

第3章

数据库设计和规范化

学习目标

- 理解数据库设计的基本概念
- 掌握数据库设计方法和数据库设计步骤
- 理解函数依赖的基本概念
- 理解范式和规范化
- 掌握关系的分解

3.1 数据库设计概述

3.1.1 数据库设计

在数据库中,数据库应用系统(DBAS)是指能够在数据库管理系统(DBMS)支持下建立的计算机应用系统。从实现技术角度而言,以数据库为基础的图书管理系统、人事管理系统、图书馆管理系统、电子商务系统等都是计算机应用系统。

数据库设计是指对于一个给定的应用环境,构造(设计)优化的数据库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和管理数据,满足各种用户的应用需求,包括信息管理要求和数据操作要求。通常把在数据库中需要存储和管理的数据对象称为信息管理要求,数据操作是指数据对象需要进行的增、删、改、查等操作。

早期的软件工程疏漏了在申请中数据语义的分析和抽象,传统的数据库设计比较重视研究数据模型及建模方法,往往忽视了数据库设计中结构特性设计与行为特性设计相结合。其中,结构特性设计是指数据库的总体框架设计,行为特性设计是指实现用户事务处理操作的应用程序设计。用户一般通过应用程序来访问数据库并运行数据库操作。

数据库设计的目标是为用户提供高效的信息管理系统,为应用系统提供一个良好的运行环境。一个优良的数据库设计是应用系统的基础,是信息系统开发和建设的重要组成部分,在实际的系统开发中,数据库结构与应用系统密不可分、相辅相成。

优良的数据库设计与糟糕的数据库设计对比如表 3-1 所示。

表 3-1 数据库设计对比

优良的设计	糟糕的设计
数据的访问效率高	数据的访问效率低
减少数据维护异常	存在数据增、删、改、查、统计等异常
存储空间利用率高	浪费大量存储空间
数据冗余少	存在大量数据冗余

数据库建设是硬件、软件和干件(即技术与管理)的结合。一个好的数据库建设体现在“三分技术,十分管理,十二分基础数据”。即数据库设计要以开发技术为基础,同时重视数据库建设项目、企业、应用部门业务的管理,数据的收集、整理、结合和不断更新是数据库设计中的重要环节。

3.1.2 数据库设计方法和技巧

之前数据库主要采用手工试凑法,但其设计过程缺乏科学理论依据和工程原则方法的支持,这与工程质量和设计人员的经验和水平有直接关系,导致数据库的正式运行阶段经常出现各种问题,增加了维护代价。经过人们的努力探索,如今主要采用以下几种规范设计法设计数据库。

- 基于 3NF(第三范式)的设计方法。
- 新奥尔良(New Orleans)设计方法。
- 基于 E-R 模型的数据库设计方法。
- 计算机辅助数据库设计方法。
- 视图模型化及视图汇总设计方法。
- 关系模式的设计方法。
- 基于抽象语法的设计方法。

以上数据库设计法中较著名的是新奥尔良设计方法,它将数据库设计分为四个阶段,即需求分析、概念设计、逻辑设计和物理设计。S. B. Yao 等又将数据库设计分为五个步骤,后又有 I. R. Palmer 把数据库设计作为若干步骤的过程,并采用一些辅助设计实现每一步过程。

其中,基于 E-R 模型数据库设计方法是逻辑阶段考虑采用的有效方法,基于 3NF 的设计方法在概念设计阶段广泛采用。以下各设计方法都是数据库设计不同阶段支持实现的具体技术和方法。规范设计法的基本思想是“过程迭代,逐步求精”的过程。

3.1.3 数据库设计步骤

数据库设计方法在设计步骤上难免存在差别,通过分析、对比和整合各种常见的数据库规范设计方法,数据库设计可分为以下六个阶段。

1. 需求分析阶段

需求分析阶段要求设计者对用户需求进行了解和分析,双方密切合作,明确系统总体设计方案,包括确定和分配数据与处理由人工完成、确定人机接口界面。需求分析是整个设计

过程的基础,此阶段的设计结果将直接影响后面各个阶段的实施效率。

2. 概念结构设计阶段

概念结构是对现实世界的抽象。概念结构设计是将需求分析得到的用户需求建立抽象的信息模型(即概念模型),它是现实世界的模型,易与不熟悉计算机的用户交流,是整个数据库设计的关键,可用 E-R 图完成概念设计。

3. 逻辑结构设计阶段

逻辑结构设计阶段能用某个具体的 DBMS 实现用户需要,将概念结构转换相应的数据模型,并根据用户处理要求、安全性考虑,在基本表的基础上建立必要的视图,并对数据模型进行优化。

4. 数据库物理设计阶段

数据库物理设计阶段对于给定的逻辑数据模型选择一个高效的、最适合的物理结构(数据库的物理结构主要指数据库的存储结构和存储方法),对物理结构进行评价,评价的重点为时间和空间效率。

5. 数据库实施阶段

数据库实施阶段使用 DBMS 提供的数据库语言、工具及宿主语言,结合以上阶段的设计结果建立相应的用户数据库结构,编制、调试应用程序,组织数据入库,并进行数据库测试运行。

6. 数据库运行和维护阶段

数据库进入运行阶段,标志着设计的基本完成和维护工作的开始,对系统的性能指标进行评价、测试、调整等维护工作是一项长期的任务,是设计工作的延续和改进。

设计一个完善、优良的数据库应用系统需要对以上六个阶段进行不断重复。

3.2 数据库系统规划

3.2.1 系统规划的任务

系统规划的任务包括确定明确的功能和性能,要求设计者和用户密切合作、充分沟通,共同分析和收集数据管理信息;不仅要满足用户当前和近期需求,还要对长期发展可能存在的问题有相应的处理方案;要明确系统的名称和范围,确定系统边界;明确数据库所管理系统所需的数据将覆盖哪些部门、岗位,数据来自何处,信息处理完毕后输出何种信息到何处等;明确资源的分配,包括人员、资金、设施等的分配;明确预算系统所需资源和资金;明确系统实施计划和实施进度;分析设计数据库的必要性和可行性,预估系统能实现的经济效益和社会效益等。

3.2.2 系统规划的成果

系统规划的成果是系统的规划报告,主要内容如下。

- (1) 系统简述: 包括系统名称、系统任务、系统范围等,要求语言精练。
- (2) 系统功能说明: 设计系统的主要功能、各个分功能模块,并逐层分解,以求功能明确具体。
- (3) 所需资源: 对所需人员、资金、设施的现实情况、需求情况和落实情况进行说明。
- (4) 成本预算: 对估算的总成本分配到各个资源、工作细目中,从而建立预算、标准和检测系统的过程。通过此过程可对系统的成本进行衡量和管理,以事先弄清问题,及时采取纠正措施。其内容涉及各个阶段所需的人力资源,包括各个人员所花费的时间及费用和必要的物质资源消耗。
- (5) 效益评估: 预估系统能实现的经济效益和社会效益等。
- (6) 可行性分析: 包括技术可行性、系统开发可行性和运行环境可行性等分析。

3.3 需求分析

3.3.1 需求分析的主要任务

- (1) 详细调查数据库系统的现实对象(如用户、组织、部门、企业等,是利用数据库进行管理信息的单位)的工作概况、业务流程,每个岗位的职责、每个环节的步骤及各种人员在整个系统活动过程中的作用。
- (2) 充分了解原手工系统或计算机系统,以明确用户的各种需求,并确定新系统的功能。
- (3) 掌握系统中所要存储的数据信息的产生、存档或消亡的过程,并了解其存储特点及生命周期,充分考虑今后可能的扩充和改变。
- (4) 主要目标是编写需求说明书,包括数据流图、建立的数据字典等。

数据流图(Data Flow Diagram,DFD)是描述实际业务流程及业务与数据联系的一种图形表示法。数据流图表示了数据与处理关系及其数据流动方向,如图 3-1 和图 3-2(以办理取款手续为例)所示。

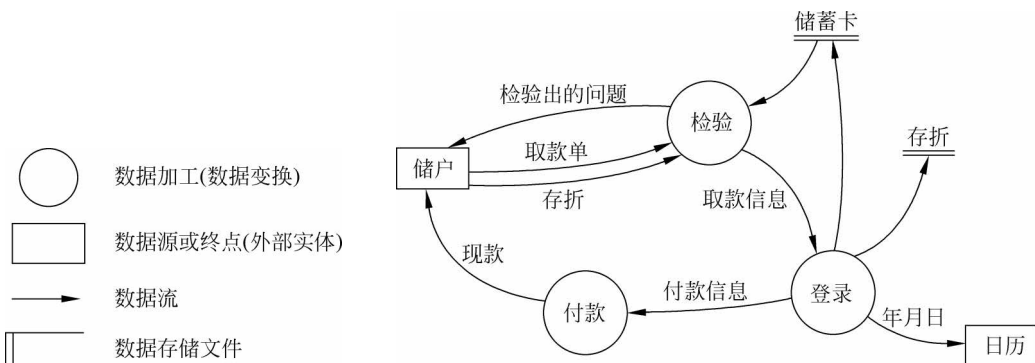


图 3-1 数据流图中的主要数据元素

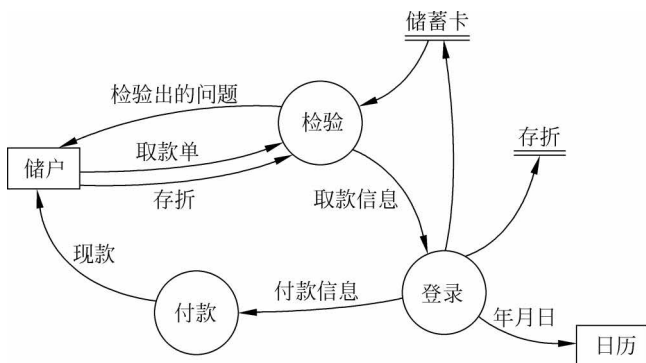


图 3-2 取款手续数据流图

3.3.2 需求分析的工作和方法

需求分析可使系统开发人员根据实际情况制订和调整方案。下面对需求分析的具体工作进行描述。

1. 制订需求分析计划

需求分析人员首先要制订工作计划,如工作时间、工作地点,需要哪些用户协助,需要哪些开发人员一同工作等。不断反思、总结工作进度和工作成果,并根据实际情况做出相应的调整。

2. 选择适当的需求分析方法和需求分析工具

选择最合适的数据库需求分析方法,如原型化分析方法、结构化分析方法等。同时,为提高系统需求分析效率,需求分析人员应尽量使用系统需求分析工具。

3. 调查现存系统

需求分析人员应认真收集、分析现存系统中的数据,避免有遗漏或错误。

4. 充分与用户沟通,理解用户的数据管理内容及目标

需求分析人员应充分了解用户需求,充分尊重用户意见,通过及时沟通了解用户的想法,即用户希望系统实现哪些功能模块,尽可能满足用户需求,并提出自己的建议,说明哪些要求可以实现,解释一些要求为什么不能实现。理解用户的数据管理内容及目标,可以帮助设计人员设计出用户满意的目标系统。

5. 分析并确认系统需求,需要变更立即联系

消除现存系统中的数据中错误、冗余、不完整、不准确等现象,可通过修改数据结构、合并或分解数据等方式,获得准确需求,减少甚至消除需求变更。

6. 撰写数据库需求说明书

需求说明书是需求分析的目标,它完整、清晰、准确地描述了系统的各种需求。用户可同需求分析人员一起反复修改,不断完善需求说明书。

7. 评审需求说明书

评审的主要内容包括系统需求是否与用户需求一致、内容是否齐全、所有图表是否合理、解释是否充分等。为保证评审质量,应由需求分析人员、用户、系统设计和测试人员一同评定。根据评定结果修改、完善需求说明书,最终获得共同批准的系统需求说明书。

如果用户缺少计算机知识,在表述需求时不能明确计算机能达到自己的哪些需求,在需求分析过程中需要不断确定用户需求,解释哪些需求可以被计算机实现,哪些需求不能或不应该被实现。而如果设计人员缺少用户的专业知识,则容易误解用户需求,所以需求分析人员必须耐心、细致地与用户沟通,同时根据不同的现状使用不同的调查方法。在需求分析阶

段常用的调查方法有以下六种。

- 跟班作业。通过亲自参与企业业务工作来了解业务活动情况。此种方法可帮助需求分析人员准确了解用户需求,但耗时耗力。
- 调查会。通过与用户开调查会来了解用户活动和用户需求。
- 专人介绍。通过专业人士的介绍来了解业务流程等。
- 询问。对某些调查中不确定、不理解的问题可以找专人询问。
- 设计调查表请用户填写。若调查表设计合理,站在设计者的专业角度对用户需求进行剖析,易于被用户接受。
- 查阅记录。通过查阅原计算机或手工系统查有关的数据记录。

3.3.3 数据字典

数据字典(Data Dictionary,DD)是进行数据收集和分析后所获得的成果;是对数据项、数据结构、数据流、数据存储、处理逻辑、外部实体等进行定义和描述,而不是数据本身,其目的是对数据流程图中的各个元素做出详细说明。简言之,数据字典是描述数据的信息集合,在需求分析阶段建立数据字典,在数据库设计过程中不断完善。数据字典由数据项、数据结构、数据流、数据存储和处理过程五部分组成。

1. 数据项

数据项是不可再分的数据单位,数据项描述包括数据项名、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,数据库之间的联系。其中,取值范围、与其他数据项的逻辑关系(如该数据项等于其他几个数据项的平均值、该数据项只等于某两数据项的和等)定义了数据的完整性约束条件,可作为设计数据检验功能的依据。

可根据关系规范化理论,把数据项之间的联系用数据依赖的概念分析表示。即按照实际语义写出各个数据项之间的数据依赖,可作为数据库逻辑设计阶段中数据模型优化的依据。

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。数据结构描述包括数据结构名、含义、组成成分(数据项或数据结构)等。

3. 数据流

数据流是数据结构在系统内传输的路径。数据流描述包括数据流名、说明、数据流来源,数据流去向、组成成分(数据结构)、平均流量、高峰期流量等。其中,数据流来源是要说明该数据流来自哪个过程;数据流去向是要说明该数据流要到哪个过程;平均流量是要说明在单位时间(每天、每周、每月等)内的传输次数;高峰期流量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。它不仅可以是计算机文档,还可以是手工文档或凭单。数据存储描述包括数据存储名、说明、编号、输入的数据流、输出的数据流、组成成分(数据结构)、数据量、存取频度、存取方式等。其中,输入的数据流要说明其来源;输出的数据流要说明其去向;存取频度指单位时间(每时、每天或每周等)存取次数及每次存取的数据量等信息;存取方式中要指明是顺序检索还是随机检索,是批处理还是联机处理,是检索还是更新等。

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,处理过程描述包括处理过程名、说明、输入的数据流、输出的数据流、处理(简要说明)等。其中,简要说明指应该说明该处理过程的功能及相应处理要求。功能是指该处理过程的目的。处理要求是指处理读要求,如单位时间内处理多少事物、通过多少数据量、对响应时间的要求等。这些处理要求可作为数据库物理设计的输入及性能评价的依据。

3.4 概念结构设计

3.4.1 概念模型

概念模型设计即完成概念结构的设计。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。概念模型是面向用户和现实世界的数据库模型,是独立于支持数据库的 DBMS,它比数据模型更独立于计算机、更抽象,从而更加稳定,它主要用来描述一个单位的概念化结构。采用概念模型,数据库设计人员可在设计的概念结构设计阶段,将主要精力放在分析和描述现实世界上,而将涉及 DBMS 的一些技术性问题推迟到后面的设计阶段去考虑。

概念模型的特点如下。

(1) 现实世界的一个真实模型,能准确描述事物与事物之间的联系,表达用户对数据处理的需求,能真实、充分地反映现实世界。

(2) 易于理解。数据库设计成功的关键是用户的积极参与,可以用概念模型在数据库管理员、系统开发人员和不熟悉计算机的用户之间交换意见。

(3) 易于更改,容易对概念模型更新和修改,以适应用户需求和环境的变化。

(4) 易于转换为各种数据模型,如关系模型、网状模型、层次模型等。

3.4.1.1 数据抽象

常用数据抽象分为如下三类。

(1) 分类:定义了某一类概念作为现实世界中一组对象的类型,抽象了对象值与型之间的联系,可用 is member of 判定。如张三、李四都是学生,其与“学生”构成分类关系。

(2) 聚集：定义了某一类型的组成成分，具有 is part of 的联系。如学生与学号、姓名、年龄等是聚集的联系。

(3) 概括：定义了类型间的一种子集联系，具有 is subset of 的联系，具有继承性。如专科生和本科生都是学生，且都是集合，因此它们之间是概括的联系。

例如，狗和动物之间是概括的联系，张三家中那只名叫旺财的狗与狗之间是分类的联系，旺财的品种和旺财之间是聚集的联系。

3.4.1.2 概念结构设计方法

概念结构的设计方法有以下四种。

- 自顶向下：首先定义全局概念结构的框架，然后逐步细化，得到全局概念结构。
- 自底向上：首先定义各局部应用的概念结构，然后将它们集成起来，得到全局概念结构。
- 逐步扩张：首先定义最重要的核心概念结构，然后向外扩充，以滚雪球的方式逐步生成其他概念结构，直至总体概念结构。
- 混合策略：将自顶向下方法和自底向上方法相结合，用自顶向下策略设计一个全局概念结构的框架，以它为骨架集成由自底向上策略中设计的各全局概念结构。

常用策略是运用自顶向下方法进行需求分析，运用自底向上方法设计概念结构。

自底向上设计概念结构的步骤如图 3-3 所示，先进行抽象数据并设计局部视图，再集成局部视图，得到全局概念结构。

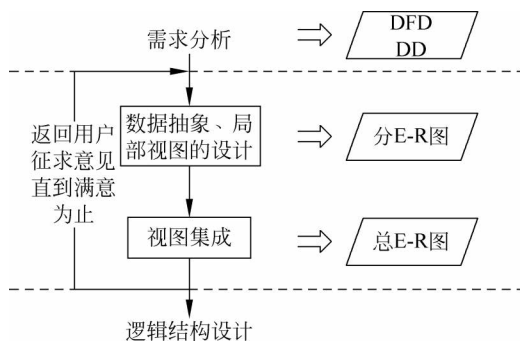


图 3-3 自底向上设计概念结构的步骤

3.4.1.3 概念设计方法(E-R 模型)

E-R(Entity-Relationship)模型即实体-联系图(Entity Relationship Diagram)，提供了表示实体、属性和联系的设计方法，是一种用于描述现实世界的信息模型，实体、属性和联系是它的三个基本元素。具体内容已在第 1 章的 1.3 节详述。在 E-R 模型中，用菱形表示联系，内部写明联系的名称(用动词表示)，并分别将有联系的实体用无向线段连接起来，同时，在无向线段的旁边标明联系的类型(1:1、1:n 或 m:n)。产品销售 E-R 图的设计如图 3-4 所示。

E-R 图的作图步骤：先确定所有的实体、实体包含的属性以及实体之间的联系；再确定实体的关键字，并用下画线在属性上标明主码；最后确定联系类型并标注。

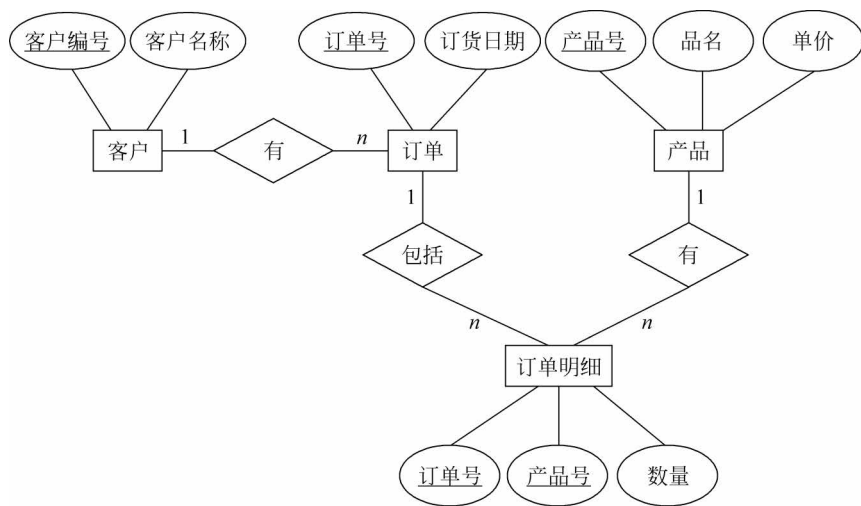


图 3-4 产品销售 E-R 图的设计

3.4.2 E-R 模型设计过程

根据需求分析的结果找出所有数据实体是采用 E-R 方法的数据库概念设计的主要任务和目标。数据实体包括一般实体和联系实体。设计出对应的 E-R 模型,要求首先设计出局部 E-R 模型,然后将各个局部 E-R 图合并起来形成全局 E-R 模型,最后将全局 E-R 模型进行优化,评审后得到最终的 E-R 模型,即概念模型。

注意:

- (1) 为简化 E-R 图的处理,尽量减少实体(集)数量,能作为属性时不要作为实体(集)。
- (2) 属性不能具有需要描述的性质,必须是不可分割的数据项。属性也不能与其他实体有联系,实体之间才有联系。
- (3) 由局部 E-R 图到整体 E-R 图不是简单的叠加。在此过程中,同名实体只能出现一次,并去掉不必要的联系,以便消除冗余。

3.4.2.1 局部 E-R 模型设计

设计局部 E-R 模型步骤如下。

1. 选择局部应用

根据需求分析说明书中描述的整个系统,主要以多层数据流图和数据字典为依据,选择多层数据流图中的一个适当层次的数据流图,使这组图中每个部分对应一个局部应用,并以这一层次的数据流图出发点,设计局部 E-R 图。高层数据流图只能反映系统的概貌,低层数据流图过细,中层数据流图能较客观地反映系统中各局部应用的子系统组成。

2. 逐一设计局部 E-R 图

以数据字典为出发点定义 E-R 图,定义实体和属性,确定实体之间的联系,并对属性进

行分配。

【例 3-1】以学籍管理中的班级管理为例,描述设计 E-R 图的步骤。

步骤一:确定涉及的实体。学籍管理主要涉及的实体有学生、班级、班主任。

步骤二:确定实体间的联系。班级和学生之间是 $1:n$ 的关系,即一个班级有若干名学生,而一名学生只能属于某个班;班主任和学生之间是 $1:n$ 的关系,即班主任可以管理多名学生,一名学生只对应一位班主任;班主任与班级之间是 $1:1$ 的关系,即班主任只对应一个班,一个班只有一位班主任。

步骤三:确定实体的属性。

学生实体的属性:(学号,姓名,所在专业,性别,出生日期)。其中,学号为该实体的主键。

班级实体的属性:(班级号,学生人数)。其中,班级号为该实体的主键。

班主任实体的属性:(职工号,姓名,所在班级,联系电话)。其中,职工号为该实体的主键。

步骤四:确定联系类型的属性。

学生与班级之间的联系包含属性:(学生号,班级号)。

学生和班主任之间的联系包含属性:(学号,职工号)。

班主任与班级之间的联系包含属性:(班级号,职工号)。

步骤五:根据实体和联系类型画出局部 E-R 图,如图 3-5 所示。

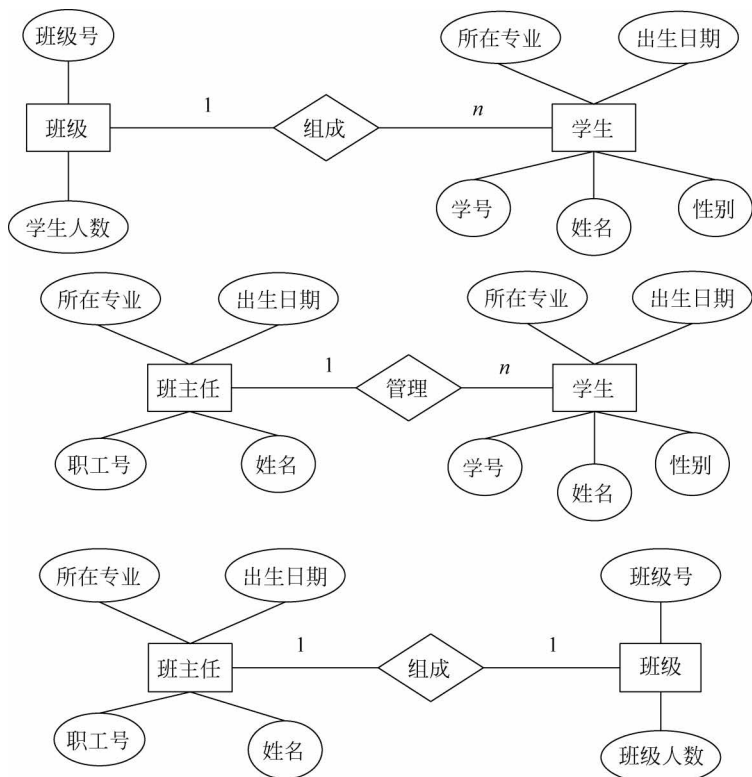


图 3-5 班级管理局部 E-R 图

【例 3-2】以学籍管理中的课程管理为例,描述设计 E-R 图的步骤。

步骤一:确定涉及的实体。学籍管理主要涉及的实体有学生、教师、课程。

步骤二:确定实体间的联系。教师和学生之间是 $1:n$ 的关系,即一个教师可以教授若干名学生,而某门课程学生只能被一名老师教授;教师和课程之间是 $1:n$ 的关系,即一名教师可以讲授多门课程,一门课程只能由一名教师讲授;学生与课程之间是 $m:n$ 的关系,即一名学生可以学习多门课程,一门课程可以由若干名学生学习。

步骤三:确定实体的属性。

学生(学号,姓名,所在专业,年龄,出生日期)。

教师(职工号,姓名,职称)。

课程(课程号,课程名,成绩)。

其中,下画线标注的为该实体的主键。

步骤四:确定联系类型的属性。

学生与教师之间的联系包含属性:(学号,职工号)。

教师与课程之间的联系包含属性:(职工号,课程号)。

课程与学生之间的联系包含属性:(课程号,学号,成绩)。

步骤五:根据实体和联系类型画出局部 E-R 图,为节省篇幅,省略了实体的属性描述后如图 3-6 所示。

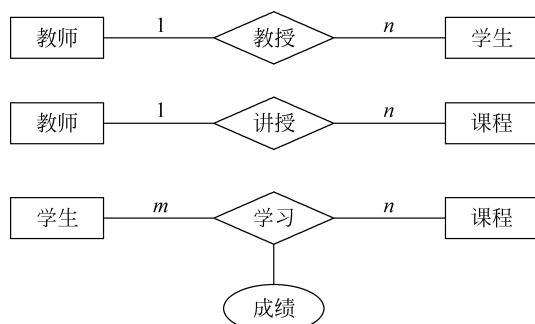


图 3-6 课程管理局部 E-R 图

3.4.2.2 整体 E-R 模型设计

本阶段是将各个局部 E-R 图合并为整体 E-R 图。设计整体 E-R 模型的具体步骤如下。

步骤一:合并。

确定各个局部 E-R 图的公共实体类型,以公共实体类型为单位合并,直到所有相同的实体类型都被合并,得到全局 E-R 图。

步骤二:消除冲突。

各个局部 E-R 图所面向的问题不同,通常由不同的设计人员进行局部 E-R 图设计,因此合并为整体 E-R 图时,可能会存在若干不同之处,称为冲突。

冲突的种类包括属性冲突、结构冲突、命名冲突。

属性冲突包括属性域冲突和属性取值单位冲突。其中属性值的类型不同、取值范围不同和取值集合不同会造成属性域冲突。

结构冲突分为以下三类。

- 同一对象在不同应用中具有不同的抽象。如某事物在某一局部 E-R 图中被当作实体,而在另一局部 E-R 图中被当作属性。
- 同一实体在不同局部 E-R 图中所包含的属性个数或属性排列次序不完全相同。
- 实体之间的联系在局部 E-R 图中呈现不同的联系类型。

命名冲突分为同名异义和异名同义(一义多名)。同名异义指不同意义的对象在不同布局 E-R 图中具有相同的名称。异名同义指同一意义的对象在不同的局部 E-R 图中具有不同的名称。

步骤三:消除冗余。

冗余分为冗余属性和冗余联系。冗余属性是指可由基本数据导出的数据。冗余联系是指可由其他联系导出的联系。因冗余数据和冗余联系容易破坏数据库的完整性,增加数据库维护成本,所以在合并局部 E-R 图时,要根据数据字典和数据流图,参考数据字典中关于数据项之间的逻辑关系,消除不必要的冗余。

【例 3-3】 将某工厂的物资管理、销售管理和劳动人事管理的局部 E-R 图集成为整体 E-R 图,如图 3-7~图 3-10 所示。

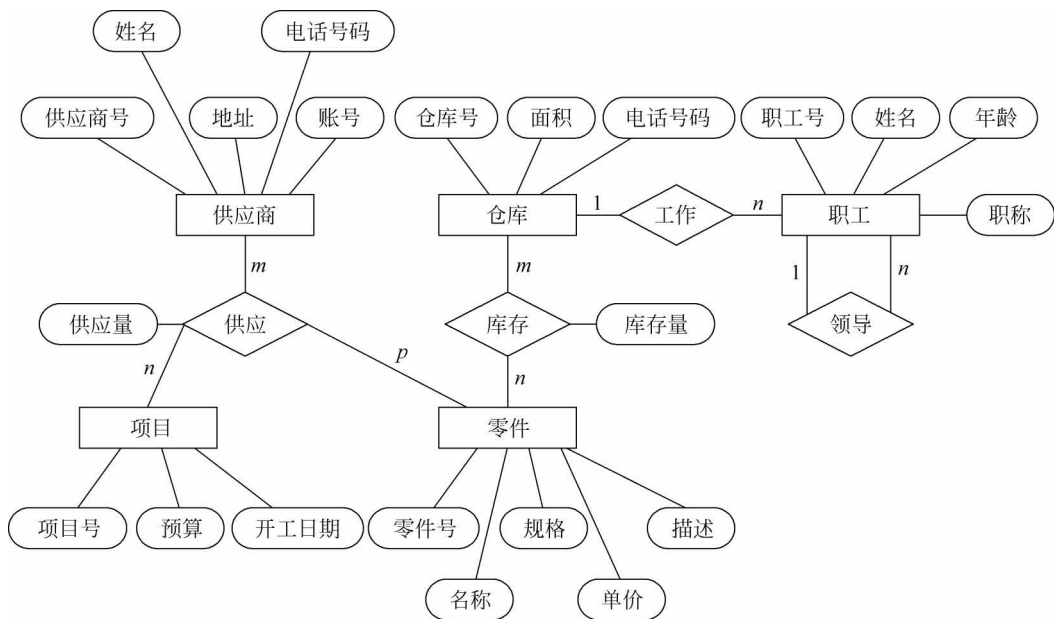


图 3-7 工厂物资管理局部 E-R 图

注:图中省略了实体的属性。

在合并过程中,解决了以下问题。

- (1) 异名同义问题,项目和产品含义相同。
- (2) 取消了库存管理中职工与仓库的工作联系,因为其已经包含在劳动人事管理的部门与职工之间的联系之中。
- (3) 取消了职工之间领导与被领导关系,因为其可由部门与职工(经理)之间的领导关系,以及部门与职工之间的从属关系之中导出。

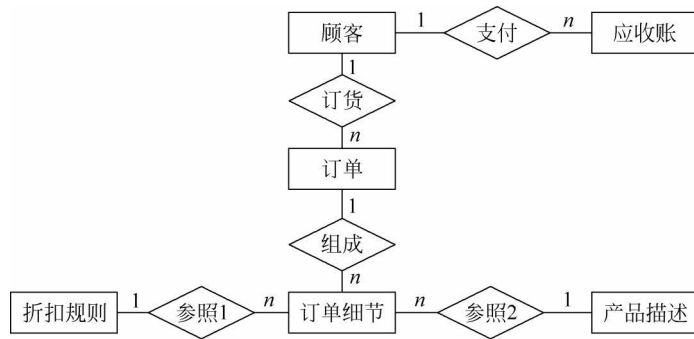


图 3-8 工厂销售管理局部 E-R 图

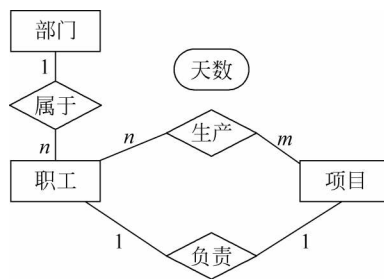


图 3-9 工厂劳动人事管理局部 E-R 图

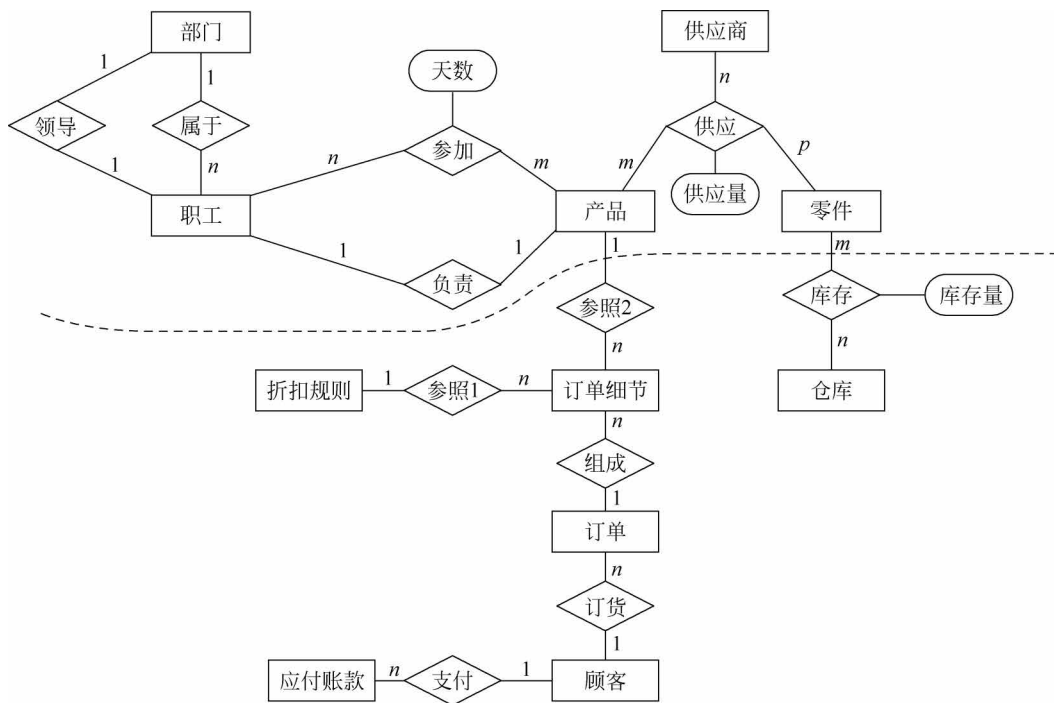


图 3-10 系统的整体 E-R 图

3.4.2.3 验证整体 E-R 模型

整体 E-R 模型设计完成后,应对其进行进一步验证,确保它能够满足下列条件。

- (1) 整体概念模型内部必须具有完整性和一致性,不存在互相矛盾的描述。
- (2) 整体 E-R 图能够准确反映原来的每个局部 E-R 图结构,包括属性、实体及实体间的联系。
- (3) 整体概念模型能够满足需求分析阶段所确定的所有要求。

整体概念模型最终应该提交给用户确认,充分询问和征求用户、有关设计和开发人员的意见,进行评审、修改和调整优化,最终把它确定下来,作为该数据库系统的概念结构,作为进一步设计数据库系统的依据。

3.5 逻辑结构设计

3.5.1 逻辑结构设计的步骤

逻辑结构设计的任务就是把概念结构设计阶段完成的整体 E-R 模型转换为与选用的 DBMS 产品所支持的数据模型相符合的逻辑结构。

通常情况下,逻辑结构的设计应选取最适合相应概念结构的数据模型,然后对所有支持这种数据模型的 DBMS 进行对比,从中选出最合适的 DBMS。目前 DBMS 产品支持的三种模型为关系模型、网状模型、层次模型。而对某一种数据模型,计算机系统有若干不同的限制,提供不同的环境和工具。所以设计逻辑结构时要分三步进行。

- (1) 根据需要,将概念模型转换为一般的关系模型、网状模型或层次模型。
- (2) 将转换好的三种模型向特定的 DBMS 所支持的数据模型转换。
- (3) 对完成的数据模型进行优化。

在数据库的逻辑结构设计完成之后,为进一步提高数据库应用系统的性能,还应该根据实际应用适当地修改、调整数据模型的结构,这就是数据模型的优化。关系数据模型的优化通常以规范化理论为指导,即消除冗余和异常、改善存储效率等,优化时可采用关系模式规范。若优化过程中发现该数据库的逻辑结构有诸多问题,需要返回上一阶段修改或重新设计数据库的逻辑结构。

3.5.2 E-R 模型转换为关系模型

进行数据库的逻辑设计,要将 E-R 模型向关系模型转换。实际上就是将属性、实体及实体之间的联系转换为关系模式。其中,关系都可以用实体和实体之间的联系表示,E-R 图中的属性可以转换为关系的属性。

3.5.2.1 实体的转换

一个实体型向一个关系模式转换,则实体的属性就可作为关系的属性,实体的主码就可作为关系的主码。实体的转换方法为:首先找出主码,主码的表示方法为属性名加下划线;然后再找出属性间的依赖关系,函数依赖关系用箭头线表示;最后将其表示为关系模式。

【例 3-4】 将图 3-11 中的实体转换为关系模式。

司机(司机编号,姓名,电话),司机编号为关系的主码。

车辆(车牌照号,厂家,出厂日期),车牌照号为关系的主码。

车队(车队号,车队名),车队号为关系的主码。

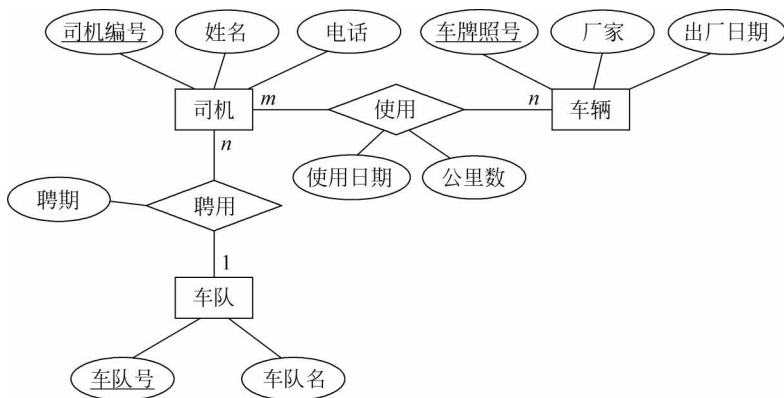


图 3-11 车辆管理 E-R 图

3.5.2.2 实体之间联系的转换

(1) 一个 1:1 联系可以向一个独立的关系模式转换,也可以与任意一端实体所对应的关系模式合并。如果转换为一个独立的关系模式,则关系的属性由该联系相连的各实体的主码以及联系本身的属性转换而来,关系的主码就是每个实体的主码。如果是与联系的任意一端实体所对应的关系模式合并,则需要加入另一个实体的主码和联系本身的属性到关系模式的属性中。

对于图 3-12 所示的 E-R 图,如果将联系与班长一端对应的关系模式合并,则转换为以下两个关系模式。

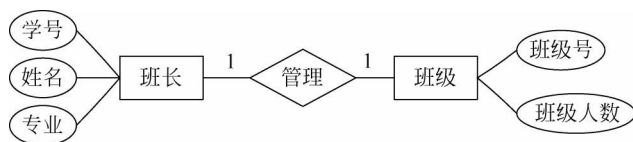


图 3-12 1:1 联系实例

班长(学号,姓名,专业,班级号),其中学号为主码,班级号为引用班级的外键。

班级(班级号,班级人数),其中班级号为主码。

如果将联系与班长一端所对应的关系模式合并,则转换为以下两个关系模式。

班长(学号,姓名,专业),其中学号为主码。

班级(班级号,班级人数,学号),其中班级号为主码,学号为引用班长的外键。

如果将联系向一个独立的关系模式转换,则转换为以下关系模式。

班长(学号,姓名,专业),其中学号为主码。

班级(班级号,班级人数),其中班级号为主码。

管理(学号,班级号),其中学号为主码,学号和班级号为外键。

或：管理(班级号,学号),其中班级号为主码,班级号和学号为外键。

(2) 一个 $1:n$ 联系可以向一个独立的关系模式转换,也可以与 n 端实体所对应的关系模式合并。如果向一个独立的关系模式转换,则关系的属性由与该联系相连的各实体的主码以及联系本身转换而来,主码是 n 端实体的主码。如果与 n 端实体所对应的关系模式合并,将联系本身的属性和 1 端实体的主码加入 n 端对应关系模式中。

例如,对于图 3-11 所示的 E-R 图中车队和司机的 $1:n$ 联系,如果与 n 端实体司机所对应的关系模式合并,则转换为以下两个关系模式。

司机(司机编号,姓名,电话),其中司机编号为主码。

车队(车队号,车队名,司机编号),其中车队号为主码,司机编号为引用车队的外键。

如果将联系向一个独立的关系模式转换,则转换为以下关系模式。

司机(司机编号,姓名,电话),其中司机编号为主码。

车队(车队号,车队名),其中车队号为主码。

聘用(司机编号,车队号,聘期),其中司机编号为主码,司机编号和车队号均为外键。

(3) 一个 $m:n$ 联系可以向一个独立的关系模式转换,关系的属性由与该联系相连的各实体的主码以及联系本身的属性转换而来,各实体主码的组合是该关系的主码或关系主码的一部分。

例如,对于图 3-11 所示的 E-R 图中司机和车辆的 $m:n$ 联系,将其向一个独立的关系模式转换,则转换为如下的关系模式。

使用(司机编号,车牌照号,使用日期,公里数),其中(司机编号,车牌照号)为主码,同时也均为外键。

(4) 三个或三个以上的实体间的一个多元联系可以向一个关系模式转换,关系的属性均由与该多元联系相连的各实体的主码以及联系本身的属性转换而来,各实体主码的组合是关系的主码或关系主码的一部分。

(5) 具有相同主码的关系模式可合并。

例如,对于图 3-10 虚线上方的 E-R 图转换为关系模型,关系的主码用下画线表示。

部门实体对应的关系模式:部门(部门号,部门名,部门领导的职工号,...)。

其中,部门关系模式已经包括领导联系所对应的关系模式,部门领导的职工号是关系的候选码。

职工实体对应的关系模式:职工(职工号,部门号,姓名,职务)。

其中,职工关系模式已经包括属于联系所对应的关系模式。

产品实体对应的关系模式:产品(产品号,产品名,产品组长的职工号,...)。

供应商实体对应的关系模式:供应商(供应商号,姓名,地址,电话号码,账号)。

产品实体对应的关系模式:零件(零件号,名称,规格,单价,描述)。

参加联系对应的关系模式:参加(职工号,产品号,天数)。

供应联系对应的关系模式:供应(产品号,供应商号,零件号,供应量)。

3.6 关系规范化

在以上数据库逻辑结构设计中,已经把 E-R 图向初始的关系模式转换,设计者必须思考应构造几个关系模式,以及每个关系由哪些属性组成,以构造一个好的、合适的数据库模

式,这就是关系数据库逻辑设计问题。

在设计过程中,把现实世界表示为关系模式的问题十分值得重视。而对于如何构造合适的数据模式(逻辑结构)这一问题,通常以关系模型为背景,使用关系规范化理论来指导关系模式的结构,对其进行修改、调整和规范,所以说关系数据库的规范化理论是数据库逻辑设计的一个有力工具。关系数据库设计理论主要包括三个方面,即数据依赖、范式和模式设计方法。

在关系中,属性相当于数学上的变量,属性在一个元组上的取值相当于属性变量的当前值。元组中一个属性或一些属性值对另一个属性值的影响相当于自变量值对函数值的影响。数据依赖研究数据之间的关系,强调一个关系内部属性与属性之间的一种制约关系,是一种语义体现,也是现实世界属性间相互联系的抽象。数据依赖分为函数依赖及多值依赖,本节重点介绍函数依赖。

3.6.1 函数依赖

数据依赖有诸多类型,其中最重要的是函数依赖,它是进行关系分解的依据,已经普遍存在于现实生活中。

例如,在设计职工表时,职工有职工号、姓名、部门、年龄等属性。职工号在此公司是标识职工的唯一属性,因此职工号能决定职工的姓名,可称为姓名依赖于职工号。对于现实来说,如果知道一名职工的职工号,就一定能知道职工的姓名、部门、年龄等其他属性信息,这就是姓名(或部门、年龄)依赖于职工号,是函数依赖。

1. 一般函数依赖

函数依赖的定义:设一个关系为 $R(U)$, X 和 Y 为属性集 U 上的子集,若对于元组中 X 上的每个值都有 Y 上的一个唯一值相对应,则 X 和 Y 之间存在着函数依赖,并称 X 函数决定 Y ,又称 Y 函数依赖于 X ,记作 $X \rightarrow Y$,称 X 为决定因素。

例如,职工名 $\rightarrow$ 部门这个函数依赖只有在该公司没有同名同姓职工的条件下才能成立。如果允许同名同姓的职工存在,则部门就不再依赖于职工名了。如果做出不允许同名同姓的人出现这类对现实世界的强制性规定,则职工名 $\rightarrow$ 部门这个函数依赖成立。如此一来,当插入某个元组时,这个元组上的属性值必须满足规定的函数依赖,若发现有同名同姓的人存在,则拒绝插入该元组。

例如:设一个职工关系为(职工号,姓名,性别,年龄,职务)。

解析:职工号函数决定姓名,或姓名函数依赖于职工号,记作“职工号 $\rightarrow$ 姓名”,职工号为该函数依赖的决定因素。

- 若姓名和职工号一一对应,它们就相互成为决定因素,记作“职工号 $\leftrightarrow$ 姓名”。
- 若在一个关系中, X 不能函数决定另一个属性子集 Y ,则记作 $X \nrightarrow Y$ (X 不能函数决定 Y)。

注意,函数依赖不是指关系模式 R 的某个或某些关系满足的约束条件,而是指 R 的一切关系均要满足的约束条件。

2. 函数依赖的分类

1) 平凡函数依赖

定义: 设一个关系为 $R(U)$, X 和 Y 为属性集 U 上的子集, 若 $X \supseteq Y$ 则称 $X \rightarrow Y$ 为平凡函数依赖。对于任一关系模式, 平凡函数依赖必然成立, 不反映新的语义。若无特别说明, 讨论非平凡函数依赖即可。平凡函数依赖称为函数依赖的自反性规则。

2) 非平凡函数依赖

定义: 设一个关系为 $R(U)$, X 和 Y 为属性集 U 上的子集, 若 $X \rightarrow Y$ 且 $X \not\supseteq Y$, 则称 $X \rightarrow Y$ 为非平凡函数依赖。

例如: 设职工关系(职工号, 姓名, 性别, 年龄, 职务)。

解析: (职工号, 性别) $\rightarrow$ 职工号和 (职工号, 性别) $\rightarrow$ 性别, 都为平凡函数依赖。

(职工号, 姓名) $\rightarrow$ 性别和 (职工号, 姓名) $\rightarrow$ (年龄, 职务), 都为非平凡函数依赖。

3) 完全函数依赖

定义: 设一个关系模式为 $R(U)$, X 和 Y 为属性集 U 上的子集, 若 $X \rightarrow Y$, 不存在一个真子集 X' , 使得 X' 也能够函数决定 Y , 则称 X 完全函数决定 Y , 或 Y 完全函数依赖于 X 。

4) 部分函数依赖

定义: 设一个关系模式为 $R(U)$, X 和 Y 为属性集 U 上的子集, 若 $X \rightarrow Y$, 同时 X 的一个真子集 X' 也能够函数决定 Y , 即存在 $X' \rightarrow Y$, 则称 X 部分函数决定 Y , 或 Y 部分函数依赖于 X 。 X 到 Y 的部分函数依赖也称为局部函数依赖。

例如: 设一个教师任课关系为(教工号, 姓名, 职称, 课程号, 课程名, 课时数, 课时费)。

解析:

(职称, 课程号)完全函数决定课时费。

(教工号, 课程号)部分函数决定姓名。

(教工号, 课程号)部分函数决定课时数。

注: 在关系中, 通常只存储基本数据, 而不存储能计算出来的数据, 如课时数 $\times$ 课时费就是总酬金。

5) 传递函数依赖

定义: 设一个关系为 $R(U)$, X 、 Y 和 Z 为属性集 U 上的子集, 其中存在 $X \rightarrow Y$ 、 $Y \rightarrow Z$, 但 $Y \not\rightarrow X$ 、 $Y \not\supseteq Z$, 则存在 $X \rightarrow Z$, 称此为传递函数依赖, 即 X 传递函数决定 Z , Z 传递函数依赖于 X 。

例如: 设一个学生关系为(学号, 姓名, 性别, 系号, 系名, 系主任名), 一个学生只属于一个系, 每个系对应一个系号。

解析: 学号函数决定系号, 系号函数决定系名和系主任名, 在学生关系中还存在学号 $\rightarrow$ 系名和学号 $\rightarrow$ 系主任名这两个函数依赖, 由于它们都是通过从学号开始的间接函数依赖得到的, 所以系名和系主任名是传递依赖于学号。

3. 函数依赖的规则

下面介绍函数依赖的一些常用规则。设一个关系为 $R(U)$, X 、 Y 、 Z 、 W 是 U 上的子集, 则:

- 自反性: 若 $X \supseteq Y$, 则存在 $X \rightarrow Y$ 。
- 增广性: 若 $X \rightarrow Y$, 则存在 $XZ \rightarrow YZ$ 。
- 传递性: 若 $X \rightarrow Y$ 和 $Y \rightarrow Z$, 则存在 $X \rightarrow Z$ 。
- 合并性: 若 $X \rightarrow Y$ 和 $X \rightarrow Z$, 则存在 $X \rightarrow YZ$ 。
- 分解性: 若 $X \rightarrow Y$, 且 $Y \supseteq Z$, 则存在 $X \rightarrow Z$ 。
- 伪传递性: 若 $X \rightarrow Y$ 和 $WY \rightarrow Z$, 则存在 $WX \rightarrow Z$ 。
- 复合性: 若 $X \rightarrow Y$ 和 $Z \rightarrow W$, 则存在 $XZ \rightarrow YW$ 。
- 自增性: 若 $X \rightarrow Y$, 则存在 $WX \rightarrow Y$ 。

例如: 学生关系(学号, 姓名, 性别, 年龄, 所在专业)。

根据增广性规则, 若学号 $\rightarrow$ 所在专业成立, 则(学号, 性别) $\rightarrow$ (所在专业, 性别)也必然成立。

根据合并性规则, 若学号 $\rightarrow$ 姓名和学号 $\rightarrow$ 性别成立, 则学号 $\rightarrow$ (姓名, 性别)也成立。

根据分解性规则, 学号 $\rightarrow$ (姓名, 性别)也成立, 学号 $\rightarrow$ 姓名和学号 $\rightarrow$ 性别也同样成立。

4. 属性关系中的函数依赖

对于属性之间的三种关系, 并不是每一种关系都存在函数依赖。设 $R(U)$ 是属性集 U 上的关系模式, X 、 Y 是 U 的子集。

(1) 若 X 和 Y 之间是 $1:1$ 关系(一对一关系), 则存在函数依赖 $X \rightarrow Y$ 和 $Y \rightarrow X$ 。如班级和班长之间是 $1:1$ 关系, 则存在班级 $\rightarrow$ 班长, 班长 $\rightarrow$ 班级。

(2) 若 X 和 Y 之间是 $1:n$ 关系(一对多关系), 则存在函数依赖 $Y \rightarrow X$ 。如班长和学生之间是 $1:n$ 关系, 则存在学生 $\rightarrow$ 班长。

(3) 若 X 和 Y 之间是 $m:n$ 关系(多对多关系), 则 X 和 Y 之间不存在函数依赖。如学生和课程之间是 $m:n$ 关系, 它们之间不存在函数依赖。

3.6.2 范式

在创建一个数据库的过程中, 范化是将其转化为一些表的过程, 这种方法可以使从数据库得到的结果更加明确。即关系数据库中的关系必须满足一定的要求, 满足的要求不同则范式不同。这样可能使数据库产生重复数据, 从而导致创建多余的表。范化是在识别数据库中的数据元素、关系, 以及定义所需的表和各表中的项目这些初始工作之后的一个细化的过程。一个低级范式的关系模式通过模式分解可向若干个高级范式的关系模式逐级转换, 这种过程称为规范化。

关系规范化分为六个级别, 从低到高依次为第一范式(1NF)、第二范式(2NF)、第三范式(3NF)、Boyce-Codd 范式(BCNF)、第四范式(4NF)和第五范式(5NF)。通常只要求规范到第三范式就可以了, 并且前三个范式能够保持数据的无损连接和函数依赖性。

1. 第一范式(1NF)

定义: 设一个关系为 $R(U)$, 若 U 中的每个属性都是不可再分的, 或者说都是不被其他属性所包含的独立属性, 则称关系 $R(U)$ 是符合第一范式的, 即 $R \in 1NF$ 。关系实例如图 3-13 所示。

规范化解决方法: 一是增加独立属性, 取消分栏; 二是采用把原关系分解为多个关系。

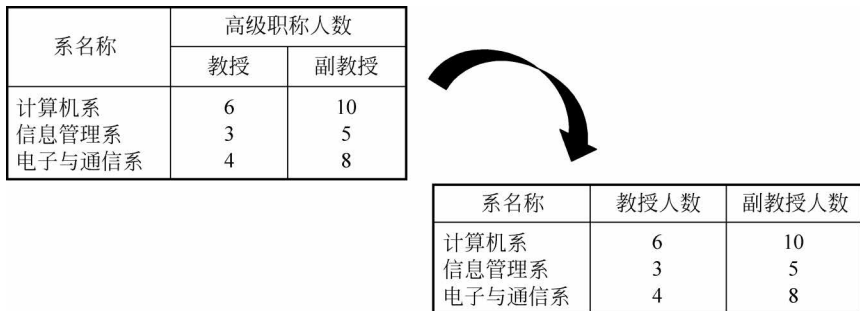


图 3-13 转换为符合第一范式的关系实例

2. 第二范式(2NF)

定义：设一个关系为 $R(U)$ ，满足第一范式，若 R 中不存在非主属性对候选码的部分函数依赖，则称该关系是符合第二范式的，即 $R \in 2NF$ 。

推论：若关系模式 $R \in 1NF$ ，且它的每一个候选码都是单码，则 $R \in 2NF$ 。

【例 3-5】 将表 3-2 所示的学生信息规范化。

表 3-2 学生成绩信息

学号	姓名	性别	所在专业	课程号	课程名	成绩
003014	张三	男	计算机	C01	C 语言	81
004001	李四	男	电子	D04	电路分析	77
004005	王苗	女	计算机	C08	操作系统	69

解析：姓名部分函数依赖于(学号,课程号)，因此表 3-2 符合第一范式但不符合第二范式，需分解为三个关系。

学生(学号,姓名,性别,所在专业)。

课程(课程号,课程名)。

选课(学号,课程号,成绩)。

经过分解，三个关系可以连接后仍得到原关系，且为无损分解和无损连接。

【例 3-6】 说明职工(职工号,姓名,所属部门,项目号,项目名称,项目角色)是否符合第二范式。

解析：不符合第二范式。其中存在三个问题，即插入异常、删除异常和修改异常。

当职工关系中项目名称发生变化时，参与该项目的人员很多，每人都有一条记录信息，若修改项目信息，需对每一个参加该项目的人员信息进行修改，加大了工作量，存在遗漏的可能，同时数据一致性也有可能被破坏。

可把上述职工关系分解成如下三个关系：

职工(职工号,姓名,所属部门)。

参与项目(职工号,项目号,项目角色)。

项目(项目号,项目名称)。

3. 第三范式(3NF)

定义：设一个关系为 $R(U)$ ，满足第一范式，若 R 中不存在非主属性对候选码的传递函数依赖，则称该关系是符合第三范式的，即 $R \in 3NF$ 。

推论 1：如果关系模式 $R \in 1NF$ ，且它的每一个非主属性既不部分依赖，也不传递依赖于任何候选码，则 $R \in 3NF$ 。

推论 2：不存非主属性的关系模式一定为 3NF。

【例 3-7】 修改假定学生关系(学号, 姓名, 年龄, 所在学校, 学校地址, 学校电话)，使其符合 3NF。

解析：(学号)→(姓名, 年龄, 所在学院, 学院地点, 学院电话)和(学号)→(所在学校名)→(学校地址, 学校电话)存在非主属性对候选码的传递函数依赖。

要将学生关系表分为如下关系的两个表。

学生(学号, 姓名, 年龄, 所在学校)。

学校(校名, 地址, 电话)。

4. Boyce-Codd 范式(BCNF)

定义：若一个关系为 $R(U)$ ，它是满足第一范式的，当 R 中不存在任何属性对候选码的传递函数依赖时，则称 R 符合 BCNF，即 $R \in BCNF$ 。

对于满足 Boyce-Codd 范式，也可做这样理解：如果 R 中的所有属性都完全依赖于候选码，或 R 的最小函数依赖集中的所有函数依赖的决定因素都是候选码，即 $R \in BCNF$ 。

【例 3-8】 说明关系模式授课(教工号, 学号, 课程号)是否符合 Boyce-Codd 范式。

解析：每名教师只教授一门课程，每门课程由若干名教师教授，若某一学生选定某门课程，就确定了一名教师；同样地，某名学生选修了某名教师的课程就确定了所选课程的名。即(教工号, 学号)→课程, (学号, 课程号)→教师, 教工号→课程号。

因为(教工号, 学号)和(学号, 课程号)都可以作为候选码，教工号、学号、课程号都是主属性，所以授课关系符合第三范式，但不符合 Boyce-Codd 范式。

可将授课分解为两个关系模式，即(学号, 课程号)和(教工号, 课程号)，它们之中不存在任何属性对候选码的部分函数依赖和传递函数依赖，所以符合 Boyce-Codd 范式。

规范化的方法是进行模式分解，且确保模式分解后原来的语义不被破坏，还要保证原来的函数依赖关系不被丢失。需要注意的是，并不是规范级别越高，模式越好。合理选择数据库模式必须结合应用环境和现实世界的具体情况。一般情况下，关系模式达到 3NF 就可以不再分解。

3.7 物理结构设计

3.7.1 物理设计的内容

数据库的物理设计分为两个部分，首先要确定数据库的物理结构，在关系数据库中主要指数据的存取方法和存储结构；其次是对所涉及的物理结构进行评价，评价的重点是系统

的时间和空间效率。通过评价结果完善、修改或重新设计物理结构,甚至返回逻辑设计阶段修改数据模型,直至评价结构满足设计要求,才进入下一实施阶段。

设计人员必须在确定数据库的物理结构之前,详细了解给定的 DBMS 的功能和特点,特别是该 DBMS 所提供的物理环境和功能,必须熟悉应用环境,充分了解所设计的应用系统中各部分的重要程度、处理频率、对响应时间等的要求,它们将作为物理设计过程中平衡时间和空间效率的依据,还应了解外存设备的特性,如 I/O 设备特性等。在对上述问题进行了解之后,即可进行物理结构的设计。

3.7.2 物理设计的步骤

数据库的物理设计分为两部分,如图 3-14 所示。

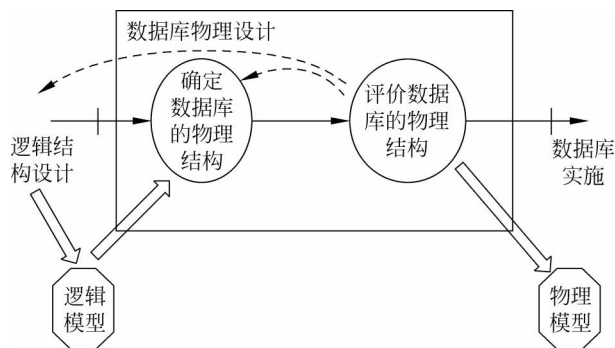


图 3-14 数据库物理设计

3.7.2.1 确定数据库的物理结构

1. 存储记录结构设计

在物理结构中,存储记录是数据的基本存取单位。存在了逻辑记录结构后,即可进行存储记录结构设计。一个存储记录可以对应于一个或多个逻辑记录。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都是存储记录结构的组成部分。

为方便对数据进行压缩和进行应用程序设计,需要分析属性特征、分割存储记录结构。其中分割方法有垂直分割和水平分割两种方法。垂直分割方法适用于含有较多属性的关系,按其中属性的使用频率不同进行分割;水平分割方法适用于含有较多记录的关系,按某些条件进行分割。决定数据的存储结构时还需要将存取时间、存储空间和维护代价间的平衡考虑在内。

2. 存储方法设计

存储方法是快速存取数据库中数据的技术。存储方法有顺序存放、HASH(散列)存放、聚簇存放和索引存放,常用的方法有聚簇存放和索引存放两种。

- (1) 顺序存放: 记录顺序存放,记录个数的二分之一为平均查询比较次数。
- (2) HASH 存放: 由 HASH 算法决定记录存放位置和查询比较次数。
- (3) 聚簇存放: 为了提高查询速度,将一个(或一组)属性上具有相同值的元组集中地

存放在一个连续的物理块中。若存放不下,可以存放在相邻的物理块中。这个(或这组)属性称为聚簇码。

聚簇功能适用于单个关系和经常进行连接操作的多个关系。使用聚簇后,聚簇码相同的元组就会集中在一起,因此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,可节省存储空间。另外,按照聚簇功能每读取一个物理块可得到多个满足查询条件的元组,从而大大提高按聚簇码进行查询的效率。

(4) 索引存放: 根据实际需要确定对要建立索引关系的属性列、建立组合索引的属性列,以及要建立唯一索引的属性列等就是索引存取方法。为提高查询速度,避免关系中主键的重复录入,确保了数据的完整性,经常在主关键字上建立唯一索引。建立索引的一般原则如下。

- 如果某个(或某组)属性经常作为查询条件,则会考虑在这个(或这组)属性上建立索引。
- 如果某个(或某组)属性经常作为表的连接条件,则会考虑在这个(或这组)属性上建立索引。
- 如果某个属性经常作为分组的依据列,则会考虑在这个属性上建立索引。
- 对于经常进行连接操作的表考虑建立索引。

建立多个索引文件可以提高查询性能,缩短数据的存取时间,但同时也会增加存放索引文件所占用的存储空间,增加建立和维护索引的成本。在修改数据时,系统要同时对索引进行维护,使索引与数据保持一致,导致建立索引后会使得数据修改性能下降。因此,关系上定义的索引数不是越多越好,要权衡数据库的操作之后再决定是否建立索引以及建立多少个索引。应该根据实际需要综合考虑,如果查询操作较多,并且对查询的性能要求高,则应考虑多建一些索引;如果数据修改操作较少,并且对修改的效率要求高,则应考虑少建一些索引。

3. 数据存放位置设计

将存取时间、存储空间利用率以及维护代价这三方面的因素综合考虑从而确定数据的存放位置和存储结构。其中,确定数据存放位置应该根据应用情况,如数据的易变部分和稳定部分不宜一起存放,经常存取部分和存取频率较低部分不宜一起存放,以提高系统性能。对于有多个磁盘的计算机存放位置的分配方案如下。

(1) 在不同的磁盘上分别存放表和索引,使两个磁盘驱动器并行工作,在查询时可以提高物理读写的速度。

(2) 在两个磁盘上分别存放较大的表,以加快存取速度,在多用户环境下效果更佳。

(3) 在不同的磁盘上存放备份文件、日志文件与数据库对象(如表、索引等)备份等。

4. 系统配置设计

通常关系数据库管理系统产品为设计人员和数据库管理员都提供了一些系统配置变量和存储分配参数,便于他们对数据库进行物理优化。系统都为这些变量赋予了合理的初始值,由于应用环境复杂多变,这些值不一定适合现状,为改善系统的性能,在进行物理设计时需要重新对这些变量赋值。

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,存储分配参数,内存分配参数,物理块的大小,物理块装填因子,缓冲区分配参数(使用的缓冲区长度、个数),时间片大小,数据库大小,锁的数目等,存取时间和存储空间的分配受这些参数值的影响。为改进系统性能,在物理设计时就要根据应用环境确定这些参数值。物理设计只需对系统配置变量进行初步的调整,为使系统性能最佳,需要在系统运行时根据系统实际运行情况再做调整。

3.7.2.2 评价数据库的物理结构

数据库物理设计过程中需要考虑多方面的因素,如需要对时间效率、空间效率、维护代价和用户各种要求进行权衡,其结果可以产生多种物理设计方案。数据库设计人员必须多方面、多角度对这些方案评价,最终数据库的物理结构要从中选取一个较优的方案。

对于完全依赖于所选用的具体 DBMS 的物理结构设计评价方法,系统的时间和空间效率应为评价的重点,具体分为如下几类。

- 查询和响应时间。从查询开始到开始显示查询结果所经历的时间为响应时间。一个好的应用程序设计可以减少 CPU 时间和 I/O 时间。
- 更新事务的开销。主要指修改索引、重写物理块或重写文件以及校验等方面的开销。
- 生成报告的开销。主要指索引、重组、排序和显示结果方面的开销。
- 主存储空间的开销。包括应用程序和数据所占用的空间。为减少空间开销,数据库设计者通常会对缓冲区做适当的控制或调整,包括控制缓冲区个数和大小。
- 辅助存储空间的开销。辅助存储空间分为数据块和索引块两种,数据库设计者可对控制索引块的大小、索引块的充满度等进行控制。

评价数据库的物理结构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,并对估算结果进行权衡、比较,从中挑选一个较优的、合理的物理结构。如果评价结果是该结构不符合用户需求,则需要修改或重新设计。

3.8 数据库的实施和维护

3.8.1 数据库的实施

数据库实施阶段主要完成两项任务,即数据的载入和应用程序的编码与调试。

通常情况下,完成数据库定义后,还要输入各种实际数据。数据库系统中数据量都很大,组织数据入库烦琐。因为新设计的数据库系统中数据的组织方式、结构和格式不同,可能出现源数据与新数据库结构不相容的情况。同时数据来源多样,可能来自单位中的各个不同的部门。所以组织数据载入需要从各个局部应用中抽取将各类源数据,存入计算机,再把它们分类转换,综合为符合新设计的数据库结构的形式,输入到数据库中再进行数据校验。

当原系统是手工数据处理系统时工作更加复杂,因为各类数据不集中,包含在各种不同的原始表单、单据、凭证之中。要先处理大量的纸质文件,才能向新的数据库系统中输入数

据,这种数据转换、组织入库的工作相当耗时、费力。

为提高数据输入工作的效率和质量,完成初始数据输入时,应该针对具体的应用环境设计一个数据录入子系统,使数据入库的任务由计算机来完成。为防止错误的数据入库,在源数据入库之前要采用多种方法对其进行检验,数据校验的工作在整个数据输入子系统中十分值得重视。

通常情况下,不同关系数据库管理系统之间数据转换的工具都会在现有的关系数据库管理系统中被利用,若原先是数据库系统,就要充分使用新系统提供的数据库转换工具。在组织数据入库的同时还需要调试应用程序,因为数据库应用程序的设计与数据库设计同时进行。

3.8.2 数据库的试运行

当应用程序调试完成并有一小部分数据已入库后,就可以开始对数据库系统进行试运行了,也称为数据库的联合调试。

试运行要测试应用程序的功能是否满足设计要求,具体做法为实际运行数据库应用程序,执行对数据库的各种操作。对应用程序不足的部分则要修改、调试,直至达到设计要求。

除对应用程序进行测试外,还要对系统的性能指标进行测试,分析其是否达到设计目标。一般情况下,设计结果会和实际系统运行有一定的差距,因此需要实际测量和评价系统性能指标,经过运行调试后能找到有些参数的最佳值。

在数据库的试运行阶段需要注意以下几点。

(1) 组织数据入库应分期分批。先输入小批量数据调试,待试运行基本正常后逐渐增加数据量,大批量输入数据,完成运行评价。

(2) 做好数据库的转储和恢复工作。试运行阶段系统不稳定,随时可能发生软硬件故障。系统操作人员对新系统还不熟悉,可能出现误操作。数据库的转储和恢复工作能使发生故障的数据库尽快恢复,使数据库的破坏减少。

3.8.3 数据库的运行与维护

数据库试运行合格后,标志着数据库开发工作基本完成。维护工作也要在应用环境和数据库运行过程中物理存储的变化中开始。维护工作主要由数据库管理员完成,本阶段工作主要包括以下几方面。

(1) 数据库的转储和恢复。要求制订不同的转储计划以适应不同应用,发生故障能尽快将数据库恢复到某种一致的状态,使数据库的破坏降到最低。

(2) 控制数据库的安全性、完整性。修改原有的安全性控制以适应数据的访问、用户的密级等变化。

(3) 数据库性能的监督、分析和改造。

(4) 数据库更新事务。包括检索、重组、排序、显示结果等方面。

(5) 数据库的重组与重构造。数据库进过一段时间的运行,数据不断变化,数据的存取效率降低,数据库性能下降,要求数据库管理员对数据进行重组或重构造。数据库的重组不修改原设计的逻辑和物理结构,数据库的重构造是指部分修改数据库的模式和内模式。

3.9 本章小结

本章主要介绍了数据库设计的概念、方法和步骤,详述了数据库设计的几个阶段(需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库的实施和维护)的主要工作、设计步骤和注意事项。数据库设计过程中最重要的两个环节是概念结构设计和逻辑结构设计,本章也作为重点进行了描述。

学习本章内容,要掌握书中介绍的基本方法,并将其运用到实际工作中,设计出符合用户需求的数据库应用系统。

3.10 习题

一、简答题

1. 简述数据库系统设计的各个阶段。
2. 简答关系规范化的级别;第一范式、第二范式、第三范式、Boyce-Codd 范式的特点。
3. 设有关系 $P(A, B, C, D, E)$, 且有函数依赖集合 $F = \{A \rightarrow B, A \rightarrow C, C \rightarrow D, D \rightarrow E\}$, 今若分解关系 P 为 $P_1(A, B, C)$ 和 $P_2(C, D, E)$, 试确定 P_1 和 P_2 的范式等级。
4. 简答数据库的概念结构;试述数据库概念结构设计的步骤。
5. 简答数据库的逻辑结构设计以及将 E-R 图转换为关系模型的一般规则。

二、应用实践

1. 请设计一个图书馆数据库。此数据库中对每个借阅者保存读者信息,包括读者号、姓名、地址、性别、年龄、单位。对每本书存有书号、书名、作者、出版社。对每本被借出的书存有读者号、借出日期和应还日期。要求:画出 E-R 图,再将其转换为关系模型。

2. 建立关于系、学生、班级、社团等信息的一个关系数据库。一个系有若干个专业,每个专业每年只招一个班,每个班有若干学生,一个系的学生住在同一宿舍区,每个学生可以参加若干社团,每个社团有若干学生。

描述学生的属性:学号、姓名、出生年月、系名、班级号、宿舍区。

描述班级的属性:班级号、专业名、系名、人数、入校年份。

描述系的属性:系名、系号、系办公地点、人数。

描述社团的属性:社团名、成立年份、地点、人数、学生参加某社团的年份。

请给出关系模式,写出每个关系模式的最小函数依赖集,指出是否存在传递函数依赖,对于函数依赖是多属性的情况,讨论函数依赖是完全函数依赖还是部分函数依赖;指出各关系的候选键、外部键,有没有全键存在。

3. 设大学里教学数据库中有三个实体集:一是“课程”实体集,属性有课程号、课程名称;二是“教师”实体集,属性有教师工号、姓名、职称;三是“学生”实体集,属性有学号、姓名、性别、年龄。

设教师与课程之间有“主讲”联系,每位教师可主讲若干门课程,但每一门课程只有一位

主讲教师,教师主讲课程将选用某本教材;教师与学生之间有“指导”联系,每位教师可指导若干学生,但每个学生只有一位指导教师;学生与课程之间有“选课”联系,每个学生可选修若干课程,每门课程可由若干学生选修,学生选修该门课程有一个成绩。

试画出 E-R 图,并在图上注明属性、联系类型、实体标识符并将其转换为关系模式。

4. 参考如下示例,遵循需求分析文档格式,给出包含功能需求的需求分析文档,并利用绘图软件绘制出该系统的概念结构设计。系统设计及绘图软件操作可观看微课视频学习。

(1) “学校教材订购、查询系统”简介。

本系统主要具有两个功能:销售和查询。具体做法是,学生可以网上查询是否还有需要购买的教材,若有,则去书库购买,书库管理人员收钱、开发票、发书并修改书库信息;若教材脱销,则通知书库采购人员采购。采购人员买来新书后,即进行修改书库信息。

(2) 技术要求和限制条件。

- 当书库中的各种书籍数量发生变化(包括进书和出书)时,都应修改相关的书库记录,如库存表。
- 在实现上述销售和采购的工作过程时,需考虑有关的合法性验证(即操作权限和数据安全性)。
- 系统的外部项至少包括学生(买书者)、书库管理人员和书库采购人员。