

第3章 数据库设计

本章主要介绍数据库设计的基本概念,包括数据库设计的内容及方法,并详细介绍数据库设计的各个步骤。

3.1 数据库设计概述

数据库是现代各种计算机应用系统的核心。数据库所存储的信息能否正确地反映现实世界,能否在运行中及时、准确地为各个应用程序提供所需的数据,关系到以此数据库为基础的应用系统的性能。换句话说,设计一个能够满足应用系统中各个应用要求的数据库,是数据库应用系统设计中的关键问题。

3.1.1 数据库设计的内容

数据库的设计是从用户的数据需求、处理要求和建立数据库的环境条件,如硬件特性、操作系统和 DBMS 特性及其他限制等出发,把给定的应用环境(现实世界)内存在着的数据合理地组织起来,逐步抽象成已经选定的某个 DBMS 能够定义和描述的具体数据结构的过程。

数据库设计的主要内容如下。

① 静态特性设计:又称结构特性设计,即根据给定的应用环境、用户的数据需求,设计数据库的数据模型(即数据结构)或数据库模式。静态特性设计包括数据库的概念结构设计和逻辑结构设计两方面。

② 动态特性设计:即根据应用处理要求,设计数据库的查询、事务处理和报表处理等应用程序。动态特性设计反映了数据库在处理上的要求,即动态要求,所以又称数据库的行为特性设计。

③ 物理设计:根据动态特性(应用处理要求),在选定的 DBMS 环境下,把静态特性设计中得到的数据库模式加以物理实现,即设计数据库的存储模式和存取方法。

数据库设计的主要特点是,首先从数据模型即数据结构开始设计,并以数据模型为核心展开。数据库的一个重要优点是减少数据冗余,实现数据共享。因此,设计出包含各个用户视图的统一数据模型就成为数据库设计中的核心问题。同时,数据库设计应该和应用系统设计相结合,整个设计过程把结构设计和行为设计密切结合起来,这是建立一个数据库应用系统行之有效的办法。

从上述可知,数据库设计的成果是数据库模式和以数据为基础的应用程序。它们分别反映了对数据库的静态要求和动态要求。但应用程序是随着应用的发展而不断变化的。在有些以实时访问为主的数据库中,事先很难编出所需的应用程序或事务,因此数据库设计的最基本成果是数据库模式。数据库模式的设计必须反映数据处理的要求,保证常用的或大多数的数据处理使用方便,性能满意。

3.1.2 数据库设计的方法

数据库设计的方法目前可分为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。直观设计法又称单步设计法，它依赖于设计者的经验和技巧，因此越来越不适应信息管理系统发展的需要。为了改变这种情况，1978年10月来自30多个欧美国家的主要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，提出了数据库设计规范，把数据库设计分成公司需求分析(分析用户需求)、信息分析和定义(建立公司的组织模式)、设计实现(逻辑设计)和物理数据库设计(物理设计)4个阶段。目前，常用的规范设计方法大多起源于新奥尔良方法。

1. 基于3NF的数据库设计方法

基于3NF的数据库设计方法是由S. Atre提出的数据库设计的结构化方法，其基本思想是在需求分析的基础上，识别并确认数据库模式中的全部属性和属性间的依赖，将它们组织成单一的关系模式，然后再分析模式中不符合3NF的约束条件，用投影和连接的办法将其分解，使其达到3NF条件。

具体设计步骤分为5个阶段：

- ① 企业模式设计。利用上述得到的3NF关系模式画出企业模式。
- ② 数据库概念模式设计。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 数据库存储模式(物理模式)设计。
- ④ 对物理模式进行评价。
- ⑤ 数据库实现。

2. 基于实体联系的数据库设计方法

基于实体联系(E-R)的数据库设计方法是由P. P. S. Chen在1976年提出的，其基本思想是在需求分析的基础上，用E-R图构造一个纯粹反映现实世界实体之间内在关系的企业模式，然后再将此企业模式转换成选定的DBMS上的概念模式。

3. 基于视图概念的数据库设计方法

基于视图概念的数据库设计方法先从分析各应用的数据着手，为每个应用建立各自的视图，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。合并时必须注意解决下列问题。

- 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。

除了以上介绍的方法外，还有属性分析法、实体分析法以及基于抽象语义规范的设计法等，这里不再介绍。

在实际的设计过程中，各种方法可以结合起来使用，例如，在基于视图概念的设计方法中可用E-R图的方法来表示各个视图。

3.1.3 数据库设计的步骤

目前，分步设计法已在数据库设计中得到广泛的应用并获得较好的效果，此方法遵循自顶

向下、逐步求精的原则,将数据库的设计过程分解为若干相互独立又相互依存的阶段,每一阶段采用不同的技术与工具,解决不同的问题,从而将问题局部化,减少局部问题对整体设计的影响。

在分步设计法中,通常将数据库的设计分为4个阶段:需求分析、概念设计、逻辑设计和物理设计,如图3.1所示。

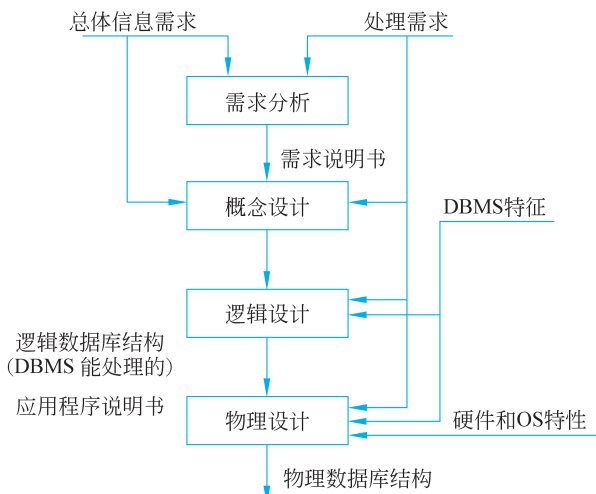


图 3.1 数据库设计的步骤

1. 需求分析

需求分析的目标是通过调查研究,了解用户的数据要求和处理要求,并按一定的格式整理形成需求说明书。需求说明书是需求分析阶段的成果,也是今后设计的依据,包括数据库所涉及的数据、数据的特征、数据量和使用频率的估计,如数据名、属性及其类型、主关键字属性、保密要求、完整性约束条件、使用频率、更改要求、数据量估计等。这些关于数据的定义描述称为元数据(Meta Data)。在设计大型数据库时,这些数据通常由称为数据字典(Data Dictionary, DD)的计算机软件(专用软件包或 DBMS)管理。用数据字典管理元数据有利于避免数据的重复或重名,以保持数据的一致性及提供各种统计数据,因而有利于提高数据库设计的质量,同时可以减轻设计者的负担。

2. 概念设计

概念设计是数据库设计的第二阶段,其目标是对需求说明书提供的所有数据和处理要求进行抽象与综合处理,按一定的方法构造反映用户环境的数据及其相互联系的概念模型,即用户的数据模型或企业数据模型。这种概念数据模型与 DBMS 无关,是面向现实世界的数据库模型,极易为用户所理解。为保证所设计的概念数据模型能正确、完全地反映用户(一个单位)的数据及其相互关系,便于进行所要求的各种处理,在本阶段设计中可吸收用户参与和评议设计。在进行概念设计时,可设计各个应用的视图(View),即各个应用所看到的数据及其结构,然后再进行视图集成(View Integration),以形成一个单位的概念数据模型。这样形成的初步数据模型还要经过数据库设计者和用户的审查和修改,最后形成所需的数据库模型。

3. 逻辑设计

逻辑设计阶段的设计目标是把上一阶段得到的与 DBMS 无关的概念数据模型转换成等价的并为某个特定的 DBMS 所接受的逻辑模型表示的概念模式,同时将概念设计阶段得到的应用视图转换成外部模式,即特定 DBMS 下的应用视图。在转换过程中要进一步落实需求说

明,并满足 DBMS 的各种限制。逻辑设计阶段的结果是 DBMS 提供的数据库定义语言(DDL)写成的数据模式。逻辑设计的具体方法与 DBMS 的逻辑数据模型有关。

4. 物理设计

物理设计阶段的任务是把逻辑设计阶段得到的逻辑数据库在物理上加以实现,其主要内容是根据 DBMS 提供的各种手段,设计数据的存储形式和存取路径,如文件结构、索引等,即设计数据库的内模式或存储模式。数据库的内模式对数据库的性能影响很大,应根据处理需求及 DBMS、操作系统和硬件的性能进行精心设计。

在数据库设计的基本过程中,每一阶段设计基本完成后,都要进行认真检查,看是否满足应用需求,是否符合前面已执行步骤的要求和满足后续步骤的需要,并分析设计结果的合理性。在每一步设计中,都可能发现前面步骤遗漏或处理不当之处,此时往往需要返回去重新处理并修改设计和有关文档。所以,数据库设计过程通常是一个反复修改、反复设计的迭代过程。

3.2 需求分析

3.2.1 需求分析的任务

需求分析是数据库设计的第一阶段,这一阶段收集到的基础数据和一组数据流图(Data Flow Diagram,DFD)是下一步设计概念结构的基础。

从数据库设计的角度考虑,需求分析阶段的目标是:对现实世界要处理的对象(组织、部门、企业等)进行详细调查,在了解原系统的概况和确定新系统功能的过程中,收集支持系统目标的基础数据并进行相应处理。

调查的重点是“数据”和“处理”,通过调查要从中获得每个用户对数据库的下列要求。

① 信息要求。用户将从数据库中获得信息的内容、性质。由信息要求导出数据要求,即在数据库中需存储哪些数据,如何处理这些数据等。描述数据间本质上和概念上的联系,描述信息的内容、结构以及信息之间的联系等。

② 处理要求。处理要求定义未来系统处理数据的操作功能,描述操作的优先次序,包括操作执行的频率和场合以及操作与数据之间的联系。处理要求还包括弄清楚用户要完成什么样的处理功能,对某种处理功能的响应时间,处理的方式是批处理还是联机处理。

③ 安全性和完整性的要求。在众多分析和表达用户需求的方法中,结构化分析(Structured Analysis,SA)方法是一个简单实用的方法。SA 方法用自顶向下、逐层分解的方式分析系统,用数据流图、数据字典描述系统。

3.2.2 需求分析的基本步骤

需求分析大致分为需求信息的收集、分析整理和评审三步。

1. 需求信息的收集

需求信息的收集又称系统调查。为了充分了解用户可能提出的要求,在调查研究之前,要做好充分的准备工作,明确调查的目的、内容和方式。

首先,要了解组织的机构设置,主要业务活动和职能;其次,要确定组织的目标,大致的工作流程和任务范围划分。调查的内容包括外部要求、业务现状、组织机构和规划中的应用范围

及要求。外部要求一般包括信息的性质,响应的时间、频度和发生的规则,以及经济效益的考虑和要求,安全性及完整性要求。业务现状包括信息的种类、信息的流程、信息的处理方式、各种业务工作过程和各种票据。调查方式可采用开座谈会、跟班作业、请调查对象填写调查表、查看业务记录和票据以及个别交谈等形式。

2. 需求信息的分析整理

要想把收集到的信息(如文件、图表、票据、笔记)转换为下一阶段设计工作可用的信息形式,必须对需求信息做分析整理的工作。

① 业务流程分析。业务流程分析的目的是获得业务流程及业务与数据联系的形式描述。一般采用数据流分析法,分析结果用数据流图表示。

② 分析结构的描述。除了数据流图以外,还要用一些规范的表格进行补充性描述。一般有数据清单(数据元素表)、业务活动清单(事务处理表)、完整性及一致性要求、响应时间要求、预期变化的影响等。可使用数据字典和需求分析语言来描述。

3. 需求信息的评审

评审的目的在于确认某一阶段的任务是否全部完成,以避免重大的疏漏或错误。评审要有项目组以外的专家和主管部门负责人参加,以保证评审工作的客观性和质量。

需求分析阶段的成果是产生系统需求说明书。系统需求说明书主要包括数据流图、数据字典的表格、各类数据的统计表格、系统功能结构图,并加以必要的编辑说明。系统需求说明书将作为数据库设计全过程的重要依据。

3.3 概念设计

概念模型设计的任务是,在需求分析产生的需求说明书的基础上,按照一定的方法抽象出满足应用需求的用户(单位)的信息结构,即通常所称的概念模型。

概念模型的设计过程也就是正确地选择设计策略、设计方法和概念数据模型并加以实施的过程。

3.3.1 概念设计的目标和策略

概念模型设计的目标是产生一个用户易于理解的、反映系统信息需求的整体数据库概念模型。概念模型是系统中各个用户共同关心的信息结构。它独立于计算机的数据模型,独立于特定的数据库管理系统,独立于计算机的软硬件系统。

概念结构独立于数据库的逻辑结构,独立于支持数据库的 DBMS,其作用如下。

① 提供能够识别和理解系统要求的框架。因此,必须弄清每个应用的重要方面及各个应用的细微差别,否则就设计不出适用的概念模型。

② 概念模型为数据库提供一个说明性结构,作为设计数据库逻辑结构即逻辑模型的基础。对概念模型的要求是:

- 能充分地反映现实世界(包括实体和实体之间的联系);能满足用户对数据处理的要求,是现实世界的一个真实模型。
- 易于理解,从而可以和不熟悉计算机的用户交换意见。用户能否积极参与是数据库设计成功与否的关键。
- 易于变动,当现实世界改变时容易修改和扩充。

- 易于向关系、网状或层次等各种数据模型转换。

概念结构是各种数据模型的共同基础,它比数据模型更独立于机器、更抽象,从而更加稳定。

设计概念结构的策略有以下 4 种。

- ① 自顶向下。首先定义全局概念结构的框架,然后逐步细化。
- ② 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,得到全局概念结构。
- ③ 由里向外。首先定义最重要的核心概念结构,然后向外扩充,生成其他的概念结构。
- ④ 混合策略。采用自顶向下和自底向上相结合的方法。

使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局部概念结构。最常用的概念结构设计策略是自底向上的设计策略。

3.3.2 采用 E-R 方法的数据库概念设计

概念模型设计的常用方法是实体联系方法(E-R 方法)。用实体联系方法对具体数据进行抽象加工,将实体集合抽象成实体类型,用实体间的联系反映现实世界事物间的内在联系。利用 E-R 方法进行数据库的概念模型设计,可以分三步进行:首先设计局部 E-R 模型,然后把各局部 E-R 模型综合成一个全局 E-R 模型,最后对全局 E-R 模型进行优化,得到最终的 E-R 模型,即概念模型。

1. 设计局部 E-R 模型

每个数据库系统都是为多个不同用户服务的。各个用户对数据的观点可能不一样,信息处理需求也可能不同。在设计数据库概念结构时,为了更好地模拟现实世界,一般先分别考虑各个用户的信息需求,形成局部概念结构,然后再综合成全局结构。在 E-R 方法中,局部概念结构又称局部 E-R 模型,其图形表示称为局部 E-R 图。局部 E-R 模型的设计过程如下。

(1) 确定局部结构范围

设计各个局部 E-R 模型的第一步就是确定局部结构的范围划分,划分的方式一般有两种。一种是依据系统的当前用户进行自然划分。如一个企业的用户有不同部门,各部门对信息的内容和处理的要求明显不同,应分别为它们设计各自的局部 E-R 模型。另一种是按用户要求将数据库提供的服务归纳成几类,使每一类应用访问的数据显著地不同于其他类,并且为每一类应用设计一个局部 E-R 模型。

(2) 定义实体

每一个局部结构都包括一些实体类型。实体定义的任务就是从信息需求和局部范围定义出发,确定每一个实体类型的属性和键。

实体类型确定后,其属性也随之确定。实体类型的命名应反映实体的语义性质,在一个局部结构中应是唯一的。键可以是单个属性,也可以是属性的组合。

(3) 定义联系

定义联系的一种方式是依据需求分析的结果,考查局部结构中任意两个实体类型之间是否存在联系。若存在联系,进一步确定是一对一、一对多还是多对多的联系。还要考查一个实体类型内部是否存在联系,两个实体类型之间是否存在联系,多个实体类型之间是否存在联系等。

在确定联系类型时,应注意防止出现冗余的联系,即可以从其他联系导出的联系,如果存在,要尽可能识别并消除这些冗余联系,以免影响全局 E-R 模型。

(4) 属性的分配

确定了实体与联系后,可用属性描述局部结构中的其他语义信息。首先确定属性,然后把属性分配到有关的实体和联系中去。

确定属性的原则是:属性应该是不可再分解的语义单位;实体与属性之间的关系只能是一对多的关系;不同实体类型的属性之间应没有直接的关联关系。

当多个实体类型用到同一属性时,将导致数据冗余,从而可能影响存储效率和完整性约束,因而需要确定把它分配给哪个实体类型。一般把属性分配给那些使用频率最高的实体类型,或分配给实体值少的实体类型。

2. 设计全局 E-R 模型

所有局部 E-R 模型设计好之后,就需要把它们综合成单一的全局概念结构。全局概念结构不仅要支持所有的局部 E-R 模型,还必须合理地表示一个完整、一致的数据库概念结构。全局 E-R 模型的设计过程如下。

(1) 确定公共实体类型

当系统较大时,可能有很多局部模型,且这些局部 E-R 模型是由不同的设计人员确定的,因而对同一现实世界的对象可能给予不同的描述。有的作为实体类型,有的作为关系类型或属性。即使都表示成实体类型,实体类型名和键也可能不同。一般把同名实体类型或具有相同键的实体类型作为可能的公共实体类型。

(2) 局部 E-R 模型的合并

局部 E-R 模型合并的原则是:首先进行两两合并,然后再合并那些现实世界中有关系的局部结构;合并从公共实体类型开始,最后再加入独立的局部结构。

(3) 消除冲突

将局部 E-R 模型合并成全局 E-R 模型时,应消除以下三种冲突。

① 属性冲突:首先包括属性域的冲突,即属性值的类型、取值范围或取值集合不同,如对于零件号,不同的部门常采用不同的编码方式;其次是属性取值单位的冲突,如质量单位有的用千克,有的用克。

② 结构冲突:包括同一对象在不同应用中的不同抽象,如职工在某个应用中为实体,在另一应用中为属性;其次是同一实体在不同局部 E-R 模型中属性组成不同,如属性个数、次序等。再者实体之间的联系在不同的局部 E-R 模型中可能会呈现不同的类型。

③ 命名冲突:包括属性名、实体名、联系名之间的冲突。同名异义,即不同意义的对象具有相同的名字;异名同义,即同一意义的对象具有不同的名字。

属性冲突和命名冲突通常采用讨论、协商等行政手段解决,结构冲突则需要认真分析后才能解决。例如,把实体变换为属性或把属性变换为实体,使同一对象具有相同的抽象。又如,取同一实体在各局部 E-R 模型中属性的并作为集成后该实体的属性集,对属性的取值类型进行统一协调。

3. 全局 E-R 模型的优化

一个好的全局 E-R 模型,除能准确、全面地反映用户功能的需求外,还应满足下列条件:实体类型的个数尽可能少;实体类型所含属性个数尽可能少;实体类型间的联系无冗余。但是这些条件不是绝对的,要视具体的信息需求与处理需求而定,优化原则如下。

(1) 实体类型的合并

一般把具有相同键的实体类型以及具有 1:1 联系的两个实体类型合并。

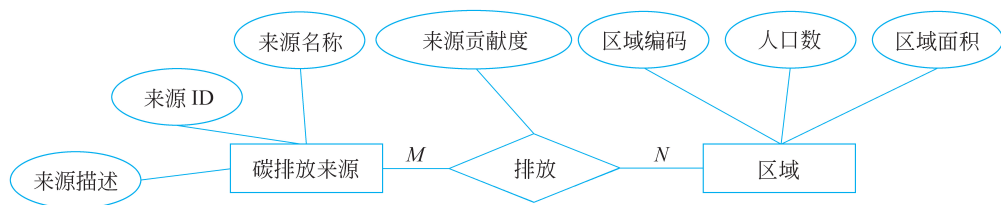
(2) 冗余属性的消除

通常在各个局部结构中是不允许冗余属性存在的。但在综合成全局 E-R 模型后,可能产生全局范围内的冗余属性。一般当同一非键的属性出现在几个实体类型中,或者一个属性值可从其他属性值导出时,应把冗余属性从全局模式中去掉。

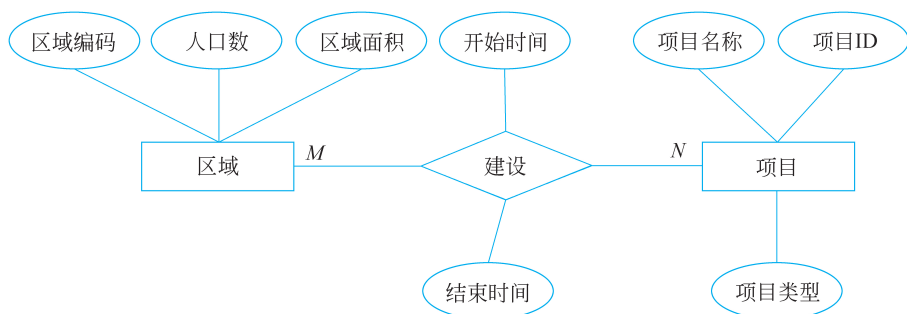
(3) 冗余联系的消除

在全局模式中可能存在冗余的联系,通常利用规范化理论中函数依赖的概念消除冗余联系。

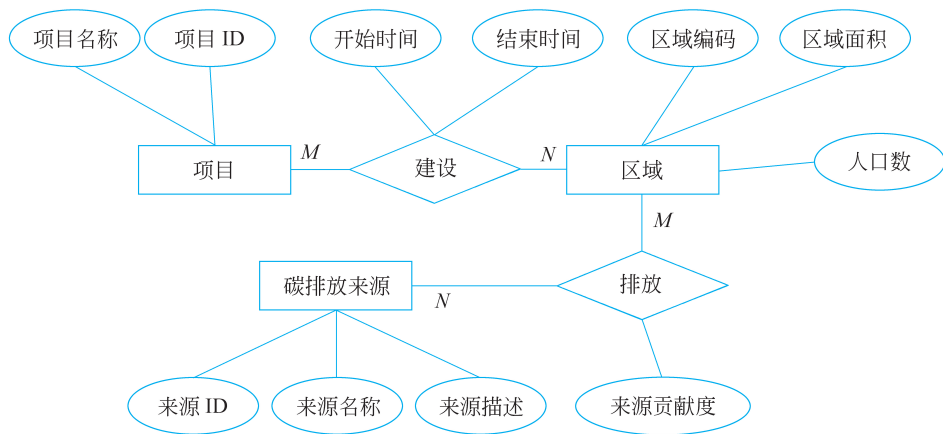
图 3.2 是由两个局部 E-R 模型集成为全局 E-R 模型的例子,该例为各区碳排放统计。



(a) 局部E-R模型1



(b) 局部E-R模型2



(c) 全局E-R模型

图 3.2 局部 E-R 模型集成为全局 E-R 模型

3.4 逻辑设计

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。

逻辑模型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。

3.4.1 逻辑设计的步骤

逻辑设计的主要任务是:将概念数据模型转换成目标 DBMS 所支持的数据模型;开发目标 DBMS 下的数据库模式和子模式,即使用选定的 DBMS 的数据定义语言来描述数据模型;同时与应用程序设计活动相结合,给出应用程序的设计指南。此外,完成这些任务的一个先决条件就是根据应用环境的特征、数据特点来确定所需要的 DBMS 功能与特征,并选择目标 DBMS。

数据库的逻辑设计可分为以下步骤。

- ① 模型转换。按不同的转换规则将 E-R 图转换为某一种结构数据模型,现有的 DBMS 支持层次模型、网状模型和关系模型。
- ② 模型评价。检查转换后的模型是否满足用户对数据的处理要求,主要包括功能要求和性能要求。
- ③ 模型修正。根据模型评价的结果调整和修正数据模型,以提高系统性能。修改后的模型要重新进行评价,直到满意为止。

3.4.2 E-R 模型向关系数据模型的转换

E-R 模型可以向现有的各种数据库模型转换,对不同的数据库模型有不同的转换规则,以下介绍向关系模型转换的规则。

E-R 模型中的主要成分是实体类型和联系类型,因此转换过程分为以下两步。

- ① 对于实体类型,可以将每个实体类型转换成一个关系模式,实体的属性即为关系模式的属性,实体标识符即为关系模式的键。

- ② 对于联系类型,要视 $1:1$ 、 $1:N$ 和 $M:N$ 三种不同的情况进行不同的处理。

若实体间的联系是 $1:1$,可以在两个实体类型转换成的两个关系模式中的任意一个关系模式的属性中加入另一个关系模式的键和联系类型的属性。

例如,要求项目的负责人与项目间存在着 $1:1$ 的关系,其 E-R 图如图 3.3 所示。在将其转换为关系模式时,项目与负责人各为一个模式。如果用户在查询项目信息时查询其负责人信息,那么可在项目模式中加入负责人 ID 和职责,负责人 ID 为外键,其关系模式设计如下:

负责人模式(负责人 ID,姓名,联系方式)

项目模式(项目 ID,项目名称,负责人 ID,项目职责)

若实体间的联系是 $1:N$ 。可以在 N 端实体类型转换成的关系模式中加入 1 端实体类型转换成的关系模式的键和联系类型的属性。

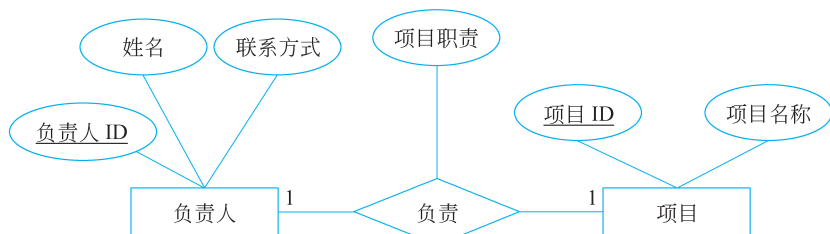


图 3.3 负责人与项目的实体联系

例如,部门与项目间存在 1 : N 联系,其 E-R 图如图 3.4 所示,转换成的关系模式为:

项目模式(项目 ID,项目名称,部门 ID,开始日期,结束日期)

部门模式(部门 ID,部门名称)

项目模式中的部门 ID 为外键。

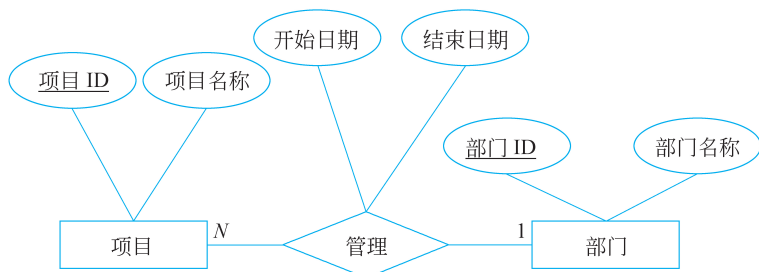


图 3.4 部门与项目的实体联系

若实体间的联系是 $M : N$,可以将联系类型转换成一个新关系模式,其属性为两端实体类型的键加上联系类型的属性,而键为两端实体键的组合。

在第 1 章介绍过的思政实践教育基地与高校的合作关系就是这种情况,即思政实践教育基地与高校之间的 $M : N$ 联系可用下面三个关系模式表示:

基地模式(基地 ID,基地名称,基地地址,联系电话,负责人,开设课程)

高校模式(高校 ID,高校名称,高校负责人)

合作模式(合作 ID,合作内容,合作时长,基地 ID,高校 ID)

合作模式中的基地 ID 和高校 ID 是两个外键。

3.4.3 关系数据库的逻辑设计

逻辑设计可以运用关系数据库模式设计理论,使设计过程形式化地进行,并且结果可以验证。关系数据库逻辑设计的过程如图 3.5 所示。

从图 3.5 中可以看出,概念设计的结果直接影响逻辑设计过程的复杂性和效率。在概念设计阶段已经把关系规范化的某些思想用作构造实体类型和联系类型的标准,在逻辑设计阶段仍然要使用关系规范化理论来设计模式和评价模式。关系数据库逻辑设计的结果是一组关系模式的定义。

关系数据库的逻辑设计过程如下。

(1) 模式转换

将概念设计的结果(即全局 E-R 模型)按转换规则转换成初始关系模式。

(2) 规范化处理

规范化的目的是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。规范化的过程分为两个步骤：

① 确定规范级别。规范级别取决于两个因素，一是归结出来的数据依赖的种类；二是实际应用的需要。在仅有函数依赖时，一般达到 3NF 或 BCNF 即可。

② 实施规范化处理。确定规范级别之后，逐一考查关系模式，判断它们是否满足规范要求。若不符合上一步所确定的规范级别，则利用相应的规范算法将关系模式规范化。

(3) 模式评价

模式评价的目的是检查已给出的数据库模式是否满足用户的功能要求，是否具有较高的效率，并确定需要加以修正的部分。模式评价主要包括功能评价和性能评价两方面。

① 功能评价：对照需求分析的结果，检查规范化后的关系模式集合是否支持用户所有的应用需求。关系模式必须包括用户可能访问的所有属性。

② 性能评价：对于目前得到的数据库模式进行性能评价是比较困难的，因为缺乏有关的物理设计因素和相应的评价手段。但可以利用逻辑记录访问算法进行估算，以给出改进建议。

(4) 模式修正

根据模式评价的结果，对已生成的模式集进行修正。修正的方式依赖于导致修正的原因，如果因为需求分析、概念设计的疏漏导致某些应用得不到支持，则应相应地增加新的关系模式或属性；如果因为性能上的考虑而要求修正，则可采用合并、分解或选用另外结构的方式进行。

① 合并：如果有若干关系模式具有相同的键，并且对这些模式的处理主要为查询操作，当同时涉及多个关系的查询占有相当比例时，可对这些模式按组合使用的频率进行合并。这样，可减少联接操作，提高查询效率。在有些特殊情况下，对即使不具有相同键的模式，也可以采用合并方式提高查询速度，但这样可能会影响规范化的等级。

② 分解：已经达到规范化要求的模式，仍然可能由于某些属性值的重复而占用过多的存储空间。如有的属性值有较少的不同值，且每一个值的长度较长，此时可对属性值实现代码化，构造一个代码转换的关系模式，以便使占用的空间达到极小化。

在经过模式评价及修正的多次反复后，最终的数据模式得以确定，全局逻辑结构设计即告结束。

在逻辑设计阶段，还要设计出全部子模式。子模式是面向各个最终用户或用户集团的局部逻辑结构设计的。子模式体现了各个用户对数据库的不同观点，它并不决定物理存放的内容，而仅是用户的一个视图(View)。

关系子模式除了指出某一类型的用户所用到的数据类型外，还要指出这些数据与模式中相应数据的关系和对应性。

逻辑设计的结果是逻辑结构设计说明书。逻辑结构设计说明书包括应用设计指南、物理设计指南、模式及子模式的集合。

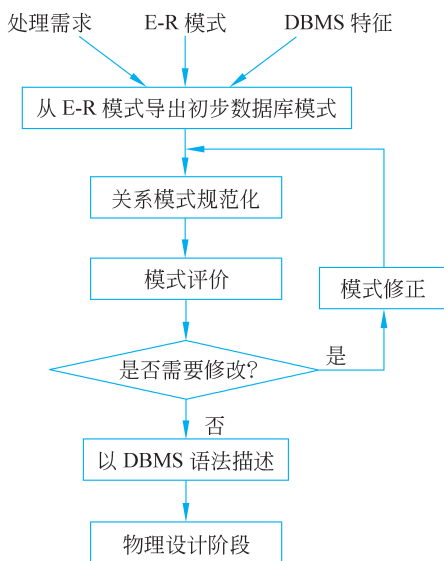


图 3.5 关系数据库逻辑设计的过程

3.5 物理设计

数据库物理设计的任务就是为上一阶段得到的逻辑数据库选择一个最适合应用环境的物理结构,也就是确定在物理设备上能有效地实现一个逻辑数据模型所必须采取的存储结构和存取方法,然后对该存储模式进行性能评价和修改设计。经过多次反复,最后得到一个性能较好的存储模式。

数据库物理设计的主要目标是提高数据库的性能,节省存储量。在这两个目标中,提高数据库性能更为重要。因为在目前的大多数数据库系统中,性能仍然是主要的薄弱环节,也是用户最关切的问题。

3.5.1 物理设计的内容

一般说来,物理设计就是根据一个满足用户信息需求的已确定的逻辑数据库结构研制出一个有效的、可实现的物理数据库结构的过程。物理设计通常包括满足某些操作约束,如存储空间的限制和响应时间的要求。

数据库的物理设计与具体 DBMS 有关,主要包括物理数据库结构设计的三个内容和涉及约束及程序设计的两个内容。

1. 确定记录存储格式

数据库中每条记录数据项的类型和长度要根据用户要求及数据值的特点来确定。一般 DBMS 提供多种数据类型可以进行选择。如字符型的数据可用字符或二进制位串来表示,如果数据项的位在一个不大的有限集内,用二进制位串来表示可以节约存储空间。

为加快存取速度,可把记录数据按不同应用进行水平或垂直分割,把它们分别存储在不同的设备或同一设备的不同位置上,尽可能地使应用程序访问数据库的代价最小。

2. 选择文件的存储结构

文件存储结构的选择与对文件进行的处理有关。对需要成批处理的数据文件,可选用顺序存储结构,而经常需要随机查询某一记录时,则选用散列方式的存储结构比较合适。对一些 DBMS,有多种存储结构可供选择。如 IMS 中有 4 种存储结构:层次顺序存取法、层次索引顺序存取法、层次直接存取法和层次索引直接存取法。

在有些 DBMS 中还支持聚集索引,采用聚集索引可使记录的物理存储顺序与主关键字值顺序相同,从而可以提供按主关键字查询的最高效率。

3. 决定存取路径

一个文件的记录之间以及不同文件的记录之间都存在着一定的联系。因此,对于一个记录的存取可根据应用的不同而选择不同的存取路径,以提高处理效率。物理设计的任务之一就是要确定和建立这些存取路径。

在关系数据库系统中,可通过建立索引来提供不同的存取路径。需要在哪些属性上建立索引,哪些是主索引,哪些是次索引,索引的键是单属性还是属性的组合,这些都是设计中需要解决的问题。

当然,索引的建立会增加系统开销,数据更新时要同时更新索引,降低了数据更新操作的效率。

4. 完整性和安全性

数据库在物理设计时,同样必须在系统的完整性、安全性等方面进行分析,并产生多种方案。在实施数据库前,对这些方案进行细致的评价以选择一个较优的方案是十分必要的。

5. 程序设计

逻辑数据库结构确定以后,就可以开始应用程序的设计了。从理论上说,数据库的物理数据独立性的目的,是消除由于物理结构设计决策的变化而引起的对应用程序的修改。但是,当物理数据独立性未得到保证时,可能会发生对程序的修改。

3.5.2 物理设计的性能

假设数据库性能用“开销”(Cost)来描述,在数据库应用系统生存期中,总的开销包括规划开销、设计开销、实施和测试开销、操作开销、运行维护开销。

对物理设计者来说主要考虑操作开销,即为用户获得及时、准确的数据所需的开销和计算机资源的开销。开销可分为如下几类。

1. 查询和响应时间

响应时间定义为从查询开始到查询结果开始显示之间所经历的时间,包括 CPU 服务时间、CPU 队列等待时间、I/O 服务时间、I/O 队列等待时间、封锁延迟时间和通信延迟时间。

一个好的应用程序设计可以减少 CPU 服务时间和 I/O 服务时间。例如,有效地使用数据压缩技术、选择好的访问路径和合理安排记录的存储等。

2. 更新事务的开销

应用程序的执行划分为若干比较小的独立程序段,这些程序段称为事务。事务的开销是用从事务的开始到完成的时间来度量的。更新事务的开销主要指修改索引、重写物理块或文件、写校验等方面的开销。

3. 报告生成的开销

报告生成是一种特殊形式的查询检索,它花费的时间和查询、更新是相同的,都是从数据输入的结束到数据显示的开始这段时间,主要包括检索、重组、排序和结果显示。

4. 主存储空间开销

主存储空间开销包括程序和数据所占有的空间,一般地,对数据库设计者来说,可以对缓冲区分别进行适当的控制,包括控制缓冲区的个数和大小。

5. 辅助存储空间

辅助存储空间分为数据块和检索块两种,块中的开销包括标志、计数、指针和自由空间等。设计者可以控制的是索引块的大小、装载因子、指针选择项和数据冗余等。

实际上,数据库设计者能有效控制的是 I/O 服务和辅助空间,有限控制的是封锁延迟、CPU 时间和主存空间,完全不能控制的是 CPU 和 I/O 队列等待时间、数据通信延迟时间。

物理设计的结果是物理设计说明书,包括存储记录格式、存储记录位置分布以及存取方法,并给出对硬件和软件系统的约束。

3.6 实现与维护

3.6.1 数据库的实现

对数据库的物理设计步骤初步评价完成后,就可以建立数据库了。在这一阶段,设计人员

运用 DBMS 提供的数据库定义语言,将逻辑设计和物理设计的结果严格地描述出来,成为 DBMS 可接受的源代码,经过调试产生目标模式,然后组织数据入库。

根据逻辑设计结果和物理设计结果,在计算机上建立实际的数据库结构,装入数据,并测试和运行的过程称为数据库的实现。该阶段的主要工作如下。

(1) 建立实际数据库结构

用 DBMS 提供的数据库定义语言编写描述逻辑设计和物理设计结果的程序,经计算机编译处理和执行后,即建立了实际的数据库结构。

(2) 试运行

数据库结构建立好后,应装入实验数据,进入数据库的试运行阶段。该阶段的主要工作是实际运行应用程序,执行对数据库的各种操作,测试应用程序的功能;测量系统的各项性能指标,检查对空间的占用情况,分析是否符合设计目标。

(3) 装入实际数据并建立实际的数据库

向数据库中装入数据又称数据库加载。在加载之前要对数据作严格的检验和整理,并建立严格的数据登录和校验规范,设计出完善的数据检验和校正程序,尽可能在加载之前把不合格的数据排除掉。然后通过系统提供的工具程序或自编的专门装入程序,将数据装入数据库。在数据库加载过程中,还必须做好数据库的转储和恢复工作。

3.6.2 数据库的其他设计

数据库的其他设计工作包括加强数据库的安全性、完整性控制,保证一致性、可恢复性等,这些设计总是以牺牲效率为代价的。设计人员的任务就是要在提高效率和实现尽可能多的功能之间进行合理平衡。其他设计如下。

(1) 数据库的再组织设计

对数据库的概念、逻辑和物理结构的改变称为再组织(Reorganization),其中改变概念或逻辑结构又称再构造(Restructuring),改变物理结构称为再格式化(Reformatting)。再组织通常是由于环境需求的变化或性能变化等原因而引起的。一般 DBMS 特别是 RDBMS 都提供数据库的再组织实用程序。

(2) 故障恢复方案设计

在数据库设计中考虑的故障恢复方案,一般是基于 DBMS 提供的故障恢复手段,如果 DBMS 已提供了完善的软硬件故障恢复和存储介质故障恢复手段,那么设计阶段的任务就简化为确定系统登录的物理方案参数,如缓冲区的个数、大小、逻辑块的长度、物理设备等,否则就要制定人工备份方案。

(3) 安全性考虑

许多 DBMS 都有描述各种对象(记录、数据项)的存取权限的功能。在设计时根据用户需求分析,规定相应的存取权限。子模式是实现安全性要求的一个重要手段。也可在应用程序中设置密码,对不同的使用者给予一定的密码,用密码控制使用级别。

(4) 事务控制

大多数 DBMS 都支持事务的概念,以保证多用户环境下的数据完整性和一致性。事务控制有人工控制和系统控制两种控制方法,系统控制以数据操作语句为单位,人工控制则以事务的开始和结束语句来实现。大多数 DBMS 也提供封锁粒度的选择,封锁粒度一般有库级、记录级和数据项级。粒度越大控制越简单,但并发性能差,这些在设计中都要统筹考虑。

3.6.3 数据库的运行与维护

数据库正式投入运行标志着数据库运行与维护工作的开始,但并不标志着数据库设计工作的结束。数据库维护工作不仅是维持其正常运行,而且是设计工作的继续和提高。

数据库运行维护阶段的主要工作如下。

① 维护数据库的安全性、完整性及系统的转储和恢复。按照系统提供的安全规范和故障恢复规范,经常核查系统安全性是否受到侵犯,及时调整授权和密码,实施系统转储与备份,发生故障后应及时恢复。

② 性能的监督、分析与改进。利用系统提供的性能分析工具对数据库的存储空间及响应时间进行分析、评价,并结合用户意见确定改进措施,实施重新构造或重新格式化。

③ 增加新功能。根据用户的意见,在不损害原系统功能和性能的情况下,对原有功能进行扩充。

④ 发现错误,修改错误。及时发现系统运行中出现的错误,并修改错误,保证系统正常运行。

由于数据库应用环境发生变化,需要增加新的应用或新的实体,实体与实体的关系也会发生相应的变化,原设计不能很好地满足新的需求,不得不适当调整数据库的外模式和内模式。当然,数据库重新构造的程序功能是有限的,只能进行部分的修改和调整,若应用变化太大,重新构造也无能为力了,则表明原数据库应用系统生存期的结束,应该重新设计数据库,开始一个新的数据库应用系统的生存期。

3.7 习题

一、选择题

- 关于 E-R 模型向关系数据模型转化的说法,错误的是()。
 - 一个实体类型转换成一个关系模式
 - 一个 1:1 关系只能转换为一个独立的关系模式,不能与其他联系的关系模式合并
 - 一个 1:N 关系可以转换为一个独立的关系模式,也可以与联系的任意 N 端实体所对应的关系模式合并
 - 三个或三个以上的实体间的多元关系转换为一个关系模式
- 将 E-R 模型按转换成关系模式后进行的规范化处理,一般达到()即可。
 - 1NF
 - 2NF
 - 3NF
 - BCNF
- 对物理设计主要考虑数据库性能的()。
 - 操作开销
 - 运行维护开销
 - 规划开销
 - 设计开销
- 数据库实现阶段不包括()工作。
 - 建立实际的数据库结构
 - 装入实验数据对应用程序进行测试
 - 装入实际数据,建立起实际的数据库
 - 增加新的功能
- 由局部 E-R 模型生成初步 E-R 模型的主要任务是()。
 - 消除不必要的冗余
 - 消除属性冲突
 - 消除结构冲突和命名冲突
 - B 和 C
- 不属于数据库设计内容的是()。

- A. 静态特性设计 B. 物理设计 C. 动态特性设计 D. 数据库测试
7. 将 E-R 模型转换为关系模式的数据库设计阶段是()。
- A. 需求分析 B. 概念设计 C. 逻辑设计 D. 物理设计
8. 同一实体在不同局部 E-R 模型中属性组成不同,属于()。
- A. 结构冲突 B. 属性冲突 C. 命名冲突 D. 实体冲突
9. 改变数据库概念或逻辑结构称为()。
- A. 再组织 B. 再构造 C. 再格式化 D. 再重用
10. 数据的最小单位是()。
- A. 数据项 B. 数据结构 C. 数据存储 D. 数据流

二、填空题

1. _____是从“数据”和“处理”两方面表达数据处理过程的一种图形化的表示方法。
2. 最常用的概念设计策略是_____的设计策略。
3. E-R 模型是描述概念世界、建立_____的使用工具。
4. 用户获得及时、准确的数据所需的开销和计算机资源的开销称为_____。
5. 在需求分析阶段,通过调查要从用户获得数据库的_____需求、_____需求和_____需求。
6. 数据库设计步骤包括规划、_____、概念设计、逻辑设计、物理设计、实现与维护。
7. 需求分析阶段需形成的文档是_____。
8. 将局部 E-R 模型合并成全局 E-R 模型时,应消除_____冲突、_____冲突和_____冲突。
9. 对数据库的概念、逻辑和物理结构的改变称为_____。
10. 生成 E-R 模型的数据库设计阶段是_____。

三、简答题

1. 试述数据库设计的主要内容。
2. 试述数据库设计的基本步骤。
3. 试述需求分析的步骤。
4. 试述概念模型的创建过程。
5. 举例说明由 E-R 模型向关系数据模型的转换规则。

第4章 SQL Server 2022 概述

本章主要介绍 SQL Server 的发展、特点、组件和技术,SQL Server 2022 的特点、版本,SQL Server 2022 的软硬件安装环境、安装过程及主要实用工具的用途,SQL Server 2022 注册服务器及 Management Studio 的使用,以及 SQL、Transact-SQL 语言概述。

4.1 SQL Server 简介

SQL Server 是 Microsoft 公司的一个关系数据库管理系统(RDBMS),是基于客户机/服务器(Client/Server,C/S)应用模式的系统,能直接处理 XML 数据,可以与 Internet 紧密结合。用户可以通过图形和命令两种方式使用 SQL Server 进行检索和更新数据库中的数据等操作。

4.1.1 SQL Server 的发展

SQL Server 是当今使用广泛的关系数据库系统之一。它最初是由 Microsoft、Sybase 和 Ashton-Tate 三家公司共同开发的,于 1988 年推出了第一个 OS/2 版本。在 Windows NT 推出后,Microsoft 与 Sybase 公司在 SQL Server 的开发上选择了不同的平台。Microsoft 公司将 SQL Server 移植到 Windows NT 系统上,并专注于开发推广 Windows 操作系统上的 SQL Server 版本,而 Sybase 公司则专注于 SQL Server 在 UNIX 操作系统上的开发与应用。

1996 年 Microsoft 公司推出了 SQL Server 6.5 版本,1998 年推出了 SQL Server 7.0 版本。随后,经过不断的修改完善,2000 年 Microsoft 公司推出了 SQL Server 2000 版本,引入了 XML 支持、存储过程、触发器等新特性;SQL Server 2005 则增加了数据分区、数据库镜像、T-SQL 查询增强等功能。随着云计算和大数据技术的快速发展,SQL Server 也在不断演进,以适应新的技术趋势和业务需求。例如,SQL Server 2012 引入了 AlwaysOn 高可用性解决方案、列存储索引等新技术,以提高数据库的性能和可靠性;SQL Server 2016 则进一步增强了与云计算的集成能力,提供了对 Linux 的支持,并引入了内存中 OLTP 等新技术。2022 年 11 月 16 日,微软正式发布了 Microsoft SQL Server 2022,旨在为企业和组织提供高效、可靠和智能的数据存储和管理解决方案,帮助用户更好地处理和分析数据,从而推动业务发展和创新。

4.1.2 SQL Server 的特点

1. 支持客户机/服务器结构

SQL Server 是支持客户机/服务器结构的数据库管理系统。客户机/服务器结构把整个数据处理的任务划分为在客户机上完成的任务和在数据库服务器上完成的任务。客户机用于运行数据库应用程序,服务器用于执行 DBMS 功能。在客户机上的数据库应用程序也称为前端系统,它负责系统与用户的交互和数据显示,在服务器上的后端系统负责数据的存储和管

理。客户机/服务器结构的优点是,数据库服务器仅返回用户所需要的数据,在网络上的数据流量将大大减少,可以加速数据的传输;数据存储在服务服务器上,而不是分散在各个客户机上,这样所有用户都可以访问到相同的数据,而且数据的备份和恢复也很容易。

2. 分布式数据库功能

SQL Server 支持分布式数据库结构,可以将逻辑上的一个整体数据库的数据分别存放在各个不同的 SQL Server 服务器上,客户机可以分别或同时向多个 SQL Server 服务器存取数据,这样可以降低单个服务器的处理负担,提高系统执行效率。

分布式查询可以引用来自不同数据源的数据,而且这些对用户来说是完全透明的。分布式数据库将保证任何分布式数据更新的完整性。通过复制可以使用户能够维护多个数据副本,这些用户能够自主地进行工作,然后再将所做的修改合并到发布数据库中。

3. 与 Internet 集成

SQL Server 数据库引擎提供完整的 XML 支持。它还具有构成最大的 Web 站点的数据存储组件所需的可伸缩性、可用性和安全功能,使用户很容易将数据库中的数据发布到 Web 页面上。

4. 可伸缩性和可用性

同一个数据库引擎可以在 Windows 不同版本及大型多处理器计算平台上使用。SQL Server 企业版支持联合服务器、索引视图和大型内存支持等功能,使其得以升级到最大 Web 站点所需的性能级别。

5. 数据仓库功能

SQL Server 提供用于析取和分析汇总数据功能,以与联机分析处理的工具连接。SQL Server 中还包括一些工具,可用来直观地设计数据库并通过 English Query 来分析数据。

4.1.3 SQL Server 的组件和技术

1. SQL Server 数据库引擎

数据库引擎是用于存储、处理和保护数据的核心服务。数据库引擎提供了受控访问和事务处理,以满足企业内最苛刻的数据消费应用程序的要求。数据库引擎还通过业务连续性和数据库恢复,为保持业务连续性提供全面的支持。

2. 机器学习服务

SQL Server 机器学习服务(数据库内)将 R 语言和 Python 语言与 SQL Server 集成,方便用户通过调用存储过程,轻松生成、重新定型模型,并对模型评分。

3. SQL Server Integration Services

SQL Server Integration Services(SSIS)是一个可用于构建高性能数据集成解决方案的平台,其中包括为数据仓库提供提取、转换和加载(ETL)处理的包。

4. SQL Server Analysis Services

SQL Server Analysis Services(SSAS)是一个针对个人、团队和公司商业智能的分析数据平台和工具集。服务器和客户端设计器通过使用 PowerPivot、Excel 和 SharePoint Server 环境,支持传统的 OLAP 解决方案、新的表格建模解决方案以及自助式分析和协作。Analysis Services 还包括数据挖掘,以便用户可以发现隐藏在大量数据中的模式和关系。

5. SQL Server 复制

复制是在数据库之间对数据和数据库对象进行复制和分发,然后在数据库之间进行同步

以保持一致性的一组技术。使用复制可以将数据通过局域网、广域网、拨号连接、无线连接和 Internet 分发到不同位置,以及分发给远程用户或移动用户。

6. Master Data Services

Master Data Services 是用于主数据管理的 SQL Server 解决方案。基于 Master Data Services 生成的解决方案可帮助用户确保报表和分析均基于适当的信息。使用 Master Data Services,用户可以为数据创建中央存储库,并随着主数据随时间变化而维护一个可审核的安全对象记录。

7. SQL Server Reporting Services

SQL Server Reporting Services 提供支持 Web 的企业级报表功能。从而使用户可以创建从多个数据源提取内容的报表,发布各种格式的报表,以及集中管理安全性和订阅。

8. Data Quality Services

Data Quality Services(简称 DQS)提供知识驱动型数据清理解决方案。DQS 使用户可以生成知识库,然后使用此知识库,同时采用计算机辅助方法和交互方法,执行数据更正和数据消除重复的操作。用户可以使用基于云的引用数据服务,并可以生成一个数据管理解决方案将 DQS 与 SQL Server Integration Services 和 Master Data Services 相互集成。

4.1.4 SQL Server 2022 的特点

SQL Server 2022 (16.x)在早期版本的基础上构建,旨在将 SQL Server 发展成一个平台,以提供开发语言、数据类型、本地或云环境以及操作系统选项。

1. 通过 Azure 实现业务连续性

通过 Azure SQL 托管实例中的链接功能,在云中实现完全托管的灾难恢复,确保正常运行时间,实现连续将数据复制到云和从云中复制数据。

2. 对本地操作数据进行无缝分析

打破操作与分析存储之间的壁垒,实现近实时的洞察。通过 Azure Synapse Link 在云中运行 Spark 和 SQL 运行时来分析所有数据。

3. 直观了解用户整个数据资产

使用 Microsoft Purview 管理和治理整个数据资产,以克服数据孤岛。

4. 安全可靠性能提高

SQL Server 被评为过去 10 年间最不易受攻击的数据库,实现安全性和合规性目标。使用不可变账本来帮助保护数据不被篡改。

5. 行业领先的性能和可用性

SQL Server 利用性能和可用性方面的改进来提高查询速度,并确保业务连续性。无须更改代码即可提高查询性能和加速优化调整。当用户位于多个位置时,保持多写入环境平稳运行。

4.2 SQL Server 2022 的安装准备

4.2.1 SQL Server 2022 的版本

根据应用程序的需要,安装要求可能有很大不同。SQL Server 2022 的不同版本能够满

是企业和个人独特的性能、运行时的需求以及价格要求。需要安装哪些 SQL Server 2022 组件也要根据企业或个人的需求而定。SQL Server 版本介绍如表 4.1 所示。

表 4.1 SQL Server 版本介绍

版本	定 义
Evaluation Edition	Evaluation Edition(试用版)包含最大的 SQL Server 功能集。仅供测试使用,不能用于正式的商业环境中,试用期 180 天,试用期过后需付费激活
Enterprise Edition	作为高级产品/服务,Enterprise Edition(企业版)提供了全面的高端数据中心功能,具有极高的性能和无限虚拟化,还具有端到端商业智能,可为任务关键工作负载和最终用户访问数据间接提供高级别服务。企业版可用于评估,评估部署的有效期为 180 天
Standard Edition	Standard Edition(标准版)提供了基本数据管理和商业智能数据库,使部门和小型组织能够顺利运行其应用程序并支持将常用开发工具部署于内部或云,有助于以最少的 IT 资源获得高效的数据库管理
Web Edition	对于 Web 主机托管服务提供商和 Web VAP 而言,Web Edition 版本是一项总拥有成本较低的选择,可针对从小规模到大规模 Web 资产等内容提供可伸缩性、经济性和可管理性能力
Developer Edition	Developer Edition(开发人员版)支持开发人员基于 SQL Server 构建任意类型的应用程序。它包括 Enterprise 版的所有功能,但有许可限制,只能用作开发和测试系统,而不能用作生产服务器
Express Edition	Express Edition(简化版)是入门级的免费数据库,是学习和构建桌面及小型服务器上数据驱动应用程序的理想选择

4.2.2 SQL Server 2022 的安装环境

安装环境是 SQL Server 2022 对硬件、软件环境的要求,这些要求也是 SQL Server 2022 运行时必需的条件。在以下的环境中可以安装、使用 SQL Server 2022 数据库系统。本书主要介绍 64 位 SQL Server 2022 的安装环境。

1. 硬件环境

硬件配置的高低会直接影响软件的运行速度,用 SQL Server 存储和管理数据的特点是数据量大,且对数据进行的查询、修改和删除等操作频繁,更主要的是要保证多人同时高效地访问数据库,这对硬件性能要求较高。

(1) 存储

SQL Server 2022 要求最少 6GB 的可用硬盘驱动器空间。磁盘空间要求随所安装的 SQL Server 组件不同而有所不同。

(2) 显示器

SQL Server 2022 要求有 Super-VGA (800 像素×600 像素)或更高分辨率的显示器。

(3) Internet

使用 Internet 功能需要连接 Internet。

(4) 内存

最低要求: Express Edition 为 512MB,所有其他版本为 1GB;推荐: Express Edition 为 1GB,所有其他版本至少为 4GB,并且应随着数据库大小的增加而增加以确保最佳性能。

(5) 处理器

① 处理器速度: 其最低要求是, x64 处理器, 主频 1.4GHz, 推荐: 处理器速度为 2.0GHz 或更快;

② 处理器类型: 采用 x64 处理器, AMD Opteron、AMD Athlon 64、支持 Intel EM64T 的