Foxbase+ 2.10原理与在财经管理中的应用

姜灵敏　黄大明　江　文
邓正清　陈业菊　编著

湖南出版社

前　言

电子计算机的高效率、高速度等优良特性所展示的魅力已为越来越多的人所赏识，因而它在我国得到了迅速的推广、发展和应用。尤其在各行各业的事务管理工作中扮演着十分重要的角色，创造了巨大的社会效益和经济效益，使它更受人们的青睐。

在电子计算机的推广应用过程中，以严密的数学理论——集合论为基础、由 Ashton-Tate 公司开发的关系型数据库管理系统 dBASE 可算是百花丛中的一朵“阆苑奇葩”，而 dBASE Ⅲ更是 dBASE 家族中的一位佼佼者。它因程序结构精巧、信息处理能力强、操作简便、用户界面好、易学易懂、运行效率高等多方面的优点而在我国ＭＩＳ的开发应用中曾一度占据了主导地位。然而，dBASE Ⅲ自身的弱点，如计算能力较弱、处理速度较慢、没有数组和自定义函数等等，则随着计算机应用领域的拓展而暴露无遗了。这些弱点，不但限制了dBASE Ⅲ在更多应用领域的发展，而且导致了用dBASE Ⅲ开发的应用项目往往只停留在比较低的水平上，难以达到综合信息管理等对其更高的要求。为此，美国Fox Software公司的FoxBASE+关系型数据库管理系统就应运而生了。作为关系型数据库管理系统dBASE Ⅲ的更新换代产品，FoxBASE+ 独领风骚， 它比dBASE Ⅲ功能更强、速度更快、效率更高、用户界面更友好，而且与dBASE Ⅲ完全兼容，因而正逐步取代dBASE Ⅲ而成为当今主流关系型数据库管理系统，成为人们开发管理信息系统的主要工具和得力助手。

FoxBASE+自问世以来，已经多次更新了版本。而目前在我国广泛流行的最新版本，当推FoxBASE+ 2.10版本。

集多年教学和管理信息系统开发的经验，博采众家之长，以目前流行的关系型数据库管理系统FoxBASE+ 2.10版为基础，我们编写了这本书，旨在帮助更多的人了解、学习和掌握FoxBASE+。我们对全书进行了精心的组织，深入浅出、循序渐进，并分散穿插了大量具有实用性、技巧性、趣味性的实例。其目的是为了方便教学和自学，帮助读者建立和提高学习的兴趣与信心，学习程序设计的方法和技巧；启迪思维，开阔视野，在学习中领略乐趣，在编程中提高兴致。本书全面介绍了FoxBASE+ 2.10的功能及在财经管理中的应用。读者学完全书后，不仅可掌握FoxBASE+的全部内容，而且还可开发财经管理或其它管理中的应用程序。本书最后一章给出的帐务子系统是根据财政部一九九三年颁布的新会计制度而设计的，结合一个单位的实际情况，对其进行修改、补充和完善，就能开发出你所期望的财务管理信息系统来。

学习任何一门知识，都得要付出辛勤的劳动。FoxBASE+虽然简单易学，但想不花一番苦功就掌握其精髓是不可能的。计算机课程的实践性很强，特别强

调编写程序和上机操作的实际动手能力。我们认为，只要有兴趣、肯动手，就不难掌握FoxBASE+的全部内容，掌握程序设计的方法和技巧。假以时日，你就能登堂入室，跻身于应用计算机的行家高手之列。

本书第一、二、三章由黄大明编写，第五章由邓正清编写，第六章由陈业菊编写，第十一章由江文编写，姜灵敏编写了其余各章、全部练习及附录。并由姜灵敏主编并最后修改定稿。书中所有例题均在多用户FoxBASE+ 2.10环境下上机调试通过。

我们企望本书作为一朵小葩开放在姹紫嫣红、五彩缤纷的计算机应用的百花园中，不为争奇斗艳，但求增色添辉。如果能为读者学习FoxBASE+提供一些帮助，为计算机进一步普及应用，为FoxBASE+发挥更大的作用，为加速我国会计电算化和其它管理信息系统的建设略尽绵薄之力，则于愿足矣！

作 者

1993年7月于长沙

目　　录

第一章　数据库系统概述

本章主要阐述有关数据库的基本概念，如数据库的特点，数据库管理系统(DBMS)，数据模型，数据库系统的种类等；介绍数据库的部分理论知识，如关系型数据库的理论基础，数据库的保护机制及数据库设计所用的实体—联系方法(E—R)方法；此外，还要在论述过程中介绍数据库的一般设计方法。

本章是我们进一步学习关系型数据库管理系统FoxBASE+2.10的理论基础。

§1.1　信息、数据、数据处理与数据库

信息(Information)这一概念有各种不同含义，这里，我们将信息定义为提供关于现实世界新的事实的知识，而数据则定义为用以载荷信息的物理符号，两者是不可分离而又有一定区别的概念，一方面，信息消化了数据，另一方面，数据则是信息的具体表现，是信息的量化形式。信息比数据更基本地，直接地反映现实的概念。

有了数据(或信息)就有了数据(或信息)处理问题，早期的各种初级计算工具，如算盘、手摇计算机、电动计算机、笔算和纸上记录是手工数据处理阶段。随着生产的发展，社会产生的信息量急剧增加。开发信息资源，充分利用加工处理过的信息资源，对于推动生产和科技的发展将产生巨大的威力。如果说电子计算机硬件技术的飞速发展为快速准确地处理繁杂、大量的数据提供了可能性，那么数据库就被历史地推到数据处理的核心地位。

数据库的产生和发展大约经历了如下三个阶段：

一、人工管理阶段

这一阶段(50年代中期以前)的背景是计算机主要应用于科学计算，硬件背景是外存只有磁带、卡片、纸带等，没有磁盘等直接存取存储设备。从软件方面讲，没有操作系统，没有管理数据的软件，数据处理的方式是批处理。这个时期的管理特点是：

1.数据不保存，因为计算机主要应用于科学计算，重在求得计算结果，一般不需要将数据长期保存。

2.没有软件系统对数据进行管理，程序员不仅规定数据的逻辑结构，而且在程序中还要设计物理结构，包括存储结构、存取方法、输入输出方式等。程序中存取的子程序也随着存储的改变而改变，即数据与程序不具独立性。这样，不仅程序员必须花费许多精力在数据的物理布置上，而且数据在存储上有一些改变，就必须修改程序。

3.基本上没有文件概念，即使有文件，也大都是顺序文件。

4.一组数据对应于一个程序，即数据是面向应用的。因为各应用程序处理的数据不会全无关系，程序与程序之间就会有大量重复数据。

二、文件系统阶段

这一阶段(50年代后期—60年代中期)的背景是计算机不仅用于科学计算，还大量用于管

理，外存储器有了磁盘，磁鼓等直接存取存储器设备。在软件方面已经有了专门的管理数据的软件，一般称为文件系统，或信息管理模块，文件系统包含在操作系统之中。从处理方式讲有：计算化文件批处理，联机实时处理。

在这种背景下，使这一阶段的数据管理形成了如下几个特点：

1.由于计算机大量用于数据处理等方面，数据需要长期保留在外存上反复处理，即对文件进行查询、修改、插入、删除等操作。

2.由于有软件进行数据的管理，程序和数据之间有存取方法进行转换，有共同的数据查询、修改的例行程序存放在程序中。文件的逻辑结构与存储结构有一定的区别，即程序与数据有一定的独立性。这样使程序员可以集中精力于算法，而不必过多地考虑物理细节。

3.文件已经多样化。由于已有了直接存取存储设备，也有了索引文件、链接文件、直接存取文件等而且用上倒排文件进行多码检索。数据的存取基本上是以记录为单位。

上述各点都比第一阶段有很大的改进，但这种方法仍然存在着很大的缺陷，即文件本身还是基本上对应于一个或几个应用程序，或者说数据还是面向应用的。从逻辑结构上讲与前阶段没有什么变化，尽管程序不必要直接和文件打交道而有软件作为接口，但它仍然是一个不具有弹性的无结构的信息集合，存在着冗余度大、空间浪费、文件不易扩充、修改费时、有可能引起不相容等缺点，从而反映不了现实世界之间广泛的内在联系。

三、数据库系统阶段

数据库系统是在文件系统的基础上发展起来的，它克服了文件系统的缺点，其发展也日趋完善。它经历了如下几个发展时期：

1.摇篮时期(60年代中期)。数据库系统开始萌牙，数据库概念开始形成，比较有影响的工作包括：

(1)1968年网状数据系统TOTAL等开始出现；

(2)1969年MCGEE等人开发的层次数据库系统 。IBM公司的IMS(Information management system)发表。

2.发展时期(70年代)。数据库技术日益发展，并有了坚实的理论基础。

(1)出现了以DBTG(Database task group)为典型代表的数据库网状模型；

(2)出现了数据库的关系模型，开始了数据库关系方法和关系数据理论的研究。

3.成熟与集成时期(80年代至今)。这一时期，数据库技术迅速发展，大量商品化的数据库系统广泛应用于各种领域、各种硬件环境，数据库理论更加充实，并且伴随着计算机技术与通讯技术的发展，数据库系统产生了许多集成产品，这一阶段的成果表现在：

(1)丰富的关系型数据库产品。从大型机到微机，从UNIX操作系统到DOS操作系统 ，推出了许多出色的成熟的数据库系统，如：ORACLE、INFORMIX、SYBASE、UNIFY/ACCELL、RDB、PROGRESS、PARADOX、DBASE、FOXBASE、FOXPRO等等。

(2)由于网络通讯技术的发展，在集中式数据库成熟的基础上，产生和发展了分布式数据库系统 (Distriruted database system)，网络结点上的用户可以存取异型机上的任何数据而不用知道该数据存储在哪一物理设备上。因而大型的数据库系统都推出了网络和分布式版本。

(3)为了提高数据库应用系统的开发效率，提高软件人员的生产率，许多软件公司基于数据库系统生产了大量的 CASE(Computer assisting software engineering)工具，第四代开

发环境和第四代语言(Forth generation language，又称面向问题语言)。如INFORMIX的4GL及OA软件，ORACLE的CASE产品系列，Foxbase的FoxVIEW、FoxCODE、FoxDOC、FoxGRAPH等。

随着数据库应用领域的不断扩大，数据库系统已能进行多媒体(MULTIMEDIA)处理，不仅能处理常规的数据，而且还能将图象、声音等作为数据进行处理，如INFORMIX的ONLINE等。

数据库技术的发展日趋完善和成熟，在任何进行事务处理的计算机系统中都不可缺少。有朝一日，数据库系统也会象文件系统一样，融入到操作系统之中。

§1.2 数据库系统的特点和数据模型

一、数据库系统的特点

数据库系统是当代计算机系统的一个重要组成部分，其内涵丰富。数据库技术本身也是逐渐形成的，且直到今天还在发展之中。所以人们从不同角度以不同观点来看待数据库，从而产生的定义也各式各样，五花八门。这里，我们从数据库组成特点的角度给出如下定义：

数据库是按一定的数据模型进行组织，存储在一起的相关的数据的集合，这些数据无不必要的冗余，为多种应用服务，数据的存储独立于使用它们的程序。

从数据库应用特点出发定义如下：

数据库是管理大量的、持久的、可靠的和共享的数据之工具。从这一定义可见，数据库系统是一种管理数据的工具，它的管理对象具有如下特征：

1.“大量”：是大量数据的集合，因其大量，故不能放在通常的内存中，需要大容量的外存作支持；

2.“持久”：指数据必须长久的保留，这表明数据不是简单的为某一特定的应用而准备的，不是象科学计算那样，处理完成数据就随之消失；

3.“可靠”：指一旦系统发生可能的软、硬件故障，可以恢复数据库；

4.“共享”：指若干用户可按照一定有序的方式存取数据库中的数据，避免同步存取可能造成的错误。

与文件系统相比数据库系统有如下显著的特点：

1.数据结构化

在文件系统中，从总体上讲，数据是无结构的，即文件的记录之间没有联系。文件系统只关心记录内数据项之间的联系，它无法实现文件之间的联系。

而数据库系统则不仅要考虑数据项之间的联系，更要考虑记录型之间的联系（是一对一的联系、还是一对多、多对多），因为在客观世界中，实体和实体之间是有联系的，所以反映实体属性的数据(虽然存放在不同的文件内)也是有联系的。在文件系统阶段只有靠应用程序才能实现这种联系，而在数据库阶段则是首先描述这种联系之后通过存取路径来实现。

例如，在产品材料耗用情况的管理中，需要知道产品的基本情况，原材料的基本情况，以及产品消耗某种原材料的数量。可以建立三个文件存放这三方面的数据，显然这三者是有密切联系的，比如要“查询某产品耗用原材料”情况，“耗用某原材料的产品名”等信息，在文件系统阶段必须针对每一查询设计相应的程序，而数据库阶段可以组织成如图 1.1。

对于复杂的查询或其它处理，通过图中的两条存取路径，从一个记录型走到另一个记录

型就可以完成。可以说通过存取路径表示自然的数据联系，从而组织出高度结构化的数据模型，这是数据库系统与文件系统的根本区别。

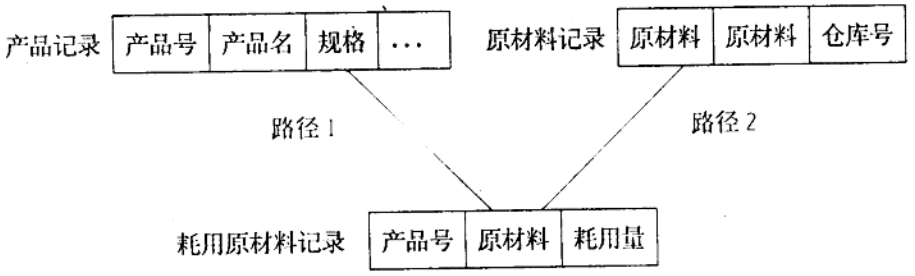


图 1.1

2.数据冗余度小，易扩充

所谓冗余数据，是指可由基本数据导出的数据，这正同数学里的“向量线性相关”的概念一样。

由于数据库里面的数据不再是面向某一应用，而是面向整个系统，因此，就大大的减少了数据冗余，从而就解决了由于数据冗余带来的诸多问题：如空间浪费、存放时间长、数据的不相容和不一致性。

数据库是面向某个系统，还带来了灵活的应用方式：既可以取整体数据的各种合理子集用于不同的应用系统，又可以在应用需求改变或增加时，重新选取不同的子集或加上一小部分数据来满足变化的需要，这就是所谓“弹性大”、“易扩充”。也正是这一特点，才使多用户、多应用共享数据成为现实。

3.具有较高的数据与程序的独立性

数据的独立性是数据库的另一重要特征。

在使用数据库时，应用程序对存储结构和存取方法有较高的独立性。因为系统提供了映象功能(或转换功能)，这种功能分成两种，一是物理独立性，即存储结构与逻辑结构之间由系统提供映象，存储结构或者说物理结构改变了，逻辑结构可以不改变，从而应用程序不必改变。这就是数据与程序的物理独立性。二是逻辑独立性，即当总体逻辑结构改变时，通过对映象的相应改变而保持局部的逻辑结构不变。而程序员是根据局部逻辑结构编写程序的，所以程序也可以不改。这就是逻辑独立性。

4.由于在数据库中强调共享，而计算机的共享一般是并发的，即许多用户同时使用数据库，因此必须提供以下三个方面的控制功能。

(1)安全性(Securily)

数据库的安全性是指保护数据以防止不合法的使用。这就要采取一定的安全保密措施，如用口令或其它手段检查用户身份，检查用户能否查询或修改数据。检查通过才能执行允许的操作。

(2)完整性(Integrily)

数据的完整性是指数据的正确性、有效性与相容性。系统提供必要的功能，保证数据库系统中的数据在输入、修改过程中始终符合原来的定义和规定。

⑶并发控制(Concurrency)

当多用户的并发进程同时存取，修改数据库同一目标时，可能会发生相互干扰而得到错误的结果，并使数据库完整性遭到破坏，因此必须对多用户的并发操作加以控制、协调。

综上所述，数据库是个通用化的、综合性的数据集合，可以提供各种用户共享而具有最小的冗余度和较高的数据与程序的独立性，而且由于多种程序并发的使用数据库，应能有效及时的处理数据，并提供安全性和完整性。所以必须要有一个软件系统—DBMS (Data Base Management System)在建立、运用和维护时进行集中控制。

二、数据模型

在介绍本节之前，先给出几个基本概念的定义：

数据项(或叫字段，域，基本项等)：是数据库中可进行处理的最小数据单位，它可以包含多个字节或位。例如用数据项来描述品名，编号或价格等产品特征。数据项有型和值两个方面。例如在设备管理系统中，厂家是型，而某一具体工厂则为其值。

记录：若干相互关联的数据项的集合。记录是数据库系统中存取数据的基本单位。例如在工资管理数据库中，对某人工资数据的完整描述就构成了一条记录。

文件：是由同类记录组成的信息(数据)的集合。如某单位所有人的工资记录就构成了工资数据文件。

关键字：用来识别一条记录的一条或一种数据项叫关键字。关键字应具有唯一标识性和无冗余性。如工资管理数据库文件中可以用编号作为关键字。此外还可以指定其它辅助关键字。

实体：客观存在并可以相互区分的事物。实体可以指人，指物，也可以指实际的对象或某些概念，还可以指事物本身，甚至事物与事物之间的联系。实体对应于数据库中的一个表(文件)。

属性：实体所具有的某一特性。一个实体可以由若干个属性来刻划，例如，产品这一实体可以由产品号，产品名，规格等属性来描述。属性与数据库中表的字段相对应。

联系：主要是指实体与实体之间的联系。在文件系统中，实体间的联系只有靠应用程序来实现，而在数据库系统中不仅要描述实体间的联系，还要存储这种联系。

数据模型是数据库系统中用于提供信息表示和操作手段的形式构架，是对客观世界事物的抽象。客观世界中，事物之间有一定的联系，由于对这些联系处理问题的方式不同，可以产生不同类型的数据模型。下面对于三类不同的数据模型予以阐述。

1.层次模型(Hierarchical Model)

用如 MS-DOS 磁盘目录那样的树形结构来表示实体之间联系的数据模型叫层次模型。树的节点是实体，树的枝是联系，树中只有顶上的一个唯一的结点无父结点，称之为树根，又叫根结点。这种数据结构就如一棵倒置的树，它有这样的特点：

所有结点(除树根)有且仅有一个父结点。同一个父结点的所有结点处在同一层中，称为兄弟结点。所以，这种树形结构很形象的表示了一种层次结构关系。例如，一个企业的组织机构就是一种层次关系，如图 1.2，工厂是树根；一车间、二车间、三车间是工厂的子女，同处于第二层；班组 1 至 K 都是无子女的结点，称为树叶。因此，用层次关系来表达数据之间的逻辑关系比较自然，容易理解。但是在层次模型中，查询很不方便，需按一定的路径进行，这种路径是由数据间的层次关系确定的，层次较多时，层次模型的操作就变得很复杂，

从而导致了用于层次数据库的操作语句也变得复杂了，给应用带来困难。

2.网状模型(Nelwork Model)

如果取消层次模型中的约束条件，即允许每个结点有多个父结点；至少有一个以上的结点无父结点，就成为网状模型。如图 1.3 所示，产品A和产品B和原材料（或零配件)之间就构成了一个网络模型。由图可见，子结点到父结点的联系不是唯一的，所以不能象层次模

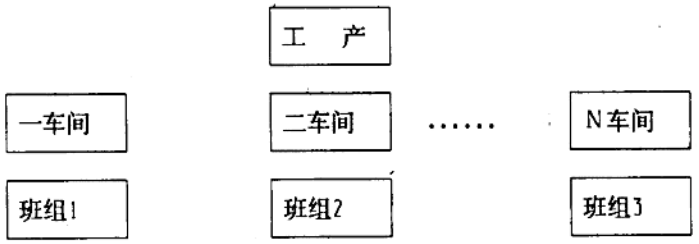


图 1.2

型那样只用父结点来描述记录的联系，而是给每一种联系一个名字，利用这个名字来查找。网络模型可表达复杂的逻辑结构，查询也很灵活，但控制较为复杂。

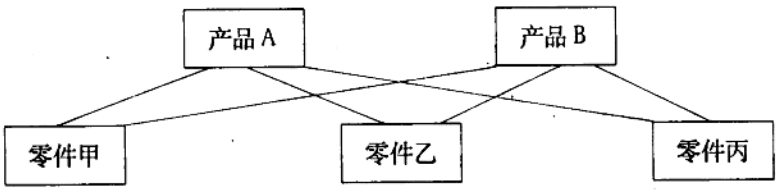


图 1.3

3.关系模型(Relational Model)

关系模型将实体间的联系用二维表来描述。每个二维表称为一个关系。如表 1.1的设备登记表就构成了一张二维表。二维表中的每一行称为一条记录(又叫元组)，每一列是一个属性或字段(又叫域)，即为记录中的数据项，列标题就是对应的字段名，数据项是二维表中不能再分割的最小数据单位。

表 1.1

产品代号	产品名	型号	单位	单价	数量
FD125	发电机	JQG5	台	9985	125
DD205	电动机	JQB6	台	8756	100
BY10	变压器	GB20	台	9650	100

关系模型的特点是：

⑴每一列是类型相同的数据；

⑵列的顺序可以任意；

⑶行的顺序也可以任意；

⑷表中的分量都是不可再分的最小数据项；

⑸表中任意两行的记录不能完全相同，表中不允许再有表；

凡满足上述条件所建立的二维

表即称为关系模型。

按层次和网络模型组建的数据库效率比较高，但是结构不够灵活，不便更动，修改和扩展，不直观，难以掌握。而关系模型是数据的一种最自然，最简单的表达方式，概念清晰、易懂易用、符合自然习惯、加之又具有严格的数学理论基础，因而成了当前数据库领域内应用最广泛的数据模型。

相应于这三种数据模型，有三种数据库管理系统。本书的任务，就是对关系数据库管理系统之———FoxBASE+ 2.10 予以阐述。

§1.3 E—R 方法与数据库设计

在上一节里，我们对“联系”的概念进行了描述，在开展本节之前，有必要对实体间联系的分类作进一步的说明。实体间的联系可分为三类：

1.一对一联系(1:1)—实体集 A 中的每一实体都只与实体集 B 中的一个实体相联系，反之亦然，则两个实体集中的实体是一对一联系。

2.一对多联系(1:N)—实体集 A 中的每一实体，实体集 B 中有多个与之对应，反之不然，则称实体集 A 与实体集 B 有一对多联系。

3.多对多联系(M:N)—实体集 A 中的每一实体，实体集 B 中有多个与之对应，反之亦然，则称实体 A 与实体集 B 有多对多的联系。

例如，一个工厂有一个厂长，且每一个厂长只在一个工厂任职，则工厂与厂长之间具有一对一联系；一个工厂有若干工人，而每个工人只在一个工厂工作，则工厂与工人之间是一对多的联系；一个产品耗用多种原材料，且每种原材料可能耗用到多种产品上去，所以产品与原材料之间是多对多联系。

图 1.4 说明了上述三种联系关系。

在下面将要阐述的 E—R 方法中，描述概念模型有如下规定：

1.用长方形表示实体，在框内写上实体名。

2.用椭圆表示实体属性，并用无向边把实体与其属性连接起来。

3.用菱形表示实体间的联系，菱形框内写上联系名。用无向边把菱形分别与有关实体相连接，在无向边旁标上联系的类型 (1:N, 1:1, M:N)。若实体之间联系也具有属性，则把属

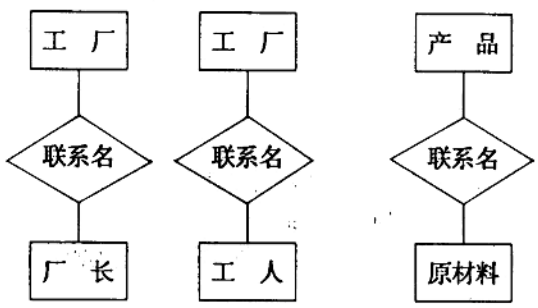


图 1.4

性和菱形也用无向边连接上。如产品与原材料之间的联系有“耗用量”，要将它与表示联系的棱形联接起来。

一、E—R 方法

在数据库中是用数据模型(Data Model)来对现实世界进行抽象，进而表示成为机器中存取的数据。而数据模型是通过概念模型转换而成的。现实世界中的认识对象首先是通过人脑的抽象表示成为概念模型(或信息模型)之后经过进一步的转换才成为机器世界的数据模型。因此对数据模型的研究，必先讨论一下概念模型。

概念模型是数据库设计人员在认识现实世界中的实体及实体之间联系后所进行的抽象，并用一种简洁，清晰的工具进行表示，是设计者与用户之间进行交流的语言。

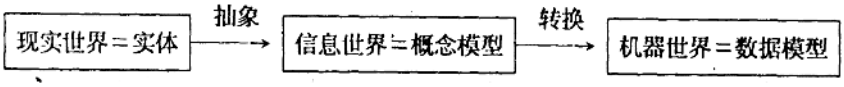


图 1.5

概念模型的表示方法最常用的是实体→联系方法(Entity—Relationship mode)，简称 E—R 方法。

E—R方法是进行数据库设计时使用的主要工具，它往往在管理信息系统研制的系统分析与数据流程图DFD一起画出，是进行系统设计的依据。

例如：某个工厂的供销部门要对该厂产品及其所耗用的原材料进行核算，经过对具体事物的抽象整理得到有关实体与联系的如下资料：

实体：

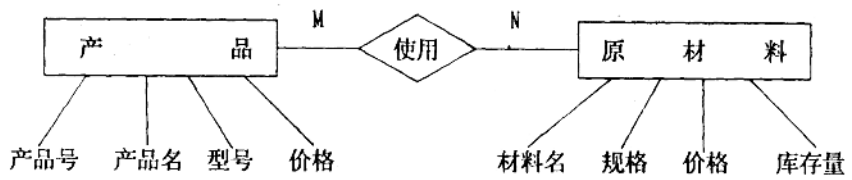
· 产品，属性包括产品号、产品名、型号、价格。

· 原材料，属性包括材料名、规格、价格、库存量。

在经过调查与分析的基础上得到实体间的联系如下：

一种产品可以耗用多种原材料，并且每一种原材料可能用于多种产品，则产品与原材料的联系是多对多(M:N)的联系。

以上这些实体及其联系可以用E—R图简洁清晰地描述出来：



有了这一 E—R 图，在设计产品及原材料核算的数据库时就有了依据。

综上所述，E—R图是对现实世界的抽象，它描述了相应的概念模型。概念模型与具体的 DBMS 所支持的数据模型相独立，是各种数据模型的共同基础，是数据库设计的依据和工具。

二、数据库设计

数据库设计是指对一个给定的应用环境，构造最优的、最有效的数据库模式，建立数据

库及其应用系统，使之能够高效率地存取数据，满足各种用户的应用需求。

数据库一般采用规范化方法，运用软件工程的思想，依据各种设计准则和规程进行。

通常将其分为四大阶段：需求分析，概念设计，逻辑设计和物理设计。之后，数据库便进入了实现、运行和维护阶段，如图1.6所示。

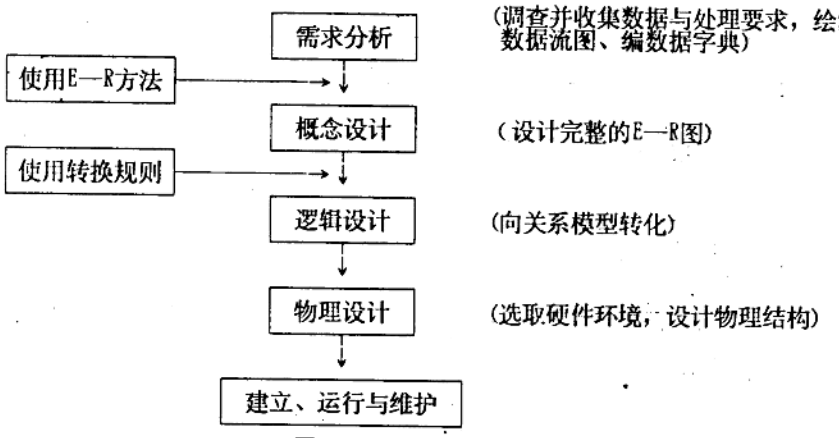


图 1.6

数据库设计往往是在一个通用的DBMS支持下进行的，这里我们都以FoxBASE+为基础设计关系型数据库。

1.需求分析

用户的需求是设计的起点也是设计的目标。为此，要设计好一个数据库，必须了解和分析用户要求，包括数据要求，加工要求和限制条件等。需求分析的成果要写成书面报告。整理后交用户确认，一旦认可，就应把它作为用户与设计者共同遵守的规范。

2.概念结构设计

在需求分析的基础上，依据数据流程图来设计概念结构，概念结构是在需求分析的基础上对客观世界的进一步描述，其主要作用是详细描述实体之间的联系，实体的属性构成，以便向其它数据模型,(这里主要是关系模型)转化,从而设计出具体FoxBASE+下的数据库。

概念结构独立于数据库逻辑结构。概念结构的具体作法如下：

第一步：选择局部应用，设计分E—R图。

由于我们所设计的数据库往往很大，涉及的方面很广，一下要画出其数据库所用的全局E—R图是不可能的。为此，可以根据数据流图DFD的划分，根据实际情况，逐个画出相应的E—R 图，称为分E—R图。

例如，对某工厂的生产和物资供应进行管理时，可设计为产品和零配件两个实体，且知一种产品需多种零配件，每一种零配件可用于多种产品，显然两实体之间是多对多的联系。属性与实体联系如图 1.7。

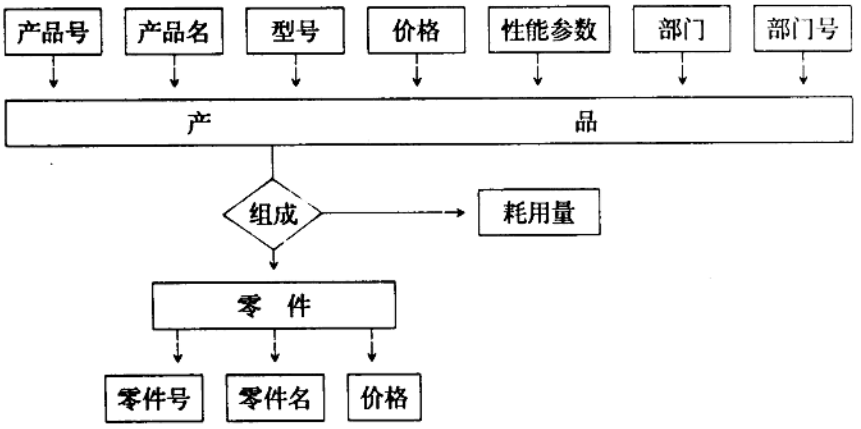


图 1.7

在对零配件进行管理时，假设可涉及到如下三个实体：零配件、进货单位、存放的库房号。设某种零配件只来源于某一产地，而某一产地只能向该厂提供某种零配件，则它们之间是(1:1)的联系。又设每一库房可存放多种零配件， 某种零配件可同时存放于多个库房，则它们之间的联系是(M:N)。可得E—R图如图 1.8。

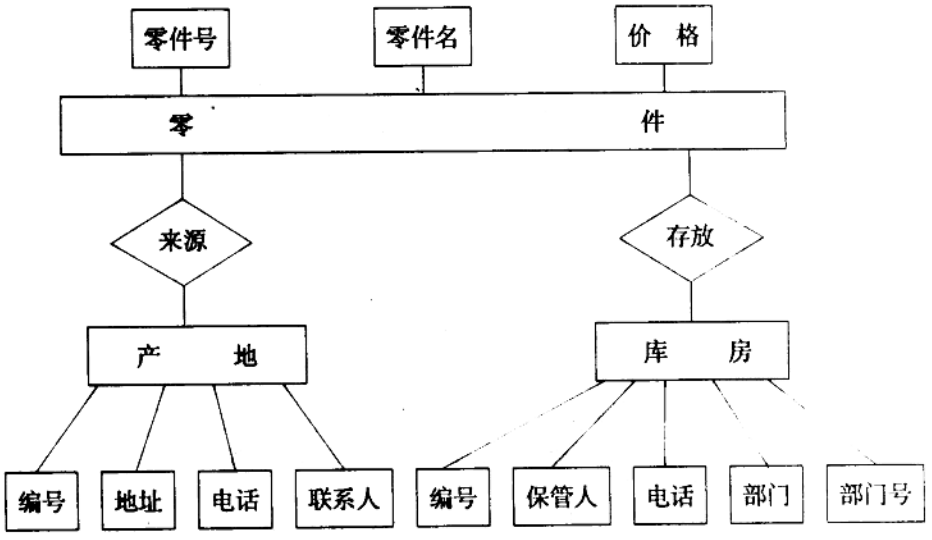


图 1.8

第二步：E—R图的合并。

在分 E—R 图的基础上进行合并，使之成为描述整个组织内实体之间联系的总的概念结构。合并按下列原则进行：

⑵消除分E—R图之间的不一致，并且互相补充。

⑵消除不必要的冗余，设计基本E—R图。

由以上E—R图可见，库房和产品的属性中都有部门和部门号，它们肯定是重复的冗余数据项。可以设想：产品是由某一部门生产，库房属于某一部门，一个部门能生产多种产品，有多间库房，则可把部门单独作为一个实体，它与产品和库房之间是(1:N)的联系。因此，可得总的E—R图如图 1.9。

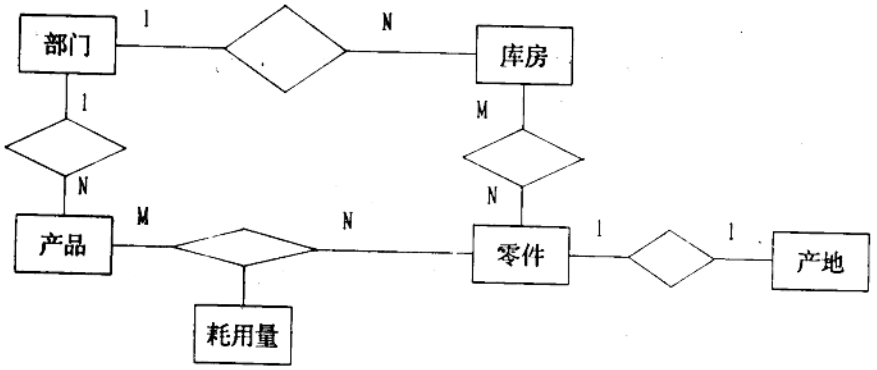


图 1.9

图中，各实体的属性为：

部门：(部门号、部门名、负责人、地址、电话)

产品：(产品号、产品名、型号、价格、性能参数)

库房：(编号、保管人、电话)

零件：(零件号、零件名、价格)

产地：(编号、地址、电话、联系人)

这些属性的具体描述都在数据字典中说明。

3.逻辑结构设计

逻辑结构设计的任务是把独立于任何一种具体数据模型的概念结构(基本E—R图)转换为与选用的DBMS支持的数据模型相符合的过程。本书使用的是关系型DBMS—FoxBASE，所以这里只讲关系数据模型的转化。

转化规则如下(只有通过这样的转换我们才能在FoxBASE+的点状态下使用“CREATE”命令建立相应的库结构)：

规则 1：一个实体转换为一个关系表。实体的属性就是关系的字段，实体的码(或关键字，即唯一能标识一个实体的属性)就是关系的码或关键字段。

规则 2：实体间的联系，也用关系表实现，与该联系相连的各实体的码以及联系即属性转换为关系表的字段。具体情况有三种：

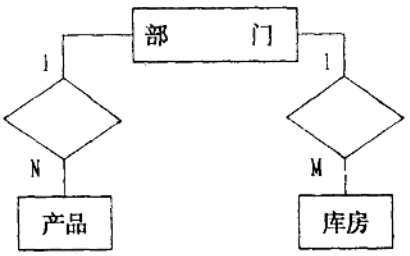
⑴如果两实体的联系是(1:1)，依规则 1，两个实体建立两个关系表，只要将1:1两端的任何一个实体码放入另一个实体中作为属性，就可实现1:1的联系，可以不建另外的关系表描述1:1的联系。

⑵如果两实体间联系是1:N，则两个实体建立两个关系表，并且把1端的实体码放到N端的实体中，作为N端实体的属性之一(即N端关系表的字段之一)。这样，1:N 的联系也不用建额外的关系表来描述。

⑶如果两实体间联系为M:N，则除两个实体建立两个关系外，还要建立另外一个关系表用来描述多对多的联系，这一关系的码是两个实体码的组合，联系的属性作为关系的属性。

根据以上规则，即可把前面的基本 E—R 图转换成相应的关系表，并填入数据。

部门与库房、与产品是1:N的联系，属于第二种情况，因此，可以建立与三个实体相对应的三个关系表，并且把1端“部门”的码—“部门号”放入产品和库房关系表中，作为其字段之一。



部门:

部门号	名　称	负责人	地址	电话
FAC01	第一分厂	张XX	1区306	446846
FAC02	第二分厂	李XX	2区402	414749
……	……	……	……	……

产品:

产品号	产　品　名	型号	价格(元)	部门号
JQ106	钻石碑吊扇	56英寸	110	FAC01
JK100	钻石碑台扇	18英寸	160	FAC02
……	……	……	……	……

库房:

编号	保管人	电话	部门号
001	刘XX	224763	FAC01
002	王XX	224764	FAC01
……	……	……	……

产品与零件的联系，库房与零件的联系都是M:N的联系，属于第三种情况，要建立新的关系描述。新关系表的属性应包含两实体的码(即两实体的关键字)以及联系的属性(若有的话)。