

# 第1章

## 数据库系统基础知识

### 本章学习目标

- 了解数据库技术的发展过程及特点
- 了解数据库应用系统的开发方法和开发步骤
- 了解 SQL Server 的性能和特点
- 掌握数据库系统模型及其体系结构
- 掌握数据库、数据库管理系统、数据库系统等基本概念

### 1.1 数据库系统概述

数据库技术是计算机科学技术发展最快、应用最广的领域之一,是计算机信息系统与应用系统的核心技术和重要基础。数据库系统是指在计算机系统中引入数据库技术的系统,本节简要介绍数据库管理技术的产生和发展,以及数据库系统涉及的基本概念和数据库系统的特点。

#### 1.1.1 数据库管理技术的产生和发展

数据库管理技术是应数据管理需要而产生的,数据库管理技术包括数据处理和数据管理。数据处理是对各种数据进行收集、存储、加工和传播的一系列活动的总和;数据管理则是数据分类、组织、编码、存储、检索和维护技术的总和。数据库管理技术的发展是与计算机技术的发展紧密联系在一起,数据库管理技术主要经历了人工管理、文件管理和数据库系统三个发展阶段。

##### 1. 人工管理阶段

20 世纪 50 年代中期,计算机在诞生之初主要用于科学计算。从硬件上讲,只有卡片、纸带、磁带等顺序存取存储设备。从软件上讲,没有操作系统,没有专门的数据管理的软件。数据处理方式是批处理。图 1-1 是人工管理阶段程序与数据的对应关系。人工管理阶段数据处理有如下特点:(1)数据不能保存;(2)程序和数据放在一起;(3)数据不能共享;(4)数据没有结构;(5)数据没有独立性。

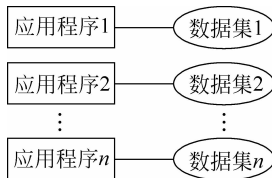


图 1-1 人工管理阶段程序与数据的对应关系

## 2. 文件管理阶段

20 世纪 50 年代后期至 60 年代中期,始于操作系统产生之后,这个时期的计算机不仅用于科学计算,还大量用于数据管理。计算机的硬件系统已经有了磁盘、磁鼓等直接存取存储设备。并且有了以操作系统为核心,并包含文件系统的软件系统。在文件系统的支持下,数据可以与程序分离,形成独立的数据文件,但数据文件仍然无结构。图 1-2 是文件管理阶段应用程序与数据的对应关系。文件管理阶段数据处理有如下特点:(1)数据能够长期保存;(2)程序和数据能够分离;(3)数据能够共享,但冗余度大;(4)数据文件形式多样;(5)数据没有独立性。

## 3. 数据库管理阶段

20 世纪 60 年代后期,从这个时期开始,计算机迅速应用于各个领域,计算机用于管理的数据规模越来越大。在硬件方面,出现了大容量且价格低廉的磁盘存储设备。在软件方面,出现了专门的数据库管理系统软件。在数据库管理系统的技术支持下,数据文件采用一定的数据模型,完全独立于应用程序,并有安全性和完整性保障。图 1-3 是数据库管理阶段程序与数据的对应关系。数据库管理阶段数据处理有如下特点:(1)数据结构化;(2)数据冗余低;(3)数据独立性高;(4)数据由 DBMS 统一管理控制;(5)为用户提供友好接口。

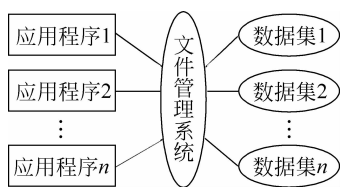


图 1-2 文件管理阶段程序与数据的对应关系

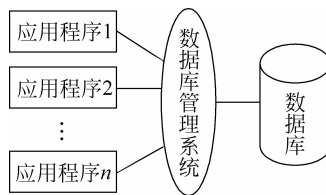


图 1-3 数据库管理阶段程序与数据的对应关系

### 1.1.2 数据库系统基本概念

在数据库系统的基础知识中,数据、数据库、数据库管理系统、数据库系统和数据库应用系统是 5 个密切相关的基本概念,是学习和理解数据库管理技术的基础,必须认真理解和掌握。

#### 1. 数据(Data)

数据定义为描述事物的符号记录。这些符号记录在形式上可以是文字、图形、图像和声音等,在内容上可以是学生成绩、职工档案等,数据有多种表现形式,经过加工处理就可以存入计算机系统中。数据是数据库中存储的对象。

#### 2. 数据库(DataBase,DB)

数据库是数据库系统的核心,是数据库管理的对象。所谓数据库是指长期存储在计算机内、有组织的、可共享的数据的集合。数据库中的数据按一定的数据模型组织、描述和存储,具有较小的冗余度,较高的数据独立性和易扩展性,并可为各种用户共享。

### 3. 数据库管理系统(DataBase Management System,DBMS)

前面介绍了数据和数据库的概念,数据是描述事物的符号记录,数据库是存储在计算机内、有组织的、可共享的数据集合。数据库管理系统就是科学地组织和存储、高效地获取和维护数据库的一个系统软件。

数据库管理系统是位于用户与操作系统之间的一层数据管理软件,其功能是提供对数据库的统一管理和控制。它的主要功能包括以下几个方面。

#### 1) 数据定义功能

数据库管理系统提供相应的数据语言来定义数据库结构,使用它用户可以方便地对数据库中的数据对象进行定义。数据定义功能刻画数据库的框架,并被保存在数据字典中。

#### 2) 数据操纵功能

数据库管理系统提供数据操纵语言,实现对数据库数据的基本存取操作,如检索、更新、插入、修改和删除等。

#### 3) 数据库运行管理功能

数据库管理系统提供数据控制功能,即从数据的安全性、完整性和并发控制等方面对数据库运行进行有效地控制和管理,以确保数据正确有效。

#### 4) 数据库的建立和维护功能

数据库管理系统提供数据库初始数据的输入、转换功能,数据库的转储、恢复、重新组织,系统性能监视、分析等功能。这些功能常由一些实用程序完成。

### 4. 数据库系统(DataBase System,DBS)

数据库系统指在计算机系统中引入数据库后构成的系统,主要由 5 部分组成:数据库、数据库管理系统、数据库应用系统、数据库管理员和用户。图 1-4 为数据库系统的组成框图。

#### 1) 数据库(DB)

数据库是存储在外存介质(磁带、磁盘、光盘等)上、按一定结构组织在一起、可共享的的相关数据的集合。

#### 2) 数据库管理系统(DBMS)

数据库管理系统是一个能够描述、管理、维护数据库的系统软件,其功能是对数据库进行统一管理和控制。

#### 3) 数据库应用系统(DataBase Application System,DBAS)

数据库应用系统即用户应用程序,是指系统开发人员使用开发工具,利用数据库系统资源开发出来的,面向某一类实际应用问题的应用软件系统,例如以数据库为基础的学籍管理系统、财务管理系统、图书管理系统等都是数据库应用系统(用户应用程序)。

#### 4) 数据库管理员(DataBase Administrator,DBA)

数据库管理员负责建立、使用和维护数据库,是负责管理和维护数据库服务器的人员。根据数据库管理员对数据库的掌握情况,可将其分成三个等级:初级、中级和高级。数据库

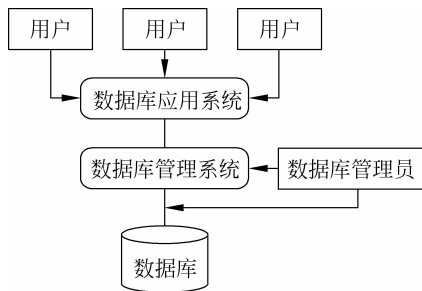


图 1-4 数据库系统的组成框图

管理员的具体职责包括：

- (1) 决定数据库中的信息内容和结构；
  - (2) 决定数据库的存储结构和存取策略；
  - (3) 定义数据的安全性要求和完整性约束条件；
  - (4) 监控数据库的使用和运行；
  - (5) 负责数据库的改进和重组重构工作。
- 5) 用户(User)

在数据库系统中具有使用数据库应用系统软件权限的部门或人员称为用户,用户通过数据库应用系统提供的用户界面使用数据库,完成应用程序能够实现的各种任务。

### 1.1.3 数据库系统的特点

数据库技术发展到今天,仅有 50 余年的历史,但发展速度之快,应用范围之广,是其他技术所不及的,这是因为在数据库技术在发展过程中克服了人工管理阶段和文件管理阶段的缺点,在数据库管理阶段体现出许多优点。

#### 1. 采用数据模型方式组织数据

数据库系统中的数据是有结构的,即数据的结构化,这是数据库系统与文件系统的根本区别,数据的结构化是由 DBMS(数据库管理系统)所支持的数据模型表现出来。使用数据模型方式组织数据,数据库系统不仅可以表示事物内部各数据项之间的联系,而且可以表示事物与事物之间的联系,即不但描述记录内部的结构性,还要描述数据记录之间的关系。任何一个数据库管理系统都支持一种抽象的数据模型,以此反映出现实世界事物之间的联系。

#### 2. 数据共享性高,冗余度低,容易扩充

##### 1) 共享性高

共享性高是指在数据库系统中,数据库中的一片独立的数据可以被其他几个不同的用户共同使用,即每个用户都可以存取该片数据用于不同的目的。这是因为在数据库系统中,数据的定义和描述已经从用户应用程序中分离出来,数据库系统从整体角度看待和描述数据,数据不再面向某个应用而是面向整个系统,因此数据可以被多个用户共享使用。

##### 2) 冗余度低

冗余度是指同一数据被重复存储的程度。数据共享可以大大减少数据冗余,避免数据之间的不相容性与不一致性。冗余度低并不是没有一点重复,而是重复得很少,去掉了那些多余的和有害的数据。

##### 3) 容易扩充

由于数据面向整个系统,是有结构的数据,不仅可以被多个应用共享使用,而且容易增加新的应用,这就使得数据库系统弹性大,易于扩充,可以适应各种用户的要求。

#### 3. 数据有较高的独立性

数据的独立性是数据库系统中一个常用的术语,包括物理独立性和逻辑独立性。数据独立性是指数据库应用系统读取数据不是直接从数据库中读取,而是通过 DBMS 间接读

取,所以保持了数据库应用系统与数据库中的数据的物理独立性和逻辑独立性。

#### 1) 物理独立性

物理独立性是指数据库应用系统与存储在磁盘(物理结构)上的数据库中的数据相互独立。也就是说,数据库中的数据在磁盘上怎样存储是由 DBMS 管理的,用户和用户应用程序不需要了解和掌握,数据库应用程序要处理的只是数据的逻辑结构,这样当数据的物理存储改变时,数据库应用程序不用改变。

#### 2) 逻辑独立性

逻辑独立性是指数据库应用程序与数据库的逻辑结构是相互独立的,也就是说,当数据的逻辑结构改变时,数据库应用程序可以不变。

在数据库系统中,数据独立性是通过数据库管理系统 DBMS 提供映像功能来实现的。数据与程序的独立,使得用户应用程序在数据的逻辑结构与物理存储结构之间具有较高的独立性。这样用户在操作数据时,不用考虑数据在存储器上的物理位置与结构,只需通过简单的逻辑结构对数据进行操作,从而简化应用程序的编制,降低了应用程序的维护和修改成本。

### 4. 数据由 DBMS 统一管理和控制

数据库可以被多个用户应用程序共享,因而存在着并发控制问题。数据库管理系统提供必要的控制措施,不仅包括并发访问控制功能,而且还包括数据安全性控制功能和数据完整性控制功能等。

#### 1) 数据的安全性控制

数据的安全性是指保护数据以防止不合法的使用造成的数据的泄密和破坏,使每个用户只能按规定对某些数据以某些方式进行使用和处理。

#### 2) 数据的完整性控制

数据的完整性指数据的正确性、有效性和相容性。完整性检查将数据控制在有效的范围内,或保证数据之间满足一定的约束关系。

#### 3) 并发控制

当多个用户的并发进程同时存取、修改数据库时,可能会发生相互干扰而得到错误的结果或使得数据库的完整性遭到破坏,因此必须对多用户的并发操作加以控制和协调。

#### 4) 数据库恢复

计算机系统的硬件系统故障、软件系统故障、用户操作失误以及人为的破坏也会影响数据库中数据的正确性,甚至造成数据库部分或全部数据的丢失。DBMS 必须具有将数据库从错误状态恢复到某一已知的正确状态(也称为完整状态或一致状态)的功能,这就是数据库的恢复功能。

## 1.2 数据库系统的数据模型

### 1.2.1 客观世界与数据模型

在数据库中引入计算机技术,信息处理的对象是描述客观世界的数据。数据库中的

数据是某个企业、公司、组织或部门等所涉及业务的综合反映。由于计算机不能直接处理现实世界中的具体事物,人们必须事先把具体的事物转换成计算机能够处理的数据,存储到计算机中,然后用计算机技术模拟客观世界,完成各种应用任务,实现人们不同的目的和要求。在数据库中用数据模型工具来抽象、表示和处理客观世界中的数据和信息。因此数据模型应满足三方面的要求,一是能比较真实地模拟客观世界;二是容易被人们理解;三是便于在计算机上实现。一种数据模型要很好地满足这三方面的要求,目前很困难,在数据库中针对不同的使用对象和应用目的,采用不同的数据模型。数据模型按不同的应用层次分成三种类型,分别是概念数据模型(简称概念模型)、逻辑数据模型(简称逻辑模型)、物理数据模型(简称物理模型)。图 1-5 是现实世界与数据模型的关系示意图。

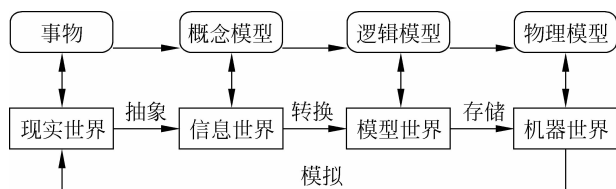


图 1-5 现实世界与数据模型的关系示意图

概念模型是面向数据库用户的现实世界的模型,主要用来描述世界的概念化结构,它使数据库的设计人员在设计的初始阶段摆脱计算机系统及数据库管理系统的具体技术问题,集中精力分析数据以及数据之间的联系等概念模型,它与具体的数据库管理系统无关。概念模型必须换成逻辑模型,才能在数据库管理系统中实现。

逻辑模型是用户从数据库所看到的模型,是具体的数据库管理系统所支持的数据模型,如关系数据模型、网状数据模型、层次数据模型等。逻辑模型既要面向用户,又要面向系统,主要用于数据库管理系统(DBMS)的实现。

物理模型是面向计算机物理表示的模型,描述了数据在储存介质上的组织结构,它不但与具体的 DBMS 有关,而且与操作系统和硬件有关。每一种逻辑数据模型在实现时都有相对应的物理数据模型。DBMS 为了保证其独立性与可移植性,大部分物理数据模型的实现工作由系统自动完成,而设计者只设计索引、聚集等特殊结构。

从概念模型到逻辑模型的转换是由数据库设计人员完成的,从逻辑模型到物理模型的转换是由 DBMS 完成的,普通人员不需要考虑物理模型实现的细节,因此逻辑数学模型是数据库系统的基础,也是数据库应用过程中必须考虑的核心问题。

### 1.2.2 概念模型

从图 1-5 可以看出,概念模型是现实世界到机器世界的一个中间过程,概念模型用于信息世界的建模,是从现实世界到信息世界的抽象过程,是数据库设计人员进行数据库设计的有力工具,是数据库设计人员与用户之间进行交流的语言,因此概念模型不仅应该具有较强的语义表达能力,能够方便、直接地表达应用中的各种语义知识,而且还应该简单、清晰,使用户易于理解。下面介绍概念模型中的基本术语和概念模型的表示。

## 1. 概念模型中的基本术语

### 1) 实体(Entity)

现实世界存在着各种事物,而事物与事物之间是彼此相互区别和联系的。我们把客观存在并可相互区别的事物称为实体。实体可以是具体的人,如学生、教师、职工等;也可以是具体的事和物,如书、课程、学号、演出等。它不仅指现实世界的客观对象,也可以指事物与事物的联系,如学生的选课、教师的授课等都是实体。

### 2) 属性(Attribute)

实体所具有的某一特性称为属性,实体通过属性来描述。一个实体可以包含若干个属性,如学生实体可以由学号、姓名、性别、出生年份、学院、入学时间等属性组成,教师实体可以由教工号、姓名、性别、出生年份、职称、入校时间等属性组成。

### 3) 码(Key)

码是唯一标识实体的最小属性集。例如,学号是学生实体的码,学号可以唯一标识一名学生;教工号是教师实体的码,教工号可以唯一标识一名教师。

### 4) 域(Domain)

属性的取值范围称为该属性的域。例如,学号的域为 8 位整数,姓名的域为字符串集合,年龄的域为 18~60 的整数。

### 5) 实体型(Entity Type)

实体型就是实体的结构描述,通常是实体名和属性名的集合。具有相同属性的实体有相同的实体型。例如,学生(学号,姓名,性别,出生年份,学院,入学时间)就是一个实体型,教师(教工号,姓名,性别,出生年份,职称,入校时间)也是一个实体型。

### 6) 实体集(Entity Set)

性质相同的同型实体的集合称为实体集。例如,全体学生就是一个实体集,全校的教师也是一个实体集。

### 7) 联系(Relationship)

在现实世界中,事物内部以及事物之间是有联系的,这些联系在概念世界中反映为实体型内部的联系和实体型之间的联系。实体型内部的联系通常是指组成实体型的各属性之间的联系。实体间的联系通常是指不同实体集之间的联系。实体型之间有各种各样的对应关系,归纳起来有三种类型,如图 1-6 所示。

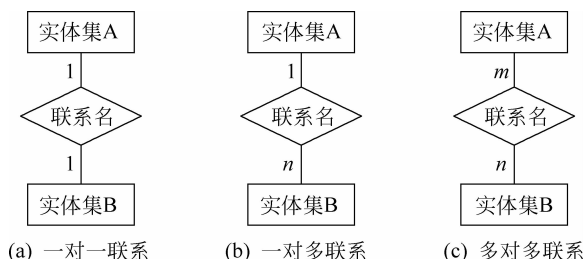


图 1-6 两个实体型之间的联系示意图

### (1) 一对一联系

如图 1-6(a)所示,如果实体集 A 中的每一个实体,在实体集 B 中仅有一个实体与之联系,反之亦然,则实体集 A 与实体集 B 具有一对一联系,记为 1:1。例如,一个公司只有一个经理,一个经理只在一个公司任职,则经理与公司之间的联系是一对一的联系。

### (2) 一对多联系

如图 1-6(b)所示,如果实体集 A 中的每一个实体,在实体集 B 中有  $n(n \geq 0)$  个实体与之联系;而实体集 B 中的每一个实体,在实体集 A 中仅有一个实体与之联系,则实体集 A 与实体集 B 是一对多联系,记为  $1:n(n \geq 0)$ 。例如,一个公司有许多员工,但一个员工只能在一个公司任职,则公司与职员之间的联系是一对多的联系。

### (3) 多对多联系

如图 1-6(c) 所示, 如果实体集 A 中的每一个实体, 在实体集 B 中有  $n(n \geq 0)$  个实体与之联系; 而实体集 B 中的每一个实体, 在实体集 A 中有  $m(m \geq 0)$  个实体与之联系, 则实体集 A 与实体集 B 是多对多的联系, 记为  $m : n$ 。例如, 图书馆的一种图书可由多位读者借阅, 一个读者可以借阅多种图书, 则图书和读者之间是多对多联系。

其实,三种联系之间有着一定的联系,一对一联系是一对多联系的特例,而一对多联系又是多对多联系的特例。

## 2. 概念模型的表示

概念模型是对信息世界的建模,是反映实体之间联系的模型。数据库设计的重要任务就是建立概念模型。概念模型的表示方法很多,其中最为著名和最为常用的是实体-联系方法(Entity-Relationship Approach)。该方法用 E-R 图来描述现实世界的概念模型,E-R 图提供了表示实体型、属性和联系的方法,E-R 方法也称为 E-R 模型。

- (1) 实体型：用矩形表示，矩形框内写明实体名。
- (2) 属性：用椭圆形表示，并用无向边将其与相应的实体连接起来。
- (3) 联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时，在无向边旁标上联系类型（1:1、1:n 或 m:n）。联系本身也可以看成一种实体，如果联系具有属性，则这些属性也用无向边与该联系相连接。

如图 1-7 所示是一个班级与学生的 E-R 图。下面是班级管理概念模型涉及的实体和属性：

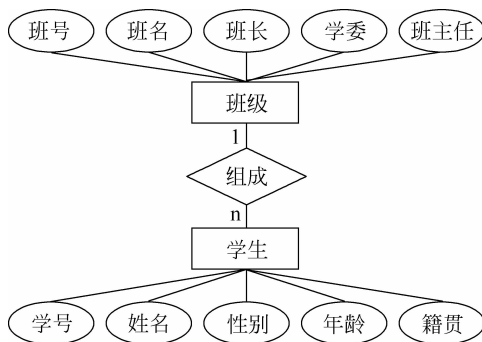


图 1-7 班级与学生的 E-R 图



班级：属性有班号、班名、班长、学委、班主任。

学生：属性有学号、姓名、性别、年龄、籍贯。

### 1.2.3 逻辑模型

逻辑模型即逻辑数学模型,是计算机世界的数学模型,是数据库系统的核心和基础,在计算机上实现的 DBMS 系统软件都基于某种逻辑数据模型。逻辑数学模型是严格定义的一组概念的集合,这些概念精确地描述了系统的静态特性、动态特性和完整性约束条件。逻辑数据模型通常由数据结构、数据操作和完整性约束 3 部分组成。

#### 1. 数据结构

数据结构用于描述系统的静态特性,是所研究的对象类型的集合。这些对象是数据库的组成成分,它们包括两类,一类是与数据类型、内容、性质有关的对象,如关系模型中的域、属性、关系等;一类是与数据之间的联系有关的对象,如关系模型中反映联系的关系等。

数据结构是刻画一个逻辑模型性质最重要的方面。在数据库系统中通常按照其数据结构的类型来命名逻辑模型。常见的 4 种类型结构是层次结构、网状结构、关系结构和面向对象结构,它们对应的逻辑模型分别名为层次模型、网状模型、关系模型和面向对象模型。

#### 2. 数据操作

数据操作用于描述系统动态特性。数据操作是指对数据库中各种对象的实例允许执行的操作的集合,包括操作及有关的操作规则。数据库主要有检索和更新(包括插入、删除、修改)两大类操作。逻辑数据模型必须定义这些操作的确切含义、操作符号、操作规则(如优先级)以及实现操作的语言。

#### 3. 数据的约束条件

数据的约束条件是一组完整性规则的集合。完整性规则是给定的逻辑数据模型中数据及其联系所具有的制约和依存规则,用以限定符合逻辑模型的数据库状态以及状态的变化,以保证数据的正确、有效及相容。

逻辑模型规定该模型必须遵守的基本的完整性约束条件。如在关系模型中,任何关系必须满足实体完整性和参照完整性两个条件;此外,逻辑模型还应该提供定义完整性约束条件的机制,以反映具体应用所涉及的数据必须遵守的特定的语义约束条件。如在校学生的数据库中要规定大学生入学年龄不得小于 18 岁等。

### 1.2.4 常用逻辑数据模型

#### 1. 层次模型(Hierarchical Model)

层次模型是数据库系统中最早出现的数据模型。层次模型用树形结构来表示各类实体以及实体间的联系,层次模型适合于表示数据之间具有从属关系的结构。层次模型像一棵

倒立的树,根结点在最上层,子结点在下层,逐层逐级排列,最低层是叶结点。上层结点与下层结点之间为一对多的关系。现实世界中许多实体之间的联系本来就呈现出一种很自然的层次关系,如某部门的行政机构、家族中的族谱都是层次关系。

例如,使用层次模型建立某商业大学的行政组织机构的层次模型如图 1-8 所示。其中“商业大学”为根结点,根结点的下一层是各二级学院,最低层是“教研室”为叶结点。

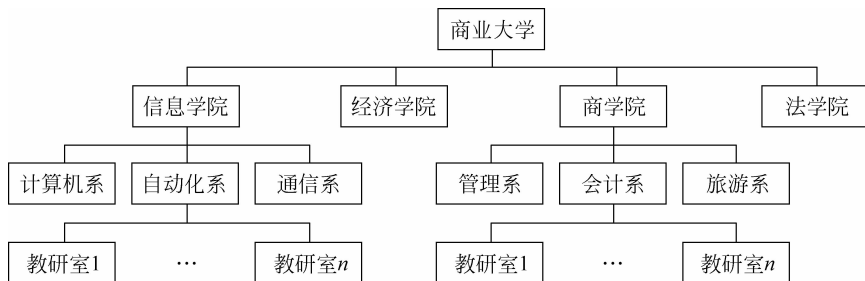


图 1-8 某商业大学组织机构的层次模型

从图 1-8 可以看出,层次模型具有以下特征。

- (1) 仅有一个根结点且无父结点。
- (2) 根结点以下的子结点向上有一个父结点,向下有若干子结点。
- (3) 最低层为叶结点且无子结点。

层次模型具有层次清晰、构造简单、易于实现等优点,但其本身的特性也限制了它可以方便地表示一对一和一对多的实体联系,而不能直接表示多对多联系。对于多对多联系,必须分解为几个一对多的联系才能表示出来。因而对复杂的数据库实现起来比较麻烦,这是层次模型的局限性。

层次数据库系统采用层次模型作为数据的组织方式,层次数据库系统的典型代表是 IBM 公司的 IMS(Information Management System)数据库管理系统,这是 1968 年 IBM 公司推出的第一个大型商用数据库管理系统,曾经得到广泛的使用。

## 2. 网状模型(Network Model)

网状模型是用网状结构表示实体与实体之间联系的模型,是层次模型的扩展。它可以表示多个从属关系的层次结构,其表现为一种交叉关系的网络结构。如城市交通系统的每一条道路都可能与另外几条道路相联系,这样的联系就需要用网状模型来描述。

网状模型具有以下特征。

- (1) 有一个以上的结点无父结点。
- (2) 至少有一个结点有多个父结点。

网状模型是一种比层次模型更具普遍性的结构,允许多个结点没有父结点,允许结点有多个父结点,此外它还允许两个结点之间有多种联系。因此网状模型可以更直接地去描述现实世界。而层次模型实际上是网状模型的一个特例。

由通用电气公司 1961 年开发成功的 IDS(Integrated Data Store,集成数据存储)是世界上第一个使用网状数据模型的数据库管理系统,也是第一个数据库管理系统。

### 3. 关系模型(Relational Model)

关系模型是目前最重要的一种数据模型,用二维表的形式表示事物之间联系的数据模型称为关系模型,而通过关系模型建立的数据库则称为关系数据库。关系模型是目前数据库产品的主流数据模型,例如,目前流行的数据库产品 IBM DB2、Oracle、Sybase、Microsoft SQL Server 和 Microsoft Access 等都是关系数据库管理系统。数据库领域当前的研究工作也都是以关系方法为基础,因此本书的重点也是关系模型。下面以表 1-1 为例讲解关系模型的基本概念和基础知识。

表 1-1 学生基本信息表

| 学号       | 姓名  | 性别 | 出生日期     | 党员否 | 班级      | 生源 | 入学成绩 | 备注    |
|----------|-----|----|----------|-----|---------|----|------|-------|
| 20080022 | 李中华 | 男  | 05/27/90 | 是   | 会计 0801 | 北京 | 628  | 校三好学生 |
| 20080033 | 章硕  | 男  | 02/20/90 | 否   | 会计 0801 | 天津 | 600  |       |
| 20080044 | 王红  | 女  | 08/15/90 | 是   | 会计 0801 | 河北 | 590  |       |
| 20080055 | 李明燕 | 女  | 05/20/90 | 是   | 会计 0801 | 山西 | 618  |       |
| 20080111 | 郑海  | 男  | 12/10/89 | 否   | 会计 0802 | 山东 | 598  |       |
| 20080112 | 张梅  | 女  | 11/09/89 | 是   | 会计 0802 | 辽宁 | 580  | 市三好学生 |
| 20080113 | 彭惠  | 女  | 03/13/90 | 否   | 会计 0802 | 天津 | 610  |       |
| 20081101 | 张鹏  | 男  | 07/17/90 | 是   | 经济 0801 | 湖南 | 601  |       |
| 20081102 | 江锦添 | 男  | 08/15/90 | 是   | 经济 0801 | 河北 | 632  | 优秀干部  |
| 20081103 | 王晓  | 女  | 09/10/89 | 否   | 经济 0801 | 上海 | 612  |       |

#### 1) 关系模型的基本术语

##### (1) 关系—表

一个关系对应一个二维表,由若干行和列组成,每个关系都有一个名称叫表名。例如,表 1-1 是一张学生基本信息表,表名为“学生基本信息”。

##### (2) 属性—字段

在二维表中,每一列称为一个属性,属性由属性名和属性值构成,在关系数据库中通常将属性称为字段,把属性名称称为字段名。例如,“学生基本信息”表中的“学号”、“姓名”和“性别”等均为字段名,“20080022”、“李中华”、“男”为相对应的字段值。

##### (3) 元组—记录

在二维表中,每一行称为一个元组或一条记录,即表中每个属性取一个值所形成的一组数据称为一个记录。例如,“学生基本信息”表中的第一行数据“20080022、李中华、男、05/27/90、是、会计 0801、北京、628、校三好学生”称为一个元组或一条记录。表 1-1 中共有 10 条记录。

##### (4) 域

在二维表中,属性的取值范围称为域。例如,“学生基本信息”表中“姓名”字段的取值范围是文字字符,“性别”字段只能从“男”、“女”两个汉字中取其一,“入学成绩”的取值为数值型。

### (5) 主键

在一个二维表中,某一个属性可以用来唯一地标识一条记录,就称该属性为主键。例如,学生基本信息表中的“学号”字段可以作为主键,因为学号是学生的唯一标识,没有也不允许有重复值。而“姓名”、“性别”和“出生日期”等等字段则无法作为识别一个学生的唯一标识,因此不能作为主键。

### 2) 关系模型的操纵与完整性约束

关系模型的操作主要包括查询、插入、删除和修改数据。这些操作必须满足关系的完整性约束条件。关系的完整性约束条件包括三大类,即实体完整性、参照完整性和用户定义的完整性,其具体含义将在后面介绍。

### 3) 关系模型的基本演算

关系数据库主要支持 3 种基本关系运算,它们是选择、投影和连接。

(1) 选择: 从一个关系中找出满足给定条件的记录行的操作称为选择运算。选择运算是从行的角度对关系进行的筛选,选择运算的结果构成新的关系,其关系模式不变,而其中的记录是原关系的一个子集。例如,从学生基本信息表中筛选“经济 0801”班的学生信息的操作就是一个选择运算。

(2) 投影: 从一个关系中找出若干字段的操作称为投影运算。投影是从列的角度对关系进行筛选或重组,进而构成新的关系,其关系模式往往是原关系模式的子集。例如,从学生基本信息表中筛选“学号、姓名、性别和班级”4 列数据的操作就是一个投影运算。

(3) 连接: 将两个关系中的记录按一定条件连接,组成一个新的关系的操作称为连接运算。

总之,利用关系的选择、投影和连接运算,可以灵活地在一个或多个关系中访问所需数据,从而实现对关系数据库的各种操作。

### 4) 关系模型的优缺点

关系模型具有下列优点。

(1) 关系模型与非关系模型不同,它是建立在严格的数学概念的基础上的。

(2) 关系模型的概念单一,无论实体还是实体之间的联系都用关系表示。对数据的检索结果也是关系(表),所以其数据结构简单清晰,用户易懂易用。

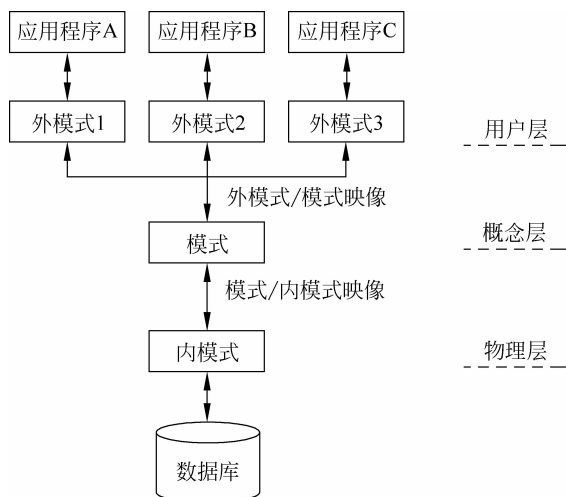
(3) 关系模型的存取路径对用户透明,从而具有更高的数据独立性、更好的安全保密性,也简化了程序员的工作和数据库开发建立的工作。

关系模型最主要的缺点是由于存取路径对用户透明,查询效率往往不如非关系数据模型,因此为了提高性能,必须对用户的查询请求进行优化,增加了开发数据库管理系统的难度。

## 1.3 数据库系统的体系结构

为有效地组织和管理数据,提高数据库的逻辑独立性和物理独立性,人们为数据设计了一个严谨的体系结构。1978 年美国国家标准协会数据库管理系统研究小组提出了标准化建议,将数据库结构分为三级模式结构和两级映像。三级模式结构包括外模式、模式和内模式,外模式对应用户层,面向具体的数据用户;模式对应概念层,面向数据库建立维护人员;

内模式对应物理层,面向系统程序员。两级映像分别是外模式/模式映像、模式/内模式映像,两级映像确保数据库系统中数据具有较高的逻辑独立性和物理独立性。图 1-9 为数据库系统体系结构框图。



### 1.3.1 数据库系统模式的概念

在数据模型中有“型”和“值”的概念。型是指对某一类数据的结构和属性的说明,值是型的一个具体赋值。例如,学生记录定义(学号,姓名,性别,出生日期,党员否,班级,生源,入学成绩,备注)是记录型,而(20080022,李中华,男,05/27/90,是,会计 0801,北京,628,校三好学生)是该记录型的一个记录值。

模式是数据库中全体数据的逻辑结构和特征的描述,它仅仅涉及型的描述,不涉及具体的值。模式的一个具体值称为模式的一个实例。同一个模式可以有很多实例。模式是相对稳定的,而实例是相对变动的,因为数据库中的数据是在不断更新的。模式反映的是数据的结构及其联系,而实例反映的是数据库某一时刻的状态。

虽然实际的数据库管理系统产品种类很多,它们支持不同的数据模型,使用不同的数据库语言,建立在不同的操作系统之上,数据的存储结构也各不相同,但它们在体系结构上通常都具有相同的特征,即采用三级模式结构和两级映像。

### 1.3.2 数据库系统的三级模式结构

#### 1. 外模式(External Schema)

外模式也称用户模式,是用户与数据库系统的接口,是用户用到的那部分数据的描述。外模式通常是逻辑模式的子集,由若干个外部记录型组成。由于每个用户均需使用一个外模式,而用户对数据的需求往往不同,因此多个用户可以有多个不同的外模式。每个外模式可为多个用户使用,而同一个用户也可以使用不同的外模式,但每一个应用程序只能使用一

个外模式。

外模式是保证数据库安全的一个有力措施,因为每个用户只能看见和访问所对应的外模式中的数据,数据库中的其余数据则是不可见的。DBMS 提供外模式描述语言来严格定义外模式。

## 2. 模式 (Schema)

模式也称逻辑模式或概念模式,是数据库中全体数据的逻辑结构和特征的描述,是所有用户的公共数据视图。它是数据库系统模式结构的中间层,既不涉及数据的物理存储细节和硬件环境,也与具体的应用程序和所使用的应用开发工具或程序设计语言(如 Visual C++)无关。

模式实际上是数据库数据在逻辑级上的视图。一个数据库只有一个模式。数据库模式以某一种数据模型为基础,统一综合地考虑了所有用户的需求,并将这些需求有机地结合成一个逻辑整体。定义模式时不仅要定义数据的逻辑结构,例如,数据记录由哪些数据项构成,数据项的名字、类型、取值范围等,而且要定义数据之间的联系,定义与数据有关的安全性、完整性要求。DBMS 提供模式描述语言来严格地定义模式。

## 3. 内模式 (Internal Schema)

内模式也称存储模式,它是数据物理结构和存储方式的描述,是数据在数据库内部的表示方式。例如,记录的存储方式是顺序存储、按照 B 树结构存储还是按 hash 方法存储;索引按照什么方式组织;数据是否压缩存储,是否加密等。

内模式是最低层,反映了数据在计算机物理结构中的实际存储形式。例如,在关系数据库系统中一个表可能由多个文件组成,如数据文件、索引文件等。

一个数据库只有一个内模式。DBMS 提供内模式描述语言来严格地定义内模式。

### 1.3.3 三级模式结构的二级映像

数据库系统的三级模式是数据在三个层次(用户层、概念层和物理层)的抽象,它把数据的具体组织留给数据库管理系统来管理,使用户能逻辑地、抽象地处理数据,而不必关心数据在计算机中的具体表示方式与存储方式。为了能够在内部实现这三层次的联系和转换,数据库管理系统在这三级模式之间提供了两级映像:外模式/模式映像和模式/内模式映像。

#### 1. 外模式/模式映像

模式描述的是数据的全局逻辑结构,外模式描述的是数据的局部逻辑结构。对应于同一个模式可以有任意多个外模式。对于每一个外模式,数据库系统都有一个外模式/模式映像,它定义了该外模式与模式之间的对应关系。这些映像定义通常包含在各自外模式的描述中。

当模式改变时(如增加新的关系、新的属性、改变属性的数据类型等),由数据库管理员对各个外模式/模式的映像作相应改变,可以使外模式保持不变。应用程序是依据数据的外模式编写的,从而使应用程序不必修改,保证了数据与程序的逻辑独立性,简称数据的逻辑

独立性。

## 2. 模式/内模式映像

数据库中只有一个模式,也只有一个内模式,所以模式/内模式映像唯一的,它定义了数据库全局逻辑结构与存储结构之间的对应关系。例如,说明逻辑记录和字段在内部是如何表示的。该映像定义通常包含在模式描述中。当数据库的存储结构改变了(如选用了另一种存储结构),由数据库管理员对模式/内模式映像作相应改变,可以使模式保持不变,从而使应用程序也不必改变,保证了数据与程序的物理独立性,简称数据的物理独立性。

在数据库的三级模式结构中,数据库模式即全局逻辑结构是数据库的中心与关键,它独立于数据库的其他层次。因此设计数据库模式结构时应首先确定数据库的逻辑模式。数据库的内模式依赖于它的全局逻辑结构,但独立于数据库的用户视图即为外模式,也独立于具体的存储设备。它是将全局逻辑结构中所定义的数据结构及其联系按照一定的物理存储策略进行组织,以达到较好的时间与空间效率。数据库的外模式面向具体的应用程序,它定义在逻辑模式之上,但独立于存储模式和存储设备。当应用需求发生较大变化,相应外模式不能满足其视图要求时,该外模式就得做相应改动,所以设计外模式时应充分考虑到应用的扩充性。

## 1.4 数据库设计

数据库系统是在计算机系统中引入数据库后构成的系统,由数据库、数据库管理系统、数据库应用系统、数据库管理员和用户 5 部分组成。数据库系统设计主要是针对某个具体的应用环境和具体问题,进行信息抽象、构造优化概念模型,设计最佳的数据库逻辑模式和物理结构,并以此为依据建立数据库及其应用系统。也就是说数据库系统设计不但要设计数据库,还要设计基于数据库的应用系统,即设计整个数据库系统。

在数据库系统设计过程中,主要是数据库设计和数据库应用系统设计,其中数据库设计是开发数据库应用系统的核心和关键,本节重点介绍数据库设计基本方法和步骤。

### 1.4.1 数据库设计的任务

数据库设计的任务是指对一个给定的应用环境,设计优化的数据库的逻辑结构和物理结构,并据此建立数据库及其应用系统,使之能够有效地存储、管理和利用数据,满足各种用户的应用需求。应用需求包括数据需求、处理需求、安全性需求和完整性需求。数据库设计的任务就是把现实世界的数据库根据各种应用处理的要求,加以合理地组织,使之能满足硬件和操作系统的特性,利用已有的 DBMS 来建立实现系统目标的数据库。数据库设计的优劣直接影响信息系统的质量和运行效果。因此设计一个结构优化的数据库是对数据进行有效管理的前提和正确利用信息的保证。

### 1.4.2 数据库设计的内容

数据库设计通常是在一个通用的 DBMS 支持下进行的,即利用现有的 DBMS 作为开发

基础。数据库设计的内容包括数据库的结构设计和数据库的行为设计两个方面。

结构特性的设计是指确定数据库的数据模型。数据模型反映了现实世界的的数据及数据间的联系,要求在满足应用需求的前提下,尽可能减少冗余,实现数据共享。它包括数据库的概念模型设计、逻辑模型设计和物理模型设计,即设计数据库的框架和数据库结构。数据库结构是静态的、稳定的,一旦形成后通常情况下是不需要改变的,所以结构设计又称静态模型设计。

数据库的行为设计是指研究数据库用户的行为和动作。在数据库系统中,用户的行为和动作是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数据库的应用程序设计,即设计应用程序、数据操作、事务处理等。行为设计是动态的,因此又称为动态模型设计。开发数据库应用程序是一项庞大的软件工程,数据库设计的各阶段可以和软件工程的各阶段对应起来,软件工程的某些方法和工具同样适用于数据库应用程序设计。两者的区别在于:软件工程中比较强调行为特性的设计;在数据库系统中,由于数据库模型是一个相对稳定的并为用户共享的数据为基础,所以数据库系统中更强调对于结构特性的设计,并与行为特性的设计结合起来。

### 1.4.3 数据库设计的方法

数据库设计是一项工程技术,需要科学理论和工程方法作为指导,否则数据库工程的质量难以保证。为了数据库设计更合理、更有效,人们努力探索提出了各种各样规范化的数据库设计方法。目前大多数设计方法都起源于新奥尔良方法,并在设计过程中采用一些辅助方法加以实现。下面简要介绍几种典型的数据库设计方法。

#### 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的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特定用户的应用,确定实体、属性和实体之间的联系,产生反映数据库整体的概念模型,然后再将此概念模型转换成基于某一特定的 DBMS 的逻辑模型。E-R 方法的基本步骤是:

- (1) 确定实体类型;
- (2) 确定实体联系;
- (3) 画出 E-R 图;
- (4) 确定属性;
- (5) 将 E-R 图转换成某个 DBMS 可接受的逻辑模型;
- (6) 设计记录格式。

#### 2. 基于 3NF 的的数据库设计方法

基于 3NF 的数据库设计方法的基本思想在需求分析的基础上,确定数据库模式全部属性之间的依赖关系,将它们组成一个单一的关系模式,然后将其投影分解,消除其中不符合 3NF 的约束条件,将其规范成若干个 3NF 关系模式的集合。



### 3. 计算机辅助数据库设计方法

计算机辅助数据库设计方法是数据库设计趋向自动化的一个重要方面,其基本设计思想是在人工设计数据库的基础上提供一个交互式过程,一方面,充分利用计算机的速度快、容量大、自动化程度高的特点,完成比较规则、重复性大的工作;另一方面,充分发挥设计者的技术和经验,做出一些重大的决策,人机结合且互相渗透,帮助设计者更好地设计数据库。目前数据库设计工具已经实用化和产品化,例如,Oracle 公司和 Sybase 公司推出的数据库设计工具软件 Oracle Designer 和 PowerDesigner,这些工具软件可以自动地或辅助设计人员完成数据库设计过程中的很多任务。

#### 1.4.4 数据库设计的步骤

按照规范化的数据库设计方法,可以把数据库设计过程分为以下 6 个设计阶段: 数据需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库实施、数据库运行和维护。数据库设计步骤如图 1-10 所示。其中前两个阶段是面向用户的应用需求,面向具体问题,中间两个阶段面向数据库管理系统,最后两个阶段面向具体的实现方法。

在数据库设计开始之前,首先必须选定参加设计的人员,包括系统分析人员、数据库设计人员和程序员、用户和数据库管理员。系统分析和数据库设计人员是数据库设计的核心人员,他们将自始至终参与数据库设计,他们的水平决定了数据库系统的质量。用户和数据库管理员在数据库设计中也是举

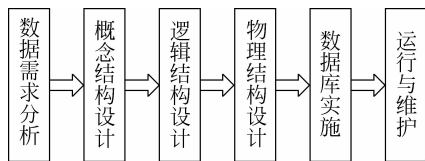


图 1-10 数据库设计步骤

足轻重的,他们主要参加需求分析和数据库的运行维护,他们的积极参与不但能加速数据库设计,而且也是决定数据库设计质量的重要因素。程序员则在系统实施阶段参与进来,负责编程序和准备软硬件环境。

如果所设计的数据库应用系统比较复杂,还应该考虑是否需要使用数据库设计工具和计算机辅助软件工程(CASE)工具,以提高数据库设计质量并减少设计工作量。下面简要介绍数据库设计步骤的具体内容。

##### 1. 数据需求分析

数据需求分析的目标是准确了解系统的应用环境,了解并分析用户对数据及数据处理的需求。数据需求分析是整个数据库设计过程中最重要的步骤之一,是其余各阶段的基础。

在数据需求分析阶段,要求从各方面对整个组织(机构)进行调研,收集和分析各项应用对信息、处理 and 安全性三方面的需求。信息要求指用户需要从数据库中获得信息的内容与性质,由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。处理要求指用户要完成什么处理功能,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。安全性要求指用户对系统信息的安全性要求等级以及信息完整性的具体要求。具体而言,需求分析阶段包括收集信息、分析信息、确定系统边界、设计数据字典和编写需求说明书等过程。

## 2. 概念结构设计

概念设计阶段的目标是把需求分析阶段得到的用户需求抽象为数据库的概念结构,即概念模型。概念模型独立于特定的数据库管理系统,也独立于数据库逻辑模型,还独立于计算机存储介质上的物理模型。

概念设计是整个数据库设计的关键,是现实世界的第一层面的抽象与模拟,最终设计出描述现实世界且独立于 DBMS 的概念模型。设计概念模型的常用方法是 E-R 方法。概念结构设计分为局部 E-R 图和总体 E-R 图,总体 E-R 图由局部 E-R 图组成;在设计过程中,一般先从局部 E-R 图开始设计,以减小设计的复杂度,最后由局部 E-R 图综合形成总体 E-R 图。

## 3. 逻辑结构设计

概念结构是独立于任何一种数据模型的信息结构。逻辑结构设计任务就是把概念结构设计阶段设计好的基本 E-R 图转换为与 DBMS 支持的数据模型相符合的逻辑结构。

目前 DBMS 产品一般支持关系、网状、层次 3 种模型中的某一种,对某一种数据模型,各个机器系统又有许多不同的限制,提供不同的环境与工具。所以进行逻辑结构设计一般要包括下面 3 步:第一步,将概念结构转换为一般的关系、网状、层次模型;第二步,将转换来的关系、网状、层次模型向特定 DBMS 支持下的数据模型转换;第三步,对数据模型进行优化。

## 4. 物理结构设计

数据库在物理设备上的存储结构与存取方法称为数据库的物理结构,它依赖于给定的计算机系统。为一个给定的逻辑数据模型选取一个最适合应用要求的物理结构的过程,就是数据库的物理设计。数据库的物理设计通常分为两步:第一步,确定数据库的物理结构,在关系数据库中主要指存取方法和存储结构;第二步,对物理结构进行评价,评价的重点是对时间效率、空间效率、维护代价和用户要求进行权衡。如果评价结果满足原设计要求,则可进入到物理实施阶段,否则就需要重新设计或修改物理结构,有时甚至要返回逻辑设计阶段修改数据模型。

## 5. 数据库实施

完成数据库的物理设计之后,设计人员就要使用具体的 DBMS 提供的数据库定义语言和其他实用程序将数据库逻辑设计和物理设计的结果严格描述出来,建立一个具体的数据库,编制调试应用程序,组织数据入库,对数据库组织测试和试运行操作。数据库实施主要包括定义数据库结构、组织数据入库、编制调试应用程序、数据库试运行和整理文档等阶段。

在数据库实施阶段,如果系统性能未达到要求,则需要返回物理结构设计阶段,修改数据库物理结构,也可能还需要返回到逻辑结构设计阶段,修改数据库逻辑结构数据模型,然后重新进行数据库实施阶段的工作。

## 6. 数据库运行和维护

数据库实施阶段完成后,数据库应用系统正式投入使用,开发工作就基本完成。但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,它包括如下主要任务。

### 1) 数据库转储和恢复

数据库的转储和恢复是系统正式运行后最重要的维护工作之一。DBA 要针对不同的应用要求制定不同的转储计划,以保证一旦发生故障能尽快将数据库恢复到某种一致的状态,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

在数据库运行过程中,由于应用环境的变化,对安全性的要求也会发生变化,比如有数据原来是机密的,现在是可以公开查询的了,而新加入的数据又可能是机密的了。系统中用户的保密等级也会改变。这些都需要 DBA 根据实际情况修改原有的安全性控制。

### 3) 数据库性能的监督、分析和改进

在数据库运行过程中,监督系统运行,对监测数据进行分析,找出改进系统性能的方法是 DBA 的又一项重要任务。目前 DBMS 产品均提供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的一系列性能参数值。

### 4) 数据库的重新组织

数据库运行一段时间后,由于记录不断增、删、改,会使数据库的物理存储情况变坏,从而降低了数据库存储空间利用率和数据的存取效率,使数据库性能下降,这时 DBA 就要对数据库进行重新组织,或部分重新组织,即重新安排数据的存储位置,回收垃圾,以改进数据库响应时间和空间利用率,提高数据库系统性能。DBMS 一般都提供重新组织数据库的实用程序,帮助 DBA 重新组织数据库。

## 1.5 数据库应用系统设计

数据库系统设计主要是数据库设计和数据库应用系统设计,数据库设计是开发数据库应用系统的核心和关键,但数据库设计和数据库应用系统设计应当结合起来,因为一个好的数据库结构是应用系统的基础,反过来具体应用需求还要对数据库结构进行有针对性的优化。数据库应用系统设计过程主要包括体系结构设计、安全体系设计、应用程序开发设计、应用程序文档设计和设计评审 5 个阶段,如图 1-11 所示。下面对各个阶段做概要介绍。

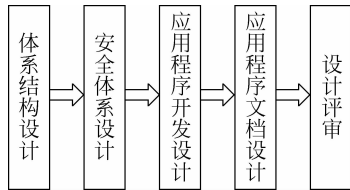


图 1-11 数据库应用系统设计过程

### 1.5.1 应用系统的体系结构设计

前面介绍的数据库系统的体系结构,即三级模式结构,是从数据库管理系统的角度来看,是数据库系统的内部系统结构。如果从数据库的最终用户角度看,一个数据库系统一般包括数据存储层、应用层与用户界面三个层次。数据库存储层由数据库管理系统来承担,完成数据的维护操作;应用层是使用数据库处理与用户有关的各种业务操作,通过程序设计语言来完成;用户界面为用户提供可视化图形操作界面,用于数据库系统与用户之间的交互。

数据库应用系统的体系结构与计算机系统的组成结构有关,通常指在数据库系统中数据存储层、应用层、用户界面层以及网络通信之间的布局与分布关系。按照目前数据库系统的应用与发展,可以分为单机结构、集中式结构、客户机/服务器(Client/Server,C/S)结构和浏览器/服务器(Browser/Server,B/S)结构等。

#### 1. 单机结构

随着 PC 性能的提高,出现了适合于 PC 的单用户数据库系统。单机结构的数据库系统将 PC 上的 DBMS 的功能和数据库应用功能结合在一个应用程序中。尽管它在数据的完整性、安全性、并发性等方面有不少缺陷,但已基本实现 DBMS 应具备的功能。现在较流行的有 Microsoft Access、Visual Foxpro 等。单机结构的数据库系统的优点是功能灵活,系统结构简洁,运行速度快;缺点是共享性、安全性和完整性等功能比较弱。

#### 2. 集中式结构

集中式数据库系统结构是大型主机带多终端的系统,它将操作系统、应用程序、数据库系统等数据和资源均放于大型主机上,而连于主机上的多个终端只是作为主机的输入输出设备。数据存储层和应用层均放在主机上,而用户界面层放在多个终端上。

在集中式数据库系统结构中,所有的处理均由主机完成,故对主机的性能要求较高,这是数据库系统初期最流行的结构,目前仍有应用。但随着计算机网络的普及和硬件价格的不断下降,这种传统的系统已逐渐被客户/服务器结构所取代。

#### 3. 客户机/服务器结构

客户机/服务器(Client/Server,C/S)结构是目前流行的数据库系统结构。采用 C/S 结构的数据库系统分为前端和后端。前端为客户机,必须安装应用程序和工具,主要负责应用处理;后端为服务器,主要负责数据库服务。

客户机/服务器结构的工作机制是,客户端向服务器发出数据库服务请求,服务器将查询结果回传给客户端,不需要传送整个文件,从而减少网络传输量。C/S 结构的本质在于通过对服务功能的分布实现分工服务。每一个服务器都为整个局域网系统提供共享服务,供所有客户机分享;客户机上的应用程序借助于服务器的服务功能以实现复杂的应用功能。在 C/S 结构中,数据存储层处于服务器上,应用层和用户界面层处于客户机上。客户机负责管理用户界面,接收用户数据,处理应用逻辑,生成数据库服务请求,将该请求发送给服务器,同时接收服务器返回的结果,并将结果按一定格式显示给用户。