

## 数据的操纵和检索——综合数据库管理系统

到目前为止，我们集中讨论了传统的文件处理。此处，文件是作为一个独立的逻辑实体来处理的，而且通常与某个特定的业务领域相联系。为了把业务上相近的那些部门的文件综合在一起，必须预先排序记录和合并文件。这样做可能耗费大量的时间，有时甚至是不可能实现的。

通常，文件是为了满足特定业务领域某部门的专门需要而设计的，如果某部门想使用保存在一个文件上的部分数据，而该文件是另一部门为自己使用而设计的，那么他们必须抽取这些数据并重新整理后才能使用。大多数部门总是宁愿建立和保持和一个类似(而不同)的文件以更好地满足自己的使用要求，而不愿兼顾其他部门的业务需要。这样做的结果形成了大量的带有重复数据的文件。前面的例子中已经提到，某所大学就保存了 75 个独立的计算机文件，而每个文件中都包含有学生记录。每当一个学生结婚了或者改换了住处时，则必须修改 75 个文件。

数据的冗余会使成本提高，而设计一个综合的资源共享数据库则可以把数据冗余度降低到最小程度。综合的资源共享数据库把公司看作一个整体来提供服务，而不只是为公司某特殊部门服务。利用数据库管理系统(DBMS)软件就可能实现一个综合的数据库。

早在 70 年代初期就出现了 DBMS 技术，但是直到 70 年代末期仍未得到普遍接受。早期的 DBMS“软件包”效率不高，实际上比传统的文件处理的要求更高的硬件能力。目前 DBMS 对硬件的要求已经降低了，甚至在小型计算机系统上也能运行 DBMS 软件，造成推迟接受和实现 DBMS 技术的主要原因可能是信息服务人员缺乏专门的知识，而且用户管理人员又不愿合作和支持建立一个共享的数据库。目前对于设计和运行 DBMS 软件已经有了广泛的了解，而且部分用户管理人员已经认识到应该把信息作为一个公司的资源来看待。

### 数据库环境的好处

1. 经济上的好处。通过先进的数据结构技术，DBMS 提供了最小化数据冗余度的可能性。当然，为了提高处理效率，即使有了 DBMS 帮助，有时还需要在共享的数据库中保留一定的重复数据。由于减少了数据的冗余度，从而也简化了收集和更新数据的过程。就前面的例子而言，只需要在一处(数据库中)更新学生地址就可以了。

2. 数据库技术也给信息服务专业人员带来的好处。综合资源共享数据库为系统分析员打开了崭新的大门。数据库能向系统分析员提供传统文件所不能提供的信息。由于有了 DBMS，就更容易得到数据，因此使程序设计任务要比原先更为简单。此外，数据是独立于程序的。这意味着用户管理人员可以增加数据库中的数据、修改库中的现有数据或者从库中删除数据，这些并不影响现有的程序，而在过去则需要测试和修改几十个(有时几百个)程序。当然，数据库管理系统还有其它重要的优点，但是这些已超出本书的范围。

3. 信息方面的好处。一个综合资源共享数据库能以有序的方式提供一种满足信息要求的结构。实际上，数据库是能够充分利用公司信息资源的唯一工具。由于数据库和 DBMS 在产生报表方面提供了极大的灵活性，因此，也给决策过程带来了同样的灵活性。

4. DBMS 软件在支持联机系统方面是特别有价值的。共享数据库不仅可供多个用户存取而且还特别能反映出数据的当前状态。对于许多查询来说，及时性是很关键的。例如，某个经理由于设备故障不得不关闭某个工作站，此时他有可能要查询其它工作站的状态，并且利用得到的信息来改变进行中的工作路线。

### 方法

DBMS 克服了传统文件处理的局限性。它依靠灵活多变的数据结构(将数据元及记录彼此联系的方法)来做到这一点。基于不同数据结构的 DBMS 之间差别很大。有三种设计 DBMS 软件的方法

1. 数据系统语言会议(CODASYL)设计并发布的基于网络或丛形数据结构的 DBMS 指南。
2. IBM 的信息管理系统(IMS)使用的一种层次结构。
3. 某些最近设计的基于关系数据结构的 DBMS。

上述的每一种方法是基本概念和术语的差别都很大。因此，为了减少混淆和便于理解，

以下的阐述和讨论只使用网络方法。

## 库设计

### 1. 数据库设计方法

我们通过下面的例子来说明数据库管理系统和一种数据库设计方法的原理。考察下述情况。

某图书馆保存了一个文件，该文件每一个记录都包含了下述数据元：

- (1) 书名
- (2) 作者
- (3) 出版社
- (4) 出版社地址
- (5) 类别
- (6) 出版年份

图书馆馆长希望在获得决策信息方面有更多的灵活性。而现在文件不可能满足其要求。经过与信息处理人员交换意见之后，他们决定实现一个基于 CODASYL 的数据库管理系统。

检查现有文件发现某些数据重复出现。每本书(每一书名)都分别有一个记录。于是，写过几本书的作者的名字在他写的每本书的记录中重复出现。在一个图书馆里可能有某出版社出版的数百本，甚至数千本书，但是在该文件的每本书的记录中都重复地出现出版社名和出版社地址。

### 2. 设立数据库的步骤

设计数据库的方法之一是使用主题词一属性矩阵作为识别记录和这些记录之间关系的一个辅助工具。以下各步骤描述了这种设计数据库的方法。

第一步，在的主题词一属性矩阵的左边列出了逻辑主题词。在本例中，这些主题词是：书名、作者和出版社。在矩阵的顶端列出了所有可能的属性。属性是提供与主题词有关的信息的任一数据元。

第二步：在属性与主题词之间的关系有一对一的，一对多的或者不存在关系。例如，一特定书名只有一个出版社。所以在书名与出版社之间存在一对一的关系。然而，一个书名可以有几个作者。因而，在书名与作者之间存在一对多的关系。用“|”表示一对一关系，用“M”表示一对多关系，而且空白表示无关系存在。

第三步：数据库记录类似于传统文件的记录，它是相关数据元的集合。可以将记录从数据库中读出来，也可以将记录写到数据库中去。数据库记录也称为节段。在这一步，用主题词一属性矩阵来标识记录。这是通过列出与每个主题词一对一关系的属性来实现的。通常，对应每个主题都有一个记录，“书名”记录包含出版社、出版年份和类别，如下所示：

书 名	类 别	出 版 年 份
作 者		
出 版 社	出 版 社 地 址	

为了将数据冗余度减至最小，在记录中将不包含一对多关系的属性，而通过数据结构来保留这些一对多的关系。在传统的文件组织中，“作者”数据元包含“书名”记录中，如下所示：

书 名	类 别	出 版 年 份	作 者 1	作 者 2	作 者 3	作 者 4
-----	-----	---------	-------	-------	-------	-------

第四步：数据库设计过程的下一步是建立记录之间的关系。图 20.6.11 给出主题一属性矩阵中指出的一对多的关系。出版社与书名之间的一对多的关系是用两个记录之间的一条连线来指出的，该连线有一箭头指向出版社记录。指向书名记录的两个箭头表示每个出版社可以有一个以上的书名。这种出版社—书名的组合称之为一个系。其它的系是：书名—作者、作者—出版社、作者—书名和出版社—作者。多对多的关系(每一端都有两个箭头)在处理时可能会引起混淆，因而是允许的。

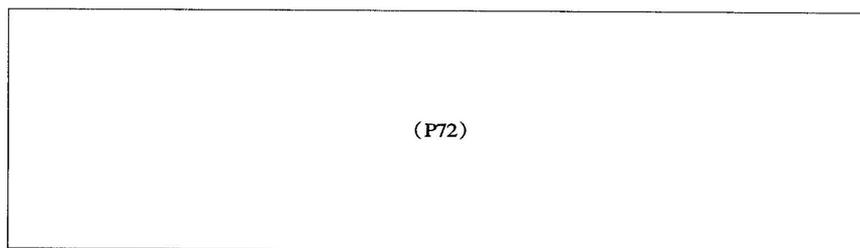


图 20.6.11 数据库记录之间的关系

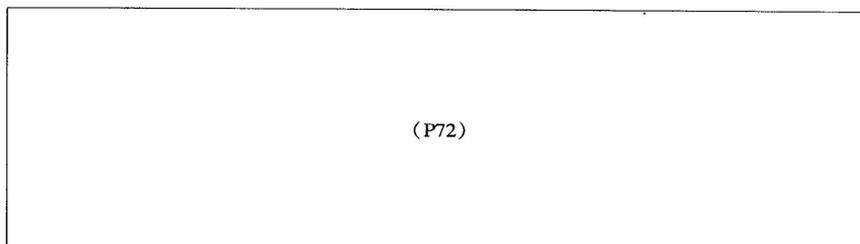


图 20.6.12 一个完整的网络模型

第五步：下一步消去冗余的属性和系。记住，我们的目的是将作者、书名和出版社记录输入到数据库，然后经由系建立的联系从不同的记录中检索数据。因此，必须消去对此目的是冗余的那些系。在本例中，显而易见作者-出版社系或作者-书名系是冗余(重复)的，由于作者与书名相匹配的机会比它与出版社匹配的机会多，因此应该消去作者-出版社系。图 20.6.12 给出了结果模式。该模式是数据库逻辑结构的一种图式表示。

现在图书馆馆长可以进行如下的查询：

- ①列出某作者写的所有书的书名；
- ②按字母顺序列出由 Pientice-Hall 在 1981 年出版的那些书的书名；
- ③列出 1987 年以来至少在三个出版社出版书的那些作者。

该数据库(正如所设计的那样)很容易实现上述查询及其它类似的查询。对图书馆原有文件来说，类似的查询不仅要求处理整个文件，而且可能还要处理好几遍。有时，还要求对文件进行排序和合并。

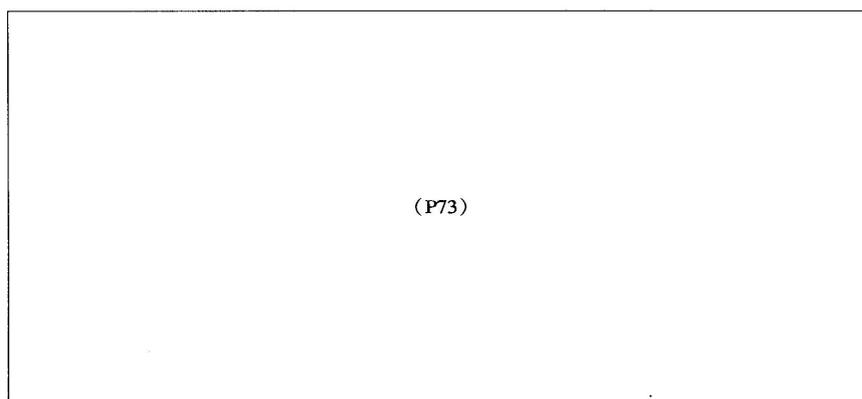


图 20.6.13 数据库结构的一个值

如果图书馆馆长在一年之后决定需要另一个记录或属性，那么，数据库管理员(DBA)可以重新产生数据库，增加属性，甚至增加系而不影响现在的程序。

图 20.6.12 表示该模式，而图 20.6.13 表示数据库结构的一个值。模式与值的关系类似于数据元与数据项的关系。一个是数据的定义，而另一个是实际的值或内容。