

数据库设计

本章学习重点：

- 需求分析的方法。
- 概念结果的设计。
- 逻辑结构的设计。
- 物理结构的设计。

数据库是数据库系统中最基本、最重要的部分。数据库性能的高低,决定了整个数据库应用系统的性能。设计一个性能优良的数据库,是满足各方面对数据需要的重要内容。本章主要介绍数据库设计的概念以及方法。

5.1 数据库设计概述

数据库中的数据不是相互独立的,数据库在系统中扮演着支持者的角色。而通常把使用数据库的各类信息系统都称为数据库应用系统。数据库设计广义地讲,是数据库及其应用系统的设计,即设计整个数据库应用系统。狭义地讲,就是设计数据库本身。

数据库设计其实就是软件设计,软件都有软件生存期。软件生存期是指从软件的规划、研制、实现、投入运行后的维护,直到它被新的软件取代而停止使用的整个期间。数据库设计方法有多种,概括起来分为 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。

按照规范化设计的方法,考虑数据库及其应用系统开发的全过程,通常将数据库设计分为 6 个阶段:需求分析阶段、概念结构设计阶段、逻辑结构设计阶段、物理结构设计阶段、数据库实施阶段以及数据库运行和维护阶段。

5.2 需求分析阶段

需求分析就是分析用户对数据库的具体要求,是整个数据库设计的起点和基础。需求分析的结果直接影响以后的设计,并影响到设计结果是否合理和实用。需求分析阶段是数据库设计的第一步,也是最困难的一步。

需求分析就是理解用户需求,询问用户如何看待未来的需求变化。让用户解释其需求,而且随着开发的继续,还要经常询问用户保证其需求仍然在开发的目的之中。了解用户业务需求有助于在以后的开发阶段节省大量的时间。同时还应该重视输入和输出,增强应用程序的可读性。需求分析主要考虑“做什么”,而不应该考虑“怎么做”。

5.2.1 需求分析的任务

需求分析的总体任务是通过详细调研现实业务处理对象,充分掌握原系统业务数据及处理流程,明确用户各种需求,经规范化和分析形成文档(报告)。数据库需求分析的重点是调查用户和系统的数据(信息)及处理要求。

1. 信息要求

确定用户需要从数据库中获得信息的具体内容与性质,从而导出各种数据要求。

2. 处理要求

确定用户具体处理要求(如处理功能、内容、方式、顺序、流程和响应时间等),最终要实现的具体处理功能、性能等。

3. 安全性和完整性要求

明确系统中不同用户对数据库的使用和操作情况,明确数据之间的关联关系及数据的用户具体定义要求。

5.2.2 需求分析的方法

进行需求分析首先要调查清楚用户的实际要求,与用户达成共识,然后分析与表达这些需求。

1. 调查用户需求的具体步骤

(1) 调查组织机构情况。包括了解该组织的部门组成情况、各部门的职责等,为分析信息流程做准备。

(2) 调查各部门的业务活动情况。包括了解各部门输入和使用什么数据,如何加工处理这些数据,输出什么信息,输出到什么部门,输出结果的格式是什么,这是调查的重点。

(3) 在熟悉业务活动的基础上,协助用户明确对新系统的各种要求。包括信息要求、处理要求、完全性与完整性要求,这是调查的另一个重点。

(4) 确定新系统的边界。对前面调查的结果进行初步分析,确定哪些功能由计算机完成或将来准备让计算机完成,哪些活动由人工完成。由计算机完成的功能就是新系统应该实现的功能。在调查过程中,可以根据不同的问题和条件,使用不同的调查方法。

2. 常用的调查方法

(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,但比较耗时。

(2) 开座谈会。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,一般可按职能部门组织座谈会。

(3) 询问或请专人介绍。一般应包括领导、管理人员、操作员等。

(4) 设计调查表请用户填写需求。如果调查表设计得合理,这种方法是很有效的,也易于被用户接受。

(5) 调阅记录。查阅与原系统有关的数据记录。

做需求调查时,往往需要同时采用上述多种方法。但无论使用何种调查方法,都必须有用户的积极参与和配合,最好能建立由双方人员参加的项目实施保障小组负责沟通联系。

5.2.3 数据流图和数据字典

数据流图(Data Flow Diagram,DFD)和数据字典(Data Dictionary,DD)是对需求分析结果进行描述的两个主要工具。

1. 数据流图

数据流图是描述数据与处理流程及其关系的图形表示,以图形的方式表示数据和数据流从输入到输出的过程中所经受的变换及过程。在结构化分析方法中,任何一个系统都可抽象成如图 5-1 所示的数据流图。

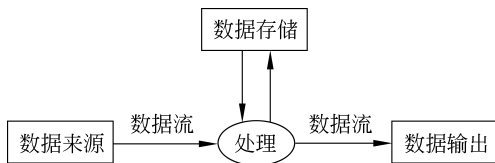


图 5-1 数据流图

1) 数据流

在数据流图中,有箭头的线段表示数据流,数据流由一组确定的数据组成。名字表示流经的数据,箭头表示数据流动的方向。

2) 处理

在数据流图中,用椭圆表示处理,处理是对数据进行的操作或处理。

3) 数据存储

数据存储是指数据保存的地方。在数据流程图中用一个双向箭头表示,箭头指出数据的流动方向,系统处理从数据存储中提取数据,也将处理的数据返回数据存储。

4) 外部实体

指独立于系统而存在的,但又和系统有联系的实体。它表示数据的外部来源和最后

的去向。确定系统与外部环境之间的界限,从而可确定系统的范围。在数据流图中,用矩形表示外部实体。外部实体可以是某种组织、系统或事物。

2. 数据字典

数据字典是系统中各类业务数据及结构描述的集合,是各类数据结构和属性清单。同数据流图互为补充,数据字典贯穿于数据库需求分析直到运行的全过程,在不同的阶段,其内容形式和用途有所差别。数据字典通常包括数据项、数据结构、数据存储、数据流和处理过程 5 部分。

1) 数据项

数据项是最小的数据单位。它通常包括属性名、含义、别名、类型、长度、取值范围、与其他数据项的逻辑联系等。

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数据项组成,也可由若干数据结构组成,或由数据项与数据结构混合组成。它包括关系名、含义、组成的成分等。

3) 数据存储

数据存储是数据结构停留并保存的地方,也是数据流的来源和去向之一。它可以是手工文档或凭单,也可以是计算机文档。它包括数据存储名、说明、输入输出数据流、组成的成分(数据结构或数据项)、存取方式、操作方式等。

4) 数据流

数据流表示数据项或数据结构在某一加工过程的输入或输出。数据流包括数据流名、说明、输入输出的加工名、组成的成分。

5) 处理过程

处理过程的具体处理逻辑一般用判定表或判定树来描述,包括处理过程名、说明、输入输出数据流、处理的简要说明等。

数据字典是关于数据库中数据的描述。数据字典是在需求分析阶段建立,在数据库设计过程中不断修改、充实、完善的。

5.3 概念结构设计阶段

概念结构设计是将需求分析中用户具体业务数据处理等实际需求,抽象为信息结构(概念模型)的过程,是现实世界(事务)到机器世界(数据及处理)的一个重要过渡层次,也是整个数据库设计的关键。

概念结构设计通常将现实世界中的客观事务(实体),先抽象为不依赖 DBMS 支持的数据模型(E-R 图),概念模型是各种数据模型的共同基础。

概念结构设计的特点及优势主要有以下 4 点。

(1) 直观易于理解,E-R 图便于研发人员和需求用户直接交换意见,用户的积极参与是数据库设计成功的关键。

(2) 可以真实且充分地描述现实世界的具体事务,包括事务及其之间的联系,可以满足用户对数据的处理要求,是对现实世界的一个真实直观模型。

(3) 易于扩充、修改和完善,当应用环境和业务需求改变时,方便对概念模型扩充、修改和完善。

(4) 便于向关系数据模型、网状数据模型和层次数据模型等进行转换。

5.3.1 概念结构设计方法

概括起来,设计概念模型的总体策略和方法可以归纳为如下 4 种。

(1) 自顶向下法。首先认定用户关心的实体及实体间的联系,建立一个初步的概念模型框架,即全局 E-R 模型,然后再逐步细化,加上必要的描述属性,得到局部 E-R 模型。

(2) 自底向上法。有时又称为属性综合法,先将需求分析说明书中的数据元素作为基本输入,通过对这些数据元素的分析,把它们综合成相应的实体和联系,得到局部 E-R 模型,然后在此基础上再进一步综合成全局 E-R 模型。

(3) 逐步扩张法。先定义最重要的核心概念 E-R 模型,然后向外扩充,以滚雪球的方式逐步生成其他概念 E-R 模型。

(4) 混合策略。将单位的应用划分为不同的功能,每一种功能相对独立,针对各功能设计相应的局部 E-R 模型,最后通过归纳合并,消去冗余与不一致,形成全局 E-R 模型。

其中最常用的策略是自底向上法,即先进行自顶向下的需求分析,再进行自底向上的概念设计。

5.3.2 概念模型设计步骤

在概念结构设计时,自底向上法可以分为两步。第一步是进行数据抽象,设计局部 E-R 模型,即设计用户视图。第二步是集成各局部 E-R 模型,形成全局 E-R 模型,即视图的集成。

1. 设计局部 E-R 模型

局部 E-R 模型的设计步骤包括如下 4 步。

1) 确定局部 E-R 模型描述的范围

根据需求分析所产生的文档,可以确定每个局部 E-R 模型描述的范围。通常采用的方法是将单位的功能划分为几个系统,每个系统又分为几个子系统。设计局部 E-R 模型的第一步就是划分适当的系统或子系统,在划分时过细或过粗都不太合适。划分过细会造成大量的数据冗余和不一致,过粗有可能漏掉某些实体。

一般可以遵循以下两条原则进行功能划分。

(1) 独立性原则。划分在一个范围内的应用功能具有独立性与完整性,与其他范围内的应用有最少的联系。

(2) 规模适度原则。局部 E-R 模型规模应适度,一般以 6 个左右实体为宜。

2) 确定局部 E-R 模型的实体

根据需求分析说明书,将用户的数据需求 and 处理需求中涉及的数据对象进行归类,指明对象的身份,是实体、联系还是属性。

3) 定义实体的属性

根据上述实体的描述信息来确定其属性。

4) 定义实体间的联系

确定了实体及其属性后,就可以定义实体间的联系了。实体间的联系按其特点可分为 3 种:存在性联系(如学生有所属的班级)、功能性联系(如教师要教学生)、事件性联系(如学生借书)。实体间的联系方式分为一对一、一对多、多对多 3 种。

设计完成某一局部结构的 E-R 模型后,再看还有没有其他的局部 E-R 模型,有则转到第 2)步继续,直到所有的局部 E-R 模型都设计完为止。

2. 局部 E-R 模型的集成

由于局部 E-R 模型反映的只是单位局部子功能对应的数据视图,可能存在不一致的地方,还不能作为逻辑设计的依据,这时可以去掉不一致和重复的地方,将各局部视图合并为全局视图,即局部 E-R 模型的集成。

一般来说,视图集成可以有两种方式:第一种是多个分 E-R 模型一次集成;第二种是逐步集成,用累加的方式一次集成两个分 E-R 模型。第一种方式比较复杂,做起来难度较大。第二种方式每次只集成两个分 E-R 模型,可以降低复杂度。

无论采用哪种集成法,每一次集成都分为两个阶段:第一步是合并,以消除各局部 E-R 模型之间的不一致情况,生成初步的 E-R 模型;第二步是优化,消除不必要的数据冗余,生成全局 E-R 模型。

5.4 逻辑结构设计阶段

数据库概念设计阶段得到的数据模式是用户需求的形式化,它独立于具体的计算机系统和 DBMS。为了建立用户所要求的数据库,必须把上述数据模式转换成某个具体的 DBMS 所支持的概念模式,并以此为基础建立相应的外模式,这是数据库逻辑设计的任务,是数据库结构设计的重要阶段。

逻辑设计的主要目标是产生一个 DBMS 可处理的数据模型和数据库模式。该模型必须满足数据库的存取、一致性及运行等各方面的用户需求。逻辑结构设计阶段一般要分为三步进行:将 E-R 模型转换为关系数据模型,关系模式的优化,设计用户外模式。

5.4.1 将 E-R 模型转换为关系数据模型

关系数据模型是一组关系模式的集合,而 E-R 模型是由实体、属性和实体之间的联系三要素组成的。所以将 E-R 模型转换为关系数据模型实际上是要将实体、属性和实体之间的联系转换为关系模式。转换过程中要遵循如下原则。

1. 实体的转换

一个实体转换为一个关系模式。实体的属性就是关系的属性。实体的主码就是关系的主码。

【例 5-1】 将图 5-2 所示的学生实体,转换为关系模式。
对应的关系模式为

学生(学号,姓名,出生日期,所在系,年级,平均成绩)

其中,学号为主码,用下画线标识。

2. 联系的转换

两实体间联系的转换根据联系类型的不同分为 3 种。

1) 1 : 1 联系的转换

两实体间 1 : 1 联系可以转换为一个独立的关系模式,也可以与任意一端对应的关系模式合并。

(1) 转换为一个独立的关系模式。

转换后的关系模式中关系的属性包括与该联系相连的各实体的码以及联系本身的属性;关系的主码为任一实体的主码。

【例 5-2】 将如图 5-3 所示的 E-R 模型转换为关系模式。

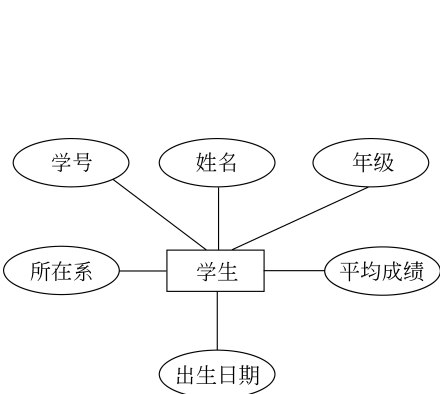


图 5-2 学生实体图

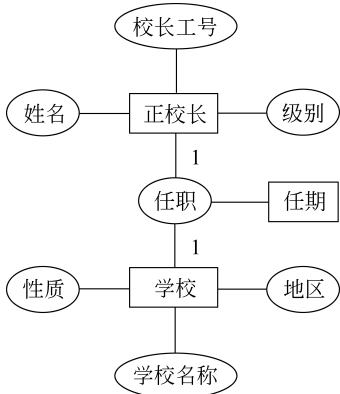


图 5-3 1 : 1 联系的 E-R 模型

转换成如下关系模式：

学校(学校名称,性质,地区)
正校长(校长工号,姓名,级别)
任职(学校名称,校长工号,任期)

(2) 与某一端对应的关系模式合并。

合并后关系的属性包括自身关系的属性和另一关系的码及联系本身的属性;合并后

关系的码不变。

【例 5-3】 将如图 5-3 所示的 E-R 模型,按照方案(2)转换为关系模式。

转换成如下关系模式:

学校(学校名称,性质,地区,校长工号,任期)
正校长(校长工号,姓名,级别)

或

学校(学校名称,性质,地区)
正校长(校长工号,姓名,级别,学校名称,任期)

2) 1 : N 联系的转换

两实体间 1 : N 联系可以转换为一个独立的关系模式,也可以与 N 端对应的关系模式合并。

(1) 转换为一个独立的关系模式。

关系的属性包括与该联系相连的各实体的码以及联系本身的属性;关系的主码为 N 端实体的码。

【例 5-4】 将如图 5-4 所示的 E-R 模型转换为独立的关系模式。

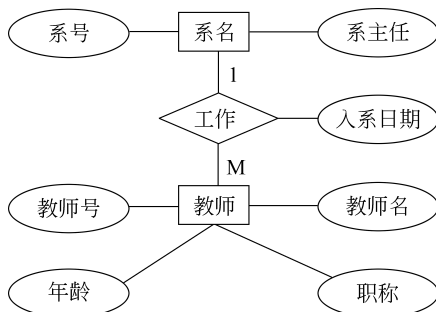


图 5-4 1 : N 联系的 E-R 模型

对应的关系模式如下:

系(系号,系名,系主任)
教师(教师号,教师名,年龄,职称)
工作(教师号,系号,入系日期)

(2) 与 N 端对应的关系模式合并。

合并后关系的属性包括在 N 端关系中加入 1 端关系的主码和联系本身的属性。合并后关系的码不变。

【例 5-5】 将如图 5-4 所示的 E-R 模型,按照方案(2)转换为合并的关系模式。

对应的关系模式如下:

系(系号,系名,系主任)
教师(教师号,教师名,年龄,职称,系号,入系日期)

注意：实际使用中通常采用这种方法以减少关系模式，因为多一个关系模式就意味着查询过程中要进行连接运算，而降低查询的效率。

3) M : N 联系的转换

两实体间 M : N 联系必须为联系产生一个新的关系。该关系中至少包含被它所联系的双方实体的主码，若联系有属性，也要并入该关系中。关系的主码为双方实体的主码的组合。

【例 5-6】 将图 5-5 所示的 E-R 模型转换成对应的关系模式。

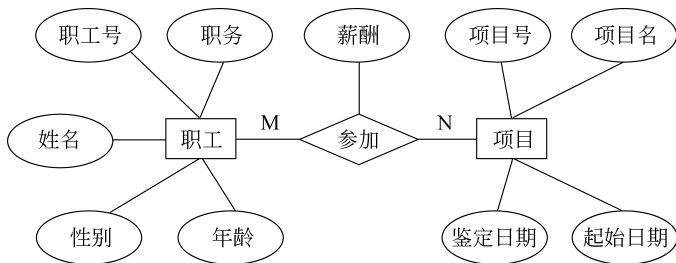


图 5-5 M : N 联系的 E-R 模型

对应的关系模式如下：

职工 (职工号, 姓名, 性别, 年龄, 职务)
项目 (项目号, 项目名, 起始日期, 鉴定日期)
参加 (职工号, 项目号, 薪酬)

5.4.2 关系模式的优化

通常情况下，数据库逻辑设计的结果不是唯一的。为了进一步提高数据库应用系统的性能，还应努力减少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重要理论基础和有力工具。为了提高数据库应用系统的性能，需要对关系模式进行修改，调整结构，这就是关系模式的优化。

关系模式的优化方法如下。

- (1) 确定数据依赖。
- (2) 对于各关系模式之间的数据依赖进行极小化处理，消除冗余的联系。
- (3) 按照数据依赖的理论对关系模式逐一进行分析，考查是否存在部分函数依赖、传递函数依赖、多值依赖等，确定各关系模式分别属于第几范式。
- (4) 按照需求分析阶段得到的各种应用对数据处理的要求，分析对于这样的应用环境这些模式是否合适，确定是否要对它们进行合并或分解。
- (5) 对关系模式进行必要的分解。

规范化理论为数据库设计人员判断关系模式优劣提供了理论标准，可用于预测模式可能出现的问题，使数据库设计工作有严格的理论基础。

5.4.3 设计用户外模式

外模式也称为子模式，是用户可直接访问的数据模式。同一系统中，不同用户可有

不同的外模式。外模式来自逻辑模式,但在结构和形式上可以不同于逻辑模式,所以它不是逻辑模式简单的子集。

外模式的作用主要有:通过外模式对逻辑模式的屏蔽,为应用程序提供了一定的逻辑独立性;可以更好地适应不同用户对数据的需求;为用户划定了访问数据的范围,有利于数据的保密等。

定义数据库全局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相对独立的,在定义用户外模式时可以注重考虑用户的习惯与方便。这些习惯与方便包括如下 3 方面。

- (1) 使用符合用户习惯的别名。
- (2) 可以对不同级别的用户定义不同的视图,以保证系统的安全性。
- (3) 简化用户对系统的使用。

如果某些局部应用中经常要使用某些很复杂的查询,为了方便用户,可以将这些复杂查询定义为视图,用户每次只对定义好的视图进行查询,大大简化了用户的使用。

5.5 物理结构设计阶段

数据库最终要存储在物理设备上。将逻辑设计中产生的数据库逻辑模型结合指定的数据库管理系统,设计出最适合应用环境的物理结构的过程,称为数据库的物理结构设计。

数据库的物理结构设计分为如下两个步骤。

- (1) 确定数据库的物理结构。
- (2) 对所设计的物理结构进行评价。

如果所设计的物理结构的评价结果满足原设计要求则可进入物理实施阶段,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。

5.5.1 确定数据库的物理结构

数据库物理结构设计包括确定数据的存储结构、设计数据的存取路径、确定数据的存放位置和确定系统配置。

1. 确定数据的存储结构

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 3 方面的因素。这 3 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。

2. 设计数据的存取路径

数据库管理系统常用存取方法有 B+树索引方法、聚簇方法和 HASH 索引方法。

1) B+树索引方法

在关系数据库中,选择存取路径主要是指确定如何建立索引。例如,应把哪些域作