

逻辑数据库 设计 方法

DATA BASE
MANAGEMENT

董永建 编译

天津市
计算机应用技术研究

验所

32

6

逻辑数据库设计方法

编 译 董永健

校 对 穆德华

温燕飞

封面设计 白风仪

天津市计算机应用技术研究所出版

编 译 前 言

在现代信息社会的电子计算机数据管理方面，特别是在庞大数据的维护和检索方面，数据库技术的重要地位是众所周知的。无论是用于数据处理的大中型计算机，还是日渐普及的微型机，如果没有数据库管理系统配置，其软件系统都将被认为是不完备的。

当前，随着计算机数据库技术的逐步推广使用，数据库的设计方法已成为一个重要课题。

我们知道，数据库的物理设计（Physical Data Base Design）主要依赖于应用环境和各自配置使用的数据库管理系统（DBMS），其物理实施由各自系统的数据库描述语言（DDL）提供。但是，数据库管理系统本身並不提供逻辑数据库设计（Logical Data Base Design）功能。对于逻辑数据库设计（或称数据库的逻辑设计），一般主要靠数据库设计人员的经验和水准。然而，在数据管理领域，目前已发展出若干逻辑数据库设计的辅助方法提供使用。这些方法在一定程度上规范了数据库的逻辑设计，使得数据的组织过程趋于简单明了。实践表明，使用这些方法，能使数据库设计更为完善，从而最终获得性能较好的数据库。

本书收选编译了两种逻辑数据库设计方法。一种是所谓E—R方法（即实体—关系方法）。该方法通过引入企业视图概念来构造概念模式，进而转换为各自系统数据库的逻辑数据结构。它简明易懂，普遍适用于各种结构（网状，层次，关系）数据库的逻辑设计。另一种方法是IBM公司提供的，我们称之为“局部视图综合法”。该方法逻辑清晰，步骤简捷。它首先用三个步骤建立各个局部视图，然后将局部视图综合为整体的系统视图。这一方法虽然是面向层次数据库设计的，但也具有相当普遍性，可供变通使用。

在开始讲述每个方法之前，对于数据库的一些有关基本概念都有

交待。特别地，对数据库理论的三模式结构，在E—R方法中有专节介绍。

本书主要选译自《The Entity—Relationship Approach To Logical Data Base Design》(By Professor Peter Chen)和《DL/1 Dos/vs Application And Data Base Design》(IBM)两书。其它参考书目在该两书中均有引介，恕未一一列举。

编译者希望，对于从事数据库应用的同志，包括从事日益普及的微机数据库应用的同志，本书将有一定参考价值。书中错谬之处，敬请识者教正。

全书由天津计委统计局电子计算站穆德华、温燕飞同志最后校订，谨致谢忱！

天津计委统计局电子计算站

董永健

目 录

第 一 部 分

逻辑数据库设计的E—R方法	1
一、介绍	1
1. 几个有关基本概念	1
2. 逻辑数据库设计与物理数据库设计	4
3. 数据库系统和数据模型	5
4. 逻辑数据库设计所牵涉的问题	6
5. 数据库设计的一个新方法:	
实体—关系方法(E—R方法)	9
6. 实体—关系方法的优点	9
二、E—R方法与ANSI/X3/SPARC建议	10
1. ANSI/X3/SPARC建议	10
2. 概念模式和企业模式	12
3. 三类数据库管理员	15
4. 小结	15
三、实体—关系图(E—R图)	16
1. 实体和关系	16
2. 实体和关系性质的描述	21
3. 特殊实体型和关系型	24
四、E—R图到数据结构图的转换	28
1. 数据结构图	28
2. E—R图到数据结构图的转换规则	35
五、逻辑数据库设计步骤	40
1. 逻辑数据库设计的主要步骤	40
2. 例一, 某装配工程公司的数据库设计	40

3. 例二, 订货帐目数据库	54
4. 例三, 图书管理数据库	60
六、数据库逻辑设计的其它考虑	67
1. 从E—R图到数据结构图的其它转换规则	67
2. 修改数据结构图以提高数据库性能和合理利用存储	69
七、层次结构数据库的设计	72
1. 转换规则	72
2. 一个将E—R图转换为层次逻辑数据结构的例子	73

第 二 部 分

逻辑数据库设计的局部视图综合法	77
------------------------	----

一、引言	77
二、局部视图的建立	77
1. 建立局部视图的三个步骤	78
2. 建立局部视图	81
三、将局部视图综合为系统视图	99
1. 数据元素之间的关系	100
2. 消除冗余关系	101
3. 标识控制键	101
4. 数据元素之间多对多关系的处理	102
5. 交互数据及其它	103
6. 综合局部视图的步骤	104
7. 将四个局部视图综合为系统视图的例子	105
8. 物理实施前的一些考虑	106

第一部分 逻辑数据库设计的E-R方法

一、介绍

1. 几个有关基本概念

本节首先概略介绍数据管理中的几个有关概念，作为以后内容讨论的引导。

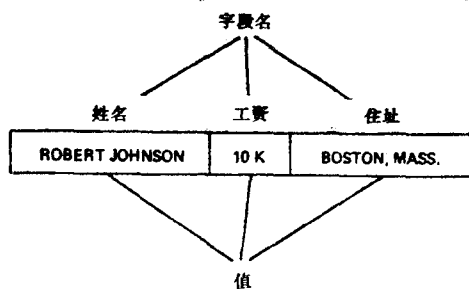
记录 (Record)：我们将一个记录定义为若干数据项的集合。例如，一个职工 (EMPLOYEE) 记录包括与某具体职工相关的姓名，工资和住址等等 (图一)。组成记录的数据项称为字段或场。这样，上述姓名、工资和住址就是职工记录的字段名或场名。对于不同职工记录，这些字段有不同的字段值。

文件 (File)：一个文件是相同类型记录的集合。例如，所有职工记录的集合组成某企业的职工文件 (图二)。

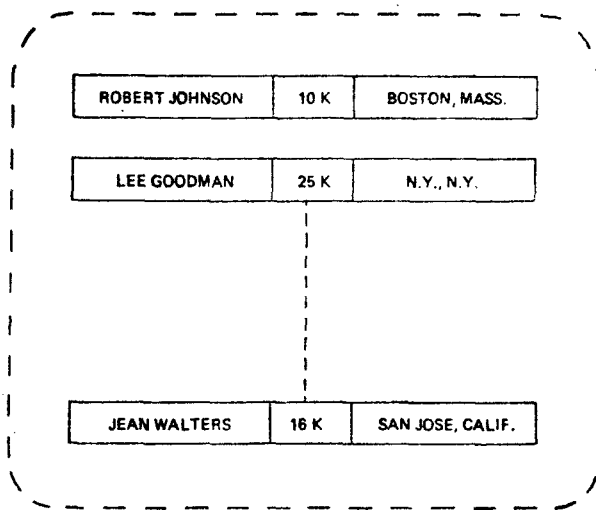
数据库 (Data Base)：数据库是不同类型记录的集合 (图三)。在数据库中，不同类型的记录彼此联结，从而其中的相关数据项可供检索。例如，一个部门 (DEPARTMENT) 的职工数据库是将在该部门工作的所有职工记录联结在一起组成的 (图四)。

图四所描述的是这一数据库的物理数据结构 (Physical Data Structure)，其中，记录间的联系通过指针链实现。如图所示，部门记录有一个指向记录链中第一个职工记录的指针。链中的每一个职工记录也都有一个指向同一链中的下一个职工记录的指针；最后一个职工记录的指针指回到部门记录。图内描述的是数据库中各记录值 (Occurrence) 之间的关系。但是，对我们来说，这种描述过于详尽，以致难以突出该数据库中对我们最为重要的关系。因此我们采用另一种较简单的方法 (图五) 来代替图四的记录组织方法。如图五所示，我们用小矩形表示一个记录型 (Record Type)，用箭头表示职工记录与其所在部门的部门记录之间的联系。此外，图五和图四还有另一点区别，由

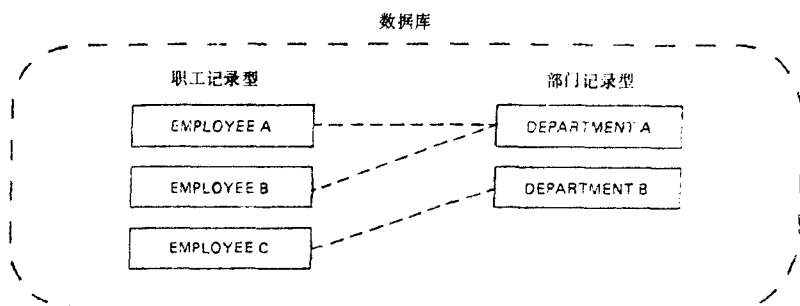
于图五仅仅表示部门记录型与职工记录型之间的联系，所以它实际上表示的是该数据库的逻辑数据结构（Logical Data Structure）。这种逻辑数据结构的表示方法不涉及记录间的联系究竟怎样去实现。



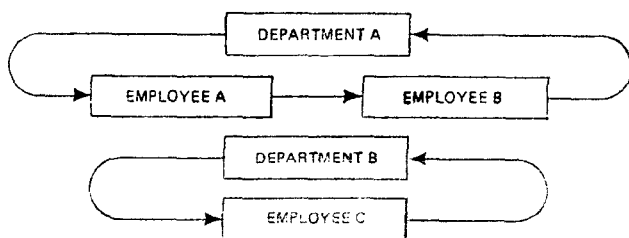
图一、一个职工记录



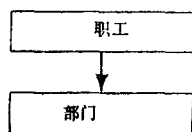
图二、一个职工文件



图三、一个包含两个记录型的数据库



图四、数据库中相关的数据记录连结在一起（数据库的物理数据结构）



图五、数据库的逻辑数据结构

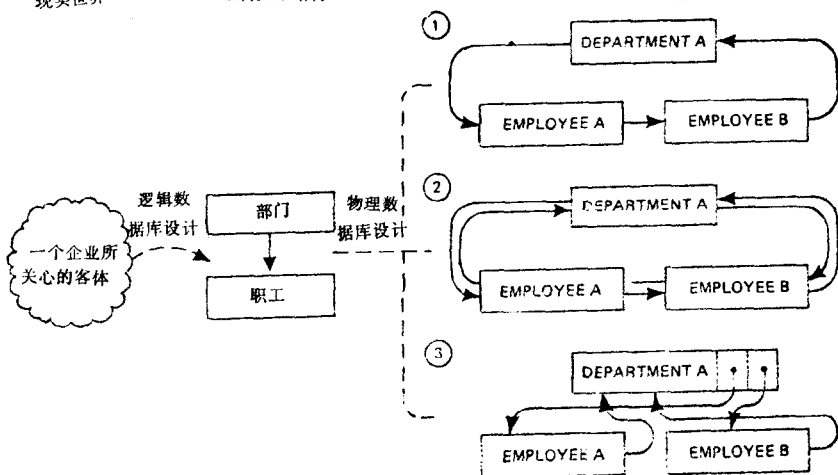
2. 逻辑数据库设计与物理数据库设计

一般来说，数据库设计分为逻辑设计和物理设计两个步骤（图六）。

物理数据库设计涉及选取一种物理数据结构来实现某一给定的逻辑数据结构。例如，对于图六所示逻辑数据结构，在 CODASYL 数据库系统中，至少有三种可供选择的物理数据结构。第一种使用向前指针连接同一部门中的所有职工记录；第二种加入了向后指针；第三种则是使用指针组，即在部门记录中含有指向所有该部门职工记录的指针。这三种物理数据结构都有各自的优点和缺点。第一种易于实施并且很适于职工记录的顺序检索处理；第二种结构对于向后检索链中的记录相对地容易，也使删除处理效率较高，当然，为了放置向后指针，这种结构增加了存储开销；对于第三种物理数据结构，其主要优点在于可以快速直接检索属于同一部门的所有职工记录。有一点十分重要，就是我们应当注意到，没有一种物理数据结构是普遍适用的最优选择。物理数据库设计的目的在于，对某一给定的应用环境，选择相对来说最适当的物理数据结构。尽管物理数据库设计也是一个重要的课题，但本书不再进一步讨论它。

逻辑数据库设计是为数据库设计逻辑数据结构的过程（图六）。这一课题既涉及对应用环境的分析，也涉及对某一给定（或选定）的数据库系统的逻辑数据结构的分析。当前在数据库技术中，已经发展出一些帮助进行逻辑数据库设计的工具。然而，数据库的设计通常更依赖于设计者的学识和经验。由于这种主观依赖，使得当今许多使用中的数据库远非完善的设计。

本书将集中讨论逻辑数据库的设计过程，并介绍给数据库设计人员二种实用而有效的逻辑数据库设计工具。



图六.

数据库的逻辑设计和物理设计示意图

3. 数据库系统和数据模型

当前世界上有许多使用中的数据库系统,主要分层次,网状,关系三类。它们之间的主要区别在于所支持的逻辑数据结构类型不同。层次数据库系统的例子有IBM的信息管理系统IMS,它要求数据记录型按层次结构组织(图七)。对于某些数据库,使用层次数据结构十分适宜,但是,当记录型之间不存在某种自然层级关系时,使用层次数据库系统来设计数据库将十分困难。网状数据库系统,(诸如CODASYL)较之层次数据库系统,它支持较复杂的逻辑数据结构,例如,网状数据库系统允许一个记录型拥有多个“双亲记录型”(图八)。关系数据库系统(如IBM的SYSTEM/R)则是使用二维表作为逻辑数据结构(图九)。

总之,逻辑数据库设计所关心的是将数据组织成一定的数据库系统所支持的逻辑数据结构形式(图十)。

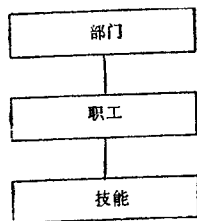
4. 逻辑数据库设计所牵涉的问题

数据库设计是一个十分复杂的过程。这是因为设计人员不仅必须考虑现实世界中的数据模型，还不得不考虑数据库系统的各种限定条件以及数据检索更新的效率等等：

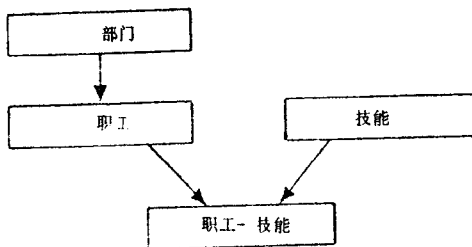
- (1) 数据库设计人员受数据库系统所支持的数据结构类型的限制。
例如，像工程和雇员之间的这种多对多关系，在某些数据库系统中就不能直接表示出来。
- (2) 数据库设计人员必须考虑记录的存取路径（即如何存取某一具体的记录型）例如，图三的数据库结构就隐含着对于职工记录的存取，必须通过相应的部门记录。
- (3) 数据库设计者还要考虑如何较快地进行检索和更新。这样，现实世界中关于一个实体的数据可能要被分为一个以上的记录，以便提高存取效率。例如，可以将一个职工的数据项组成两个记录：一个职工主记录，一个职工明细记录，

在传统的数据库设计方法中存在两个问题：

- (1) 数据库设计人员必须同时考虑许多方面许多结果，这使得设计任务非常困难。
- (2) 逻辑数据库设计过程的最后结果是用户模式（即数据库用户视图的描述），而用户模式往往将设计者引向复杂的方面，它通常难以一下搞懂，也难以改变。



图七. 层次数据结构



图八、网状数据结构

部门表

D#	BUDGET
1	10 M
5	5 M
8	20 M

职工表

NAME	SALARY	ADDRESS
JOHNSON	10K	BOSNON
GOODMAN	15K	NYC
WALTERS	16K	SANJOSE
⋮	⋮	⋮

技能表

S#	S-NAME
5	FORTRAN
2	COBOL
1	PL/I

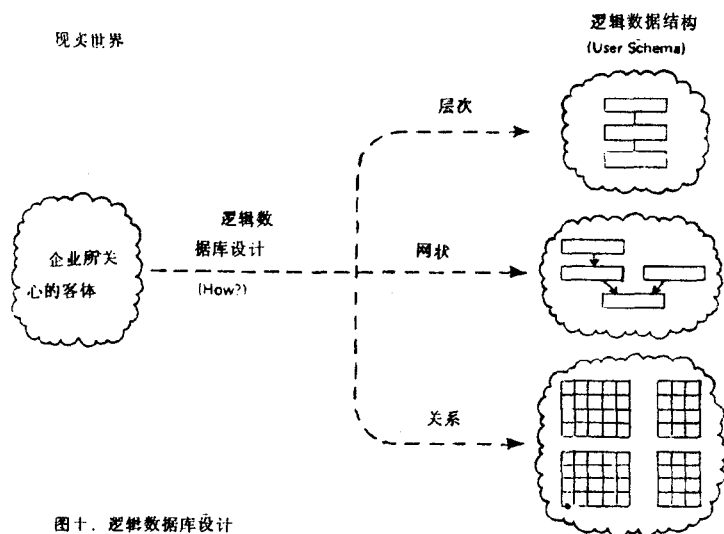
职工-技能表

NAME	S#
JOHNSON	1
JOHNSON	2
GOODMAN	1
GOODMAN	5
⋮	⋮

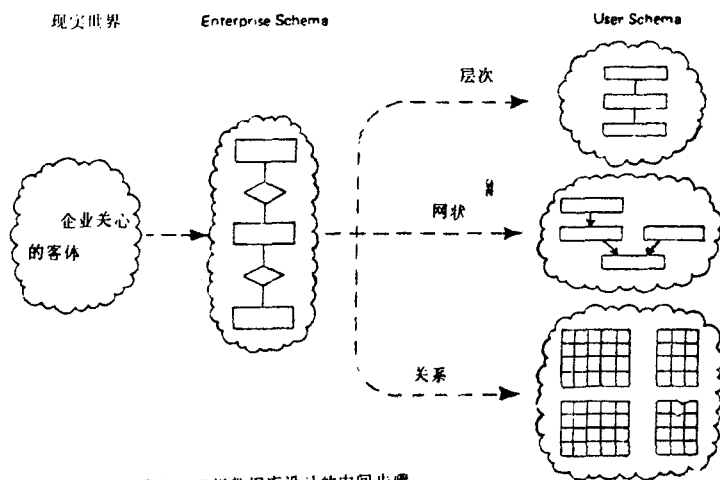
部门-职工表

D#	NAME
1	JOHNSON
1	GOODMAN
5	WALTERS
⋮	⋮

图九、关系数据结构



图十. 逻辑数据库设计



图十一. 企业模式—逻辑数据库设计的中间步骤

5. 数据库设计的一个新方法：实体—关系方法 (Entity—Relationship Approach)

我们来介绍一个逻辑数据库设计的新方法，它被称为实体—关系方法 (E—R方法)。E—R方法的主要思想是在逻辑数据库设计过程中加进一个中间阶段 (图十一)。数据库设计者首先使用实体—关系图 (E—R图) 标识出从企业角度所关心的所有实体 (Entity) 和实体间的关系 (Relationship)。在这一阶段，数据库设计人员要从整个企业或整个部门的现实去考察看待数据，而不是从具体的应用程序员的观点来看。因而，我们将企业视图的数据描述叫做“企业模式” (Enterprise Schema)。企业模式应当是对纯现实世界的表述，并且不涉及对存储空间和效率的种种考虑。我们首先设计出企业模式，随后再将其转换成一定数据库系统的逻辑数据结构 (图十一)。

6. 实体—关系方法的优点

传统的逻辑数据库设计方法通常只有一个阶段，即将现实世界中客体的信息直接映射为用户模式。E—R方法却不同，它由两个主要阶段组成：

- (1) 用实体—关系图定义企业模式。
- (2) 将企业模式转换为逻辑数据结构。

使用E—R方法进行逻辑数据库设计的优点是：

- (1) 按功能划分为两个阶段使得数据库设计过程清晰简洁，也使得设计过程便于组织。
- (2) 由于企业模式的设计不受具体数据库系统能力的限制，而且也不须考虑存储和效率等因素。所以它的设计比用户模式来得容易。
- (3) 与用户模式相比，企业模式较为稳定，由于企业模式独立于所使用的数据库系统，所以当需要从一个数据库系统改为使用另一个数据库系统时，可能要改变用户模式，而企业模式却无须改变。这时要做的只是重新将该企业模式映射到适于新数据库系统的用户模式。

- (4) 由E—R图表示的企业模式较容易为非专门电子计算机数据处理人员所理解。

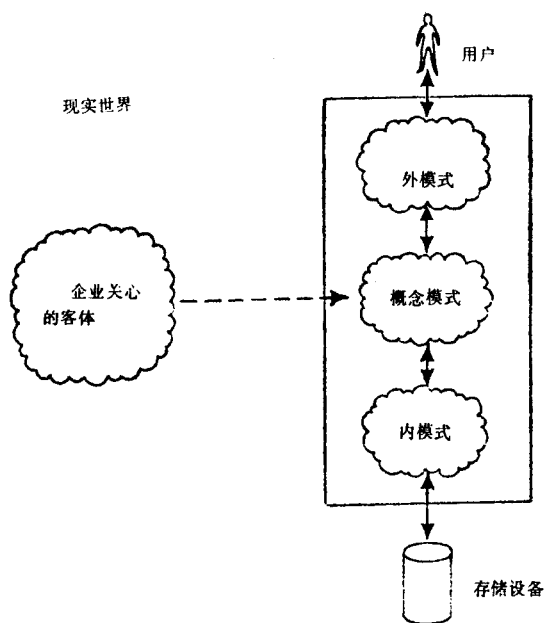
二、E—R方法与ANSI/X3/SPARC建议

1. ANSI/X3/SPARC建议

E—R方法中，企业模式的概念与ANSI/X3/SPARC提出的概念模式非常相似。下面我们讨论一下二者的联系与特点。

1971年秋，美国国家标准学会的电子计算机信息处理委员(ANSI/X3)会成立一个特别研究小组，任务是决定数据库管理系统的哪些方面适于进行标准化发展。该小组由一些用户社团，硬件厂家和大学的代表组成，名为标准规划及须求委员会(SPARC)。

这个小组花费了相当长时间，作了很大努力，仔细考虑了数据库理论的各个方面，他们的中期报告引起了数据库工作者的广泛注意。该小组意识到，对于数据库管理系统的各个成份，与其发展一套标准来规范，倒不如集中于规范这些成份相互间的配合接口。根据这一认识，他们的中期报告提出了数据库管理系统的三模式结构(图十二)。这三级结构是：外模式(External Schema)，概念模式(Conceptual Schema)和内模式(Internal Schema)。而当前的一些数据库管理系统通常只有两级结构，即逻辑结构(程序员所看到的数据结构)和物理结构(电子计算机所“看”到的数据结构)。三级结构中，外模式即用户模式，它表示用户(即程序员)的数据视图。换句话说，外模式是一种数据描述，这些数据按其数据名和性质对于应用程序来说是可见的。内模式表示在存储设备中的物理数据组织。内模式同时包括了数据完整性，恢复以及检索更新效率等细节。最后，概念模式表示整个企业看待数据的视图，它是对企业所关心的实体，属性及其关系的模型的描述。概念模式同时包括了对数据允许进行的操作，语义统一性和保密性等的要求条件。概念模式意在为数据提供一个稳定的视图。



图十二、ANSI/X3/SPARC结构