根据软件模块和模块之间的接口给出的，这里每个模块完成一个定义好的功能。RDBMSs的功能结构大致可以分为两种类型的结构，如图2.1所示：

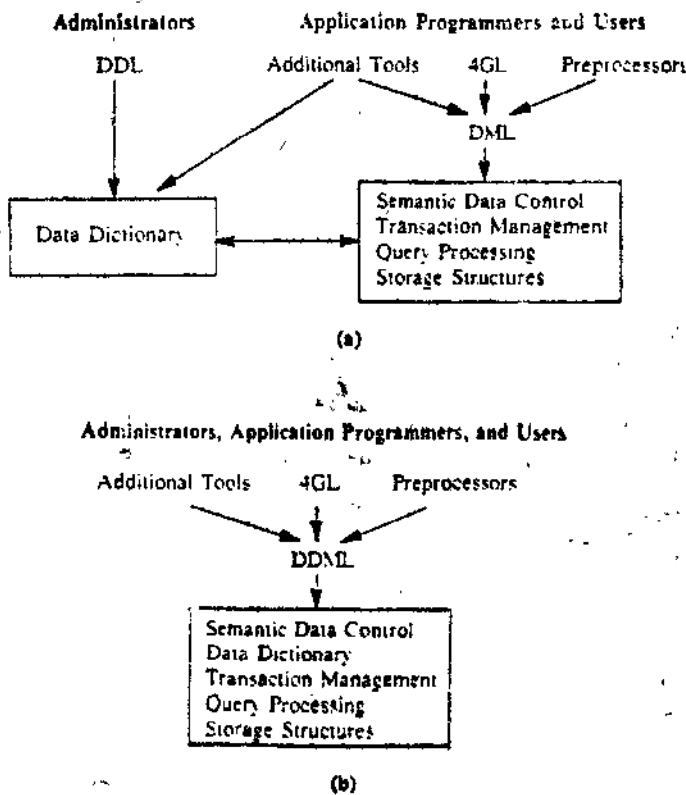


FIGURE 2.1 Functional RDBMS architectures. (a) Separate interface model and (b) uniform interface model.

第一种类型的结构是以分开的接口模型为基础的，这种模型有两种分开的RDBMS语言：DDL和数据操纵语言（DML）。不同的用户接口和附加的工具建立在DML之上。对于这种结构，数据字典一般不作为一个元组库管理，而是以要求特殊操纵能力的内部形式管理。因此，可能需要功能（或程序码）的一些副本来处理数据字典信息和关系信息。这种类型的结构一般起源于由底向上的设计方法，在这种方法中，数据字典和关系特性被加到一个已经存在的文件系统或数据库系统结构中。

第二种类型的结构是以统一接口模型为基础的，这种模型有一个唯一的数据定义和操纵语言（DDML）。对于这种类型的结构，数据字典是作为一种关系数据库来管理的。因此，数据字典管理可以完全用关系管理能力来完成。这种类型的结构起源于自顶向下的设计方法。

进程结构介绍在操作系统下多用户RDBMS的实现。有三种基本方法将操作系统进程分配给RDBMS的事务 [Stonebraker 81]，如图2.2所示。在所有方法中，每个注册用户有一个进程，称为用户进程，它用来管理用户交互活动（例如终端监视器）。

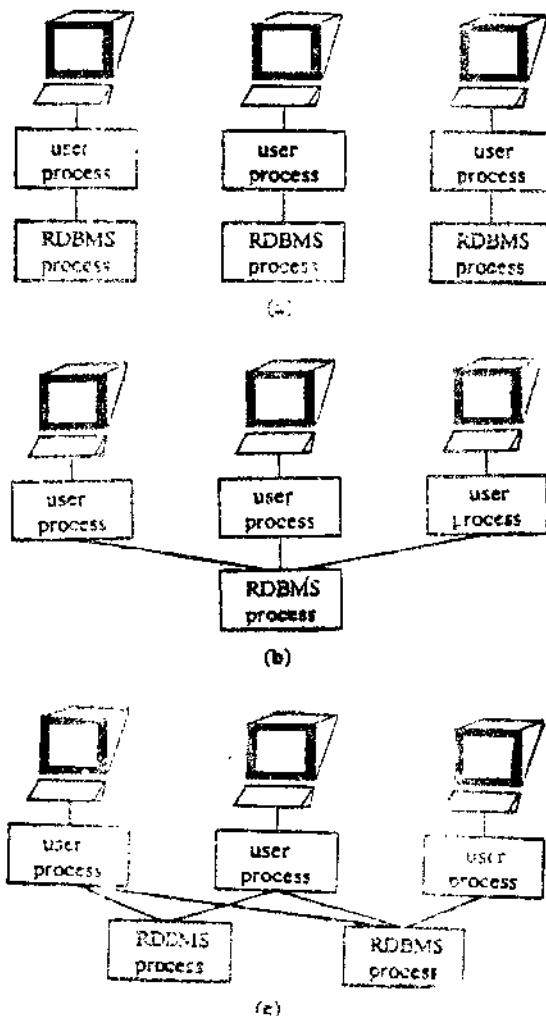


图2.2 不同RDBMS的进程结构
a. 进程模型 b. 单服务模型 c. 多服务模型

最简单的方法是进程模型方法,在这种方法中,每一个用户恰好有一个RDBMS进程。RDBMS进程共享相同的可重入RDBMS程序和一些公用的数据结构,例如缓冲区表。由于每个事务由一个进程完成,操作系统的多任务和调度程度可能被利用。这个方法的主要缺点是竞争的用户数受最大进程数的限制。请记住一个进程是一个有价值的资源。另一个问题是进程开关的开销。无论何时一个进程被中断,例如,由于磁盘存取,就会出现一个进程开关,而另一个进程被激活。在大多数操作系统中,进程开关是费CPU时间的。大多数RDBMSs使用进程模型方法。

另一个方法是单服务器模型,在这个模型中,用户进程可将请求讯号送到多RDBMS进程。这个方法解决了大多数进程模型的问题。特别地,由于有了单个RDBMS进程,进程开关的开销减小了。然而,这个方法是最复杂的,因为单个RDBMS进程必须支持几种形式的事务的多任务和调度以提供内部并行性。

第三种是多服务器模型,在这个模型中,用户进程可将请求讯号送到多个RDBMS进程。这种方法避免了单服务器模型的主要问题(操作系统程序的复制)。同时,它还提供了并行处理事务的机会,这是由于几个RDBMS进程完成相同进程的不同工作来实现的。

2.2.8 分布式数据库管理

对远程或分布式数据库的支持是DBMS的一个重要能力。自从60年代初期以来,将有组织的数据集中在一个大的和费用高的计算机上一直是数据处理的单一方法。最近在计算机网络、小型机和微型机方面的发展已经采用了分散的方法,这种方法是使数据分布在几个机器上,这是一个效率高的解决方法。分散方法有三个优点:

第一,具有一个计算机的每个组(或个人)对自己的数据直接控制,这使得数据集程度增加和数据处理更有效。

第二,与数据必须由每个组向中心计算机传送数据的集中方法相比较,通讯的费用降低。

第三,分散方法从几何分布方面看是对数据处理的自然解决。

对于放置和管理在不同计算机上的数据进行集成和共享的要求通常是非常强的。这样的数据共享要求这种计算机通过一个局部或通用网络来互相连接,并要求特殊的软件支持来管理和处理分布数据。大多数RDBMSs提供分布数据共享的解决方法。在增加功能度下提出的这些方法是远程数据库,同构分布式数据库和异构分布式数据库。

一个远程数据库是位于这样一台计算机上,在该机上,用户(或应用)不执行程序(见图2.3)。通常,用户知道远程数据库的位置,这个位置必须指定以存取数据。必须要有通讯元件以存取远程数据库。一个局部数据库可能驻留在用户使用的计算机上。用户可将远程数据库的数据卸到局部数据库上。最近在微机技术方面的进展有利于“工作站/服务器”组织,在这个组织中,远程数据库由一个RDBMS管理在一个主机服务器上,而私人数据库由相同的RDBMS的微机版本管理在工作站上。相互连接的服务器/工作站一般由一个局部网处理。

一个分布式数据库(Ceri 84, Ozsu 89)是一组驻留在不同机器上(称为位置 Sites)相互合作的数据库,这些机器是由一个计算机网络连接起来的。分布式数据库与远程数据库的主要差别在于用户不了解数据存放的位置,而把分布式数据库视为一个