第5章 数据库设计

5.1 数据库设计概述

5.1.1 数据库设计的定义和知识要求

数据库设计是指对于一个给定的应用环境，根据用户的需求，在某一具体的数据库管理系统上，构造一个性能良好的数据模式，建立数据库及其应用系统，使之能够有效地存储数据，满足各种用户的信息需求和处理需求。

(1) 信息需求。信息需求表示一个单位所需要的数据及其结构，表达了对数据库的内容及结构的要求，也就是静态要求。信息需求定义所设计的数据库将要用到的所有信息，描述实体、属性、联系的性质，描述数据之间的联系。

(2) 处理需求。处理需求表示一个单位需要经常进行的数据处理，表达了基于数据库的数据处理要求，也就是动态要求。处理需求定义所设计的数据库将要进行的数据处理，描述操作的优先次序、操作执行的频率和场合，描述操作与数据之间的联系。

因此，数据库设计就是把现实世界中的数据，根据各种应用处理的要求，加以合理地组织，使其满足硬件和操作系统的特性；同时，利用已有的 DBMS 建立数据库，使其能够实现应用系统的目标。

数据库设计是一个庞大而且复杂的工程，数据库设计人员应该具备以下知识。

(1) 计算机科学的基础知识和程序设计的方法与技巧。

作为一个数据库的设计人员，首先必须是一个懂得计算机专业的人员，而作为计算机专业的人员，最基本的就是计算机科学的基础知识，其次应该掌握关于程序设计的知识和程序设计的方法与技巧。

(2) 数据库的基本知识和数据库设计技术。

除了具有计算机的基础知识以外，作为数据库的设计人员必须具有数据库的基本知识和数据库设计技巧。

(3) 软件工程的原理和方法。

在数据库领域内，常常把使用数据库的各类系统称为数据库应用系统。数据库应用系统的开发应该遵循软件工程的方法和原理。尤其是大型数据库设计，其开发周期长、耗资大，失败的风险也大，必须把软件工程的原理和方法应用到数据库建设中来。

(4) 应用领域的知识。

应用领域的知识随着应用系统所属的领域不同而不同，如财务管理、仓库管理、人事管理、教务管理等。而且同样是教务管理，大学、中学、小学等不同类型的学校，各不相同，即使

都是高等学府各个学校的管理方式也不相同。因此,数据库设计人员必须深入实际与用户密切结合,对应用环境、专业业务流程进行详细的调查研究,才能设计出符合具体应用领域和用户要求的数据库应用系统。

5.1.2 数据库设计的内容

数据库设计包括结构设计和行为设计两方面的内容。

1. 数据库的结构设计

数据库的结构设计是指根据给定的应用环境,进行数据库的模式或子模式的设计。它包括数据库的概念设计、逻辑设计和物理设计。数据库模式是各应用程序共享的结构,是静态的、稳定的,一经形成后通常情况下是不容易改变的,所以结构设计又称为静态模型设计。

2. 数据库的行为设计

数据库的行为设计是指确定数据库用户的行为和动作。而在数据库系统中,用户的行为和动作指用户对数据库的操作,这些要通过应用程序来实现,所以数据库的行为设计就是应用程序的设计。用户的行为总是使数据库的内容发生变化,所以行为设计是动态的,行为设计又称为动态模型设计。

数据库的结构设计和行为设计是不能分离的,分离会导致数据与程序不易结合,增加数据库设计的复杂性。本章重点介绍数据库的结构设计,对于数据库的行为设计请参考“软件工程”相关书籍。

由于数据库的设计和开发是一个庞大而且复杂的工程,涉及多学科的综合性技术,所以,数据库设计是涉及硬件、软件和管理的综合技术,这也是数据库设计的另外一个特点。有人讲“三分技术,七分管理,十二分基础数据”是数据库建设的基本规律,这是有一定道理的。

5.1.3 数据库设计方法

数据库设计方法目前可分为 4 类:直观设计法、规范设计法、计算机辅助设计法和自动化设计法。

1. 直观设计法

直观设计法也叫手工试凑法,它是最早使用的数据库设计方法。这种方法依赖于设计者的经验和技巧,缺乏科学理论和工程原则的支持,设计的质量很难保证,常常是数据库运行一段时间后又发现了各种问题,这样就不得不修改原有设计,增加了系统维护的代价。因此这种方法越来越不适应信息管理发展的需要。

对于一个简单的程序设计过程来说,这样的方法具有周期短、效率高、操作简便、易于实现等优点。但是对于数据库设计,尤其是大型数据库系统的设计,由于其信息结构复杂、应用环境多样、应用需求全面等系统化综合性的要求,通常需要若干个人的共同努力、相互协调,综合多种知识才能完成,所以,在具有丰富经验和设计技巧的前提下,还应该以严格的科学理论和软件工程设计原则为依托,完成数据库设计的全过程。

2. 规范设计法

规范设计法是将数据库设计分为若干阶段,明确规定各阶段的任务,采用自顶向下、分层实现、逐步求精的设计原则,结合数据库理论和软件工程设计方法,实现设计过程的每一

细节，最终完成整个设计任务。

1978 年 10 月，来自三十多个国家的数据库专家在美国新奥尔良市专门讨论了数据库设计问题，他们运用软件工程的思想和方法，提出了数据库设计的规范，这就是著名的新奥尔良法，它是目前公认的比较完整和权威的一种规范设计法。新奥尔良法将数据库设计分成需求分析（分析用户需求）、概念设计（信息分析和定义）、逻辑设计（设计实现）和物理设计（物理数据库设计）。此后，S. B. Yao 等人提出了数据库设计的 6 个步骤：需求分析、模式构成、模式汇总、模式重构、模式分析和物理数据库设计，从而逐渐形成了数据库规范化设计方法。

目前，常用的各种数据库设计方法都属于规范设计法，即都是运用软件工程的思想和方法，根据数据库设计的特点，提出了各种设计原则与设计规程。常用的规范化设计方法主要有：基于 E-R 模型的数据库设计方法，基于 3NF 的数据库设计方法，基于视图概念的数据库设计方法等。

1）基于 E-R 模型的数据库设计方法

基于 E-R 模型的数据库设计方法是由 P. P. S. chen 于 1976 年提出的数据库设计方法，其基本思想是在需求分析的基础上，用 E-R 图构造一个反映现实世界实体之间联系的企业模式，然后再将此企业模式转换成基于某一特定的 DBMS 的概念模式。

2）基于 3NF 的数据库设计方法

基于 3NF 的数据库设计方法是一种结构化设计方法，其基本思想是在需求分析的基础上，确定数据库模式中的全部属性和属性间的依赖关系，将它们组织在一个单一的关系模式中，然后再分析模式中不符合 3NF 的约束条件，将其进行投影分解，规范成若干个 3NF 关系模式的集合。

其具体设计步骤分为以下 5 个阶段。

（1）设计企业模式，利用规范化得到的 3NF 关系模式画出企业模式；

（2）设计数据库的概念模式，把企业模式转换成 DBMS 所能接受的概念模式，并根据概念模式导出各个应用的外模式；

（3）设计数据库的物理模式（存储模式）；

（4）对物理模式进行评价；

（5）实现数据库。

3）基于视图的数据库设计方法

此方法先从分析各个应用的数据着手，其基本思想是为每个应用建立自己的视图，然后再把这些视图汇总起来合并成整个数据库的概念模式。合并过程中要解决以下问题。

（1）消除命名冲突；

（2）消除冗余的实体和联系；

（3）进行模式重构，在消除了命名冲突和冗余后，需要对整个汇总模式进行调整，使其满足全部完整性约束条件。

除了以上三种方法外，规范化设计方法还有实体分析法、属性分析法和基于抽象语义的设计方法等。规范设计法从本质上来说仍然是手工设计方法，其基本思想是过程迭代和逐步求精。

3. 计算机辅助设计法

计算机辅助设计法是指在数据库设计的某些过程中模拟某一规范化设计的方法,并以人的知识或经验为主导,通过人机交互方式实现设计中的某些部分。目前许多计算机辅助软件工程工具可以自动或辅助设计人员完成数据库设计过程中的很多任务,比如 Sysbase 公司的 PowerDesigner 和 Oracle 公司的 Oracle Designer。

4. 自动化设计法

自动化设计法是缩短数据库设计周期、加快数据库设计速度的一种方法。这种方法往往是直接用户,特别是非专业人员在对数据库设计专业知识不太熟悉的情况下,较好地完成数据库设计任务的一种捷径。例如,设计人员只要熟悉某种 MIS 辅助设计软件的使用,通过人机会话,输入原始数据和有关要求,无须人工干预,就可以由计算机系统自动生成数据库结构及相应的应用程序。由于该设计方法基于某一 MIS 辅助设计系统,从而受限于某种 DBMS,使得最终产生的数据库及其软件系统带有一定的局限性。此外,一个好的数据库模型,往往需要设计者与用户反复商讨,是在用户的参与及合作下所形成的一个最终结果,设计者的经验及对应用部门的熟悉程度,在很大程度上是数据库设计质量的关键。因此,相对于其他设计方法而言,自动化设计法并不是一种理想的设计手段。下面围绕规范化设计法,深入分析和介绍其详细设计过程。

5.1.4 数据库设计的基本步骤

按照规范设计方法,考虑数据库及其应用系统开发全过程,并仿照软件生存周期,将数据库设计分为需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库实施和运行维护 6 个阶段。数据库设计过程可以用图 5-1 表示。

1. 需求分析

需求分析是对具体应用环境的业务流程和用户提出的各种要求加以调查研究和分析,并和用户共同对各种原始数据加以综合、整理的过程,是形成最终设计目标的首要阶段,也是整个数据库设计过程中最困难的阶段。该阶段任务的完成,将为以后各阶段任务打下坚实的基础。因此,对用户的各种需求及数据,能否做出准确无误、充分完备的分析,并在此基础上形成最终目标,是整个数据库设计成败的关键。

2. 概念结构设计

概念结构设计是对用户信息需求所进行的进一步抽象和归纳,结果为数据库概念结构,通常用 E-R 模型来表示。

数据库的概念结构与 DBMS 和相关软硬件无关,它是对现实世界中具体数据的抽象,实现了从现实世界到信息世界的转换过程。概念结构设计是数据库设计的一个重要环节,是数据库的逻辑结构设计和物理结构设计的基础。

3. 逻辑结构设计

概念结构设计的结果是得到一个与 DBMS 无关的概念模式,而逻辑结构设计就是将概念模式转换为与选用的具体 DBMS 所支持的数据模型相符合的逻辑结构。所以,在逻辑结构设计阶段选择什么样的数据模型和哪一个具体 DBMS 尤为重要,它是能否满足用户各种要求的关键。

在逻辑结构设计阶段还有一个很重要的工作就是模式优化,该工作主要以用规范化理

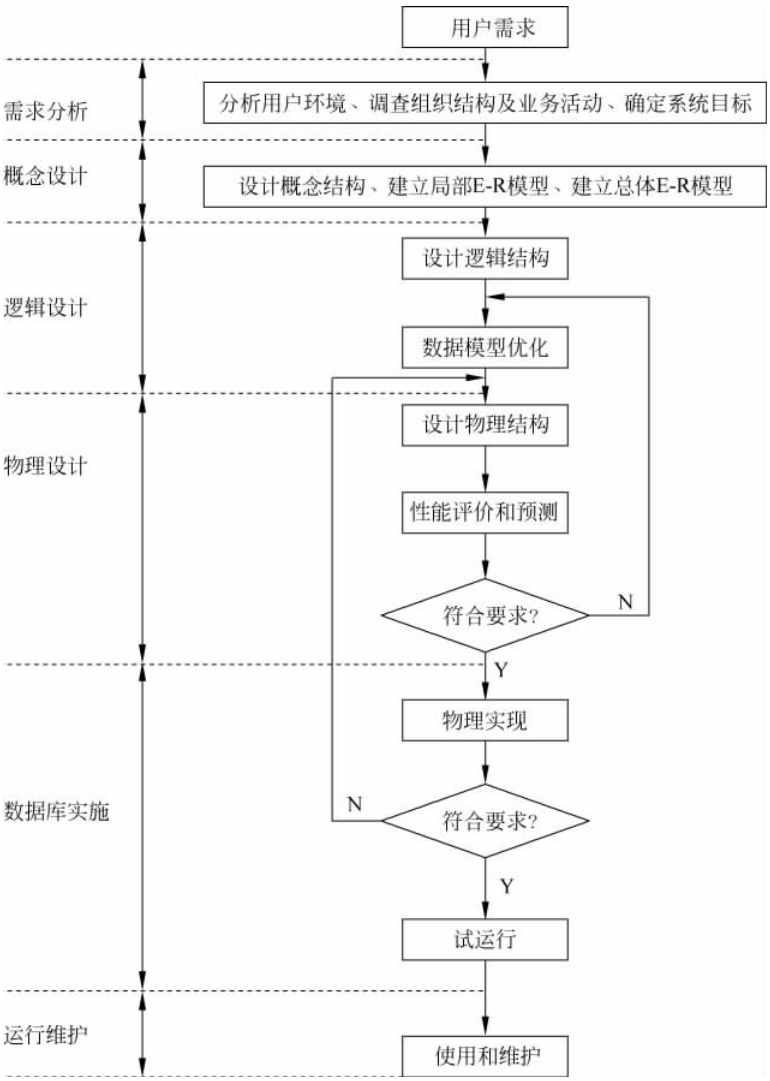

图 5-1　数据库设计步骤

论为指导，目的是能够合理存放数据集合。逻辑结构设计阶段的模式优化，已成为影响数据库设计质量的一项重要工作。

4. 物理结构设计

数据库物理设计是将逻辑结构设计阶段所产生的逻辑数据模型，转换为某一计算机系统所支持的数据库物理结构的实现过程。

数据库的物理结构主要指数据库的存储记录格式、存储记录安排和存储方法，完全依赖于给定的硬件环境、具体的 DBMS 和操作系统。

存储记录格式的设计包括记录的组成、数据项的类型、长度，以及逻辑记录到存储记录的映射。存储记录的安排是指可以把经常同时被访问的数据组合在一起。存取方法的设计主要是指存取路径，存取路径分为主存取路径与辅存取路径，前者用于主码检索，后者用于辅助键检索。

除此之外,物理结构设计还要进行完整性和安全性考虑,设计者应在完整性、安全性、有效性和效率方面进行分析,做出权衡。

完成物理结构设计后,对该物理结构做出相应的性能评价,若评价结果符合原设计要求,则进一步实现该物理结构。否则,对该物理结构做出相应的修改,若属于最初设计问题所导致的物理结构的缺陷,必须返回到概念设计阶段修改其概念数据模型或重新建立概念数据模型,如此反复,直至评价结果最终满足原设计要求为止。

5. 数据库实施

数据库实施阶段,即数据库调试、试运行阶段。一旦数据库物理结构形成,就可以用已选定的 DBMS 定义、描述相应的数据库结构,装入数据库数据,以生成完整的数据库,编制有关应用程序,进行联机调试并转入试运行,同时进行时间、空间等性能分析,若不符合要求,则需调整物理结构、修改应用程序,直至高效、稳定、正确地运行该数据库系统为止。

6. 数据库运行和维护

数据库实施阶段结束,标志着数据库系统投入正常运行的开始。严格地说,数据库运行和维护不属于数据库设计的范畴,早期的新奥尔良法明确规定数据库设计的 4 个阶段,不包括运行和维护内容。随着人们对数据库设计的深刻了解和设计水平的不断提高,已经充分认识到数据库运行和维护工作与数据库设计的紧密联系。数据库设计是一种动态和不断完善的运行过程,运行和维护阶段开始,并不意味着设计过程的结束,任何哪怕只有细微的结构改变,也许就会引起对物理结构的调整、修改,甚至物理结构的完全改变,因此数据库运行和维护阶段是保证数据库日常活动的一个重要阶段。

5.2 需求分析

5.2.1 需求分析的任务

需求分析的任务是通过详细调查现实世界要处理的对象(组织、部门、企业等),充分了解原系统(手工系统或计算机系统)工作概况,明确用户的各种需求,然后在此基础上确定新系统的功能。新系统必须充分考虑今后可能的扩充和改变,不能仅按当前应用需求来设计数据库。

需求分析的重点是调查、收集与分析用户在数据管理中的信息要求、处理要求、安全性与完整性要求。

1. 信息要求

信息要求是指用户需要从数据库中获得信息的内容与性质。由用户的信息要求可以导出数据要求,即在数据库中需要存储哪些数据。

2. 处理要求

处理要求是指用户要求完成什么处理功能,如对处理的响应时间有什么要求,处理方式是批处理还是联机处理等。新系统的功能必须能够满足用户的信息要求、处理要求。

3. 安全性与完整性要求

确定用户的最终需求其实是一件很困难的事,这是因为一方面用户缺少计算机知识,开始时无法确定计算机究竟能为自己做什么,不能做什么,因此无法一下子准确地表达自己的

需求，他们所提出的需求往往不断地变化。另一方面，设计人员缺少用户的专业知识，不易理解用户的真正需求，甚至误解用户的需求。此外，新的硬件、软件技术的出现也会使用户需求发生变化。因此设计人员必须与用户不断深入地进行交流，才能逐步确定用户的实际需求。

5.2.2 需求分析的方法和过程

需求分析常用的方法有以下几种。

(1) 跟班作业。通过亲身参加业务工作来了解业务活动的情况。这种方法可以比较准确地理解用户的需求，但比较耗费时间。

(2) 开调查会。通过与用户座谈来了解业务活动情况及用户需求。座谈时，参加者之间可以相互启发。

(3) 请专人介绍和询问。对某些调查中的问题，可以找专业人员介绍情况并进行咨询。

(4) 设计调查表请用户填写。如果调查表设计得合理，这种方法是很有效，也很易于为用户接受。

(5) 查阅记录。即查阅与原系统有关的数据记录，包括原始单据、账簿、报表等。

需求分析的过程一般如下。

(1) 分析用户活动，产生业务流程图。了解用户当前的业务活动和职能，理清其处理流程。把用户业务分成若干个子处理过程，使每个处理功能明确、界面清楚，画出业务流程图。

(2) 确定系统范围，产生系统范围图。在和用户经过充分讨论的基础上，确定计算机所能进行数据处理的范围，确定哪些工作由人工完成，哪些工作由计算机系统完成，即确定人机界面。

(3) 分析用户活动所涉及的数据，产生数据流图。深入分析用户的业务处理，以数据流图(Data Flow Diagram，DFD)的形式表示出数据的流向和对数据所进行的加工。DFD 有 4 个基本成分：数据流，加工或处理，文件，外部实体。DFD 可以形象地表示数据流与各业务活动的关系，它是需求分析的工具和分析结果的描述手段。

(4) 分析系统数据，产生数据字典(Data Dictionary，DD)。仅有 DFD 并不能构成需求说明书，DFD 只表示出系统由哪几部分组成和各个部分之间的关系，并没有说明各个成分的含义。数据字典提供对数据库时间描述的集中管理，它的功能是存储和检索各种数据描述(元数据 Metadata)，数据字典是数据收集和数据分析的主要成果，在数据库设计中占有很重要的地位。

(5) 功能分析。数据库的设计是与应用系统的设计紧密结合的过程，离开一定的功能，数据库就失去其存在的价值。数据库设计的一个重要特点是结构(数据)和行为(功能)的结合。用户希望系统能提供的功能必须有一个清晰的描述。功能分析可以采用软件结构图或模块图来表示系统的层次分解关系、模块调用关系。

5.2.3 需求分析常用工具

1. 数据流图

数据流图(Data Flow Diagram，DFD)是结构化分析方法中用于表示系统逻辑模型的一种工具，它以图形的方式描绘数据在系统中流动和处理的过程，由于它只反映系统必须完成的逻辑功能，所以它是一种功能模型。

图 5-2 是一个飞机机票预订系统的数据流图,它反映的功能是:旅行社把预订机票的旅客信息(姓名、年龄、单位、身份证号码、旅行时间、目的地等)输入机票预订系统。系统为旅客安排航班,打印出取票通知单(附有应交的账款)。旅客在飞机起飞的前一天凭取票通知单交款取票,系统检验无误,输出机票给旅客。

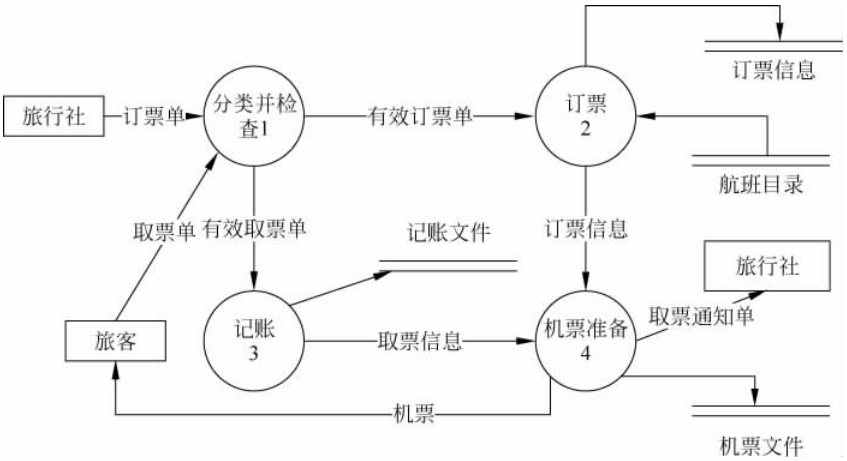

图 5-2 飞机机票预订系统数据流图

DFD 由以下 4 个要素组成。

(1) 数据流 代表数据流的有向线。数据流是数据在系统内传播的路径,因此由一组成分固定的数据组成。如订票单由旅客姓名、年龄、单位、身份证号、日期、目的地等数据项组成。由于数据流是流动中的数据,所以必须有流向,除了与数据存储之间的数据流不用命名外,数据流应该用名词或名词短语命名。

(2) 加工 代表数据处理逻辑,对数据流进行某些操作或变换。每个加工也要有名字,通常是动词短语,简明地描述完成什么加工。在分层的数据流图中,加工还应编号。

(3) 文件 代表数据存储,指暂时保存的数据,它可以是数据库文件或任何形式的数据组织。

(4) 外部实体 代表系统之外的数据提供者或使用者,是本软件系统外部环境中的实体(包括人员、组织或其他软件系统),统称外部实体。一般只出现在数据流图的顶层。

画 DFD 的步骤如下。

(1) 首先画系统的输入输出,即先画顶层数据流图。顶层流图只包含一个加工,用以表示被开发的系统,然后考虑该系统有哪些输入数据、输出数据。顶层图的作用在于表明被开发系统的范围以及它和周围环境的数据交换关系。

(2) 画系统内部结构,即画下层数据流图。不能再分解的加工称为基本加工。一般将层号从 0 开始编号,采用自顶向下,由外向内的原则。画 0 层数据流图时,分解顶层流图的系统为若干子系统,确定每个子系统间的数据接口和活动关系。

画 DFD 时注意事项如下。

(1) 命名。不论数据流、数据存储还是加工,合适的命名使人们易于理解其含义。

(2) 画数据流而不是控制流。数据流反映系统“做什么”,不反映“如何做”,因此箭头上的数据流名称只能是名词或名词短语,整个图中不反映加工的执行顺序。

(3) 一般不画物质流。数据流反映能用计算机处理的数据,并不是实物,因此对目标系统的数据流图一般不画物质流。

(4) 每个加工至少有一个输入数据流和一个输出数据流,反映出此加工数据的来源与加工的结果。

(5) 编号。如果一张数据流图中的某个加工分解成另一张数据流图时,则上层图为父图,直接下层图为子图。子图及其所有的加工都应编号。

(6) 父图与子图的平衡。子图的输入输出数据流同父图相应加工的输入输出数据流必须一致,此即父图与子图的平衡。

(7) 局部数据存储。当某层数据流图中的数据存储不是父图中相应加工的外部接口,而只是本图中某些加工之间的数据接口时,则称这些数据存储为局部数据存储。

(8) 提高数据流图的易懂性。注意合理分解,要把一个加工分解成几个功能相对独立的子加工,这样可以减少加工之间输入、输出数据流的数目,增加数据流图的可理解性。

构造 DFD 的目的是为了系统分析师与用户能够进行明确的交流,以便指导系统的设计,并为后续工作打下基础。所以要求 DFD 既要简单,又要易于理解。构造 DFD 通常从上到下,逐层分解,直到功能细化为止,形成若干层次的 DFD。

2. 数据字典

数据字典(Data Dictionary,DD)是将数据流程图中各个要素的具体内容和特征,以特定格式记录下来,所形成的文档。它主要包括:数据项、数据结构、数据流、加工、文件、外部实体等内容。在数据库设计过程中,数据字典被不断地充实、修改和完善。

对数据库设计来讲,数据字典是进行数据收集和数据分析所获得的主要成果,是各类数据描述的集合。数据字典通常包括数据项、数据结构、数据流、数据存储和处理过程 5 个部分。

1) 数据项

数据项是不可再分的数据单位。对数据项的描述通常包括以下内容。

数据项描述={数据项名,数据项含义说明,别名,数据类型,长度,取值范围,
取值含义,与其他数据项的逻辑关系}

其中,取值范围与其他数据项的逻辑关系定义了数据的完整性约束条件,是设计数据检验功能的依据。

2) 数据结构

数据结构反映了数据之间的组合关系。一个数据结构可以由若干个数据项组成,也可以由若干个数据结构组成,或由若干个数据项和数据结构混合组成。对数据结构的描述通常包括以下内容。

数据结构描述={数据结构名,含义说明,组成:{数据项或数据结构}}

3) 数据流

数据流是数据结构在系统内传输的路径。对数据流的描述通常包括以下内容。

数据流描述={数据流名,说明,数据流来源,数据流去向,
组成:{数据结构},平均流量,高峰期流量}

其中,数据流来源是说明该数据流来自哪个过程,数据流去向是说明该数据流将到哪个过程去,平均流量是指在单位时间(每天、每周、每月等)内的传输次数,高峰期流量则是指在高峰时期的数据流量。

4) 数据存储

数据存储是数据结构停留或保存的地方,也是数据流的来源和去向之一。对数据存储的描述通常包括以下内容。

数据存储描述={数据存储名,说明,编号,流入的数据流,流出的数据流,
组成:{数据结构},数据量,存取方式}

其中,数据量是指每次存取多少数据,每天(或每小时、每周等)存取几次等信息。存取方法包括是批处理,还是联机处理;是检索还是更新;是顺序检索还是随机检索等。另外,流入的数据流要指出其来源,流出的数据流要指出其去向。

5) 处理过程

数据字典中只需要描述处理过程的说明性信息,通常包括以下内容。

处理过程描述={处理过程名,说明,输入:{数据流},输出:{数据流},
处理:{简要说明}}

其中,简要说明是指该处理过程的功能及处理要求。其中,功能是指该处理过程用来做什么(而不是怎么做),处理要求包括处理频度要求,如单位时间里处理多少事务,多少数据量;响应时间要求等。这些处理要求是后面物理设计的输入及性能评价的标准。

数据字典是关于数据库中数据的描述,即元数据,而不是数据本身。数据本身将存放在物理数据库中,由数据库管理系统管理。数据字典有助于这些数据的进一步管理和控制,为设计人员和数据库管理员在数据库设计、实现和运行阶段控制有关数据提供依据。

5.2.4 需求分析实例

本节以建立“学校管理信息系统”为例,说明数据库设计的步骤,各步骤中所做的工作及产生的结果。

首先,在建立“学校管理信息系统”的需求分析阶段,为了明确系统的数据要求、处理要求、安全性和完整性,需要明确“学校”这个应用环境的组织结构和数据流程,即学校由哪些部门组成,各部门的职能,各部门接收的数据,对这些数据进行的处理,处理时有哪些要求,处理后产生的数据,这些数据的接收者是谁等。通过这些内容的调查,可以得到目标系统的系统结构图、数据流图,对于数据的要求得到数据字典等内容,为应用系统在概念结构设计阶段抽象出 E-R 图提供了基础。

某校的行政管理部门按照职能划分,其主要的组织结构图见图 5-3。

各行政管理部门由于职能的不同,管理的对象也不同,例如,研究生院管理研究生信息;人事处负责师资队伍建设及人事管理;学生工作处负责学生事务管理及学生资助;教务处负责学生档案、选课、成绩查询、教师教学管理等。各部门之间会有一定的联系,例如,教务处需要同时处理学生、课程以及教师三种对象的信息,而教师信息则主要由人事处进行管理,教务处只需要教师信息中的部分信息即可;学生工作处也需要从教务处获得学生的相

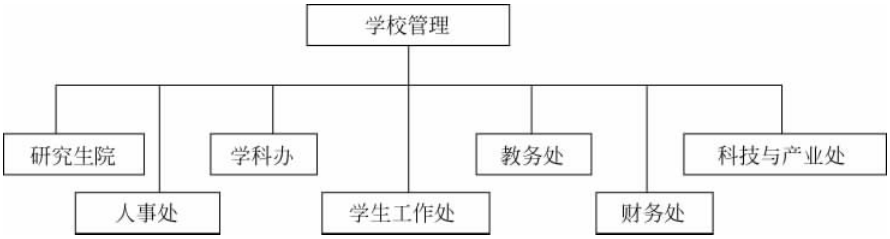

图 5-3　某校的组织结构图

关资料进行学生日常活动、评优及奖学金管理。所以，部门之间存在着数据流动和相互处理的问题，这就需要用数据流图将数据在各部门之间的流动和处理情况表示清楚，对于数据的说明则用数据字典进行表示。

在清楚了部门之间的相互关系之后，就要具体到某个部门进行详细分析了。下面以该校教务处管理系统为例，假设该系统可以分为三个主要功能：学籍管理，负责学生的基本信息采集及学籍档案管理；专业建设，管理学校各院系的专业设置、培养计划、课程管理、教学大纲、教材选用等信息；选课管理，负责所有在校本科生的选课及成绩查询等工作。

图 5-4 得到的只是高层次抽象的系统概貌，要反映更详细的内容，则需要将各功能模块的划分进一步细化，直到将系统的工作过程表述清楚为止。在处理功能逐级分解的同时，它们所使用的数据也逐级分解，形成若干层次的数据流图，数据流图表达了对数据进行处理的过程，如图 5-5～图 5-9 所示。其中，图 5-5 是学籍管理子系统的顶层数据流图，在这一层，只能看到该子系统的输入外部项（招生办、辅导员、教务处）与输出外部项（使用学籍信息的相关部门），对于学籍管理子系统中具体包含哪些功能，在顶层数据流图中是无法看到的。图 5-6 是学籍管理子系统的一级数据流图，可以看出学籍管理子系统总共包括 5 个功能：学籍基本信息管理、学籍变动管理、奖惩信息管理、统计查询及报表打印。数据存储包括：学籍信息表、学籍变动表、奖励信息表和处罚信息表，这些静态的数据存储，都是数据库设计重点考虑的内容。但是，在这一层的数据流图中，仍然无法具体地知道每个功能包含哪些操作。因此在图 5-7～图 5-9 中，对基本信息管理、学籍变动管理和奖惩信息管理三个功能进行进一步细化，得到相应功能的二级数据流图。显然，二级数据流图对各个功能具体实现的操作已经非常明确了。当然，对于图 5-9，仍然可以对奖励信息管理进一步细化，得到三级细化数据流图，但是对于数据库设计来说，二级数据流图已经可以清晰地看到需要存储和管理的数据对象了。而对于数据对象的详细说明，则使用数据字典来描述。

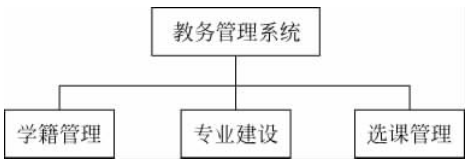

图 5-4　教务管理子系统功能结构图

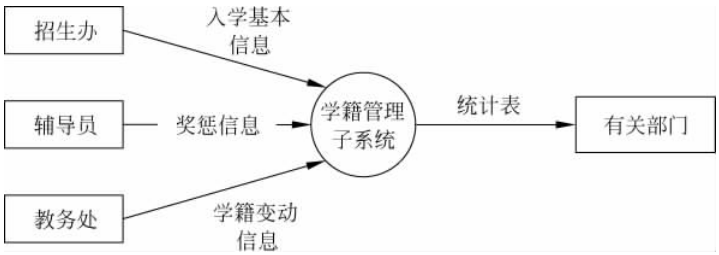

图 5-5　学籍管理子系统顶层数据流图

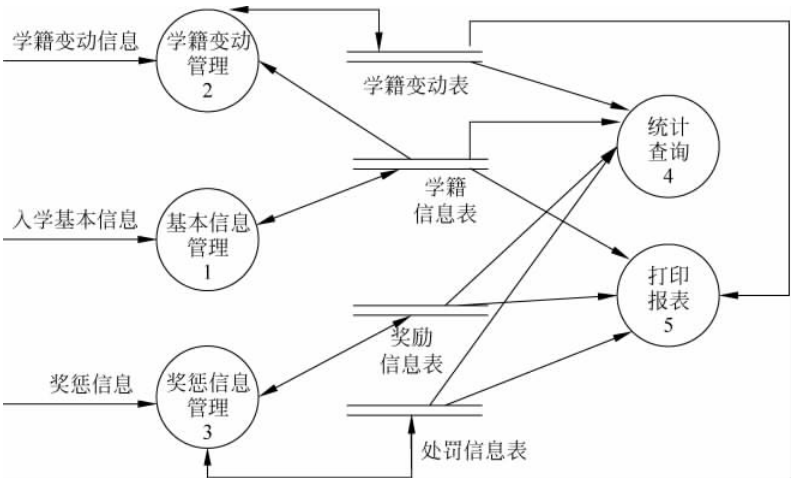

图 5-6　学籍管理子系统一层数据流图

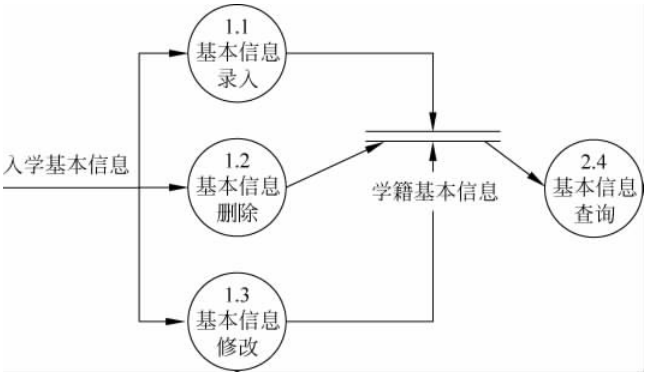

图 5-7　基本信息管理二层数据流图

下面以“学籍管理子系统”为例,简要说明如何定义数据字典。

1. 数据项

该子系统涉及很多数据项,其中,“学号”数据项非常重要,它能够唯一表示一个学生实体,作为学生实体的标识属性,可以如下描述。

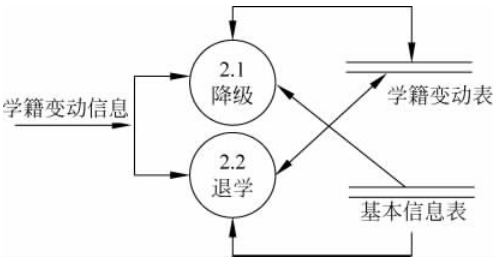

图 5-8　学籍变动管理二层数据流图

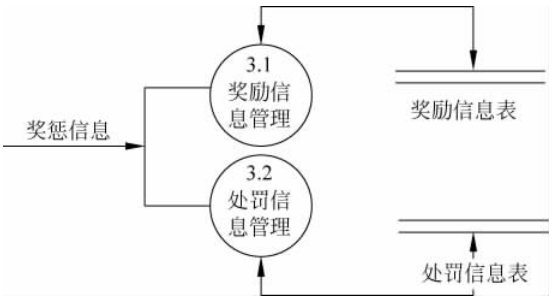

图 5-9　奖励管理二层数据流图

数据项：学号

含义说明：唯一标识某个学生

别名：学生编号

类型：字符型

长度：8

取值范围：00000000～99999999

取值含义：第 1,2 位表示该学生所在学院；第 3,4 位表示学生入学年份；后 4 位按顺序编号

与其他数据项的逻辑关系：能够决定学生的其他属性

2. 数据结构

“学生”是该系统中的重要实体，也是系统的一个核心数据结构，在实际操作中，学生实体实际上就是代表学生的一条条学籍信息，它可以如下描述。

数据结构：学生

含义说明：学籍管理子系统的主体数据结构，定义了一个学生的在校基本信息

组成：学号＋姓名＋性别＋出生日期＋年龄＋学院＋专业＋班级

3. 数据流

数据流是数据结构在系统内传输的路径，体现了系统中对数据动态的处理过程及变化。如“入学基本信息”数据流，来自系统的外部输入项“招生办公室”，流向基本信息管理模块，具体内容描述如下。

数据流：入学基本信息
含义说明：学生入学时采集到的基本信息
来源：招生办公室
去向：基本信息管理模块
组成：身份证号＋姓名＋性别＋出生日期……

4. 数据存储

系统中基本信息管理、学籍变动管理和奖惩信息管理三个功能都会对相应的数据存储进行操作，如基本信息管理模块，会对学籍信息表进行读取，同时会对学籍信息表进行添加、删除、修改等操作。数据存储中的数据，作为数据源，流向统计查询和报表显示模块。具体内容如下。

数据结构：学籍信息表
含义说明：学生在校的学籍基本信息
流入数据流：学生在校基本信息
流出数据流：具体统计和查询信息
组成：学号＋姓名＋性别＋出生日期＋学院＋专业＋班级＋宿舍

5. 处理过程

处理过程表示对数据流的处理功能和处理要求，如处理过程“1.1 基本信息录入”可如下描述。

处理过程：1.1 基本信息录入
说明：为所有新生录入在校学籍信息
输入：入学基本信息
输出：在校学籍信息
处理：在新生报到后，根据录取专业，为所有新生录入宿舍、院系、专业、班级等在校信息。

5.3 概念结构设计

5.3.1 概念结构设计的定义

概念模型可以看成是现实世界到机器世界的一个过渡的中间层次，在这个层次中，使用接近计算机存储的方式表示数据，同时又不涉及具体的 DBMS。做出概念模型后，再转换为具体的 DBMS(如 SQL Server 或 Oracle)下的模型，就成为逻辑模型。在设计数据库应用系统时，要把现实世界的事物通过认识和抽象转换为信息世界的概念模型，再把概念模型转换为机器世界的数据模型。

概念模型是对现实世界的一种抽象，即对实际的人、物、事和概念进行人为处理，抽取人们关心的共同特性，忽略非本质的细节，并把这些特性用各种概念精确地加以描述。

概念模型是数据库系统的核心和基础。由于各个机器上实现的 DBMS 软件都是基于某种数据模型的，所以在具体机器上实现的模型都有许多严格的限制。而现实应用环境是复杂多变的，如果把现实世界中的事物直接转换为机器中的对象，就非常不方便。因此，人们研究把现实世界中的事物抽象为不依赖于具体机器的信息结构，又接近人们的思维，并具有丰富语义的概念模型，然后再把概念模型转换为具体的机器上 DBMS 支持的数据模型。概念模型的描述工具通常是使用 E-R 模型图。该模型不依赖于具体的硬件环境和 DBMS。

将需求分析得到的用户需求抽象为信息结构即概念模型的过程就是概念结构设计。

对数据库概念模型有以下要求。

(1) 有丰富的语义表达能力，能表达用户的各种需求。

(2) 易于交流和理解，从而可以用它和不熟悉计算机的用户交换意见。

(3) 要易于更改。当应用环境和应用要求改变时，概念模型要能很容易地修改和扩充以反映这种变化。

(4) 易于向各种数据模型转换。

5.3.2 概念结构设计方法

概念结构设计阶段，一般使用语义数据模型描述概念模型。通常是使用 E-R 模型图作为概念设计的描述工具进行设计。用 E-R 模型图进行概念设计可以采用如下方法。

1. 集中式模式设计法

集中式模式设计法即首先定义全局概念结构的框架，然后逐步细化。例如，可以先确定几个高级实体类型，然后在确定其属性时，把这些实体类型分解为更低一层的实体类型和联系。集中式模式设计法的设计过程如图 5-10 所示。

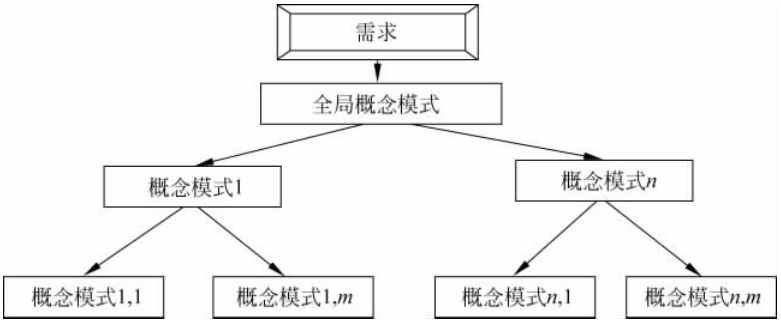

图 5-10　集中式模式设计法

2. 视图集成法

以各部分的需求说明为基础，分别设计各自的局部模式，这些局部模式相当于各部分的视图，然后再以这些视图为基础，集成为一个全部模式。视图是按照某个用户组、应用或部门的需求说明，用 E-R 数据模型设计的局部模式。现在的关系数据库设计通常采用视图集成法。视图集成法的设计过程如图 5-11 所示。

3. 混合方法

即将集中式模式设计法和视图集成法相结合，用集中式模式设计法设计一个全局概念结构的框架，以它为骨架集成由视图集成法中设计的各局部概念结构。

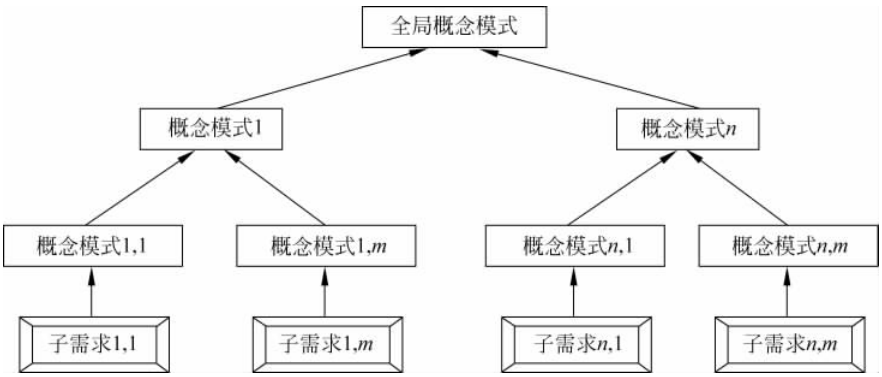

图 5-11　视图集成法

4. 由内向外法

首先定义最重要的核心概念结构,然后向外扩充,考虑已存在概念附近的新概念使得建模过程向外扩展。使用该策略,可以先确定模式中比较明显的一些实体类型,然后继续添加其他相关的实体类型,如图 5-12 所示。

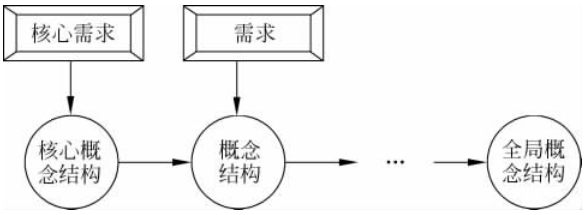

图 5-12　由内向外法

5.3.3　局部视图设计

1. 局部视图设计的方法

1) 选择局部应用

在需求分析阶段,通过对应用环境和要求进行详尽的调查分析,用多层数据流图和数据字典描述了整个系统。设计分 E-R 图的第一步,就是要根据系统的具体情况,在多层的数据流图中选择一个适当层次的(经验很重要)数据流图,使得图中每一部分对应一个局部应用,可以就这一层次的数据流图为出发点,设计分 E-R 图。一般而言,中层的数据流图能较好地反映系统中各局部应用的子系统组成,因此人们往往以中层数据流图作为设计分 E-R 图的依据。

2) 逐一设计分 E-R 图

每个局部应用都对应了一组数据流图,局部应用涉及的数据都已经收集在数据字典中了。现在就是要将这些数据从数据字典中抽取出来,参照数据流图,标定局部应用中的实体、实体的属性、标识实体的码,确定实体之间的联系及其类型(1∶1、1∶n、m∶n)。

(1) 标定局部应用中的实体。

现实世界中一组具有某些共同特性和行为的对象就可以抽象为一个实体。对象和实体

之间是“is member of”的关系。

例如，在学校环境中，可以把张三、李四、王五等对象抽象为学生实体。对象类型的组成成分可以抽象为实体的属性。

组成成分与对象类型之间是“is part of”的关系。例如，学号、姓名、专业、年级等可以抽象为学生实体的属性。其中，学号为标识学生实体的码。

(2) 标定实体的属性、标识实体的码。

实际上，实体与属性是相对而言的，很难有截然划分的界限。同一事物，在一种应用环境中作为“属性”，在另一种应用环境中就必须作为“实体”。一般说来，在给定的应用环境中，属性不能再具有需要描述的性质，即属性必须是不可分的数据项。属性不能与其他实体具有联系，联系只发生在实体之间。

(3) 确定实体之间的联系及其类型(1∶1、1∶n、m∶n)。

根据需求分析，要考察实体之间是否存在联系，有无多余联系。

2. 应用举例

继续以“教务管理系统”中的“学籍管理子系统”为例，讨论该系统概念结构设计的过程和方法。在该例中，由学籍管理的数据流图得到的局部应用中，主要涉及的实体包括：学生、学院、班级、宿舍、专业、奖励信息、处罚信息等。那么，这些实体之间的联系又是怎样的呢？

由于一个宿舍可以住多个学生，而一个学生只能住在某一个宿舍中，一个班级往往有若干名学生，而一个学生只能属于一个班级，一个学院包含多名学生，开设了多个专业方向，每个专业方向有多个班级，而每个班级又包括多名学生，因此各实体之间的关系见图 5-13。

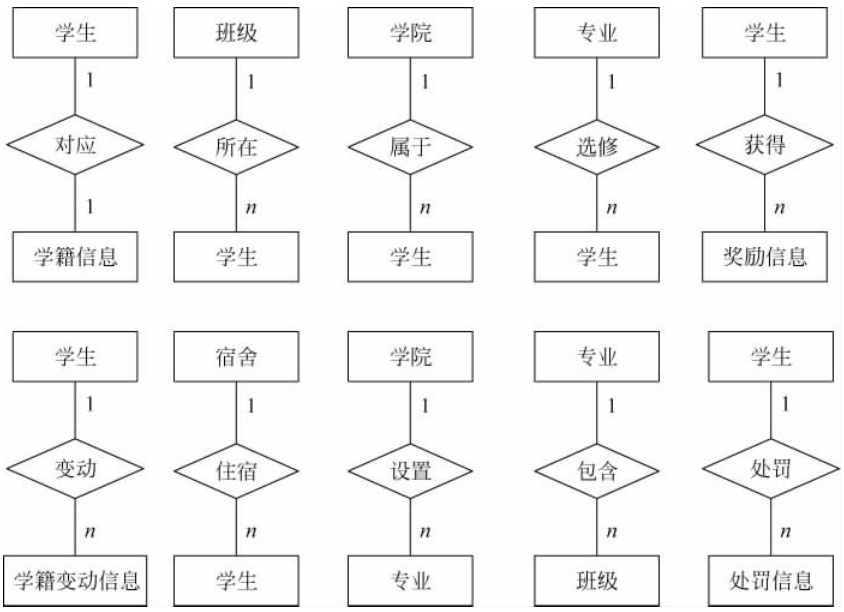

图 5-13　学籍管理局部应用的分 E-R 图

这些实体的属性分别如下(此处省略了 E-R 图对实体属性的表示)。

学生：{学号，姓名，性别，出生日期，年龄，学院，专业，班级，宿舍}

学籍信息:{学号,姓名,性别,出生日期,年龄,学院,专业,班级,宿舍}

学籍变动信息:{学籍变动编号,学号,姓名,变动类型,变动时间}

班级:{班级号,学生人数,班主任}

宿舍:{宿舍编号,地址,人数}

学院:{学院编号,学院名称,院长,办公室电话}

专业:{专业编号,专业名称}

奖励信息:{编号,学号,奖励名称,获奖时间,获奖等级,颁奖单位}

处罚信息:{编号,学号,处罚名称,处罚时间,处罚原因}

此处,学生和学籍信息为相同的内容,合并为一个实体,命名为学生。因此,学籍管理子系统中,共计 8 个实体,9 个联系。

5.3.4 集成全局视图

1. 视图集成要解决的问题

由于局部概念设计相对简单,因此简化了全局模式的设计。但是,在将局部视图合成为全局视图的时候,需要具体地解决下列问题。

1) 确定模式之间的对应和冲突

由于各子模式是分开进行设计的,因此有必要在集成之前确定各模式表示的是否是同一个现实世界的概念结构。在此过程中,模式间可能会发生如下一些冲突。

(1) 属性冲突

① 属性域冲突,即属性值的类型、取值范围或取值集合不同,如零件号,有的部门作为整数对待,有的部门则使用字符串。不同部门对零件号的编码也可能不同。

② 属性取值单位冲突,如零件重量,有的部门以千克为单位,有的部门以克为单位。

(2) 命名冲突

包括同名异义和异名同义。如科研项目,财务科称为项目,科研处称为课题,生产管理处称为工程,这就是一个异名同义的例子。

(3) 结构冲突

① 同一对象在不同应用中具有不同的抽象。如在教学管理中,职称是一个属性;而在人事管理中,因为职称与工资、住房挂钩,因此是一个实体。

② 同一实体在不同局部视图中所包含的属性不完全相同。

③ 实体间的联系在不同分 E-R 图中为不同类型,如在生产子系统分 E-R 图中,产品和零件构成 1∶n 联系。而物资子系统分 E-R 图中,产品、零件、供应商三者构成多对多联系。

2) 修改视图使得相互一致

对一些模式进行修改,以便于其他模式相符合。这一步可以解决上一步发现的冲突。

3) 合并视图

通过创建单个子视图来创建全局视图。相应的概念在全局模式中只出现一次,并且要确定子视图和全局视图之间的映射关系。在涉及数百个实体和联系的现实数据库中,这一步是最为困难的。因为牵扯到大量的人为干预和协商来解决冲突,并且要确定全局模式的一个最为合理并且能够接受的解决方案。

4）重构

该步骤是一个可选步骤，可能会对全局模式进行分析和重构，以删除任何冗余和不必要的内容。

2. 视图集成的策略

子视图的集成是一个非常复杂的过程，需要一个更加严格和系统化的方法。下面介绍一些用于视图合并的策略。

1）二元集成

首先对两个比较类似的模式进行集成。然后把结果模式和另外一个模式集成，不断重复该过程直到所有模式被集成。可以根据模式的相似程度确定模式集成的顺序。由于集成是逐步进行的，所以该策略适用于手工集成。

2）n 元集成

对视图的集成关系进行分析和说明之后，在一个过程中完成所有视图的集成。对于规模较大的设计问题，这个策略需要使用计算机化的工具，目前有一些这种工具的原型，但还没有成熟的商业产品。

3）二元平衡策略

首先将模式成对地进行集成，然后再将结果模式成对地进一步集成，不断重复该过程直至得到最终的全局模式。

4）混合策略

首先，根据模式的相似性把它们划分为不同的组，对每个组单独地进行集成。然后对中间结果进行分组并集成，重复该过程直至集成结束。

3. 应用举例

将学籍管理子系统的各分 E-R 图进行合并的主要步骤如下。

（1）第一步：确定各模式之间的对应和冲突。

学籍管理子系统各模式之间主要存在命名冲突和结构冲突：学生实体的属性{学院，专业，班级，宿舍}分别对应“学院”实体的{学院编号}、“专业”实体的{专业编号}、“班级”实体的{班级编号}和“宿舍”实体的{宿舍编号}。因此，消除冲突后学生实体表示为：

学生：{学号，姓名，性别，出生日期，年龄，学院编号，专业编号，班级编号，宿舍编号}

（2）第二步：消除存在着的冗余属性和冗余联系。

学生实体中的年龄属性可以由出生日期推算出来，属于冗余属性，应该去掉。这样不仅可以节省存储空间，而且当某个学生的出生日期有误，进行修改后，无须相应修改年龄，减少了产生数据不一致的机会。因此，消除冗余属性后学生实体表示为：

学生：{学号，姓名，性别，出生日期，学院编号，专业编号，班级编号，宿舍编号}

第三步：视图集成。集成后的学籍管理模块的 E-R 图如图 5-14 所示。

学籍管理子系统的 E-R 图还需要进一步和专业建设子系统（如图 5-15 所示）以及选课管理子系统（如图 5-16 所示）的 E-R 图合并，生成教务管理系统的 E-R 图，由于篇幅限制，这里就不再给出系统的完整 E-R 图了，待读者自己完成。

视图集成后形成一个整体的数据库概念结构，对该整体概念结构还必须进行进一步验

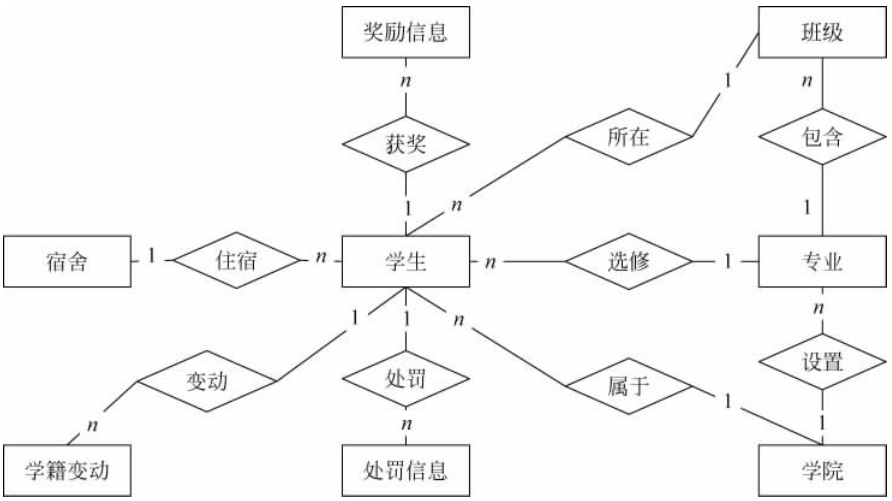

图 5-14　学籍管理模块 E-R 图

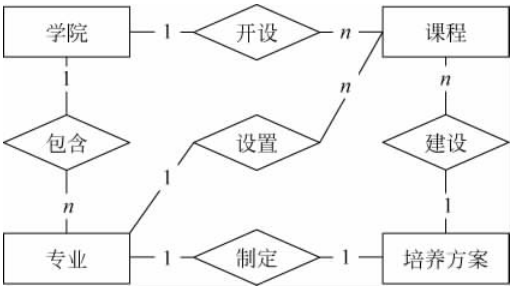

图 5-15　专业建设模块 E-R 图

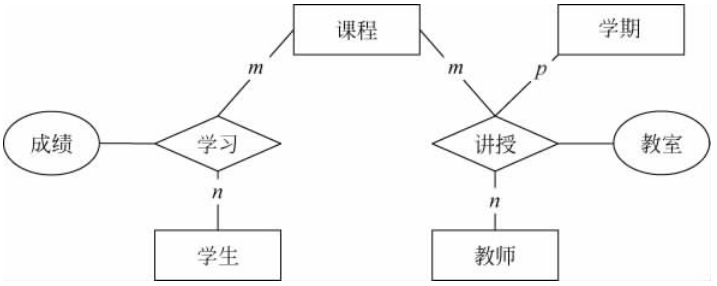

图 5-16　选课管理模块 E-R 图

证,确保它能够满足下列条件。

(1) 整体概念结构内部必须具有一致性,即不能存在互相矛盾的表达。

(2) 整体概念结构能准确地反映原来的每个视图结构,包括属性、实体及实体间的联系。

(3) 整体概念结构能满足需求分析阶段所确定的所有要求。

5.4 逻辑结构设计

5.4.1 逻辑结构设计的任务和步骤

概念结构设计阶段得到的 E-R 模型是用户的模型，它独立于任何一种数据模型，独立于任何一个具体的 DBMS。数据库逻辑结构设计的任务是将概念结构转换成特定 DBMS 所支持的数据模型的过程。从此开始便进入了“实现设计”阶段，需要考虑到具体的 DBMS 的性能、具体的数据模型特点。

从 E-R 图所表示的概念模型可以转换成任何一种具体的 DBMS 所支持的数据模型，如网状模型、层次模型和关系模型。逻辑结构设计应该选择最适于描述与表达相应概念结构的数据模型，然后选择最合适的 DBMS。

逻辑结构设计阶段需要完成的任务和步骤如下。

(1) 将 E-R 模型转换为等价的关系模式。

(2) 按需要对关系模式进行规范化。

(3) 对规范化后的模式进行评价。

(4) 根据局部应用的需要，设计用户外模式。

要使计算机能够处理 E-R 模型中的信息。首先必须将它转换为具体的 DBMS 能处理的数据模型。E-R 模型可以向现有的各种数据模型转换。而目前市场上 DBMS 大部分是基于关系数据模型的，所以本书只详细讲解 E-R 模型向关系数据模型的转换方法。

从 E-R 图中可以看出，E-R 模型实际上是实体型及实体间联系所组成的有机整体，而前面也学过，关系模型的逻辑结构是一系列关系模式的集合。所以将 E-R 模型转换为关系模型，实质上就是将实体型和联系转换为关系模式。也就是如何用关系模式来表达实体型以及实体集之间的联系的问题。

5.4.2 E-R 图向关系模型的转换原则

关系模型的逻辑结构是一组关系模式的集合。而 E-R 图则是由实体、实体的属性和实体之间的联系三个要素组成的。所以将 E-R 图转换为关系模型实际上就是要将实体、实体的属性和实体之间的联系转换为关系模式。

E-R 图向关系模型的转换一般应遵循如下原则。

1. 实体的转换

一个实体型转换为一个关系模式。实体的属性就是关系的属性，实体的码就是关系的码。

例如，“学生”实体可以转换为如下关系模式，其中，学号为学生关系的码。

学生：(学号，姓名，性别，出生日期，学院编号，专业编号，班级编号，宿舍编号)

同样，学院、专业、班级、宿舍、学籍变动信息、奖励信息和惩罚信息等几个实体都分别转换为一个关系模式。

2. 联系的转换

一个联系转换为一个关系模式，与该联系相连的各实体的码以及联系的属性转换为关

系的属性,该关系的码则有以下几种情况。

(1) 若联系为 1∶1,则每个实体的码均是该关系的候选码。

(2) 若联系为 1∶n,则关系的码为 n 端实体的码。

(3) 若联系为 m∶n,则关系的码为诸实体码的组合。

(4) 三个或三个以上实体间的多元联系、同一实体集内的自反联系的转换规则与二元联系相同。

下面以二元联系为例,对联系的转换规则进行详细的解释。

1) 联系为 1∶1

(1) 转换为独立的关系模式。如果转换为一个独立的关系模式,则与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性,每个实体的码均是该关系的候选码。

例如,图 5-15 中专业和培养方案之间的联系“制定”为 1∶1 类型,则该联系转换为独立的关系模式时,形式为:

制定(专业编号,培养方案编号)

该联系的属性,包含专业和培养方案两个关系的码及联系的属性(此处没有联系的属性)。该联系的码,为任意一方关系的码,即专业编号或者培养方案编号。此处,选择专业编号为主码。

(2) 与关系模式合并。如果与某一端对应的关系模式合并,则需要在该关系模式的属性中加入另一个关系模式的码和联系本身的属性。

例如,图 5-15 中专业和培养方案之间的联系“制定”,可以与专业或者培养方案两个关系模式合并,合并结果为下面两种情况之一。

专业(专业编号,专业名称,所属学院,培养方案编号)

培养方案(培养方案编号,内容,制定日期,负责人,专业编号)

在上面的关系模式中,用下画线表示主码,波浪线表示外码,该外码即为并入的另一方关系模式的主码。后续的章节中,如不做特殊说明,主外码均用这种方式表示。

2) 联系为 1∶n

(1) 转换为独立的关系模式。如果转换为一个独立的关系模式,则与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性,而关系的码为 n 端实体的码。

例如,图 5-14 中专业和学生之间的联系“选修”为 1∶n 类型,则该联系转换为独立的关系模式时,形式为:

选修(学生编号,专业编号)

该联系的属性,包含专业和学生两个关系的码及联系的属性(此处没有联系的属性),该联系的码,为 n 方关系的码,即学生编号。

(2) 与关系模式合并。如果与 n 端对应的关系模式合并,则在 n 端实体对应模式中加入 1 端实体所对应关系模式的码,以及联系本身的属性。而关系的码为 n 端实体的码。

例如,图 5-14 中专业和学生之间的联系“选修”,只能与表示 n 方的学生关系模式合并,形式为:

学生:(学号,姓名,性别,出生日期,学院编号,专业编号,班级编号,宿舍编号)

3) 联系为 m∶n

m∶n 联系的转换方法只有一种,就是转换为一个独立的模式。与该联系相连的各实

体的码以及联系本身的属性均转换为关系的属性。而关系的码为各实体码的组合。

例如，图 5-16 中学生和课程之间的“选课”联系是一个 $m:n$ 联系，可以将它转换为如下关系模式，其中，学号与课程号为关系的组合码，成绩则是联系的属性。

学习(学号，课程号，成绩，学期)

3. 具有相同码的关系模式可合并

为了减少系统中的关系个数，如果两个关系模式具有相同的主码，可以考虑将它们合并为一个关系模式。合并方法是将其中一个关系模式的全部属性加入到另一个关系模式中，然后去掉其中的同义属性(可能同名也可能不同名)，并适当地调整属性的次序。

按照上述转换方法，学籍管理模块中的 9 个联系，若 1∶1 联系和 1∶n 联系按照转换为独立关系模式的方法，可以得到下列 9 个关系模式，最终逻辑结构转换的结果是 9 个联系的关系模式和 8 个实体的关系模式的集合，共计 17 个关系模式。

住宿(学生编号，宿舍编号)

变动(学籍变动编号，学生编号)

获奖(奖励编号，学生编号)

处罚(处罚编号，学生编号)

所在(学生编号，班级编号)

选修(专业编号，学生编号)

属于(学生编号，学院编号)

包含(班级编号，专业编号)

设置(专业编号，学院编号)

若按照合并的方式，学籍管理模块中的 9 个联系转换得到如下模式。

学生：(学号，姓名，性别，出生日期，学院编号，专业编号，班级编号，宿舍编号)

学籍变动信息：(学籍变动编号，学号，姓名，变动类型，变动时间)

班级：(班级号，班级名称，学生人数，专业编号)

专业：(专业编号，专业名称，学院编号)

奖励信息：(奖励编号，学号，奖励名称，获奖时间，获奖等级，颁奖单位)

处罚信息：(处罚编号，学号，处罚名称，处罚时间，处罚原因)

由于合并后的关系模式包含部分实体模式，因此最终逻辑结构转换的结果，还应添加余下的两个实体关系模式，因此这种转换方法共计得到 8 个关系模式。

宿舍：{宿舍编号，地址，人数}

学院：{学院编号，学院名称，院长，办公室电话}

从上面两种转换方法的结果可以看出，独立关系模式转换方法操作简单，结果清晰，但是会增加系统的存储空间，且产生较多的连接运算，降低系统查询效率；合并转换方法得到的关系模式数量较少，查询效率高，但是转换过程容易发生遗漏。

5.4.3 逻辑结构的优化

应用规范化理论对逻辑设计阶段产生的逻辑模式进行初步优化，以减少乃至消除关系模式中存在的各种异常，改善完整性、一致性和存储效率。规范化理论是数据库逻辑设计的指南和工具，规范化过程分为两个步骤：确定范式的级别和实施规范化处理(模式

分解)。

1. 确定范式级别

考察关系模式的函数依赖关系,确定范式等级。找出所有"数据字典"中得到的数据之间的依赖关系,对各模式之间的数据依赖进行极小化处理,消除冗余的联系。按照数据依赖理论对关系模式逐一进行分析,考察是否存在部分函数依赖、传递函数依赖和多值依赖等,确定各关系模式属于第几范式。

例如,在学籍变动信息模式中,学号和姓名之间存在函数依赖:学号→姓名,因此该关系模式存在传递依赖关系,属于 2NF。

学籍变动信息:(学籍变动编号,学号,姓名,变动类型,变动时间)

2. 实施规范化处理

确定范式级别后,根据应用需求,判断它们对于这样的应用环境是否合适,确定对于这些模式是否进行合并或分解。对于上例中的学籍变动信息中存在的传递依赖关系,若学生姓名在此关系中经常需要被查询,则可以不对其进行优化,保留在原关系模式中;若学生姓名较少成为查询目标,则可以对将该关系模式进行优化,结果为:

学籍变动信息:(学籍变动编号,学号,变动类型,变动时间)

5.4.4 设计用户外模式

前面根据用户需求设计了局部应用视图,这种局部应用视图只是概念模型,用 E-R 图表示。在将概念模型转换为逻辑模型后,即生成了整个应用系统的模式后,还应该根据局部应用需求,结合具体 DBMS 的特点,设计用户的外模式。

目前关系数据库管理系统一般都提供了视图概念,支持用户的虚拟视图。可以利用这一功能设计更符合局部用户需要的用户外模式。

定义数据库模式主要是从系统的时间效率、空间效率、易维护等角度出发。由于用户外模式与模式是独立的,因此在定义用户外模式时应该更注重考虑用户的习惯与方便。包括:

(1) 使用更符合用户习惯的别名。

(2) 对不同级别的用户定义不同的外模式,以满足系统对安全性的要求。

(3) 简化用户对系统的使用。

5.5 物理结构设计

数据库最终要存储在物理设备上。对于给定的逻辑数据模型,选取一个最适合应用环境的物理结构的过程,称为数据库物理设计。物理设计的任务是为了有效地实现逻辑模式,确定所采取的存储策略。此阶段是以逻辑设计的结果作为输入,结合具体 DBMS 的特点与存储设备特性进行设计,选定数据库在物理设备上的存储结构和存取方法。

数据库的物理设计可分为以下两步。

(1) 确定物理结构,在关系数据库中主要指存取方法和存储结构。

(2) 对物理结构进行评价,评价的重点是时间和空间效率。

5.5.1 确定数据库的物理结构

1. 确定数据库的存储结构

确定数据库存储结构时要综合考虑存取时间、存储空间利用率和维护代价三方面的因素。这三个方面常常是相互矛盾的，例如，消除一切冗余数据虽然能够节约存储空间，但往往会导致检索代价的增加，因此必须进行权衡，选择一个折中方案。

在物理结构中，数据的基本存取单位是存储记录。有了逻辑记录结构以后，就可以设计存储记录结构，一个存储记录可以和一个或多个逻辑记录相对应。存储记录结构包括记录的组成、数据项的类型和长度，以及逻辑记录到存储记录的映射。某一类型的所有存储记录的集合称为“文件”，文件的存储记录可以是定长的，也可以是变长的。

文件组织或文件结构是组成文件的存储记录的表示法。文件结构应该表示文件格式、逻辑次序、物理次序、访问路径、物理设备的分配。物理数据库就是指数据库中实际存储记录的格式、逻辑次序和物理次序、访问路径、物理设备的分配。

决定存储结构的主要因素包括存取时间、存储空间和维护代价三个方面。设计时应当根据实际情况对这三个方面进行综合权衡。一般 DBMS 也提供一定的灵活性可供选择，包括聚簇和索引。

1）聚簇

聚簇就是为了提高查询速度，把在一个（或一组）属性上具有相同值的元组集中地存放在一个物理块中。如果存放不下，可以存放在相邻的物理块中。其中，这个（或这组）属性称为聚簇码。聚簇有以下两个作用。

（1）使用聚簇以后，聚簇码相同的元组集中在一起了，因而聚簇值不必在每个元组中重复存储，只要在一组中存储一次即可，因此可以节省存储空间。

（2）聚簇功能可以大大提高按聚簇码进行查询的效率。例如，假设要查询学生关系中计算机系的学生名单，设计算机系有 300 名学生。在极端情况下，这些学生的记录会分布在 300 个不同的物理块中，这时如果要查询计算机系的学生，就需要做 300 次的 I/O 操作，这将影响系统查询的性能。如果按照系别建立聚簇，使同一个系的学生记录集中存放，则每做一次 I/O 操作，就可以获得多个满足查询条件的记录，从而显著地减少了访问磁盘的次数。

2）索引

存储记录是属性值的集合，主键可以唯一确定一个记录，而其他属性的一个具体值不能唯一确定是哪个记录。在主键上应该建立唯一索引，这样不但可以提高查询速度，还能避免关系键重复值的录入，确保了数据的完整性。

在数据库中，用户访问的最小单位是属性。如果对某些非主属性的检索很频繁，可以考虑建立这些属性的索引文件。索引文件对存储记录重新进行内部链接，从逻辑上改变了记录的存储位置，从而改变了访问数据的入口点。关系中数据越多索引的优越性也就越明显。

建立多个索引文件可以缩短存取时间，但是增加了索引文件所占用的存储空间以及维护的开销。因此，应该根据实际需要综合考虑。

2. 确定访问方法

访问方法是为存储在物理设备（通常指辅存）上的数据提供存储和检索能力的方法。一个访问方法包括存储结构和检索结构两个部分。存储结构限定了可能访问的路径和存储记

录；检索结构定义了每个应用的访问路径，但不涉及存储结构的设计和设备分配。

存储记录是属性的集合，属性是数据项类型，可用作主码或辅助键。主码唯一地确定了一个记录。辅助键是用作记录索引的属性，可能并不唯一确定某一个记录。

访问路径的设计分成主访问路径与辅访问路径的设计。主访问路径与初始记录的装入有关，通常是用主码来检索的。首先利用这种方法设计各个文件，使其能最有效地处理主要的应用。一个物理数据库很可能有几套主访问路径。辅访问路径是通过辅助键的索引对存储记录重新进行内部链接，从而改变访问数据的入口点。用辅助索引可以缩短访问时间，但增加了辅存空间和索引维护的开销。设计者应根据具体情况做出权衡。

3. 确定数据的存放位置

为了提高系统性能，应该根据应用情况将数据的易变部分、稳定部分、经常存取部分和存取频率较低部分分开存放。

例如，目前许多计算机都有多个磁盘，因此可以将表和索引分别存放在不同的磁盘上，在查询时，由于两个磁盘驱动器并行工作，可以提高物理读写的速度。

在多用户环境下，可能将日志文件和数据库对象(表、索引等)放在不同的磁盘上，以加快存取速度。另外，数据库的数据备份、日志文件备份等，只在数据库发生故障进行恢复时才使用，而且数据量很大，可以存放在磁带上，以改进整个系统的性能。

4. 确定系统配置

DBMS 产品一般都提供了一些系统配置变量、存储分配参数，供设计人员和 DBA 对数据库进行物理优化。系统为这些变量设定了初始值，但是这些值不一定适合每一种应用环境，在物理设计阶段，要根据实际情况重新对这些变量赋值，以满足新的要求。

系统配置变量和参数很多，例如，同时使用数据库的用户数、同时打开的数据库对象数、内存分配参数、缓冲区分配参数(使用的缓冲区长度、个数)、存储分配参数、数据库的大小、时间片的大小、锁的数目等，这些参数值影响存取时间和存储空间的分配，在物理设计时要根据应用环境确定这些参数值，可以存放在辅助存储设备上，以改进整个系统的性能。

5.5.2 评价物理结构

数据库物理设计过程中需要对时间效率、空间效率、维护代价和各种用户要求进行权衡，其结果可以产生多种方案，数据库设计人员必须对这些方案进行细致的评价，从中选择一个较优的方案作为数据库的物理结构。

评价物理数据库的方法完全依赖于所选用的 DBMS，主要是从定量估算各种方案的存储空间、存取时间和维护代价入手，对估算结果进行权衡、比较，选择出一个较优的合理的物理结构。如果该结构不符合用户需求，则需要修改设计。

5.6 数据库实施

数据库实施是指根据逻辑设计和物理设计的结果，在计算机上建立起实际的数据库结构、装入数据、进行测试和试运行的过程。

数据库实施主要包括以下工作。

1. 建立实际数据库结构

确定了数据库的逻辑结构与物理结构后，就可以用所选用的 DBMS 提供的数据定义语言(DDL)来严格描述数据库结构。可使用 SQL 定义语句中的 CREATE TABLE 语句定义所需的基本表，使用 CREATE VIEW 语句定义视图。

2. 组织数据入库

装入数据又称为数据库加载(Loading)，是数据库实施阶段的主要工作。在数据库结构建立好之后，就可以向数据库中加载数据了。

由于数据库的数据量一般都很大，它们分散于一个企业(或组织)中各个部门的数据文件、报表或多种形式的单据中，存在着大量的重复，并且其格式和结构一般都不符合数据库的要求，必须把这些数据收集起来加以整理，去掉冗余并转换成数据库所规定的格式，这样处理之后才能装入数据库。因此，需要耗费大量的人力、物力，是一种非常单调乏味而又意义重大的工作。

由于应用环境和数据来源的差异，所以不可能存在普遍通用的转换规则，现有的 DBMS 并不提供通用的数据转换软件来完成这一工作。

对于一般的小型系统，装入数据量较少，可以采用人工方法来完成。其步骤如下。

1) 筛选数据

需要装入数据库中的数据通常都分散在各个部门的数据文件或原始凭证中，所以首先必须把需要入库的数据筛选出来。

2) 转换数据格式

筛选出来的需要入库的数据，其格式往往不符合数据库要求，还需要进行转换。这种转换有时可能很复杂。

3) 输入数据

将转换好的数据输入计算机中。

4) 校验数据

检查输入的数据是否有误。

但是，人工方法不仅效率低，而且容易产生差错。对于数据量较大的系统，应该由计算机来完成这一工作。通常是设计一个数据输入子系统，其主要功能是从大量的原始数据文件中筛选、分类、综合和转换数据库所需的数据，把它们加工成数据库所要求的结构形式，最后装入数据库中，同时还要采用多种检验技术检查输入数据的正确性。

为了保证装入数据库中数据的正确无误，必须高度重视数据的校验工作。在输入子系统的设计中应该考虑多种数据检验技术，在数据转换过程中应使用不同的方法进行多次检验，确认正确后方可入库。

如果在数据库设计时，原来的数据库系统仍在使用，则数据的转换工作是将原来老系统中的数据转换成新系统中的数据结构。同时还要转换原来的应用程序，使之能在新系统下有效地运行。

数据的转换、分类和综合常常需要多次才能完成，因而输入子系统的设计和实施是很复杂的，需要编写许多应用程序，由于这一工作需要耗费较多的时间，为了保证数据能够及时入库，应该在数据库物理设计的同时编制数据输入子系统，而不能等物理设计完成后才开始。

3. 编制与调试应用程序

数据库应用程序的设计属于一般的程序设计范畴,但数据库应用程序有自己的一些特点。例如,大量使用屏幕显示控制语句,形式多样的输出报表,重视数据的有效性和完整性检查,有灵活的交互功能。

数据库应用程序如果要将数据库中的数据及各种查询结果显示在应用程序中,首先需要建立应用程序与数据库之间的连接,主要的数据库连接方式有以下几种。

1) ODBC 数据库接口

ODBC 即开放式数据库互连(Open Database Connectivity),是微软公司推出的一种实现应用程序和关系数据库之间通信的接口标准。符合标准的数据库就可以通过 SQL 编写的命令对数据库进行操作,但只针对关系数据库。目前所有的关系数据库都符合该标准(如 SQL Server,Oracle,Access,Excel 等)。ODBC 本质上是一组数据库访问 API(应用程序编程接口),由一组函数调用组成,核心是 SQL 语句,其结构图如图 5-17 所示。

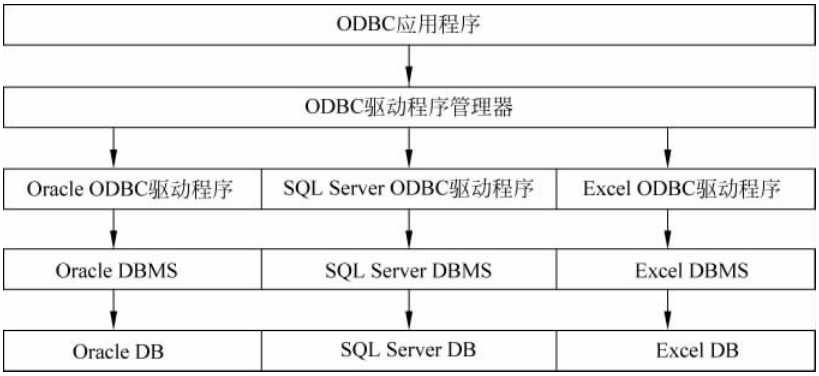

图 5-17　ODBC 访问数据库接口

2) OLE DB 数据库接口

OLE DB 即数据库链接和嵌入对象(Object Linking and Embedding DataBase)。OLE DB 是微软提出的基于 COM 思想且面向对象的一种技术标准,目的是提供一种统一的数据访问接口访问各种数据源。这里所说的"数据"除了标准的关系型数据库中的数据之外,还包括邮件数据、Web 上的文本或图形、目录服务(Directory Services),以及主机系统中的文件和地理数据以及自定义业务对象等。OLE DB 标准的核心内容就是提供一种相同的访问接口,使得数据的使用者(应用程序)可以使用同样的方法访问各种数据,而不用考虑数据的具体存储地点、格式或类型,其结构图如图 5-18 所示。

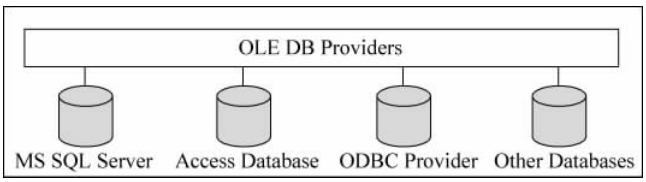

图 5-18　OLE DB 访问数据库接口

3）ADO 数据库接口

ADO(ActiveX Data Objects)是微软公司开发的基于 COM 的数据库应用程序接口，通过 ADO 连接数据库，可以灵活地操作数据库中的数据。

图 5-19 展示了应用程序通过 ADO 访问 SQL Server 数据库接口。从图中可看出，使用 ADO 访问 SQL Server 数据库有两种途径：一种是通过 ODBC 驱动程序，另一种是通过 SQL Server 专用的 OLE DB Provider，后者有更高的访问效率。

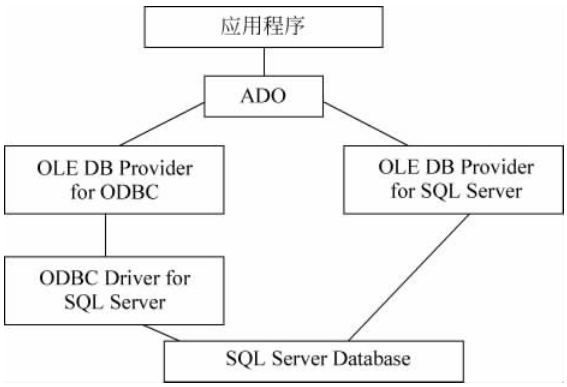

图 5-19　ADO 访问 SQL Server 的接口

4）ADO. NET 数据库接口

ASP. NET 使用 ADO. NET 数据模型。该模型从 ADO 发展而来，但它不只是对 ADO 的改进，而是采用了一种全新的技术。主要表现在以下几个方面。

（1）ADO. NET 不是采用 ActiveX 技术，而是与. NET 框架紧密结合的产物。

（2）ADO. NET 包含对 XML 标准的完全支持，这对于跨平台交换数据具有重要的意义。

（3）ADO. NET 既能在与数据源连接的环境下工作，又能在断开与数据源连接的条件下工作。特别是后者，非常适合于网络应用的需要。因为在网络环境下，保持与数据源连接，不符合网站的要求，不仅效率低，付出的代价高，而且常常会引发由于多个用户同时访问带来的冲突。因此，ADO. NET 系统集中主要精力用于解决在断开与数据源连接的条件下数据处理的问题。

ADO. NET 提供了面向对象的数据库视图，并且在 ADO. NET 对象中封装了许多数据库属性和关系。最重要的是，ADO. NET 通过很多方式封装和隐藏了很多数据库访问的细节。可以完全不知道对象在与 ADO. NET 对象交互，也不用担心数据移动到另外一个数据库或者从另一个数据库获得数据的细节问题。如图 5-20 所示为 ADO. NET 架构。

5）JDBC 数据库接口

JDBC(Java Data Base Connectivity)是由 Java Soft 公司开发的，以一组 Java 语言编写的用于数据库连接和操作的类和接口，可为多种关系数据库提供统一的访问方式。通过 JDBC 完成对数据库的访问包括 4 个主要组件：Java 应用程序，JDBC 驱动器管理器，驱动器和数据源。

在 JDBC 的 API 中有两层接口：应用程序层和驱动程序层。前者使开发人员可以通过

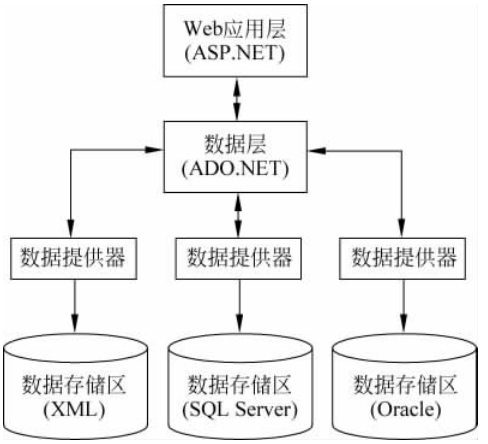

图 5-20　通过 ADO. NET 访问数据库的接口模型

SQL 调用数据库和取得结果,后者处理与具体数据库驱动程序的所有通信。

使用 JDBC 接口对数据库操作具有如下优点。

(1) JDBC 的 API 与 ODBC 十分相似,有利于用户理解;

(2) 使编程人员从复杂的驱动器调用命令和函数中解脱出来,而致力于应用程序功能的实现;

(3) JDBC 支持不同的关系数据库,增强了程序的可移植性。

使用 JDBC 的主要缺点:访问数据记录的速度会受到一定影响,此外,由于 JDBC 结构中包含不同厂家的产品,这给数据源的更改带来了较大麻烦。

4. 数据库试运行

应用程序编写完成,并有了一小部分数据装入后,应该按照系统支持的各种应用分别试验应用程序在数据库上的操作情况,这就是数据库的试运行阶段,或者称为联合调试阶段。在这一阶段要完成两方面的工作:①功能测试,实际运行应用程序,测试它们能否完成各种预定的功能;②性能测试,测量系统的性能指标,分析系统是否符合设计目标。

系统的试运行对于系统设计的性能检验和评价是很重要的,因为有些参数的最佳值只有在试运行后才能找到。如果测试的结果不符合设计目标,则应返回到设计阶段,重新修改设计和编写程序,有时甚至需要返回到逻辑设计阶段,调整逻辑结构。

由于数据装入的工作量很大,所以可分期分批地组织数据装入,先输入小批量数据做调试用,待试运行基本合格后,再大批量输入数据,逐步增加数据量,逐步完成运行评价。

数据库的实施和调试不是几天就能完成的,需要有一定的时间。在此期间由于系统还不稳定,随时可能发生硬件或软件故障,加之数据库刚刚建立,操作人员对系统还不熟悉,对其规律缺乏了解,容易发生操作错误,这些故障和错误很可能破坏数据库中的数据,这种破坏很可能在数据库中引起连锁反应,破坏整个数据库。

因此必须做好数据库的转储和恢复工作,要求设计人员熟悉 DBMS 的转储和恢复功能,并根据调试方式和特点首先加以实施,尽量减少对数据库的破坏,并简化故障恢复。

5.7 数据库的运行和维护

数据库试运行结果符合设计目标后，数据库就可以真正投入运行了。数据库投入运行标志着开发任务的基本完成和维护工作的开始，并不意味着设计过程的终结，由于应用环境在不断变化，数据库运行过程中物理存储也会不断变化，对数据库设计进行评价、调整、修改等维护工作是一个长期的任务，也是设计工作的继续和提高。

在数据库运行阶段，对数据库经常性的维护工作主要是由 DBA 完成的，它包括：

1. 数据库的转储和恢复

定期对数据库和日志文件进行备份，以保证一旦发生故障，能利用数据库备份及日志文件备份，尽快将数据库恢复到某种一致性状态，并尽可能减少对数据库的破坏。

2. 数据库的安全性、完整性控制

DBA 必须对数据库安全性和完整性控制负起责任。根据用户的实际需要授予不同的操作权限。另外，由于应用环境的变化，数据库的完整性约束条件也会变化，也需要 DBA 不断修正，以满足用户要求。

按照设计阶段提供的安全规范和故障恢复规范，DBA 要经常检查系统的安全是否受到侵犯，根据用户的实际需要授予用户不同的操作权限。

数据库在运行过程中，由于应用环境发生变化，对安全性的要求可能发生变化，DBA 要根据实际情况及时调整相应的授权和密码，以保证数据库的安全性。

同样，数据库的完整性约束条件也可能会随应用环境的改变而改变，这时 DBA 也要对其进行调整，以满足用户的要求。

另外，为了确保系统在发生故障时能够及时地进行恢复，DBA 要针对不同的应用要求定制不同的转储计划，定期对数据库和日志文件进行备份，以使数据库在发生故障后恢复到某种一致性状态，保证数据库的完整性。

3. 数据库性能的监督、分析和改进

目前许多 DBMS 产品都提供了监测系统性能参数的工具，DBA 可以利用这些工具方便地得到系统运行过程中一系列性能参数的值。DBA 应该仔细分析这些数据，通过调整某些参数来进一步改进数据库性能。经常对数据库的存储空间状况及响应时间进行分析评价；结合用户的反应情况确定改进措施；及时改正运行中发现的错误；按用户的要求对数据库的现有功能进行适当的扩充。

4. 数据库的重组织和重构造

数据库运行一段时间后，由于记录的不断增加、删除和修改，会改变数据库的物理存储结构，使数据库的物理特性受到破坏，从而降低数据库存储空间的利用率和数据的存取效率，使数据库的性能下降。因此，需要对数据库进行重新组织，即重新安排数据的存储位置，回收垃圾，减少指针链，改进数据库的响应时间和空间利用率，以提高系统性能。这与操作系统对“磁盘碎片”的处理的概念相类似。

数据库的重组只是使数据库的物理存储结构发生变化，而数据库的逻辑结构不变，所以根据数据库的三级模式，可以知道数据库重组对系统功能没有影响，只是为了提高系统的性能。

数据库应用环境的变化可能导致数据库的逻辑结构发生变化,比如要增加新的实体,增加某些实体的属性,这样实体之间的联系发生了变化,使原有的数据库设计不能满足新的要求,必须对原来的数据库重新构造,适当调整数据库的模式和内模式,比如要增加新的数据项,增加或删除索引,修改完整性约束条件等。

DBMS 一般都提供了重新组织和构造数据库的应用程序,以帮助 DBA 完成数据库的重组和重构工作。

只要数据库系统在运行,就需要不断地进行修改、调整和维护。一旦应用变化太大,数据库重新组织也无济于事,这就表明数据库应用系统的生命周期结束,应该建立新系统,重新设计数据库。从头开始进行数据库设计工作,标志着一个新的数据库应用系统生命周期的开始。

5.8 数据库设计实例

为了具体说明数据库的设计方法,本节选取两个实例对数据库的设计过程进行说明。为了突出数据库设计的特点,重点强调了概念结构设计和逻辑结构设计的方法和步骤。当然,要能够熟练掌握数据库设计的技巧,还必须综合软件工程的知识,以及实际应用环境的知识,将理论和实践相结合,才能设计出满足要求的数据库应用系统。

实例 1:某公司公开招聘职员管理系统

第一步:需求分析。

假设用户需求情况如下。

某公司准备公开招聘若干个公司部门经理和职员,为了使招聘工作公开化,公司需要进行报名、考试(笔试、面试)、公布考试结果等工作。

要求每个需要报考的人员填写报考人员登记表,登记表主要内容有准考证号、身份证号、姓名、年龄、性别、学历、单位名称、单位负责人、政治面貌;对于每个报考人员要详细填写工作经历,包括时间、地点、职务、证明人;一个人可以报考多个职位,每个职位可以有多个人报名参加考试;一个人报考一个职位就对应一个面试成绩和笔试成绩;描述报考职位的属性有职位代码、职位名称。

第二步:概念结构设计。

从需求分析的结果中,抽象出实体、实体属性和实体之间的联系,见图 5-21~图 5-24。

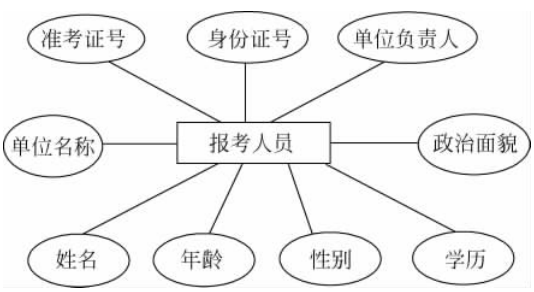

图 5-21 "报考人员"实体及属性

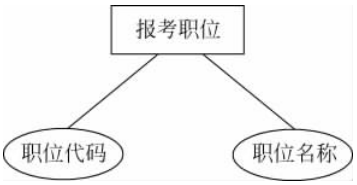

图 5-22 “报考职位”实体及属性

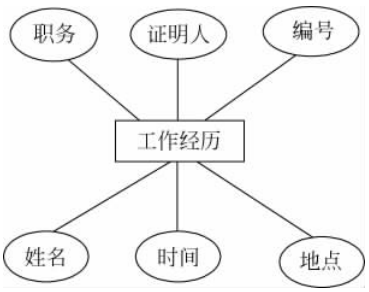

图 5-23 “工作经历”实体及属性

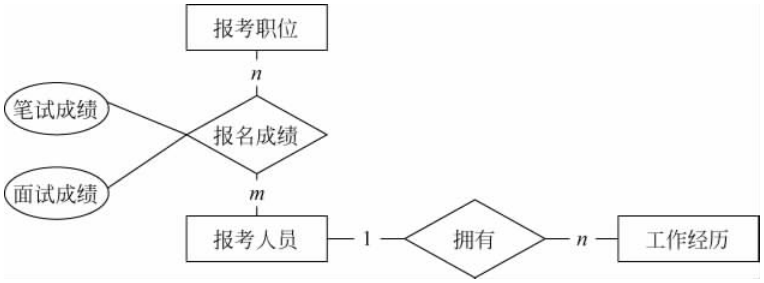

图 5-24 实体之间的联系图

合并 E-R 图后，报考职位和报考人员之间以“报名成绩”作为联系，联系的属性包括“笔试成绩”和“面试成绩”。

第三步：逻辑设计。

(1) 将上一步概念结构设计的结果(图 5-24)转换到关系模型，得到如下结果。

报考职位(职位代码，名称)

报考人员(准考证号，身份证号，姓名，年龄，性别，学历，单位名称，单位负责人，政治面貌)

工作经历(编号，开始时间，结束时间，姓名，地点，职务，证明人，准考证)

报名成绩(职位代码，准考证号，笔试成绩，面试成绩)

在上面的 4 个关系模式中，下画线标明的是关系模式的主码，波浪线标明的是关系模式的外码。特别说明对于报考人员关系模式，有两个候选码：准考证号和身份证号。在该报考管理系统中，从实际操作方面考虑，选择“准考证号”作为报考人员这个关系模式的主码。

(2) 关系模式的优化。

分析上面的关系模式，发现只有在关系模式“报考人员”中，存在着如下的传递函数依赖。

准考证号→单位名称

准考证号→单位负责人

单位名称→单位负责人

因为关系模式“报考人员”不存在部分函数依赖，所以该模式属于 2NF。将该关系模式进行分解得到下面两个子模式。

报考人员(准考证号,身份证号,姓名,年龄,性别,学历,单位名称,政治面貌)

单位(单位名称,单位负责人)

其他的关系模式均属于 3NF。

实例 2:医院管理信息系统

第一步:需求分析。

假设某医院需求情况如下。

医院有若干科室,科室包括科室编号、名称、人数、地点、负责人。医院每一个科室负责若干个病房,有若干名医生,医生包括医生编号、姓名、职务、学历、职称、简历,每个医生的简历包括简历编号、开始时间、终止时间、单位、担任职务、证明人。一个医生要负责几个病房病人的医疗工作,每个病房又可以有多个医生为病人治疗,但一个病人只能由一个医生负责。对于病人,医院关心病人编号、姓名、性别、年龄、住院时间、出院时间、病因等信息,对于病房关心病房号、床位数、床位号、床位是否为空等信息。

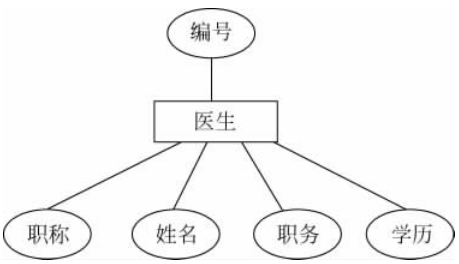

图 5-25 "医生"实体及属性

第二步:概念结构设计。

通过需求分析,可以得到系统中的实体包括医生、科室、简历、病人和病房,实体及实体的属性如图 5-25～图 5-29 所示,实体之间的联系如图 5-30 所示。

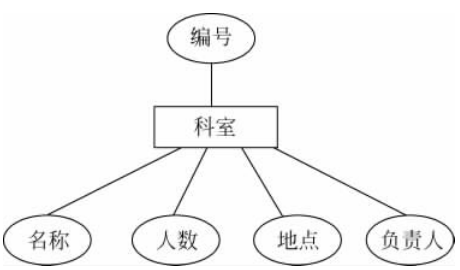

图 5-26 "科室"实体及属性

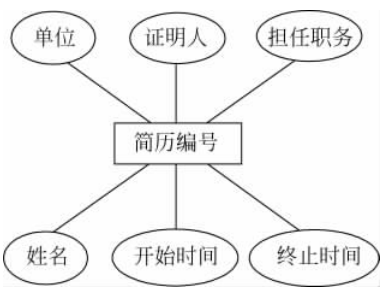

图 5-27 "简历"实体及属性

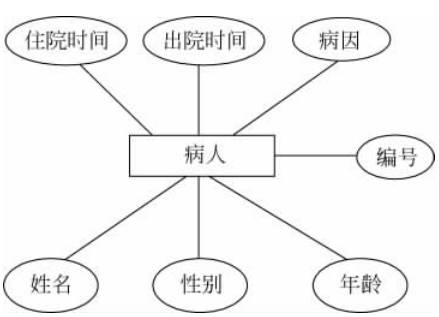

图 5-28 "病人"实体及属性

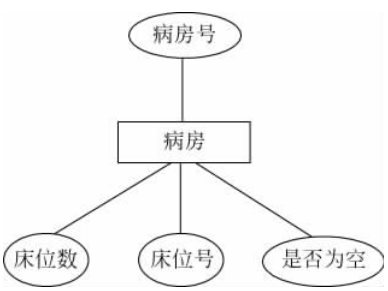

图 5-29 "病房"实体及属性

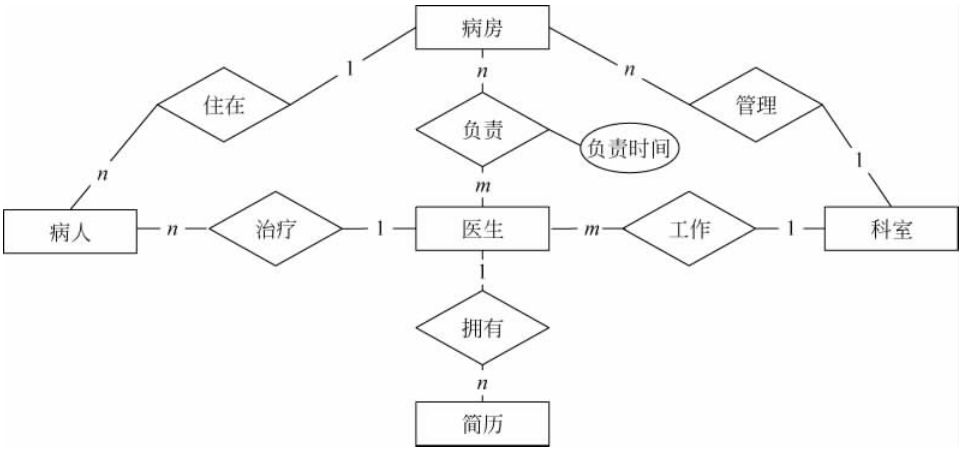

图 5-30　系统总 E-R 图

在合并各关系模式时，注意到病房的属性“床位是否为空”可以由病人的属性“病房号”“床位号”以及“住院时间”“出院时间”确定，即某个病人在住院期间的“病房号”和“床位号”可以确定该病床是否为空，所以“床位是否为空”是冗余属性，应该将其去除。

第三步：逻辑结构设计。

(1) 首先根据图 5-30 的总 E-R 图，将实体转换为对应的关系模式。

医生(医生编号，姓名，职称，职务，学历)

科室(科室编号，科室名称，人数，位置，负责人)

简历(简历编号，医生姓名，单位，职务，开始时间，终止时间，证明人)

病人(病人编号，姓名，性别，出生日期，住院时间，出院时间，病因)

病房(病房号，床位号，病床数)

其次，将联系采用合并的方式进行转换，得到下列表示联系的模式集合。

医生(医生编号，姓名，职称，职务，学历，科室编号)

简历(简历编号，医生编号，医生姓名，单位，担任职务，开始时间，终止时间，证明人)

病人(病人编号，姓名，性别，年龄，住院时间，出院时间，病因，病房编号，床位号，医生编号)

病房(病房号，床位号，病床数，科室编号)

负责(病房号，医生编号，负责时间)

最后，将实体和联系转换的结果合并，得到最终的逻辑模式集合。

医生(医生编号，姓名，职称，职务，学历，科室编号)

科室(科室编号，科室名称，人数，位置，负责人)

简历(简历编号，医生编号，医生姓名，单位，担任职务，开始时间，终止时间，证明人)

病人(病人编号，姓名，性别，年龄，住院时间，出院时间，病因，病房编号，床位号，医生编号)

病房(病房号，床位号，病床数，科室编号)

负责(病房号，医生编号，负责时间)

(2) 关系模式的优化。

在“简历”模式中,存在:“简历编号→医生姓名”的传递依赖,而“简历”模式属于 2NF,为了满足 3NF 的要求,将“医生姓名”属性从“简历”模式中删除。其他关系模式都满足 3NF,不再需要进一步优化。

小　　结

本章从“软件工程”的角度讨论了数据库设计的 6 个阶段。在本章的学习过程中,除了要掌握书中讨论的基本原理和方法外,还要主动地尝试在实际应用中运用这些思想解决具体问题,这样将实践和理论相结合,才能设计出符合应用需求的数据库应用系统。

习　题　5

5.1　名词解释。

数据库设计　基于 3NF 的数据库设计方法　基于 E-R 模型的数据库设计方法

5.2　什么是数据库设计?试述数据库设计的步骤。

5.3　试述数据库设计需求分析阶段的任务和方法。

5.4　数据流图和数据字典的内容和作用分别是什么?

5.5　视图集成时,分 E-R 图之间的冲突有哪些?解决这些冲突的方法是什么?

5.6　试述数据库逻辑结构设计的步骤。

5.7　试述 E-R 图转换成关系模型的转换规则。

5.8　规范化理论对数据库设计有什么指导意义?

5.9　试述数据库逻辑结构设计结果的优化方法。

5.10　试述数据库物理结构设计的内容和步骤。

5.11　数据库实施阶段的主要任务是什么?

5.12　数据库系统投入运行后,有哪些维护工作?

5.13　某商业集团管理系统的数据库信息如下。

该系统中包含三个实体集:一是“仓库”实体集,属性有仓库号、仓库名和地址等;二是“商店”实体集,属性有商店号、商店名、地址等;三是“商品”实体集,属性有商品号、商品名、单价。设仓库与商品之间存在“库存”联系,每个仓库可存储若干种商品,每种商品存储在若干仓库中,每个仓库每存储一种商品有存储日期及存储量;商店与商品之间存在着“销售”联系,每个商店可销售若干种商品,每种商品可在若干商店里销售,每个商店销售一种商品有月份和月销售量两个属性。

请在上述背景介绍的基础上,完成如下数据库设计。

(1) 试画出 E-R 图,并在图上注明联系类型。

(2) 将 E-R 图转换成满足 3NF 的关系模式,并标识主外键(用下画线标识主码,用波浪线标识外键)。

5.14　现针对学生参与教师的科研项目建立“科研项目管理数据库系统”,其中,学生信息包括学号,姓名,性别,所在学院;学院信息包括学院编号,学院名称,办公电话;教师信

息包括教师编号，姓名，性别，职称，所在学院；项目信息包括项目编号，项目名称，开始时间，结束时间，项目负责人，职称。各实体之间的关系为：一个学生可以参与教师的多个项目，一个项目可以有多个学生参加，每个学生选定项目后要承担相应的任务；一个教师可以主持多个项目，一个项目只能由一个教师作为项目负责人。

请在上述背景介绍的基础上，完成如下数据库设计。

(1) 画出“科研项目管理数据库系统”的 E-R 图。

(2) 将 E-R 图转换为一组符合 3NF 要求的关系模式，并标出每个关系模式的主外键(用下画线标识主码，用波浪线标识外键)。

5.15　某工厂零件管理系统的需求分析如下。

(1) 一个车间有多个工人，每个工人有职工号、姓名、年龄、性别、工种；

(2) 一个车间生产多种产品，产品有产品号、价格；

(3) 一个车间生产多种零件，一种零件也可能为多个车间制造，零件有零件号、重量、价格；

(4) 一种产品由多种零件组成，一种零件也可装配到多种产品中，产品与零件均存入仓库中；

(5) 厂内有多个仓库，仓库有仓库号、主任姓名、电话。

请在上述背景介绍的基础上，完成如下数据库设计。

(1) 请画出该系统的 E-R 图。

(2) 给出相应的关系模型，并标出每个关系模式的主外键(用下画线标识主码，用波浪线标识外键)。

5.16　设计一个学校的图书管理系统，请给出该系统的需求分析并进行数据库设计，具体要求为：

(1) 实体数不少于 5 个，每个实体有属性 3～6 个，实体之间的关系至少要包含 1∶n，m∶n 两种联系类型。

(2) 给出该系统的 E-R 图。

(3) 将 E-R 图转换为一组符合 3NF 要求的关系模式，并标识主外键(用下画线标识主码，用波浪线标识外键)。