

第五章

电子商务数据库设计

知识目标

- » 理解电子商务数据库设计的特点；
- » 掌握数据库设计的基本步骤,数据库设计各个阶段的具体设计内容、设计描述、设计方法；
- » 掌握建立概念模型的方法,E-R 图的设计方法,E-R 图向关系模型的转换,数据模型的优化；
- » 了解电子商务数据库物理设计的内容和方法,数据库的实施和维护；
- » 了解数据库设计工具软件 PowerDesigner 的使用。

技能目标

- » 理解 E-R 图向关系模型的转换；
- » 初步掌握根据实际应用设计电子商务数据库的方法。

数据库设计是电子商务系统开发和建设的重要组成部分,从广义的角度看,电子商务数据库设计是数据库及其应用系统的设计,即设计整个电子商务数据库应用系统。从狭义的角度看,电子商务数据库设计是设计数据库本身,即设计数据库的各级模式并建立数据库,这是数据库应用系统设计的一部分。本章重点介绍狭义的电子商务数据库设计。当然,设计一个好的数据库与设计一个好的电子商务数据库应用系统是密不可分的,一个好的数据库结构是电子商务应用系统的基础,在实际的系统开发中两者是密切相关,并行进行的。

第一节 电子商务数据库设计概述

电子商务数据库设计实际上就是根据实际的应用问题建立关系数据库及相应的应用系统,一个电子商务系统的各部分能否紧密地结合在一起以及如何结合,关键在于数据库。因此只有对数据库进行合理的逻辑设计和物理设计,才能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,最终开发出完善而高效的电子商务应用系统。

一、数据库设计的特点

数据库建设是指数据库应用系统从设计、实施到运行与维护的全过程。它和一般的软件系统的设计、开发与维护有许多相同之处,但也有其自身的一些特点。

(一) 三分技术、七分管理、十二分基础数据

“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。要建设好一个电子商务数据库应用系统,开发技术固然重要,但相比之下管理更为重要,管理不仅仅指项目的管理,更重要的是企业的业务管理。数据库结构的设计就是对企业中业务部门的数据以及各个业务部门之间数据联系的描述和抽象,而业务部门的内部数据以及各个业务部门之间数据的联系是与各个部门的职能,以及整个企业的管理模式密切相关的。因此企业的业务管理对数据库结构的设计有着直接影响。

在数据库设计中,基础数据在数据库建设中的地位和作用更是不容忽视,数据的收集、整理、组织和不断更新是数据库建设的重要环节。其中,基础数据的收集、入库是数据库建立初期工作量最大、最烦琐、最细致的工作;在数据库运行过程中需要不断地把新的数据加到数据库中,使数据库成为一个“活库”。数据库一旦成了“死库”,系统也就失去了应用价值。

(二) 结构设计和行为设计相结合

数据库设计要达到预期的目标,在整个设计过程中必须强调结构设计和行为设计相结合,这也是数据库设计的第二大特点。但是,由于建立数据模型的方法并没有给行为特性设计提供任何依据,导致结构设计和行为设计不得不分离进行。早期的结构化设计方法着重于数据处理过程的特性,即行为特性,一般尽量推迟数据结构设计,这种方法对数据库应用系统的设计是不妥的;而早期的数据库设计致力于数据模型和数据建模方法的研究,着重于

结构特性的设计,而忽视行为设计对结构设计的影响,此方法也是不完善的。因此,要强调在数据库设计中把结构特性和行为特性结合起来,并将其作为数据库设计的重要特点。

应该说结构设计是关键,因为数据库正是从分析用户的行为所涉及的数据中汇总出来的。整个设计过程是一种“反复探寻、逐步求精”的过程,不可能一蹴而就。同时数据库设计中,数据库设计者还应当具有战略眼光,考虑到当前、近期和远期某个时间段对系统的需求,设计的系统不仅应当完全满足用户当前和近期对系统的数据需求,而且对远期的数据需求有相应的处理方案。数据库系统设计者应充分考虑到系统可能的扩充与改变,使设计出的应用系统具有较长的生命力。

二、数据库设计的步骤

(一) 准备工作

在数据库设计之前,首先要选定参加设计的人员。参与数据库设计的人员包括系统分析员、数据库设计人员、应用开发人员、数据库管理人员和用户代表。其中系统分析员和数据库设计人员是数据库设计的核心人员,他们自始至终都参与数据库设计,其水平将决定数据库系统的质量;用户和数据库管理人员在数据库设计中也起着举足轻重的作用,他们主要参加需求分析和数据库的运行和维护,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;应用开发人员分别负责编制程序和准备软硬件环境,一般在系统实施阶段参与进来。

如果所设计的系统比较复杂,还应考虑是否需要使用数据库设计工具以及选用何种工具,以提高数据库设计质量并减少设计的工作量。

(二) 数据库设计的基本步骤

按照规范设计的方法,考虑数据库及其应用系统开发的全过程,将数据库设计分为6个阶段,如图5-1所示。

在数据库设计过程中,需求分析和概念结构设计是独立于任何DBMS进行的,逻辑结构设计与物理结构设计与选用的DBMS密切相关。

1. 需求分析阶段

需求分析是数据库设计的第一步,也是最困难、最耗时间的一步。进行数据库设计时首先必须准确了解与分析用户的数据需求 and 处理需求,弄清系统要达到的目标和实现的功能。作为“地基”的需求分析是否做得充分与准确,决定着在其上构建数据库的速度与质量。需求分析做得不好,会影响整个系统的性能,甚至会导致整个数据库设计返工重做。

2. 概念结构设计阶段

概念结构设计是整个数据库设计的关键。在概念结构设计过程中,通过对用户需求进行综合、归纳与抽象,形成一个独立于具体DBMS的概念模型。此步骤即设计E-R模型。

3. 逻辑结构设计阶段

逻辑结构设计阶段的主要任务是将概念结构转换为某个DBMS所支持的数据模型,并将其性能进行优化。即将E-R图转化为关系模型,最终生成表,并确定表中的列。

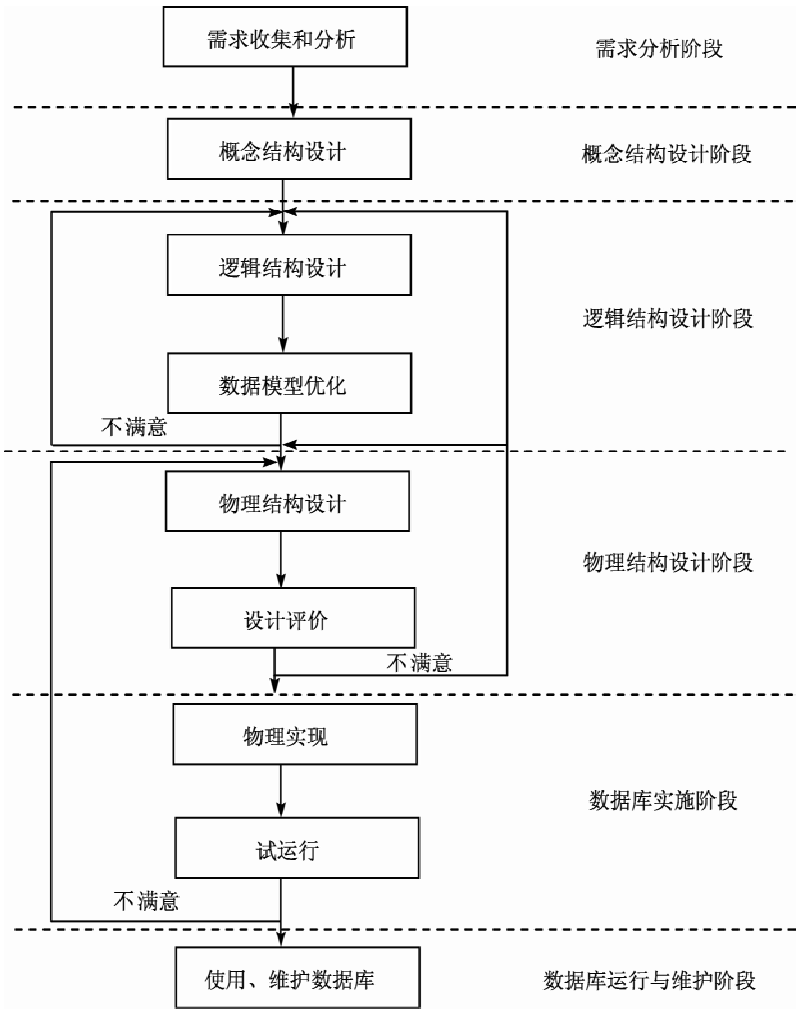


图 5-1 数据库设计步骤

4. 物理结构设计阶段

物理结构设计的主要任务是为逻辑数据模型选取一个最适合应用环境的物理结构,包括数据存储结构和存取方法,真正实现规划好的数据库。它是将一个满足用户信息需求的已确定的逻辑结构转化为一个有效的、可实现的物理数据库结构的过程。

5. 数据库实施阶段

在数据库实施阶段中,系统设计人员要运用 DBMS 提供的数据库操作语言,如 SQL,以及宿主语言,根据数据库的逻辑设计和物理设计的结果建立数据库、编制与调试应用程序、组织数据入库并进行系统试运行。

6. 数据库运行与维护阶段

数据库应用系统经过试运行后即可投入正式运行。在数据库系统运行过程中,必须不断地对其结构和性能进行评价、调整 and 修改。

设计一个完善的数据库应用系统是不可能一蹴而就的,它往往是上述 6 个阶段的不断往复。需要指出的是,这 6 个设计步骤既是数据库设计的过程,也包括了数据库应用系统的设计过程。在设计过程中,应把数据库的结构设计和数据处理的操作设计紧密结合起来,这两个方面的需求分析、数据抽象、系统设计及实现等各个阶段应同时进行,相互参照和相互补充。事实上,如果不了解应用环境对数据的处理要求或没有考虑如何去实现这些处理要求,是不可能设计出一个良好的数据库结构的。

本章主要讨论关于数据特性的描述以及如何在整个设计过程中参照处理特性的设计来完善数据模型设计等问题。关于处理特性的设计方法和原理,需参照计算机软件工程、信息系统设计课程的知识,在此不作介绍。

(三) 数据库设计过程中的各级模式

在数据库设计的不同阶段形成了数据库的各级模式,如图 5-2 所示。

图 5-2 表明,在需求分析阶段,设计的中心工作是综合各个用户的应用需求。在概念设计阶段,要形成与计算机硬件无关的、与各个 DBMS 产品无关的概念模型(即 E-R 图)。在逻辑设计阶段,要完成模式和外模式的设计工作,即首先要将 E-R 图转换成具体的数据库产品支持的数据模型,形成数据库逻辑模式,然后根据用户处理的要求、安全性的考虑,在基本表的基础上再建立必要的视图,形成数据的外模式。在物理设计阶段,要根据 DBMS 的特点和处理的需要,进行物理存储安排,建立索引,得出数据库的内模式。

本章就是以如图 5-1 所示的数据库设计步骤为主线,以某电脑配件专卖店连锁公司管理系统中的数据库设计为例,讨论数据库各设计阶段的设计内容、设计方法和工具。

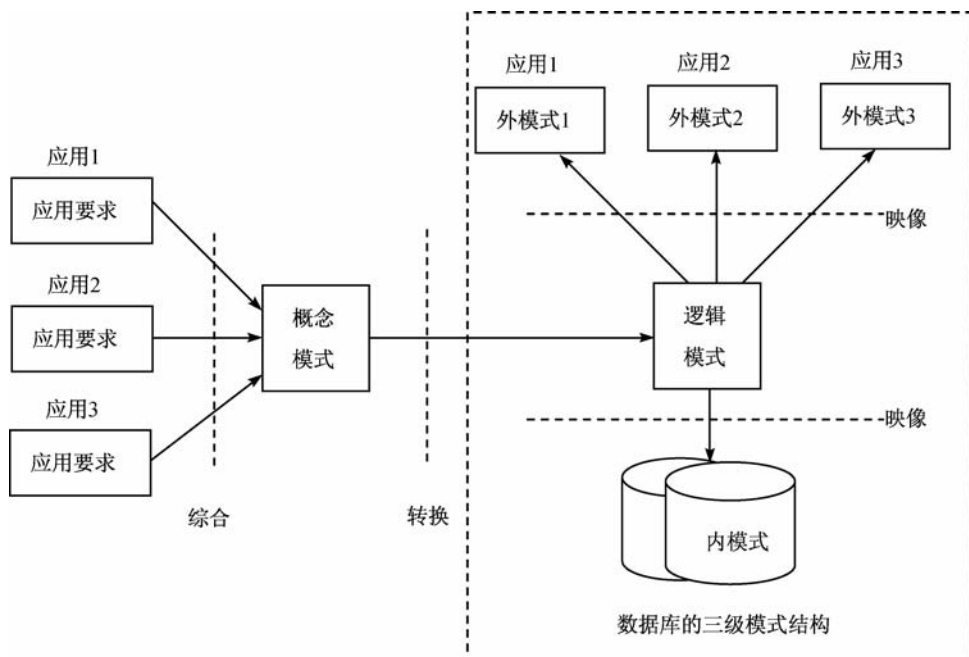


图 5-2 数据库的各级模式

第二节 需求分析

需求分析就是分析用户的要求。需求分析的结果能否准确地反映用户的实际要求,将直接影响后面各个阶段的设计,并影响系统的设计是否合理和实用。

一、需求分析的任务和方法

需求分析是数据库设计的起点和基础,也是其他设计阶段的依据。

(一) 需求分析的任务

需求分析的主要任务是:通过对现实世界电子商务应用的详细调查、研究,全面了解要处理的对象(组织、部门、企业)等;收集支持系统目标的基础数据及其处理方法,明确不同用户对数据库的各种需求,充分了解原系统(手工系统或计算机系统)的概况和发展前景;确定新系统的功能和边界。

1. 系统需求调查的内容

系统需求调查的具体内容有以下 3 方面:

(1) 数据库中的信息需求。数据库中的信息内容指出未来系统用到的所有信息及其联系,数据库中需存储哪些数据;用户希望从数据库中获取什么信息,包括用户将从数据库中直接获得或者间接导出的信息的内容和性质。

(2) 功能需求。功能需求的内容包括用户希望未来系统要完成什么数据处理功能,各功能有无优先次序;用户对数据处理的响应时间和频率要求;数据处理的工作方式是批处理还是联机处理。

(3) 数据安全性和完整性需求。数据安全性是指数据的保密措施和存取控制要求,明确哪些信息需要保密,不同用户对数据库的访问权限等;数据完整性是指数据自身的或数据间的约束限制,明确哪些数据在数据库中才算是正确的数据。

2. 系统需求的主要工作

(1) 明确未来系统所涉及的用户的当前职能,调查对象的组织机构情况、业务活动及其流程,确定系统范围。

(2) 确定用户对未来系统的各种要求,包括信息要求、处理要求、安全性和完整性要求。必须重点了解不同用户在业务活动中的数据来源,输入数据的格式、范围;确定用户希望经过系统处理后的输出数据的内容;明确哪些用户有权查看输出的数据。

(3) 深入分析不同用户的业务处理。使用数据流图工具表达对整个系统的数据流向和对数据进行的处理,描述数据和处理之间的关系。

(4) 分析系统数据,产生数据字典。

关于数据流图的详细内容需参照计算机软件工程、信息系统设计课程的知识,在此不作介绍。

（二）系统需求分析方法

调查了解了用户的需求以后,需要进一步分析和表达用户的需求。分析和表达用户需求的方法很多,常用的有结构化分析方法(简称 SA 方法),它采用自顶向下、逐层分解的方式分析系统。分析的结果用数据流图、数据字典等工具来表示。数据流图表达了数据和处理过程之间的关系,系统中的数据则借助数据字典来描述。

二、数据字典及其作用

数据字典(data dictionary,DD)是对系统中各类数据的详尽描述,是所有与系统相关的数据元素的有组织的列表,并且包含了对这些数据元素的精确、严格的定义,从而使得用户和系统分析员都对输入、输出、存储的成分,甚至中间计算结果有共同的理解。简而言之,数据字典是描述数据信息的集合,是对系统中使用的所有数据元素的定义的集合。

数据字典在数据库设计中占有很重要的地位。其主要作用有两点:其一,是系统中用户关心的各类数据描述的集合;其二,是进行详细的数据收集和数据分析后所获得的主要成果。

关于系统需求分析,最后还要强调两点:

- 需求分析阶段一定要收集将来应用所涉及的数据。
- 需求分析必须有用户参与。

下面,以某电脑配件连锁店管理系统的需求为例,介绍系统的信息需求。

三、电脑配件连锁店管理系统的信息需求

某电脑配件连锁店要设计一个电子商务数据库系统来管理所有连锁店的业务信息。该系统包括存货管理、销售管理、人事管理等子系统。为了简化说明,以该公司存货管理子系统和销售管理子系统为例,给出两个子系统的信息需求。

（一）存货管理子系统的信息需求

1. 主要管理的对象

存货管理子系统主要负责管理供应商在不同时间向公司供应相应数量的不同电脑配件信息,以及仓库存储配件的信息,管理的对象有供应商、电脑配件、仓库、仓库管理员,需要存储的信息如下:

- (1) 供应商:包括供应商编号、名称、账号、地址,通过供应商编号进行标识。
- (2) 电脑配件:包括配件编号、配件名称、类别、单位、单价,通过配件编号进行标识。
- (3) 仓库:包括仓库编号、仓库地址、负责人,通过仓库编号进行标识。
- (4) 仓库管理员:包括管理员编号、姓名、业绩,通过管理员编号进行标识。

2. 各对象间的联系

存货管理子系统中各对象间的联系如下:

- (1) 每个供应商可以供应多个型号的电脑配件,且每一种型号的配件可以由多个供应商供应。供应商供应电脑配件需标注供应日期和供应数量。
- (2) 每个型号的电脑配件可以存放在多个仓库中,一个仓库可以存放多个型号的电脑配件。

(3) 一个仓库可以有多个仓库管理员,但一个仓库管理员只能负责一个仓库。

(二) 销售管理子系统的信息需求

1. 主要管理的对象

销售管理子系统主要负责不同分店的销售信息,管理的对象有电脑配件、分店、营业员、仓库、订单、客户,需要存储的信息如下:

- (1) 电脑配件:包括配件编号、名称、类别、单位、单价,通过配件编号进行标识。
- (2) 分店:包括分店编号、名称、地址,通过分店编号进行标识。
- (3) 营业员:包括营业员编号、姓名、月薪,通过营业员编号进行标识。
- (4) 仓库:包括仓库编号、仓库地址、负责人,通过仓库编号进行标识。
- (5) 订单:包括订单编号、订单日期、客户编号等信息,通过订单编号进行标识。
- (6) 客户:包括客户编号、客户名、地址、电话等,通过客户编号进行标识。

2. 各对象间的联系

销售管理子系统中各对象间的联系如下:

- (1) 每个分店可以销售多种电脑配件,每种电脑配件可以在不同的分店销售,每个分店每销售一种电脑配件需标注销售日期、销售数量。
- (2) 每个分店有若干营业员,每个营业员只在一个分店工作,分店聘用营业员有聘期和月薪。
- (3) 每个仓库可以向多个分店进行发货,每个分店只能由一个仓库配送配件。每个仓库向分店发货需标注发货日期和发货数量。
- (4) 每个营业员可以开多张订单,一个订单中可以包含多种电脑配件,相同型号的配件可以出现在不同的订单中。
- (5) 订货时,每个客户可以有多张订单。

第三节 数据库的概念结构设计

数据库的概念结构设计是将系统需求分析得到的用户需求抽象为反映用户观点的信息结构的过程,也就是将现实事物以不依赖于任何数据模型的方式加以描述,目的在于以符号的形式正确地反映现实事物以及事物与事物间的联系。设计的结果是数据库的概念模型,即 E-R 模型。由于数据库的概念结构设计是从现实世界的角度对需求进行抽象和描述,所以它与计算机硬件、数据库逻辑结构和支持数据库的 DBMS 无关。在数据库设计中应重视概念结构设计,它是整个数据库设计的关键,为计算机存储数据做准备工作。

一、概念结构的作用和设计方法

概念结构设计的目标就是产生反映信息需求的数据库概念结构,即概念模型。概念模型最常用的表示方法就是实体联系(E-R)方法,其设计的内容就是如何根据应用需求设计系统的 E-R 模型。

（一）概念结构的作用

在早期的数据库设计中,需求分析后,直接设计 DBMS 支持的关系模式,这样,注意力往往被吸引到更多的细节限制方面,不能集中在最重要的信息组织结构和处理上。当外界环境发生变化时,设计结果就难以适应变化。

将概念结构从数据库设计中独立出来,对数据库设计人员来说,仅从用户的角度看待数据并处理需求和约束,可以使数据库设计各阶段的任务相对单一化,从而可以控制设计的复杂性,便于组织管理。其作用主要体现在以下几个方面。

1. 能真实地描述现实世界

概念结构能真实地反映现实世界事物和事物之间的联系,能满足用户对数据的处理要求,是现实世界的一个真实模型。

2. 易于理解

用 E-R 图来描述概念模型非常接近人的思维,是对现实世界的真实反映,容易为人们所理解。用户不需要掌握计算机等专业知识,也能够理解概念结构。利用概念结构,用户可以与设计者交换意见,参与到数据库的设计过程中。因此,概念结构是数据库设计人员与用户交互的最有效的工具。

3. 是各种数据模型的共同基础

概念模型最终要转化成不同 DBMS 所支持的数据模型,并且很容易就能向普遍使用的关系模型转化,所以是各种数据模型的基础。

4. 有利于修改和扩充

由于现实世界的应用环境和应用要求会发生变化,发生变化时只需要改变概念模型,易于更改的概念模型更有利于扩充。

（二）概念结构设计的方法

概念模型是数据模型的前身,它比数据模型更独立于机器、更抽象,也更加稳定。概念结构设计的方法有 4 种。

1. 自顶向下的设计方法

首先定义全局概念结构的框架,然后逐步细化为完整的全局概念结构。

2. 自底向上的设计方法

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。

3. 逐步扩张的设计方法

首先定义最重要的核心概念结构,然后向外扩充,生成其他概念结构,直至完成总体概念结构。

4. 混合策略设计的方法

混合策略设计的方法首先用自顶向下策略设计一个全局概念结构的框架,然后以它为主干,通过自底向上策略设计各局部概念结构。

最常采用的策略是自底向上的方法,即先自顶向下地进行需求分析,然后再自底向上地

设计概念结构。

二、概念结构设计的主要步骤

概念结构设计是利用抽象机制对需求分析阶段收集到的数据进行分类、组织,形成实体、属性和码,确定实体之间的联系类型。设计时,其注意力主要集中在怎样表达用户对信息的需求上,暂时不考虑如何实现的问题,先根据单个应用需求,设计能反映每一个应用的局部 E-R 图,然后集成并消除冗余和可能的矛盾,得到系统的总体 E-R 图。

(一) 设计局部 E-R 图

局部用户的信息需求是构造全局概念模式的基础。因此,需要从个别用户的需求出发,为数据的观点与数据使用方式相似的用户建立一个相应的局部概念结构。设计局部 E-R 图的具体步骤如下。

1. 划分用户组

划分用户组是设计局部 E-R 图的前提,实质是确定局部范围,一般将数据要求和处理要求接近的用户分成一组,同时为了控制局部 E-R 图的复杂性,需要考虑用户组的规模,一般实体个数为 5~9 合适。

2. 确定实体和属性

实体和属性,只能根据对客观世界的理解和思维习惯,以及数据的逻辑关系来划分。实际上,实体和属性之间并不存在形式上可以截然划分的界限。但是,为了简化 E-R 图,区分实体属性,应当遵循的一条原则是:现实世界的事物能作为属性对待的尽量作为属性对待。在给定的应用环境中,可遵循如下的基本准则来划分:

(1) 属性不能再具有需要描述的性质。属性必须是不可分割的数据项,不能包含其他属性。也就是说,属性不能是另外一些属性的聚集。

(2) 属性不能与其他实体具有联系。在 E-R 图中所有联系都必须是实体间的联系,而不能有属性与实体之间的联系。

例如,职工是一个实体,职工号、姓名、年龄和职称是职工的属性。如果职称没有与工资、福利挂钩,就没有必要进一步描述的特性,则职称可以作为职工实体集的一个属性对待;如果不同的职称有着不同的工资、住房标准和附加福利,则职称作为一个实体来考虑就比较合适。

又例如,在医院中,一个病人只能住在一个病房,病房号可以作为病人实体的一个属性。但如果病房还要与医生实体发生联系,即一个医生负责几个病房的病人的工作,则根据第二条准则,病房应作为一个实体。

3. 确定实体之间的联系、属性及其联系类型

不仅实体具有属性,而且联系也可以有属性。在确定实体和属性的同时,要通过分析确定实体之间的联系以及联系的属性,并根据语义确定联系的类型。例如,仓库和负责人两个实体具有语义:一个仓库有多个负责人,一个负责人只能管理一个仓库,则仓库和负责人的联系类型为 1:n;如果语义是一个负责人可以负责多个仓库的话,则两者的联系类型为 m:n。有时,根据具体应用环境,两个实体之间可能有多种联系。例如,图 5-3 所示的职工和

设备之间的关系。

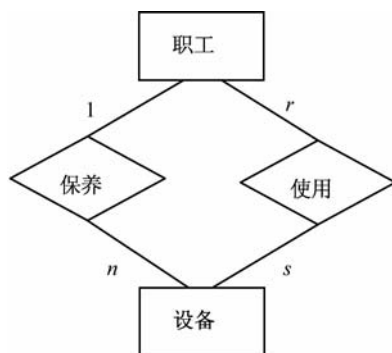


图 5-3 两个实体之间的两种联系

如果 E-R 图比较复杂,为了使 E-R 图简洁明了,常将图中实体的属性省略,而着重反映实体之间的联系。

4. 设计分 E-R 图

在确定了实体、联系以及它们的属性之后,各个局部 E-R 图的设计就水到渠成了。

(二) 全局 E-R 图的设计

全局 E-R 图的设计就是把设计好的各子系统的局部 E-R 图综合(合并)成一个系统的总 E-R 图。合并时可以有两种方法:一种方法是多个局部 E-R 图一次集成;另一种方法是逐步集成,用累加的方法一次集成两个局部 E-R 图。无论采用哪种方法,在每次集成局部 E-R 图时,都要按以下步骤进行。

1. 合并 E-R 图,生成初步 E-R 图

进行各局部 E-R 图的合并时,要解决各局部 E-R 图之间的冲突问题,并将各局部 E-R 图合并起来生成初步 E-R 图。

由于各个局部应用所面向的问题是不同,而且通常是由不同的设计人员进行不同的视图设计,这样就会导致各个局部 E-R 图之间必定会存在许多不一致的地方,即产生冲突问题。由于各个局部 E-R 图存在冲突,所以不能简单地把它们画到一起,必须先消除各个局部 E-R 图之间的不一致,形成一个能被全系统所有用户共同理解和接受的统一的概念模型,再进行合并。

局部 E-R 图之间的冲突主要有 3 类:属性冲突、命名冲突和结构冲突。

(1) 属性冲突:属性冲突包括属性域冲突和属性取值单位冲突两种情况。

属性值的类型、取值范围或取值集合的不同属于属性域冲突。例如,对于商品的属性商品号,不同的部门可能会采用不同的编码形式,而且定义的类型各不相同,有的定义为整型,有的定义为字符型,这都需要各个部门之间协商解决。

属性取值单位冲突,如商品的重量,不同的部门可能分别用斤、公斤或千克表示,结果会给数据统计造成错误。

(2) 命名冲突:命名冲突主要有同名异义冲突和异名同义冲突两种。

同名异义冲突是指不同意义的对象在不同的局部应用中具有相同的名字,即两个同名的实体、属性在不同的局部 E-R 图中含义是不同的。

异名同义冲突是指意义相同的对象在不同的局部应用中有不同的名称,即在不同的局部 E-R 图中含义相同的实体、属性,其命名不同。

命名冲突主要通过协商和调整解决。

(3) 结构冲突:结构冲突有 3 种情况。

同一对象在不同的应用中具有不同的抽象。例如,职工在某一局部应用中被当做实体对待,而在另一局部应用中被当做属性对待,这就会产生抽象冲突问题。

同一实体在不同局部 E-R 图中的属性组成不一致,即所包含的属性个数和属性排列次序不完全相同。解决这类冲突的方法是使该实体的属性取各个局部 E-R 图中属性的并集,再适当调整属性的次序,兼顾到各种应用。

实体之间的联系在不同的局部 E-R 图中呈现不同的类型。解决此类冲突的方法是根据应用的语义对实体联系的类型进行综合或调整。

2. 优化初步 E-R 图

优化初步 E-R 图是指根据数据库应用系统的需求,在初步全局 E-R 图的基础上,利用需求分析的结果,通过检测全局 E-R 图的数据冗余、联系冗余,从而消除相应的冗余数据,并最终形成独立于具体 DBMS 的整体概念结构的过程。

优化的目标是在全面准确地反映用户需求的基础上,使系统尽量满足以下要求:

- 属性尽可能少,即组成每一个实体的属性个数尽量少。
- 实体尽量少,即组成概念结构的实体个数尽量少。
- 联系尽量少,即组成概念结构的联系的个数尽量少。

优化是指修改和重组初步 E-R 图,因为在初步 E-R 图中可能存在冗余数据和实体间冗余的联系。冗余数据主要包括属性冗余、实体冗余。属性冗余是指可由基本数据导出的数据。例如,商品实体有单价、数量、总价属性,但根据分析,总价属性可由单价和数量属性导出,故应予以消除。实体冗余是指用一个实体代替两个或多个实体(即合并实体),或者在极端的情况下,可以利用多个实体导出的实体。对于合并实体,一般是指两个实体或多个实体具有相同的主关键字,则可以合并为一个实体;对于可由多个实体导出的实体,则可以直接消除。冗余的联系是可由其他联系导出的联系。

总之,冗余的存在容易破坏数据库的完整性,给数据库维护增加困难,应当加以消除。清除冗余可以借助需求分析的结果采用分析方法,也可以用第四章的规范化理论消除,其中分析方法是消除冗余的主要方法。消除了冗余的初步 E-R 图就称为基本的 E-R 图。

在实际应用中,并不是要将所有的冗余数据与冗余联系都消除。有时为了提高数据查询效率、减少数据存取次数,在数据库中有意设计了一些数据冗余或联系冗余。因此,在设计数据库结构时,冗余数据的消除或存在要根据用户的整体需要来确定。如果希望存在某些冗余,则应在数据字典中进行说明,并把保持冗余数据的一致性作为完整性约束条件。

下一步是重组 E-R 图。对于消除数据冗余后的全局 E-R 图,由于消除了不必要的冗余属性、冗余实体和冗余联系,因此需要根据应用系统的整体需求,再对全局 E-R 图进行整体统一的调整、重新组合和重新构造,从而形成优化的全局 E-R 图,即系统的整体概念结构。

3. 验证全局 E-R 图

经过优化的概念结构只有通过相关人员必要的审核和验证、确保无误后,才能作为下一

阶段数据库逻辑结构设计的依据。验证全局 E-R 图时,应确保满足下列条件:

- 全局 E-R 图必须具有一致性,不存在相互矛盾的表达。
- 全局 E-R 图能准确地反映原来每个局部应用的局部 E-R 图的结构,包括实体、属性和联系。
- 能够满足需求分析阶段所确定的所有需求。
- 全局 E-R 图必须征求用户和相关人员的意见,需要经过评审、修改、优化,再确定作为数据库的概念结构,提交给用户。

综上所述,在设计概念结构时,需要根据系统的应用需求和用户的最终需求,进行概念结构的抽象、选择局部概念结构的范围、设计局部 E-R 图,合并局部 E-R 图和消除全局 E-R 图不必要的冗余,设计的结果要向用户进行演示和解释,听取用户的意见,检查由此设计的数据库是否提供用户所需要的全部信息。在经过反复评审、修改和优化后,最后把它确定下来,从而完成最终优化的全局概念结构。

三、电脑配件连锁店管理系统的概念结构设计

(一) 存货管理子系统 E-R 图的设计

根据需求分析,设计出存货管理子系统 E-R 图,如图 5-4 所示。在其 E-R 图中,有 4 个存储数据的主要实体,分别是供应商、电脑配件、仓库实体、仓库管理员。在图中标出了每个实体的属性,其中具有下划线的属性表示实体标识符。实体标识符是能够唯一确定一个实体的属性或属性组合。

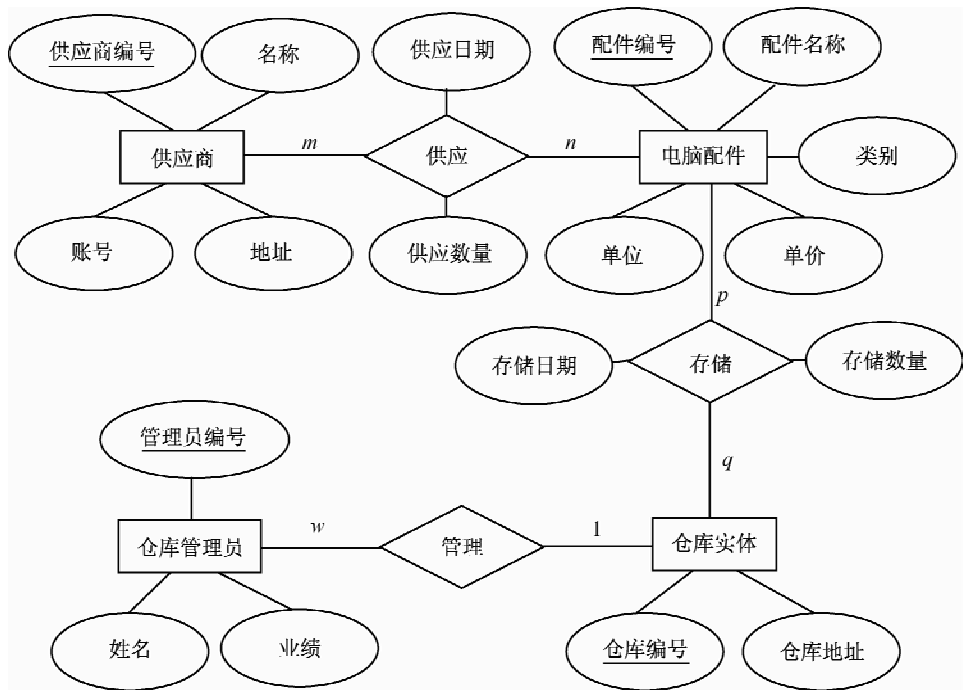


图 5-4 存货管理子系统的 E-R 图

(二) 销售管理子系统 E-R 图的设计

同样根据需求分析,设计出销售管理子系统的 E-R 图,如图 5-5 所示。在该 E-R 图中,有 6 个存储数据的主要实体,分别是分店、电脑配件、仓库、营业员、订单和客户。在图中标出了每个实体的属性,其中具有下划线的属性表示实体标识符。

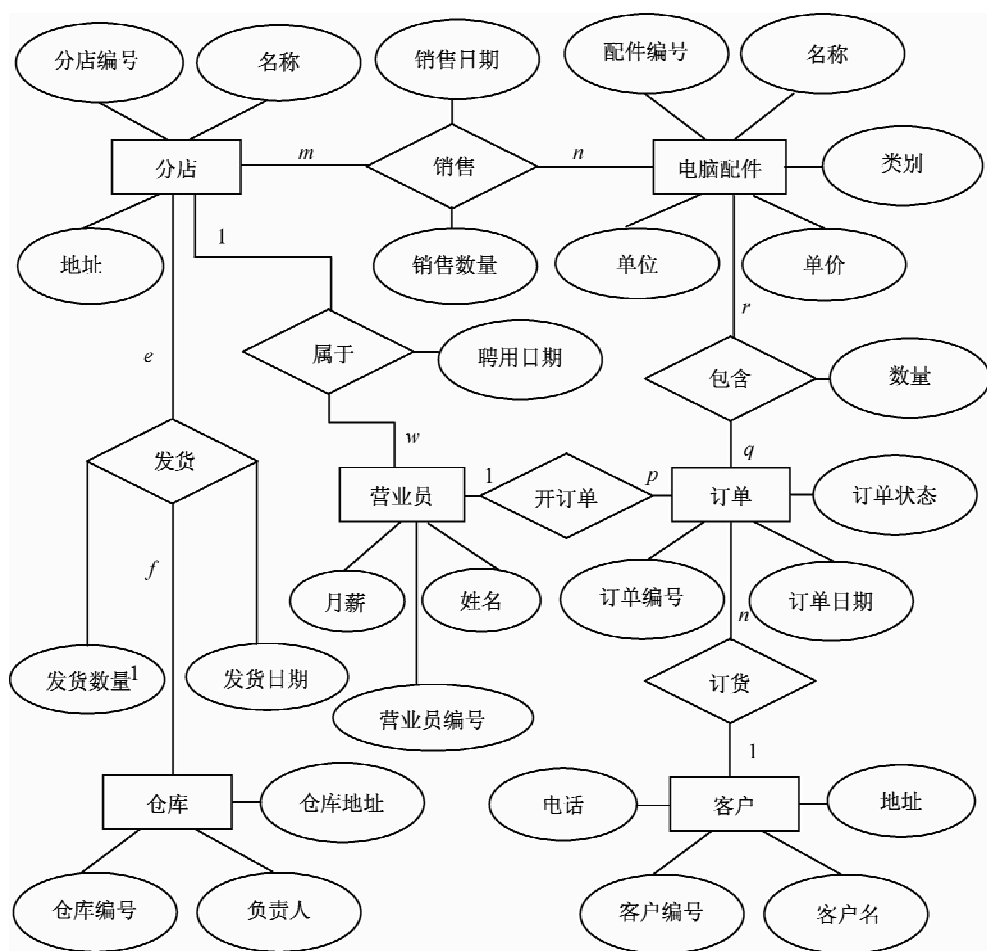


图 5-5 销售管理子系统的 E-R 图

(三) 系统全局 E-R 图的设计

系统全局 E-R 图的设计主要经过以下几个步骤。

1. 具体分析

(1) 存货管理子系统 E-R 图的“仓库实体”和销售管理子系统 E-R 图中的“仓库”为同一实体,存在实体的命名冲突,现统一将其命名为“仓库”。

(2) 在存货管理子系统 E-R 图中,电脑配件实体的属性“配件名称”和销售管理子系统 E-R 图中电脑配件实体的属性“名称”含义相同,也存在属性的命名冲突,现统一将其命名为“配件名称”。

(3) 在存货管理子系统 E-R 图中,仓库实体包括仓库编号和仓库地址,而在销售管理子系统 E-R 图中,仓库实体包括仓库编号、负责人、仓库地址。显然仓库实体在两个局部 E-R 图中存在着结构冲突。此时不能简单地将仓库实体的属性进行合并,因为在存货管理子系统 E-R 图中,把负责人抽象为一个实体仓库管理员,这样仓库实体和仓库管理员之间有 1 : n 的联系,而在销售管理子系统 E-R 图中,把负责人抽象为仓库的一个属性对待。那么负责人是抽象为属性对待,还是抽象为仓库管理员实体,要根据应用的语义,对类型进行综合的调整。在存货管理子系统有这样的语义:一个仓库可以有多个仓库管理员,但一个仓库管理员只能负责一个仓库,如果系统管理员还需要了解仓库管理员的其他信息,在合并时把负责人抽象为仓库管理员实体对待。

2. 全局 E-R 图

合并各局部 E-R 图后,生成的电脑配件连锁店管理系统全局 E-R 图如图 5-6 所示。为了使 E-R 图简洁明了,图中着重反映实体间的联系,省略各实体的属性。

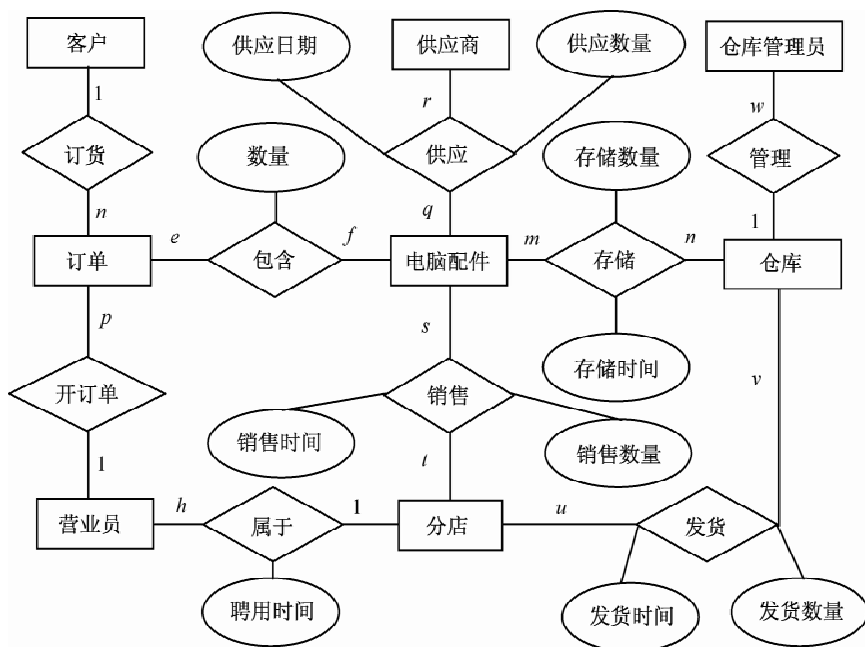


图 5-6 电脑配件连锁店管理系统的全局 E-R 图

第四节 数据库的逻辑结构设计

E-R 图表示的概念模型是用户数据要求的形式化。正如前面所述,E-R 图是独立于任何一种数据模型的概念信息结构,不为任何一个 DBMS 所支持。逻辑结构设计任务就是把设计好概念结构的基本 E-R 图转换为与选用的某个具体 DBMS 所支持的数据模型相符合的逻辑结构。如果选择的是关系型 DBMS 产品,逻辑结构的设计内容就是设计数据库中

所包含的各关系模式的结构,包括各关系模式的名称、每个关系模式中各属性的名称、数据类型、取值范围等,设计的结果是一组关系模式。

通常把概念模型向逻辑模型的转换过程分为以下 3 个步骤:

- (1) 把概念模型转换成一般的数据模型。
- (2) 将一般的数据模型转换成特定的 DBMS 所支持的数据模型。
- (3) 通过优化方法将其转化为优化的数据模型。

目前,设计的电子商务数据库系统大都采用支持关系模型的 RDBMS,所以下面只介绍由 E-R 图向关系数据模型的转换及其优化。

一、概念模型向关系模型的转换

E-R 图向关系模型转换需要解决以下两个问题:

- 如何将实体集和实体间的联系转换为关系模式?
- 如何确定这些关系模式的属性和关键字?

关系模型的逻辑结构是一组关系模式,将 E-R 图转换为关系模型实际上就是将实体集、属性以及联系转换为相应的关系模式。这种转换要遵循一定的规则。

(一) 实体集的转换规则

实体集的转换规则是 E-R 模型中的一个常规实体集转换为一个关系模式,该关系模式的属性就是原实体集中各属性的组成,关系模式的主码就是原实体集的主码,关系的结构就是关系模式。

(二) 实体集之间联系的转换规则

E-R 模型在向关系模型转换时,实体集之间的联系有以下几种情况。

1. 1 : 1 联系的转换方法

1 : 1 联系有以下两种转换方法:

(1) 将 1 : 1 联系转换为一个独立的关系,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,且每个实体的码均是该关系的候选码。

(2) 将 1 : 1 联系与某一端实体集所对应的关系合并,则需要和被合并关系中增加属性,其新增的属性为联系本身的属性和与联系相关的另一个实体集的码。

【例 5-1】 图 5-7 所示为国内某旅游管理系统数据库中涉及的旅游团和保险实体的 E-R 图,两实体的联系类型为 1 : 1 联系,将 E-R 图转换为关系模式。

首先将两个实体转换为两个关系模式:分别是“旅游团”关系模式和“保险”关系模式,其中关系模式中标有下划线的属性为候选码。

旅游团(团号,团名,地址,联系人,电话)

保险(保险单号,投保日期,保险费)

再将联系转换为关系模式,有 3 种方案。

方案 1 将联系形成一个独立的关系模式:

参加(团号,保险单号,人数)

方案 2 将“参加”和“旅游团”两个关系模式进行合并,则关系模式为

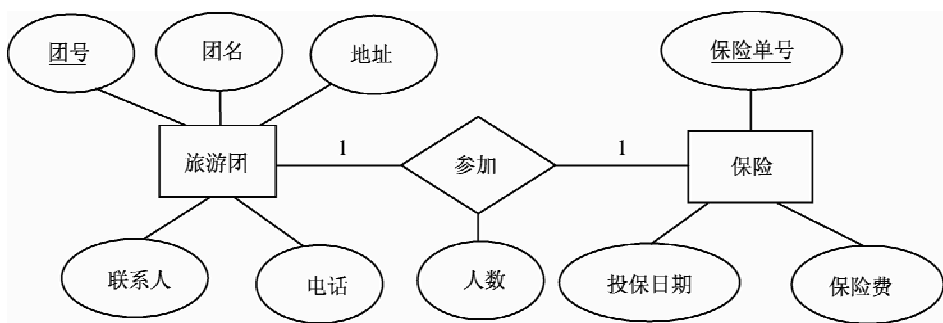


图 5-7 1:1 联系转换为关系模式的实例

旅游团(团号,团名,地址,联系人,电话,人数,保险单号)

方案 3 将“参加”和“保险”两个关系模式进行合并,则关系模式为

保险(保险单号,投保日期,保险费,人数,团号)

将上面 3 种方案进行比较,不难发现,在方案 1 中,由于关系多,增加了系统的复杂性;在方案 2 中,由于并不是每个旅游团都参加保险,这样就会造成方案 2 旅游团关系中的保险单号属性的 null 值过多;相比较起来,方案 3 比较合理。最终图 5-7 转换的关系模式集合为

{ 旅游团(团号,团名,地址,联系人,电话)
保险(保险单号,投保日期,保险费,人数,团号)

由此可以看出,“保险”关系模式不仅描述了保险单的信息,同时描述了保险和旅游团之间的关系,此联系是通过“保险”关系模式的外码(团号)属性联系起来的。

2. 1:n 联系的转换方法

E-R 模型在向关系模型转换时,实体之间的 1:n 联系可以有以下两种转换方法:

(1) 将联系转换为一个独立的关系,其关系的属性由与该联系相连的各实体集的码以及联系本身的属性组成,而该关系的码为 n 端实体集的码。

(2) 在 n 端实体集中增加新属性,新属性由联系对应的 1 端实体集的码和联系自身的属性构成,新增属性后原来关系的码不变。

【例 5-2】 图 5-8 给出了国内某旅游管理系统数据库中涉及的“旅游团”和“旅客”实体的 E-R 图,两实体的联系类型为 1:n 联系,将 E-R 图转换为关系模式。

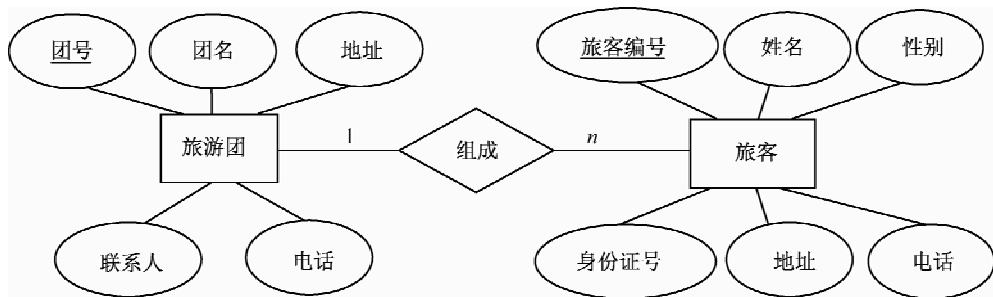


图 5-8 1:n 联系转换为关系模式的实例

首先将两个实体转换为两个关系模式,分别是“旅游团”关系模式和“旅客”关系模式。

旅游团(团号,团名,地址,联系人,电话)

旅客(旅客编号,姓名,性别,身份证号,地址,电话)

再将联系转换为关系模式,有两种方案。

方案1 将联系形成一个独立的关系模式:

组成(旅客编号,团号)

方案2 将联系“组成”形成的关系和“旅客”关系模式进行合并,则关系模式为

旅客(旅客编号,姓名,性别,身份证号,地址,电话,团号)

比较以上两种方案可以发现:方案1中使用的关系多;方案2中使用的关系少,特别适用于旅游团旅客变化小的应用场合。为了降低系统的复杂性,一般情况下采用第二种方案。

则图5-8最终转换的关系模式集合为

{ 旅游团(团号,团名,地址,联系人,电话)
旅客(旅客编号,姓名,性别,身份证号,地址,电话,团号)

同样,两个关系模式的联系是通过“旅客”关系模式的外码(团号)联系起来的。

3. $m:n$ 联系的转换方法

一个 $m:n$ 联系转换为一个关系的方法为:与 $m:n$ 联系相连的实体集的码以及联系本身的属性均转换为关系的属性,新关系的码为两个相连实体的码的组合(该码为多属性构成的组合码)。

【例5-3】 图5-9给出了某网上图书订购系统数据库所涉及的客户和图书的E-R图,两实体的联系类型为 $m:n$ 联系,将E-R图转换为关系模式。

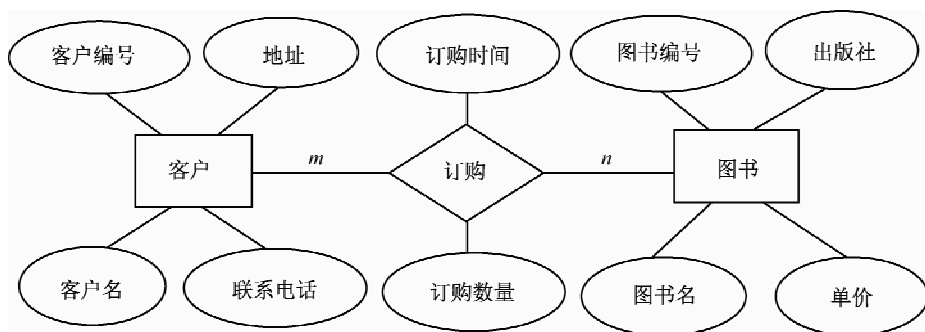


图 5-9 $m:n$ 联系转换为关系模式的实例

根据实体间 $m:n$ 联系的转换规则,转换的关系模式为

客户(客户编号,客户名,地址,联系电话)

图书(图书编号,图书名,出版社,单价)

订购(客户编号,图书编号,订购时间,订购数量)

其中“订购”关系模式的候选码是客户编号和图书编号的组合,客户编号、图书编号作为“订购”关系模式的外码,必须满足关系的参照完整性约束。

假如对“订购”联系不建立独立的关系模式,而是和“客户”或“图书”关系进行合并,则转换的关系模式为

{ 客户(客户编号, 客户名, 地址, 联系电话)
 { 图书(图书编号, 图书名, 出版社, 单价, 客户编号, 订购时间, 订购数量)
 或 { 客户(客户编号, 客户名, 地址, 联系电话, 图书编号, 订购时间, 订购数量)
 { 图书(图书编号, 图书名, 出版社, 单价)

这样,对于图书编号相同的多个客户都订购的图书,或对于一个客户在不同时间订购了多本图书的情况,都存在数据存储时冗余太大的问题,给维护带来不便。

由此可以看出,对于实体之间 $m:n$ 的联系类型,联系必须转换为一个独立的关系模式。

4. 3 个或 3 个以上实体集之间的多元联系的转换方法

(1) 对于 $1:n$ 的多元联系转换为关系模型的方法是修改 1 端实体集对应的关系,即将与联系相关的其他实体集的码和联系自身的属性作为新属性加入 n 端实体集中。

(2) 对于 $m:n$ 的多元联系,转换为关系模型的方法是新建一个独立的关系,该关系的属性为多元联系相连的各实体的码以及联系本身的属性,码为各实体码的组合。

【例 5-4】 图 5-10 给出了某公司网上采购系统数据库所涉及的“供应商”、“商品”和“采购员”实体的 E-R 图,3 个实体间属于多元联系,联系类型为 $m:n$ 联系,将其 E-R 图转换为关系模式。

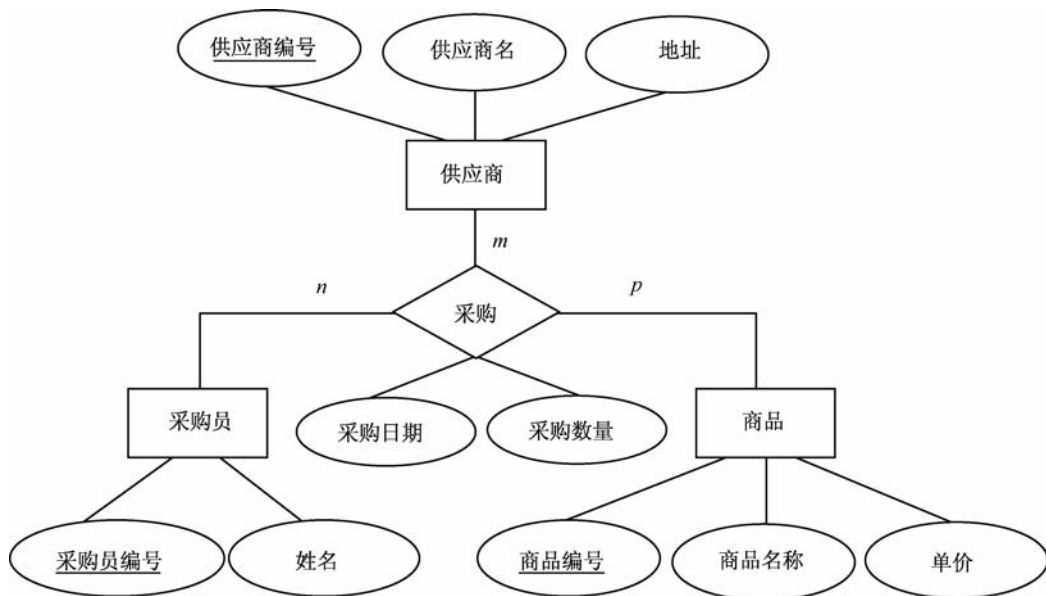


图 5-10 多实体之间多对多联系实例

根据多实体间联系的转换规则,3 个实体转换为 3 个关系模式,联系转换为一个独立的关系模式:

采购员(采购员编号, 姓名)

供应商(供应商编号, 供应商名, 地址)

商品(商品编号, 商品名称, 单价)

采购(采购员编号,供应商编号,商品编号,采购日期,采购数量)

在转换中,除了要遵循以上介绍的转换原则外,还应注意以下两点:

(1) 命名和属性域的处理。关系模式的命名,可以采用 E-R 图中原来的命名,也可以另行命名。命名应有助于对数据的理解和记忆,同时应尽可能避免重名,具体的 DBMS 一般只支持有限的几种数据类型,而 E-R 图是不受这个限制的。如果 DBMS 不支持 E-R 图中的某些属性的域,则应作相应的调整。

(2) 非原子属性的处理。E-R 模型中允许非原子属性,这不满足第一范式的条件。解决的办法是对 E-R 图中出现的非原子属性进行展开。

E-R 模型和关系模型相对于物理存储文件来说,都是对现实世界的抽象逻辑表示,例如,在数据库的三层模式中,概念模式和逻辑模式就是一回事,因此两种模型采用类似的设计原则,可以将 E-R 图设计转换为关系的设计。将数据库的表示从 E-R 图转化为表的形式是由 E-R 图产生关系数据库设计的基础。

二、关系模型优化

数据库逻辑结构设计的结果不是唯一的。为了提高数据库应用系统的性能,还应对根据 E-R 模型转换得到的初步关系模式作适当的修改,调整关系模型的结构,这就是关系模型的优化。关系模型优化通常以规范化理论为指导,在此主要介绍优化的具体方法和子模式的设计。

(一) 优化方法

1. 确定数据依赖

用数据依赖分析和表示数据项之间的联系,写出每个数据项之间的数据依赖。即根据需求分析阶段得到的语义,分别写出每个关系模式内部各属性之间的数据依赖,以及不同关系模式属性之间的数据依赖。

2. 分析函数依赖

按照函数依赖的理论,逐一分析所构造的关系模式,检查是否存在部分函数依赖、传递函数依赖等。如果存在部分函数依赖、传递函数依赖,则要消除其中的依赖关系,确定关系模式分别属于第几范式,并消除冗余的联系。

3. 判断是否合并或分解

结合实际应用环境、数据库的规模及不同应用对数据处理的要求,分析这些模式是否适合具体的应用环境,确定是否需要关系模式的合并和分解。如果两个关系模式的主码相同则可以进行合并,但是为了数据库的完整性,对于某些不是 BCNF 的关系模式,可以考虑进行分解,以提高数据操作效率和存储空间利用率。

需要注意的是,在应用规范化理论进行优化关系模式时,一定要分析应用环境,千万不要盲目追求高范式。因为并不是规范化程度越高的关系模式就越优,由于规范化程度越高,数据分离程度就越低,这会给查询操作带来不利的影响。例如,当查询涉及两个或多个关系模式的信息时,系统就必须经常使用连接运算,而连接运算恰恰是一种高代价的运算,可以说,关系模型的缺陷低效的主要原因就是连接运算引起的,如果这种操作频繁出现,必将大

大降低处理的速度,这时可以考虑将几个关系进行合并。尽管在理论上非 BCNF 的关系模式会存在不同程度的更新异常或数据冗余,但如果该关系主要用于查询,不执行更新操作,那么更新异常等问题在实际应用中就不会产生影响。

因此,对于一个具体的应用来说,关系模式到底规范化到什么程度,需要权衡响应的时间和潜在的问题两者的利弊来决定。规范化理论为数据库逻辑设计时判断关系模式的优劣提供了理论标准,可用来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。

(二) 用户子模式的设计

把 E-R 模型转换为全局的逻辑模型后,还应根据局部应用的需求,结合具体的 DBMS 的特点设计用户子模式。用户子模式也称外模式,是用户所看到的数据模式,各类用户有各自的子模式。关系数据库管理系统中提供的视图是根据用户子模式设计的。设计用户子模式时只考虑用户对数据的使用要求、习惯及安全性要求,而不用考虑系统时间效率、空间效率、易维护等问题。设计用户子模式时应注意以下问题。

1. 使用更符合用户习惯的别名

在合并各局部 E-R 图时应消除命名的冲突,以使数据库系统中同一关系和属性具有唯一的名称。这在设计数据库整体结构时是非常必要的。但命名统一后会使用户在使用上感到别扭,用户子模式(视图)的方法可以有效地解决该问题。必要时,可以设计视图重新定义命名,使其与用户习惯一致,以方便用户的使用。

2. 对不同级别的用户可以定义不同的子模式

由于视图能够对表中的行和列进行限制,所以它还具有保证系统安全性的作用。对不同级别的用户定义不同的视图,可以保证系统的安全性。

假设有关系模式:产品(产品号,产品名,规格,单价,生产车间,生产负责人,产品成本,合格率,质量等级)

为一般客户建立视图:

产品 1(产品号,产品名,规格,单价)

为产品销售部门建立视图:

产品 2(产品号,产品名,规格,单价,生产车间,生产负责人)

在建立视图后,产品 1 视图中包含了允许一般顾客查询的产品属性;产品 2 视图中包含允许销售部门查询的产品属性;而对生产部门,则可以利用产品关系查询产品的全部属性数据。这样,既方便了使用,又可以防止用户非法访问本来不允许他们查询的数据,保证了系统的安全性。

3. 简化用户对系统的操作

实际中,某些局部应用经常要使用某些复杂的查询,这些查询包括多表连接、限制、分组、统计等。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,避免每次查询都要对其进行重复描述,大大简化了用户的操作。

三、电脑配件连锁店管理系统的逻辑结构设计

由第三节设计的电脑配件连锁店管理系统的全局 E-R 图,可以得到如表 5-1 所示的一组关系模式及相关信息。表中的一行为一个关系模式。

表 5-1 电脑配件连锁店管理系统的关系模式信息

数据性质	关系名	属 性	说 明
实体	客户	<u>客户编号</u> ,客户名,地址,电话	
实体	订单	<u>订单编号</u> ,订单日期,订单状态,客户编号,营业员编号	“客户编号”为与“订货”关系合并后的新增属性,“营业员编号”为与“开订单”关系合并后的新增属性
实体	营业员	<u>营业员编号</u> ,姓名,月薪,聘用日期,分店编号	“分店编号”为与“属于”关系合并后的新增属性
实体	供应商	<u>供应商编号</u> ,名称,账号,地址	
实体	电脑配件	<u>配件编号</u> ,名称,类别,单位,单价	
实体	分店	<u>分店编号</u> ,名称,地址	
实体	仓库管理员	<u>仓库管理员编号</u> ,姓名,业绩,仓库编号	“仓库编号”为与“管理”关系合并的新增属性
实体	仓库	<u>仓库编号</u> ,地址	
1 : n 联系	订货	订单编号,客户编号	与“订单”关系合并
1 : n 联系	开订单	订单编号,营业员编号	与“订单”关系合并
1 : n 联系	属于	营业员编号,分店编号,聘用时间	与“营业员”关系合并
1 : n 联系	管理	仓库管理员编号,仓库编号	与“仓库管理员”关系合并
m : n 联系	包含	订单编号,配件编号,数量	
m : n 联系	供应	供应商编号,配件编号,供应日期,供应数量	
m : n 联系	存储	仓库编号,配件编号,存储日期,存储数量	
m : n 联系	销售	分店编号,配件编号,销售日期,销售数量	
m : n 联系	发货	仓库编号,分店编号,发货日期,发货数量	

注:带有下列线的属性为关系的码。

根据概念模型向关系模型的转换规则,8 个实体转换为 8 个关系模式,5 个 $m : n$ 联系转换为 5 个关系模式,4 个 $1 : n$ 联系形成的关系模式与相应的实体形成的关系模式进行合并。因此,经过优化后,该数据库可以设计 13 个关系模式,分别是:

- 客户(客户编号,客户名,地址,电话)
- 订单(订单编号,订单日期,订单状态,客户编号,营业员编号)
- 营业员(营业员编号,姓名,月薪,聘用日期,分店编号)
- 供应商(供应商编号,名称,账号,地址)
- 电脑配件(配件编号,名称,类别,单位,单价)

分店(分店编号,名称,地址)

仓库管理员(仓库管理员编号,姓名,业绩,仓库编号)

仓库(仓库编号,地址)

订单详细信息(订单编号,配件编号,数量)

供应(供应商编号,配件编号,供应日期,供应数量)

存储(仓库编号,配件编号,存储日期,存储数量)

销售(分店编号,配件编号,销售日期,销售数量)

发货(仓库编号,分店编号,发货日期,发货数量)

第五节 数据库的物理结构设计

数据库的物理结构主要是指数据库在物理设备上的存储结构与存取方法,它依赖于给定的计算机系统。设计的任务是选择合适的存储结构和存取路径,也就是设计数据库的内模式。内模式和逻辑模式不一样,不直接面向用户,一般的用户不一定、也不需要了解内模式的设计细节。内模式的设计可以不考虑用户理解的方便,其主要的设计目标有两个:

- 提高数据库的性能,特别是满足主要应用的性能要求。
- 有效地利用存储空间。

这两个目标中,第一个目标更为重要,因为性能仍然是当今数据库系统的薄弱环节。

一般来说,物理结构与 DBMS 的功能、DBMS 所提供的物理环境和工具、应用环境以及数据存储设备的特性都有密切关系。数据库用户通过 DBMS 使用数据库,物理结构设计比逻辑结构设计更加依赖于 DBMS,设计者只能在 DBMS 所提供的技术范围内,根据需求和实际条件适当地选择。设计者必须仔细阅读 DBMS 的有关手册,充分了解其限制条件,充分利用其提供的各种手段。

不同的计算机系统提供的物理环境、存储结构和存取方法是不相同的,没有通用的物理设计方法可供遵循,这里只介绍物理设计所要考虑的因素以及设计内容。

一、影响物理结构的主要因素

数据库的物理结构主要由以下因素决定。

(一) 数据的特性

数据的特性主要是指数据的结构、之间的联系、数据的检索频度等。例如,对经常要检索的关系建立必要的索引,以提高查询的效率。在数据库设计阶段,数据特性很难准确估计,会随着数据库状态的改变而变化。

(二) 数据的使用特性

数据的使用特性包括各个用户的应用所对应的数据视图,各种应用的处理频度、使用数据的方法、对系统的重要程度。这些是对时空效率进行平衡和优化的主要依据。一般来说,物理设计不能均等地考虑每一个用户,必须将用户分类,以便保证重点用户的重点应用。

（三）可用性要求

数据库的可用性要求是指适应用户的要求,维护数据库逻辑上、物理上的完整性的能力。用户都希望数据库有较高的可用性,但为此需要付出较大的代价,所以必须权衡得失。

（四）DBMS、OS、计算机硬件的特性

从整个计算机系统来说,数据库应用仅是其负荷的一部分,数据库的性能不仅决定于数据库的设计,而且与 OS(操作系统)、计算机系统的运行环境有关,受到硬件资源的制约;DBMS 的特性主要指 DBMS 的功能、提供的物理环境和工具,特别是存储结构和存取方法。由于每一种 DBMS 都有自己的特点和不足,只有真正了解 DBMS 的特性,才能设计出充分发挥 DBMS 特色的物理结构。

数据库的物理设计可以分为以下两步进行:

- (1) 确定数据的物理结构,即确定数据库的存取方法和存储结构。
- (2) 对物理结构进行评价,对物理结构评价的重点是时间和效率。

二、物理结构设计的内容

关系数据库的物理设计一般比层次、网状数据库的物理设计简单,既不用选择存取路径,又不用确定内部存储结构。关系数据库管理系统一般把数据的内部存储结构、存取路径完全向用户隐藏起来,一切都由系统自动进行,并且系统的内部都有优化程序,能够在进行路径选择时,先对各种路径的成本进行估算,然后自动选择最佳的路径。这就是关系数据库系统的优越性之一。

关系数据库物理结构设计的内容主要包括选择关系模式存取方法和确定数据库的存放位置和系统配置。

（一）关系模式存取方法的选择

关系数据库常用的存取方法有索引方法、聚簇方法等。

1. 索引存取方法的选择

索引是添加到数据库内部的特殊数据结构,用于加快记录的查询速度,几乎所有的关系 DBMS 都有建立索引功能。选择索引存取方法实际上就是根据应用要求确定对关系的哪些属性列建立索引,哪些属性列建立组合索引,哪些索引建立唯一索引等。即索引的选择应考虑两个问题:一是什么关系建立索引,二是选择哪个或哪些属性作为索引关键字。选择索引存取方法的基本原则是:

(1) 如果一个属性经常在查询条件中出现,则考虑在这个属性上建立索引,如果一组属性经常在查询条件中出现,则考虑在这组属性上建立组合索引。

(2) 如果一个属性经常作为最大值和最小值等聚集函数的参数,则考虑在这个属性上建立索引。

(3) 如果一个属性经常在连接操作的连接条件中出现,则考虑在这个属性上建立索引;同理,如果一组属性经常在连接操作的连接条件中出现,则考虑在这组属性上建立索引。

(4) 对很少需要执行查询、连接、统计操作且记录又较多的关系建立索引,而对经常进行插入、删除、修改操作以及记录较少的关系,应避免建立索引。关系上定义的索引数要适

当,并不是越多越好,因为系统维护索引要付出代价,更重要的是随着关系中数据的变化,索引需要维护更新,以反映数据的变化。

2. 聚簇存取方法的选择

在物理设计中,为了改善性能,提高处理效率,特别是为了提高某个属性或属性组的查询速度,把这个属性或属性组上具有相同值的元组集中存放在连续的物理块上的处理称为聚簇,这个属性或属性组称为聚簇码。一个数据库可以建立多个聚簇,但一个关系只能加入一个聚簇,选择聚簇存取方法就是确定需要建立多少个聚簇,以及每个聚簇中包括哪些关系。下面是聚簇设计的两个步骤。

(1) 根据规则确定候选聚簇。设计候选聚簇的原则是:

- 对经常在一起进行连接操作的关系可以建立聚簇。
- 如果一个关系的一组属性经常出现在相等、比较条件中,则该单个关系可建立聚簇。
- 如果一个关系的一个(或一组)属性上的值重复率很高,则此单个关系可建立聚簇。
- 如果关系的主要应用是通过聚簇码进行访问或连接的,而其他属性访问关系的操作很少时,可以使用聚簇。尤其是当 SQL 语句中包含有与聚簇有关的 ORDER BY、GROUP BY、UNION、DISTINCT 等子句或短语时,使用聚簇特别有利,可以省去对结果集的排序操作。反之,当关系较少利用聚簇码操作时,最好不要使用聚簇。

(2) 从候选聚簇中去除不必要的关系。检查候选聚簇,取消其中不必要关系的方法是:

- 从聚簇中删除经常进行全表扫描的关系。
- 从聚簇中删除更新操作远多于连接操作的关系。
- 不同的聚簇中可能包含相同的关系,一个关系可以在某一个聚簇中,但不能同时加入多个聚簇。要从这多个聚簇方案(包括不建立聚簇)中选择一个较优的,其标准是在这个聚簇上运行各种事物的总代价最小。

建立聚簇应注意以下问题:

- 聚簇虽然提高了某些应用的性能,但建立与维护聚簇的开销是相当大的。
- 对已有的关系建立聚簇,将导致关系中的元组移动其物理存储位置,这样会使关系上原有的索引无效,要想使用原索引就必须重建原有索引。
- 当一个元组的聚簇码值改变时,该元组的存储位置也要做相应移动,所以聚簇码值应当相对稳定,以减少修改聚簇码值所引起的维护开销。

(二) 确定数据库的存放位置和系统配置

确定数据的存放位置要综合考虑存取时间、存储空间利用率和维护代价 3 个方面的因素。这 3 个方面常常相互矛盾,需要进行权衡,选择一个折中方案。

1. 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与稳定部分、经常存取部分和存取频率较低部分分开存放。对于有多个磁盘的计算机,可以采用下面几种存取位置的分配方案:

(1) 将表和索引放在不同的磁盘上,这样在查询时,由于两个磁盘驱动器并行工作,可以提高物理 I/O 读写的效率。

(2) 将比较大的表分别放在两个磁盘上,以加快存取速度,这在多用户环境中特别有效。

(3) 将日志文件、备份文件与数据库对象(表、索引等)放在不同的磁盘上,以改进系统的性能。

(4) 对于经常存取或存取时间要求高的对象(如表、索引)应放在高速存储器(如硬盘)上,对于存取频率小或存取时间要求低的对象(如数据库的数据备份和日志文件备份等只在故障恢复时才使用),如果数据量很大,可以存放在低速存储设备上。

2. 确定系统配置

DBMS 产品一般都提供了一些系统配置变量和存储分配参数供设计人员和 DBA 对数据库进行物理优化。在初始情况下,系统都为这些变量赋予了合理的默认值,但这些默认值不一定适合每一种应用环境。在进行数据库的物理设计时,还需要重新对这些变量赋值,以改善系统的性能。

系统配置变量很多,如同时使用数据库的用户数、同时打开的数据库对象数、数据库的大小等,这些变量会影响存取时间和存储空间分配,在物理设计时要根据应用环境确定这些参数值,以使系统性能最佳。

在物理设计时对系统配置变量的调整只是初步的,在系统运行时还要根据系统实际运行情况作进一步的调整,以期切实改进系统性能。

三、物理结构的评价

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,其结果可能会产生多种设计方案。数据库设计人员必须对这些方案进行详细的评价,从中选择一个较优的方案作为数据库的物理结构。

评价物理结构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和比较,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。

第六节 数据库的实施

对数据库的物理设计进行初步评价以后,就可以进行数据库的实施了。数据库实施阶段的主要任务是根据数据库逻辑结构和物理结构设计的结果,在实际的计算机系统中建立数据库的结构、装载数据、测试程序、对数据库的应用系统进行试运行等。

一、建立数据库的结构

利用给定的 DBMS 所提供的命令,建立数据库的模式、子模式和内模式。对关系数据库来说,就是创建数据库,建立数据库中所包含的各个基本表、视图、索引。此部分内容将在第七章详细介绍。

二、数据的装载和应用程序的编制调试

数据的装载是数据库实施阶段最主要的工作。一般数据库系统中的数据量都很大,而且来源于各个不同部门,分散在各种不同的单据或原始凭证中,数据的组织方式、结构和格式都与新设计的数据库系统有相当的差距。数据装载就是要将各种源数据从各个局部应用中抽取出来,并输入计算机后再进行分类转换,综合成符合新设计的数据库结构的形式,最后输入数据库。因此,数据转换和组织数据入库工作是一件耗费大量人力物力的工作。

为提高数据输入的效率和质量,应该针对具体的应用环境设计一个数据录入子系统,由计算机完成数据入库的任务。为了防止不正确的数据输入数据库内,应当采用多种方法多次对数据进行检验。现有的 DBMS 一般都提供不同的 DBMS 之间数据转换的工具,若原有系统是数据库系统,就可以利用新系统的数据转换的工具,先将原系统中的表转换成新系统中相同结构的临时表,再将这些表中的数据分类、转换,综合成符合新系统的数据模式,插入相应的表中。

数据库应用程序的设计应该与数据库设计同时进行,因此,在组织数据入库时,还要调试应用程序。调试应用程序需要将少部分的、适合程序调试用的数据装入数据库。关于调试的方法、步骤在软件工程等课程中有详细讲解,这里不再阐述。

需要注意的是,装载数据时,一般是分期分批地组织数据入库,先输入小批量数据做调试用,等系统试运行结束基本合格后,再大批量输入数据,逐步增加数据量,逐步完成运行评价。

三、数据库的试运行

在原有系统的部分数据输入数据库后,就可以开始对数据库系统进行联合调试,从而进入数据库的试运行阶段。其主要工作如下。

(一) 测试应用程序功能

要实际运行数据库应用程序,执行对数据库的各种操作,首先要测试应用程序功能是否满足设计要求。如果应用程序的功能不能满足设计要求,则需要对应用程序部分进行修改、调整,直到达到设计要求为止。

(二) 测试系统的性能指标

测试系统的性能指标,分析其是否符合设计目标。由于对数据库进行物理设计时考虑的性能指标只是近似的估计,和实际系统运行总有一定的差距,因此必须在试运行阶段实际测试和评价系统性能指标。

在此阶段由于系统还不稳定,软、硬件故障随时都有可能发生。同时系统的操作人员对新系统还不熟悉,误操作也不可避免。因此,在数据库试运行时,应首先调试运行 DBMS 的恢复功能,做好数据库的转储和恢复工作,一旦故障发生,能使数据库尽快恢复,尽量减少对数据库的破坏。

第七节 数据库的维护

数据库试运行合格后,即可投入正式运行,这标志着数据库开发工作基本完成。但是,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数据库管理员完成的。数据库的维护工作包括以下 4 项。

一、数据库的转储和恢复

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。数据库管理员要针对不同的应用要求制订不同的转储计划,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。

二、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。例如,有的数据原来是机密的,现在变成可以公开查询的数据,而新加入的数据又可能是机密的了。系统中用户的权限也会变化。这些都需要数据库管理员根据实际情况修改原有的安全性控制。同样,数据库的完整性约束条件也会变化,也需要数据库管理员不断修正,以满足用户要求。

三、数据库性能的监督、分析和改造

在数据库运行过程中,监督系统运行、对监测数据进行分析并找出改进系统性能的方法是数据库管理员的又一重要任务。目前有些 DBMS 产品提供了监测系统性能参数的工具,数据库管理员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。数据库管理员应仔细分析这些数据,判断当前系统运行状态是否最佳,应当做哪些改进,如调整系统物理参数,或对数据库进行重组织或重构造等。

四、数据库的重组织与重构造

数据库运行一段时间后,由于记录不断增加、删除、修改,数据库的物理存储情况变坏,降低了数据的存取效率,数据库的性能下降。这时,数据库管理员就要对数据库进行重组织或部分重组织(只对频繁增删的表进行重组织)。DBMS 一般都提供用于数据重组织的实用程序。在重组织的过程中,按原设计要求重新安排存储位置、回收垃圾等,以提高系统性能。数据库的重组织并不修改原设计的逻辑结构和物理结构。

数据库的重构造不同于重组织,它要部分修改数据库的模式和内模式。当数据库的应用环境发生变化,如增加了新的应用或新的实体、取消了某些应用、有的实体与实体间的联系发生了变化等,使原有的数据库设计不能满足新的需求,需要调整数据库的模式和内模式

时,就需要重构造数据库。当然数据库的重构造也是有限的,只能作部分修改。如果应用变化太大,重构造也无济于事,说明此数据库应用系统的生命周期已经结束,应该设计新的数据库应用系统了。

本章小结



综合训练

一、选择题

- 概念结构设计是整个数据库设计的关键,它通过对用户需求进行综合、归纳与抽象,形成一个独立于具体 DBMS 的()。
A. 数据模型 B. 概念模型 C. 层次模型 D. 关系模型
- 数据库设计的概念设计阶段,概念结构最常用的表示方法是()。
A. 层次分析法和层次结构图 B. 数据流分析法和数据流程图
C. 实体联系方法 D. 结构分析法和模块结构图
- 在数据库设计中,将 E-R 图转换成关系数据模型的过程属于()。
A. 需求分析阶段 B. 概念设计阶段 C. 物理设计阶段 D. 逻辑设计阶段
- 数据库设计可划分为 6 个阶段,每个阶段都有自己的设计内容,“为哪些关系,在哪

些属性上建什么样的索引”这一设计内容应该属于()设计阶段。

- A. 概念设计 B. 逻辑设计 C. 物理设计 D. 全局设计

5. 数据库物理设计完成后,进入数据库实施阶段,下述工作中,()一般不属于实施阶段的工作。

- A. 建立库结构 B. 系统调试 C. 加载数据 D. 扩充功能

6. 当同一个实体集内部实体之间存在着一个 $m:n$ 的关系时,根据 E-R 模型转换成关系模型的规则,转换关系的数目为()。

- A. 1 B. 2 C. 3 D. 4

7. 在 E-R 模型中,如果有 3 个不同的实体型,3 个 $m:n$ 联系,根据 E-R 模型转换为关系模型的规则,转换为关系的数目是()。

- A. 4 B. 5 C. 6 D. 7

二、设计题

某医院病房需要设计一个数据库系统来管理该医院病房的业务信息。涉及如下的信息:

科室:科名,科地址,科电话,医生姓名

病房:病房号,床位号,所属科室名

医生:工作证号,姓名,职称,所属科室名,年龄

病人:病历号,姓名,性别,诊断,主管医生,病房号

其中,一个科室有多个病房、多个医生,一个病房只能属于一个科室,一个医生只属于一个科室,但可负责多个病人的诊治,一个病人的主管医生只有一个。

根据以上信息完成以下设计:

(1) 设计该系统的 E-R 图。

(2) 将该 E-R 图转换为关系模型的结构,并指出转换结果中每个关系模式的主码。

第六章

网络数据库管理系统 SQL Server 2005

知识目标

- » 了解 SQL Server 2005 的主要特性、安装过程和常用管理工具；
- » 掌握服务器的配置和连接方法；
- » 掌握 SQL Server 2005 数据库的基本对象和操作；
- » 掌握使用 SQL Server Management Studio 创建数据库和数据库基本对象的方法和步骤,重点掌握表的创建和维护。

技能目标

- » 掌握服务器的配置和连接方法；
- » 能够熟练运用 SQL Server Management Studio 创建数据库及数据库对象。

SQL Server 2005 是 Microsoft 公司在 SQL Server 2000 基础上开发的关系型网络数据库管理系统,它在易用性、可伸缩性、可靠性等方面的优异性能,能够为企事业单位的信息管理系统提供一个全面、安全、可靠的数据库平台,满足各种类型单位构建网络数据库的需求,使其成为客户构建、管理商业数据库的最佳选择方案之一。因此,SQL Server 2005 数据库管理系统正被越来越多的用户使用,已成为企业级数据库管理系统的主流产品。

第一节 SQL Server 2005 简介

“SQL Server 2005 是用于大规模联机事务处理、数据仓库和电子商务应用的数据库和数据分析平台。”这句话是 Microsoft 对 SQL Server 2005 的定义,可以看出,SQL Server 2005 是一个全面、集成、端到端的数据解决方案,能为用户提供一个安全可靠并且高效的数据平台,用于企业数据管理。

一、SQL Server 2005 的主要特性

SQL Server 2005 在 SQL Server 2000 的基础上,拓展了可用性、易用性、可靠性、编程能力等方面的性能,较之 SQL Server 2000,在企业数据管理、开发人员工作效率和商业智能等方面改善了数据基础架构。其特性主要体现在以下几个方面。

(一) 企业数据管理

SQL Server 2005 作为数据库管理系统,包含一整套的管理和开发工具,这些工具都具有非常友好的操作界面,既提供强大的功能,同时又方便管理和使用。

1. 易管理性

SQL Server 2005 所提供的单一管理控制平台,使得数据库管理员能够在任何地方监视、管理和调整企业中的所有数据库和相关服务,简化了管理的复杂度,使管理更为容易。

2. 可用性

SQL Server 2005 的可用性包括数据库镜像、故障转移群集、数据库快照和增强的联机帮助操作,这些有助于确保企业的关键系统可用。

3. 安全性

SQL Server 2005 自身带有支持对用户自定义数据库中存储的数据进行加密的功能。通过授权、身份验证和本机加密等特性,提供了更为精确和灵活的控制,使得企业数据更加安全。

(二) 提高开发人员的生产效率

SQL Server 2005 包含许多可以显著提高开发人员工作效率的新技术,例如,对 .NET 的支持,能与 Visual Studio 紧密集成,使开发人员能利用现有的多种开发语言的技巧开发数据库系统。

1. 扩展语言的支持

因为通用语言运行时被集成到数据库引擎中,扩展了语言支持,所以允许开发人员选择自己熟悉的语言来开发数据库应用程序,包括.NET、Transact-SQL等。

2. 改进对数据库的访问,与 Web 服务共存

SQL Server 2005 可以为应用程序提供新型的数据访问功能,无须使用 Microsoft Internet 信息服务(IIS),使用 HTTP 即可直接访问 SQL Server。

3. 引入新的应用程序框架

SQL Server 2005 引入了新的 SQL Server 应用程序框架,包括 SQL Server Express、SQL Server Mobile、Service Broker。其中,Service Broker 是一个分布式应用程序框架,可在数据库到数据库级上提供可靠的异步消息传递,从而为电子商务等大型系统提供较好的可扩展性。

(三) 商务智能

SQL Server 2005 通过在数据集成、开发工具和强大数据分析等方面的改进更好地确立了其在商务智能领域的地位,SQL Server 2005 能够把关键的信息及时地传递到企业员工的手中,从而实现可伸缩的商务智能。

其商务智能特性主要体现在集成服务、分析服务、报表服务,以及与 Microsoft Office 的集成。

二、SQL Server 2005 的安装和配置

为了更好地满足每一个用户的需求,Microsoft 设计了 SQL Server 2005 的 5 个版本,分别是企业版、开发版、标准版、工作组版和简易版,并提供一系列的解决方案。各个版本主要针对不同的具体应用需求而设计,能够满足企业和个人独特的性能、运行需求。需要安装 SQL Server 2005 的哪些组件也要根据企业或个人的需求做出最佳选择。

安装 SQL Server 2005 时须满足基本的硬件要求和软件要求。SQL Server 2005 运行在 32 位平台上的要求和运行在 64 位平台上的要求不同,详细情况可查看联机文档。

(一) SQL Server 2005 安装过程

下面以安装 SQL Server 2005 开发版为例,介绍其安装的过程。

1. 启动 SQL Server 2005 安装程序

(1) 将 SQL Server 2005 安装光盘插入光驱,启动 SQL Server 2005 安装向导。

(2) 选择“服务器组件、工具、联机丛书和示例”选项,打开“最终用户许可协议”对话框,如图 6-1 所示。

(3) 选中“我接受许可条款和条件”复选框,单击“下一步”按钮,打开“安装必备组件”对话框,当安装所需组件后,出现如图 6-2 所示的对话框。

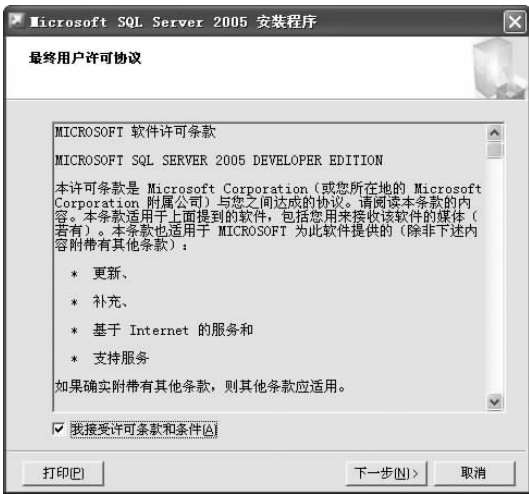


图 6-1 “最终用户许可协议”对话框

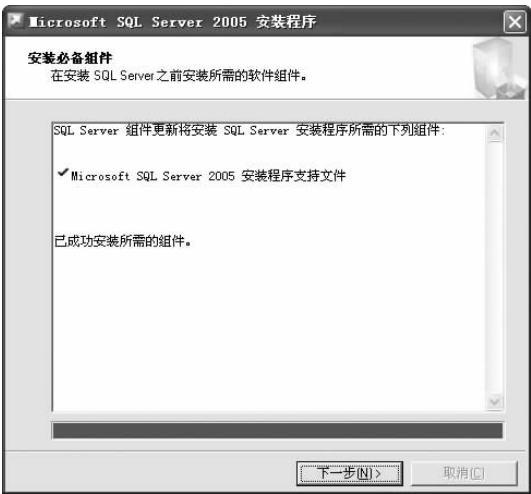


图 6-2 “安装必备组件”对话框

2. 打开安装向导的欢迎界面

(1) 在安装准备工作完成,具备了安装 SQL Server 2005 所需的必备组件后,单击如图 6-2 所示对话框中的“下一步”按钮,打开 Microsoft SQL Server 2005 安装向导欢迎界面。

(2) 单击“下一步”按钮,进行系统配置检查,以确保安装所需的条件都得到满足,包括软件、硬件等各方面条件。

(3) 系统配置检查结束后,出现如图 6-3 所示的“系统配置检查”对话框,在其中列出了系统配置检查的统计结果,从中可以知道安装条件的满足情况,从而了解系统对 SQL Server 2005 的支持程度。由于此处的机器配置较低,出现了最低硬件要求的警告信息,但不影响安装和使用,而如果有错误则不能安装。



图 6-3 “系统配置检查”对话框

3. 输入用户信息

单击“下一步”按钮,系统自动经过简短的安装准备过程,接着打开“注册信息”对话框。输入用户名、公司名称以及 SQL Server 2005 的产品密钥,如图 6-4 所示,其中公司名称可以忽略。

4. 选择 SQL Server 2005 的安装组件

(1) 单击“下一步”按钮,打开“要安装的组件”对话框,如图 6-5 所示,用户可以根据需要选择组件进行安装。

(2) 单击“高级”按钮,在打开的“功能选择”对话框的树形目录中列出了可以安装的程序功能,可以更进一步来制订安装规划,使安装更灵活、更个性化。



图 6-4 “注册信息”对话框



图 6-5 “要安装的组件”对话框

5. 命名安装实例

在图 6-5 的对话框中,单击“下一步”按钮,打开“实例名”对话框,为数据库服务器指定一个实例名,如图 6-6 所示。默认情况下,首次安装时选择“默认实例”单选按钮。

说明:SQL Server 2005 允许在一台计算机上执行多次安装,每一次安装都是一个实例。可以把实例理解为一个 SQL Server 服务器,一个实例就是一组配置文件和运行在计算机内存中的一组程序。SQL Server 2005 的默认实例由运行它的计算机的网络名称标识,一台计算机上只能有一个默认实例。

6. 选择服务账户

完成实例命名后,单击“下一步”按钮,打开“服务账户”对话框,如图 6-7 所示。SQL Server 2005 系统的账户分为“内置系统账户”和“域用户账户”两类。一般建议使用内置系统账户中的本地系统账户。

在“服务账户”对话框中,可以为所有服务指定同一个登录账号,也可以根据需要为每一个服务指定单独的账户。



图 6-6 “实例名”对话框



图 6-7 “服务账户”对话框

7. 选择身份验证模式

确定了服务账户,单击“下一步”按钮,打开“身份验证模式”对话框,如图 6-8 所示。选择“混合模式”单选按钮,并为系统默认的 sa(超级管理员,super administrator 的缩写)设置登录密码。也可选择“Windows 身份验证模式”单选按钮,在 SQL Server 2005 安装完成后,用户在 SQL Server 服务器中可根据需要重新设置用户身份验证模式。关于身份验证的内容将在第八章中介绍。



图 6-8 “身份验证模式”对话框

8. 设置排序规则与错误和使用情况报告

(1) 单击“下一步”按钮,打开如图 6-9 所示的“排序规则设置”对话框,设置 SQL Server 2005 的排序规则,即指定在 SQL Server 2005 中字符的存储形式以及字符的排序和比较规则。若无特殊要求,使用默认设置 Chinese_PRC 即可。

(2) 单击“下一步”按钮,打开如图 6-10 所示的“错误和使用情况报告设置”对话框,采用默认值即可。



图 6-9 “排序规则设置”对话框



图 6-10 “错误和使用情况报告设置”对话框

9. 安装

(1) 单击“下一步”按钮,打开如图 6-11 所示的“准备安装”对话框,进行准备安装前的最后一次信息反馈及确认。

(2) 如果确定不需要修改,单击“安装”按钮,打开如图 6-12 所示的“安装进度”对话框,开始进行文件复制和组件配置。这个过程由系统自动完成,在此可随时监视系统安装的过程。



图 6-11 “准备安装”对话框



图 6-12 “安装进度”对话框

(3) 全部安装完毕后的对话框如图 6-13 所示,单击“下一步”按钮,系统显示如图 6-14 所示的安装提示信息,单击“完成”按钮,结束整个 SQL Server 2005 的安装过程。



图 6-13 “安装进度”对话框



图 6-14 安装完提示信息

(二) SQL Server 2005 的配置

完成了 SQL Server 2005 的安装后,首要的问题是配置 SQL Server 2005。

1. SQL Server 2005 的服务配置

执行“开始”→Microsoft SQL Server 2005→“配置工具”→SQL Server Configuration Manager 命令,运行 SQL Server 配置管理器,然后单击配置管理器左窗格中的“SQL Server 配置管理器(本地)”将其展开,如图 6-15 所示。



图 6-15 SQL Server 2005 配置管理器

配置 SQL Server 2005 服务的步骤如下：

(1) 在如图 6-15 所示的 SQL Server 配置管理器的左窗格中,单击“SQL Server 2005 服务”将其展开。在右窗格中列出了当前可配置的 SQL Server 2005 服务项目。

(2) 右击要设置的服务项目,在弹出的快捷菜单中选择“属性”命令,如图 6-16 所示。打开服务项目的属性对话框,该对话框有 3 个选项卡,其中“登录”选项卡如图 6-17 所示,可以为服务指定登录身份。其中内置账户是 Windows 绑定账户,本账户由 SQL Server 管理。

(3) 单击“服务”选项卡,可以配置服务的手动、自动、已禁用 3 种启动模式,如图 6-18 所示。“高级”选项卡是服务的一些高级属性,一般情况下无须修改。



图 6-16 选择“属性”命令



图 6-17 “登录”选项卡



图 6-18 “服务”选项卡

2. SQL Server 2005 的网络配置

(1) 在如图 6-15 所示的配置管理器中,单击左窗格的“SQL Server 2005 网络配置”,将其展开可以看到“MSSQLSERVER 的协议”节点,窗口右边列出了当前实例所应用的协议及其运行状态,如图 6-19 所示。



图 6-19 配置 SQL Server 2005 的网络

(2) 如果要启用或禁用某一项网络协议,可以右击该协议,在弹出的快捷菜单中选择“启用”或“禁用”选项。如果要对选中的网络协议进行设置,可右击该网络协议,在弹出的快

捷菜单中选择“属性”命令,在打开的协议属性对话框中进行设置,如图 6-20 所示。



图 6-20 设置网络协议属性

3. SQL Server 2005 的本机客户端配置

在如图 6-15 所示的配置管理器中,单击左窗格的“SQL Native Client 配置”,将其展开后可以配置 SQL Server 2005 客户端协议,如启用、设置协议顺序等,以及根据协议设置一个预定义的客户端和服务端之间连接的别名。

第二节 SQL Server 2005 常用管理工具

SQL Server 2005 提供了各种帮助数据库管理员和开发人员提高工作效率的工具,通过这些工具可以完成数据库的配置、管理和开发等任务。因此,在使用 SQL Server 2005 之前,认识各种工具及其特性是非常重要的。

一、SQL Server Management Studio

SQL Server Management Studio 是 SQL Server 2005 中最重要的一个管理工具,用于访问、配置、控制、管理和开发 SQL Server 的所有组件,是将早期的 SQL Server 2000 中所包含的企业管理器、查询分析器和 Analysis Manager 功能整合到同一环境中,形成了用于数据库管理的功能丰富的图形工具与脚本编辑器。作为开发和管理 SQL Server 数据库对象的有效工具,SQL Server Management Studio 可以完成对 SQL Server 2005 的管理,如创建服务器组、注册服务器、配置服务器选项、创建和管理数据库对象等。

下面简单介绍 SQL Server Management Studio 的使用方法。

(1) 执行“开始”→“程序”→Microsoft SQL Server 2005→SQL Server Management Studio 命令,打开如图 6-21 所示的对话框。单击“连接”按钮,连接到 SQL Server 服务器。



图 6-21 “连接到服务器”对话框

(2) 与服务器连接之后的 SQL Server Management Studio 的集成环境如图 6-22 所示。

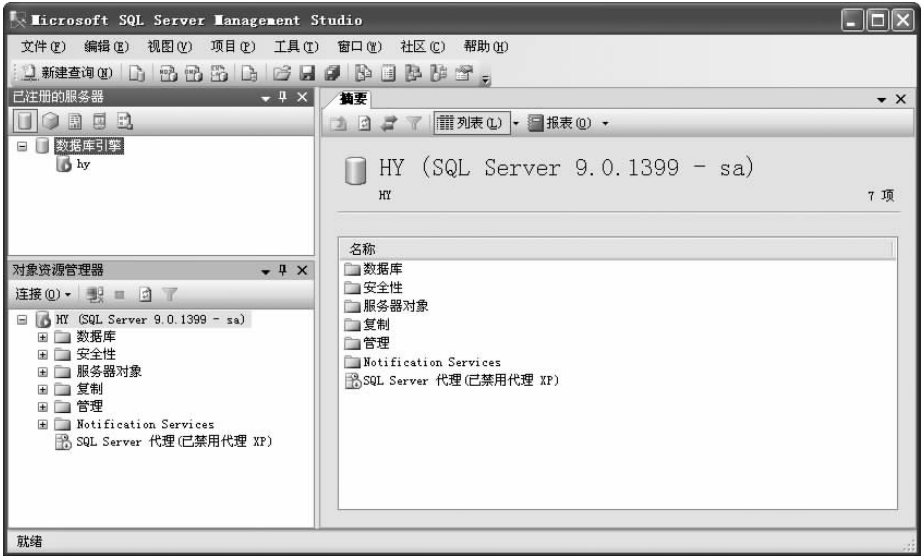


图 6-22 SQL Server Management Studio 集成环境

SQL Server Management Studio 是一个功能强大且灵活的工具,由于集成了很多窗格,如果将所有窗格都打开的话,就会占用整个屏幕空间。为了更好地利用屏幕空间,可以关闭、隐藏或移动这些组件窗格。通过菜单栏上的“视图”菜单可以进行定制。下面简单介绍 SQL Server Management Studio 的主要组成部分。

(一) “已注册的服务器”窗格

“已注册的服务器”窗格用于显示所有已注册的服务器,可以在此添加和删除服务器。

(二) “对象资源管理器”窗格

“对象资源管理器”窗格是服务器中所有数据库对象的树形结构视图。用户可以通过该窗格操作数据库,如创建、修改、删除数据库、表、视图等数据库对象,创建登录用户和授权,

进行数据库的备份和恢复等操作。

(三) 查询编辑器

SQL Server Management Studio 集成了用于编写 Transact-SQL(T-SQL)查询语句的查询编辑器。查询编辑器是一个提供了图形界面的查询管理工具,它与 SQL Server 2000 中的查询分析器类似,是数据库管理员或开发人员执行 T-SQL 语句的工具。在开发和维护应用系统时,查询编辑器是最常用的工具之一。其具体启动过程如下:

在图 6-22 的工具栏中单击“新建查询”按钮,在 SQL Server Management Studio 主窗口的右边产生一个新的查询编辑器,如图 6-23 所示。其中上面是 SQL 代码窗口,用于输入 SQL 的查询语句;下面是结果窗口,用于显示查询结果和分析结果。多次单击“新建查询”按钮,将会出现多个查询编辑器,切换查询编辑器上的选项卡可以在不同的查询编辑器中编辑 T-SQL 语句。

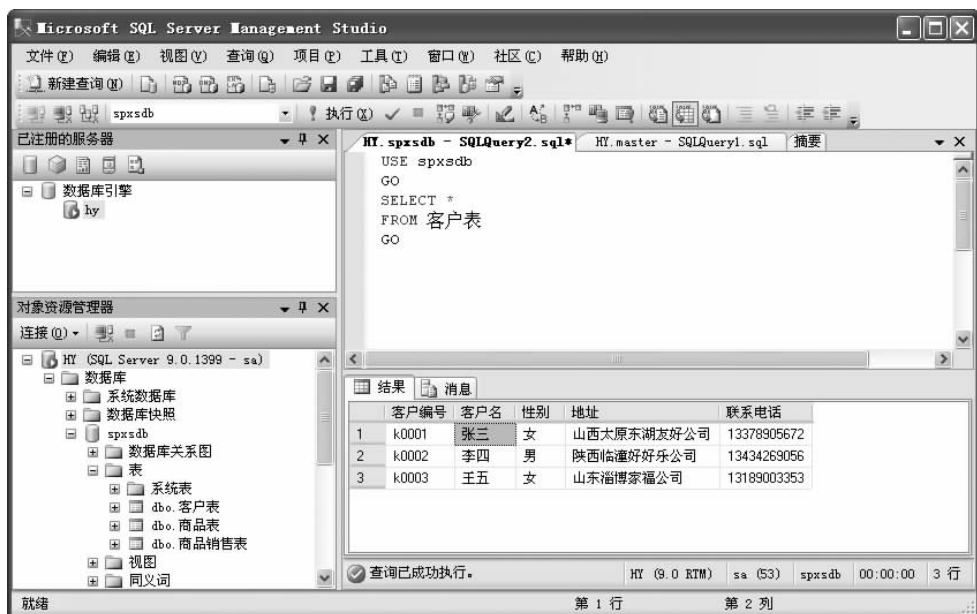


图 6-23 打开查询编辑器

要想使用查询编辑器,就必须掌握 T-SQL 语句。在查询编辑器的代码窗口中输入 T-SQL 语句,输入完毕后,单击查询编辑器上方工具栏中的“分析”按钮 ,检查输入的 T-SQL 语句是否有语法错误,如果有语法错误,则进行修改;如果语句分析正确,则单击工具栏的“执行”按钮 ,执行该 T-SQL 语句,并在结果窗口中显示执行结果。

对于 T-SQL 的执行结果,可以通过执行“查询”→“将结果保存到”命令,选择输出形式将其输出。常用的输出结果的形式有:以网格形式显示结果、以文本形式显示结果、将结果保存到文件。

二、SQL Server 2005 的配置工具

SQL Server 2005 的配置工具包括 SQL Server Configuration Manager、Reporting

Services 配置、Notification Services 命令提示、SQL Server 错误和使用情况报告和 SQL Server 外围应用配置器。

(一) SQL Server Configuration Manager

SQL Server Configuration Manager(SQL Server 配置管理器)是一个管理工具,用于管理与 SQL Server 有关的连接服务,如 SQL Server 客户端计算机与服务器端计算机的连接配置。SQL Server Configuration Manager 实际上是将 SQL Server 2000 中的“服务管理器”、“服务器网络实用工具”、“客户端网络实用工具”3 个工具集成在了同一个工具中,可以完成 3 个工具所能完成的工作。

(二) Reporting Services 配置

Reporting Services 配置(报表服务器配置)的作用是配置和管理 SQL Server 2005 的报表服务器。

(三) Notification Services 命令提示

Notification Services 命令提示即通知服务命令提示,执行“开始”→“程序”→Microsoft SQL Server 2005→“配置工具”→“Notification Services 命令提示”命令,打开如图 6-24 所示的命令提示窗口。常用的命令有 nscontrol,用来部署、配置、监视和控制 Notification Services 实例和应用程序。



图 6-24 Notification Services 命令提示窗口

(四) SQL Server 错误和使用情况报告

SQL Server 2005 的错误和使用情况报告的方式有两种:一种是将 SQL Server 2005 的所有组件和实例的错误报告发送到 Microsoft 公司或错误报告服务器;另一种是将 SQL Server 2005 的所有组件和实例的功能使用情况报告发送到 Microsoft 公司。Microsoft 公司希望可以收集到错误信息和使用情况,以便改进 SQL Server。SQL Server 2005 的错误和使用情况报告的使用方法如下:

执行“开始”→“程序”→Microsoft SQL Server 2005→“配置工具”→“SQL Server 错误和使用情况报告”命令,打开“错误和使用情况报告设置”窗口,选择想要的发送方式,单击“确定”按钮完成设置。

(五) SQL Server 外围应用配置器

SQL Server 外围应用配置器可以启用或禁用、开始或停止 SQL Server 2005 安装的一些功能和服务。配置对本地和远程 SQL Server 服务器的外围应用,是为了增加 SQL Server 的安全性,因此应该尽量减少外围应用。

三、SQL Server 2005 的性能工具

SQL Server 2005 的性能工具包括 SQL Server Profiler 和数据库引擎优化顾问。

(一) SQL Server Profiler

SQL Server Profiler(SQL Server 事件探查器)与 SQL Server 2000 中的事件探查器一样,是用于从服务器上捕获 SQL Server 2005 事件的工具。

捕获后的事件保存在一个跟踪文件中,根据这个跟踪文件,可以分析有问题的查询并找到问题的所在,查找导致 SQL Server 运行缓慢的查询,捕获导致某个问题的 T-SQL 语句,以及监视 SQL Server 的性能。

执行“开始”→“程序”→Microsoft SQL Server 2005→“性能工具”→SQL Server Profiler 命令,打开 SQL Server Profiler 窗口。在菜单栏中执行“文件”→“新建跟踪”命令,打开“连接到服务器”对话框,连接到服务器后,打开“跟踪属性”对话框。“跟踪属性”对话框有两个选项卡:“常规”选项卡用于设置跟踪名称、使用的模板、保存到文件的地址和名称等属性;“事件选择”选项卡用于设置跟踪的事件和事件列,如图 6-25 所示。设置完毕后,单击“运行”按钮,就可以根据结果分析出现问题的原因。

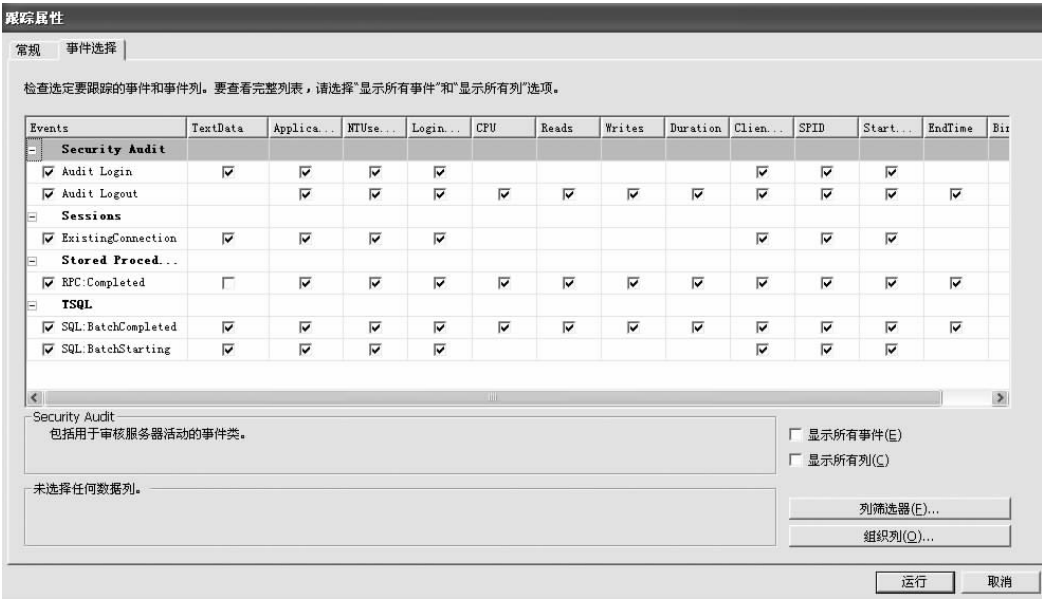


图 6-25 “事件选择”选项卡

(二) 数据库引擎优化顾问

数据库引擎优化顾问是 SQL Server 2005 相对于以前版本所提供的的一个工具,通过它可以不让数据库管理员不必精通数据结构和 T-SQL,也可以完成对数据库的优化。

执行“开始”→“程序”→Microsoft SQL Server 2005→“性能工具”→“数据库引擎优化顾问”命令,打开数据库引擎优化顾问程序。与 SQL Server 服务器建立连接后,会出现 Database Engine Tuning Advisor 窗口。在该窗口中,可以设置会话的名称、工作负荷所用的文

件或表,选择要分析的数据库和表,然后单击“开始分析”按钮,进行分析。分析完毕后可以看到 SQL Server 2005 给出的优化建议及优化报告。

四、SQL Server 2005 的文档和教程

SQL Server 2005 为用户提供了学习平台,执行“开始”→“程序”→Microsoft SQL Server 2005→“文档和教程”命令,即可启动文档和教程。

第三节 创建服务器组和注册服务器

创建服务器组可以将众多已注册的服务器进行分组管理。而通过注册服务器,可以存取服务器连接信息,以供连接服务器时使用。

一、服务器组的创建与删除

可以在 SQL Server Management Studio 中创建服务器组,并将服务器放在该服务器组中。服务器组提供了一种简便的方法,可将大量的服务器按照不同的用途和类型组织在几个易于管理的组中。

(一) 创建服务器组

创建服务器组的操作步骤如下:

(1) 执行“开始”→“程序”→Microsoft SQL Server 2005→SQL Server Management Studio 命令。

(2) 打开“连接到服务器”对话框(见图 6-21),单击“取消”按钮,打开如图 6-26 所示的 Microsoft SQL Server Management Studio 窗口。



图 6-26 Microsoft SQL Server Management Studio 窗口

(3) 在图 6-26 中执行“视图”→“已注册的服务器”命令,打开“已注册的服务器”窗格,在该窗格中选择相应的服务器类型“数据库引擎”,然后右击“数据库引擎”,在弹出的快捷菜单中选择“新建”→“服务器组”命令,如图 6-27 所示,弹出“新建服务器组”对话框。

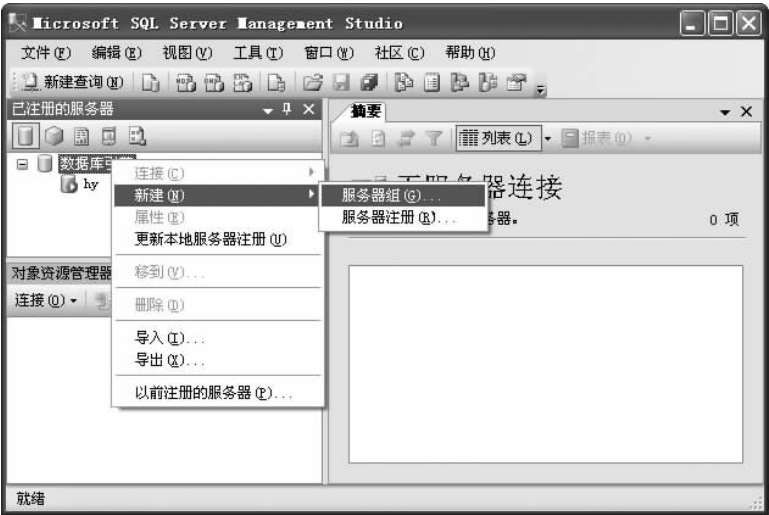


图 6-27 新建服务器组菜单

(4) 输入新建服务器组的名称 group1,如图 6-28 所示,输入服务器组的描述信息,并选择新建服务器组的位置,可以是顶层组(默认值),或者是某一个服务器组的子组,在此选择顶层组。



图 6-28 “新建服务器组”对话框

(5) 单击“保存”按钮,就成功地创建了一个服务器组 group1,该组没有数据库服务器,结果如图 6-29 所示。

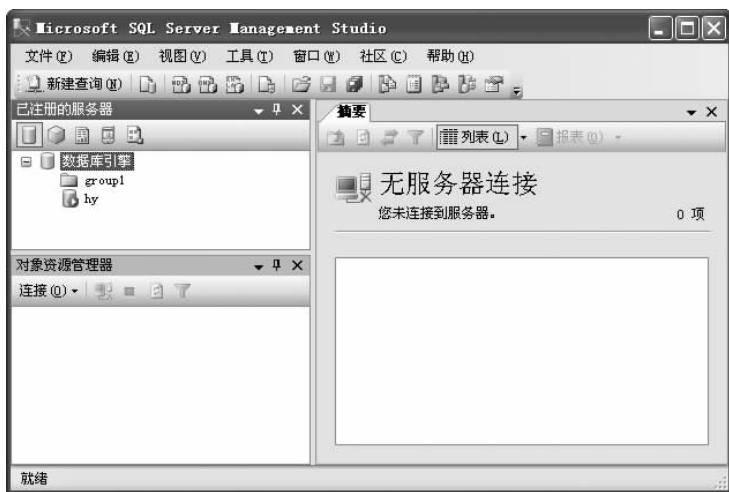


图 6-29 新建的 group1 服务器组

(二) 删除服务器组

删除服务器组的操作步骤如下：

(1) 按照创建服务器组时打开“已注册的服务器”窗格的步骤打开“已注册的服务器”窗格。

(2) 右击需要删除的服务器组，在弹出的快捷菜单中选择“删除”命令。

(3) 弹出“确认删除”对话框，单击“是”按钮即可完成服务器组的删除。

注意：在删除服务器组的同时，也会将该组内所注册的服务器一同删除。

二、服务器的注册、删除和连接

(一) 服务器的注册与删除

SQL Server 2005 可以管理多个不同的服务器实例，为了让 SQL Server 管理工具实现对后台数据库的管理，必须对需要进行管理的本地或远程服务器进行注册。在注册服务器时必须指定服务器名称、登录到服务器时使用的安全类型、注册服务器后要将该服务器加入其中的组的名称。

1. 注册服务器

注册服务器就是在 Microsoft SQL Server Management Studio 中登记服务器，然后把它加入一个指定的服务器中。注册服务器的操作步骤如下：

(1) 右击“已注册的服务器”窗格的空白处，或右击新建的服务器组 group1，在弹出的快捷菜单中选择“新建”→“服务器注册”命令。

(2) 打开“新建服务器注册”对话框，该对话框中有“常规”和“连接属性”两个选项卡。“常规”选项卡用于设置服务器的类型、名称、登录时的身份验证方式、已注册的服务器名称等信息，如图 6-30 所示。“连接属性”选项卡用于设置所要连接服务器中的数据库、使用的网络协议、连接时等待建立连接的时间、连接后等待任务执行的时间等信息，如图 6-31 所示。

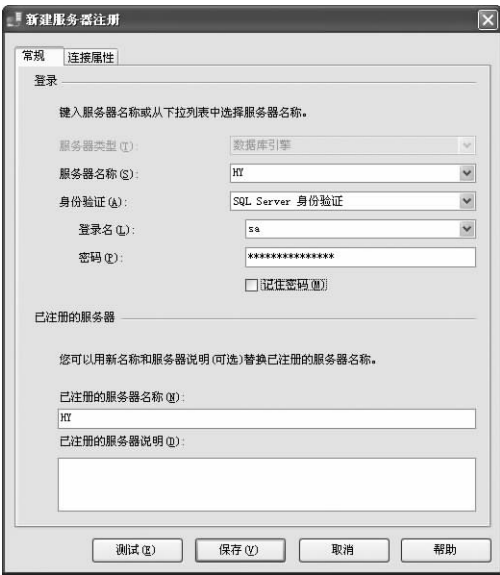


图 6-30 “常规”选项卡



图 6-31 “连接属性”选项卡

(3) 设置完成后,单击“测试”按钮测试能否与所注册的服务器连接,如果成功连接,会弹出如图 6-32 所示的提示信息,单击“确定”按钮。

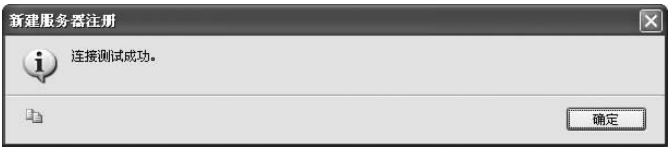


图 6-32 提示连接测试成功

(4) 返回“新建服务器注册”对话框,单击“保存”按钮,确定注册,在 Microsoft SQL Server Management Studio 窗口中会出现新注册成功的服务器图标。

2. 删除服务器

删除服务器的操作步骤如下:

- (1) 在“已注册的服务器”窗格中右击需要删除的服务器,在弹出的快捷菜单中选择“删除”命令。
- (2) 在弹出的“确认删除”对话框中单击“是”按钮,即可完成对服务器的删除。

(二) 服务器的连接

在对象资源管理器中,单击工具栏的“连接”下拉按钮,在下拉菜单中选择要连接的服务器类型(如数据库引擎),如图 6-33 所示,打开“连接到服务器”窗口,根据要连接的服务器在注册时设置的信息,正确选择服务器的类型、名称和身份验证模式。单击“连接”按钮,连接成功后,在 Microsoft SQL Server Management Studio 窗口中会出现所连接的数据库服务器上的各个数据库实例及各自的数据库对象。注册服务器连接成功后的 Microsoft SQL Server Management Studio 窗口如图 6-34 所示。



图 6-33 选择连接的服务器类型

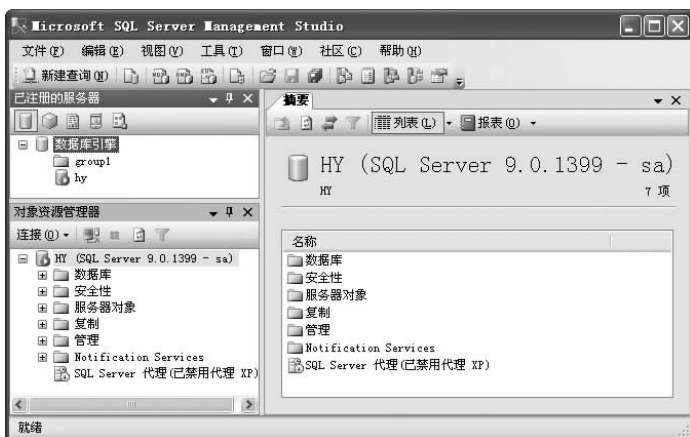


图 6-34 注册服务器连接成功后的窗口

三、启动和关闭 SQL Server 2005 服务器

通常情况下,SQL Server 服务器被设置为自动启动模式,在系统启动后,会以 Windows 后台服务的形式自动运行。但某些服务器的配置被更改后必须重新启动才会生效,此时需要数据库管理员先关闭服务器,再重新启动服务器。SQL Server 2005 的服务器可以通过 Microsoft SQL Server Management Studio、SQL Server Configuration Manager 管理工具和后台 3 种方式启动或关闭。

(一) 在 Microsoft SQL Server Management Studio 中关闭或启动服务器

在成功连接到 SQL Server 2005 数据库服务器后,打开 Microsoft SQL Server Management Studio 窗口,可以对服务器进行各种管理。

1. 关闭服务器

在“对象资源管理器”窗格中,右击要关闭的服务器,在弹出的快捷菜单中选择“停止”命令,如图 6-35 所示,出现“是否确实要停止服务器”的提示信息,单击“是”按钮,即可关闭选中的服务器,并停止相应的服务。服务器关闭后,其左侧的图标为.

2. 启动服务器

启动服务器的操作与关闭服务器类似,右击要启动的服务器,在弹出的快捷菜单中选择“启动”命令,即可启动选中的服务器。服务器启动后,其左侧的图标为.

(二) 在 SQL Server Configuration Manager 中关闭或启动服务器

利用 SQL Server Configuration Manager 启动和关闭服务器的操作类似于 SQL Server 2005 的服务配置,按照 SQL Server 2005 服务配置的步骤,展开“SQL Server 2005 服务”节点,在右侧窗格中右击要关闭的服务器,在弹出的快捷菜单(见图 6-16)中选择“停止”命令,即可关闭选中的服务器。要启动服务器,只需右击需要启动的服务器,在弹出的快捷菜单中选择“启动”命令即可。



图 6-35 选择“停止”命令

(三) 通过后台启动或关闭服务器

通过后台启动和关闭服务的操作步骤是：执行“开始”→“设置”→“控制面板”命令，在打开的“控制面板”窗口中双击“管理工具”选项，在打开的“管理工具”窗口中，双击“服务”选项，在打开的“服务”窗口中，找到并右击需要启动或关闭的 SQL Server 2005 服务器，在弹出的快捷菜单中选择“启动”或“停止”命令，如图 6-36 所示，即可启动或关闭服务器。



图 6-36 “服务”窗口

第四节 SQL Server 2005 数据库操作

数据库是 SQL Server 2005 最基本的操作对象之一,数据库的创建、删除、修改、查看和重命名是 SQL Server 2005 中最基本的操作,是进行数据库管理与开发的基础。

一、SQL Server 2005 的数据库基本结构

(一) 数据库的组成

SQL Server 2005 数据库主要由文件和文件组组成,数据库中的所有数据和对象都以文件的形式存储在磁盘中。

1. 数据库文件

在 SQL Server 2005 系统中,一个数据库在磁盘上可以保存为一个或多个文件,把这些文件称为数据库文件。数据和日志信息分别存储在不同的文件中,而且每个数据库都拥有自己的数据和日志信息文件。SQL Server 数据库文件根据其作用的不同,可以分为主数据文件、次数据文件和事务日志文件 3 种类型。

(1) 主数据文件(primary file):用来存放数据和数据库的初始化(启动)信息,是 SQL Server 数据库的主体,是每个数据库不可缺少的部分。每个数据库有且仅有一个主数据文件,用户数据和对象也可以存储在此文件中,主数据文件的扩展名为 .mdf。

(2) 次数据文件(secondary file):用来存储主数据文件没有存储的其他数据和对象。如果数据库中的数据量很大,除了将数据存储在主数据文件中以外,还可以将一部分数据存储在一次数据文件中;如果主数据文件足够大,能够容纳数据库中的所有数据,则该数据库不需要次数据文件。使用次数据文件可以将数据分散在多个磁盘上以提高读取速度。次数据文件的扩展名为 .ndf。

(3) 事务日志文件(transaction log file):用来记录数据库更新情况。SQL Server 2005 具有事务功能,可以保证数据库操作的一致性和完整性,用事务日志文件来记录所有事务及每个事务对数据库所作的修改。事务日志是数据库的重要组件,如果数据库遭到破坏,就需要事务日志恢复数据库内容。每个数据库至少拥有一个事务日志文件,也可以拥有多个数据文件。事务日志文件的扩展名为 .ldf。

在建立数据库时,需要注意以下两点:

- SQL Server 2005 不强制使用 .mdf、.ndf、.ldf 文件扩展名,但使用这些扩展名可以帮助标识文件的用途。
- SQL Server 2005 的每个数据库文件都有逻辑文件名和物理文件名两种名称。物理文件名是数据库文件在操作系统中存储的文件名,是操作系统文件的实际名称,每个物理文件名都有明确的存储位置(文件所在的路径)。其文件名称比较长,在 SQL Server 内部访问非常不便。因此,每个数据库又有逻辑文件名,逻辑文件名只在 T-SQL 语句中使用,是实际磁盘文件名的代号,比较简单,引用起来比较方便。一个

物理文件名对应一个逻辑文件名。

2. 数据库文件组

为了方便管理,可以将多个数据库文件集合起来形成一个整体,称为文件组。文件组是 SQL Server 2005 数据文件的一种逻辑管理单位,对文件分组的目的就是便于进行管理和数据分配。可以将文件组中的文件存放在不同的磁盘,以便提高数据库的访问性能。

每个文件组对应一个组名,SQL Server 2005 提供了 3 种类型的文件组:主(primary)文件组、用户自定义(user-defined)文件组、默认(default)文件组。

(1) 主文件组:主数据文件所在的组称为主文件组。当创建数据库时,如果用户没有定义文件组,系统会自动建立主文件组,当数据文件没有指定文件组时,默认都在主文件组中。主文件组包含了所有的系统表、主要数据文件和所有没有包含在其他文件组的次数据文件。

(2) 用户自定义文件组:用户定义的文件组也称为次文件组,包含所有在使用 create database 或 alter database 的 SQL 语句时,使用 filegroup 关键字来指定文件组的文件,该组包含逻辑上一体的数据文件 and 相关信息。创建用户自定义文件组的主要目的是便于数据分配。

(3) 默认文件组:包含所有在创建时没有指定文件组的表、索引等数据库对象。在每个数据库中,每次只能有一个文件组是默认文件组。可以在用户自定义文件组中指定一个默认文件组,如果没有指定默认文件组,则系统将主文件组认定为默认文件组。

大多数数据库只需要一个文件组和一个日志文件就可以很好地运行,如果数据库中的文件很多,则需要创建用户自定义文件组。

使用数据库文件和文件组时,必须遵循以下规则:

- 一个数据文件只能存在于一个文件组中,不能存在于两个或两个以上的文件组中。
- 一个文件组也只能被一个数据库使用。
- 日志文件不属于任何文件组。

(二) 常用数据库对象

SQL Server 数据库中的数据在逻辑上被组织成一系列对象,当用户连接到数据库后,所看到的是逻辑对象,而不是物理的数据库文件。数据库对象就是存储、管理和使用的不同结构形式,包括表、视图、存储过程、索引、触发器、规则、默认等。在图 6-37 所示的对象资源管理器中,可以看到 SQL Server 将服务器的数据库组织成一个树形逻辑结构,在该结构中有若干节点,每个节点又包括很多子节点,它们代表与该特定数据库有关的不同类型的对象。

(三) 数据库的种类

SQL Server 2005 中的数据库可分为示例数据库、用户数据库和系统数据库。示例数据库是系统为了用户学习和理解 SQL Server 2005 而设计的。例如,Adventure Works 和 Adventure Works DW 是 SQL Server 2005 中的示例数据库(如果安装时选择了),还有 pubs 和 northwind 数据库也是示例数据库。用户数据库是用户根据事务管理需求创建的数据库,如商品销售数据库等。系统数据库是 SQL Server 内置的,主要用于系统管理,是在安装 SQL Server 2005 时自动安装的。SQL Server 2005 中的系统数据库有 master、model、tempdb、msdb、resource 等。

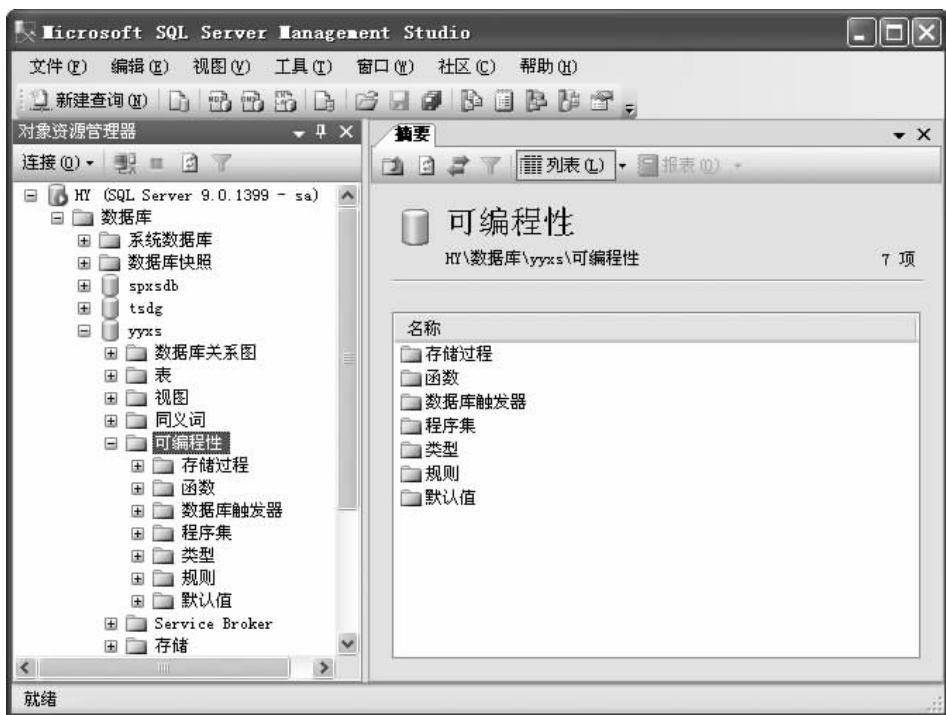


图 6-37 SQL Server 2005 的数据库对象

(四) 数据库对象的架构

架构是 SQL Server 2005 的新增功能,是一种允许用户对数据库对象进行分类的容器对象,是形成单个命名空间的数据库对象的集合。命名空间是一个集合,其中每个元素的名称都是唯一的。在同一架构中不能有相同类型、相同名称的数据库对象。例如,为了避免名称冲突,同一架构中不能有两个同名的表,两个表只有在位于不同的架构中时才可以同名。

架构对如何引用数据库对象具有很大的影响,在 SQL Server 2005 中,数据库对象除了在命名时需要遵循命名规则之外,在引用时,同样需要遵循引用规则。一个数据库对象通过 4 个命名部分组成的结构来引用,引用数据库的语法格式为

服务器名. 数据库名. 架构名. 数据库对象名

由此可以看出,数据库从属于 SQL Server 服务器,架构从属于数据库,这些实体是嵌套在一起的,架构下面可以包含很多安全对象,但不能包括其他架构。

在 SQL Server 2005 中,默认的架构是 dbo(database owner),如果在创建数据库对象时没有指定架构,那么默认的数据库对象放在 dbo 架构中。例如,引用服务器 HY 上的数据库 books 中的图书表 book 时,完整的引用为 HY. books. dbo. book。虽然 SQL Server 对数据库对象的引用包括 4 个部分,但在实际运用中,在能够区分对象的前提下,前三个部分是可以省略的。当要访问的数据库对象在当前数据库上时,则可以省略服务器名和数据库名,但要指定架构名。当要访问的数据库对象与正在使用的数据库对象有相同的架构时,则可以只写数据库对象名。例如,“dbo. 图书表”表示省略了服务器名和数据库名。

二、使用 SQL Server Management Studio 创建数据库

在介绍使用 SQL Server Management Studio 创建数据库之前,先介绍数据库的命名规则。

(一) 数据库的命名规则

在 SQL Server 中创建数据库时,其名称必须遵循 SQL Server 2005 的标识命名规则。数据库的命名规则取决于数据库的兼容级别。兼容级别可以为 60、65、80 和 90,一般来说,SQL Server 2000 使用的级别是 80,SQL Server 2005 使用的级别是 90。当兼容级别为 90 时,其命名规则如下:

- 名称长度为 1~128,不能超过 128 个字符,本地临时表的名称不能超过 116 个字符。
- 名称的第一个字符必须是英文字母、下划线、@、# 等符号;除第一个字符之外的其他字符,还可以包括数字、\$。
- 在中文版的 SQL Server 2005 中,可以直接使用中文名称。
- 名称中不能有空格,不允许使用 SQL Server 2005 的保留字,如系统数据库 master、model,这样的数据库名称都属于保留字。

需要注意的是,在 T-SQL 中,以“@”开头的变量表示局部变量,以“@@”开头的变量表示全局变量,以“#”开头的表示全局临时对象,所以用户在命名数据库时最好不要以这些字符开头,以免引起混乱。

(二) 创建数据库

在开发 SQL Server 2005 数据库应用程序之前,首先要设计数据库结构并创建数据库。创建数据库时需要对数据库的属性进行设置,包括数据库的名称、所有者、大小以及存储数据库的文件和文件组。

在 SQL Server 2005 中创建数据库有两种方法:一种是使用 SQL Server Management Studio 创建数据库,此方法采用图形化操作界面,简单、直观,适合初学者学习;另一种是使用 T-SQL 语句创建数据库,此方法难度稍大,需要对 T-SQL 语法和语句非常熟悉,但可以将数据库的脚本保存下来,在其他计算机上运行以创建相同的数据库,对于高级用户,此方法使用起来更加得心应手。

本节主要介绍使用 SQL Server Management Studio 创建数据库,第二种方法将在第七章介绍。

在 SQL Server Management Studio 中创建数据库的具体步骤如下:

(1) 启动 SQL Server Management Studio,并连接到 SQL Server 2005 中的数据库,在“对象资源管理器”窗格中展开服务器→“数据库”节点,右击“数据库”节点,在弹出的快捷菜单中选择“新建数据库”命令,如图 6-38 所示。打开如图 6-39 所示的“新建数据库”窗口。该窗口的“选择页”列表框中有常规、选项和文件组 3 个选项。

(2) 单击“常规”选项,在“数据库名称”文本框中输入要创建的数据库名称 books。SQL Server 2005 会自动在数据库文件列表中产生一个主数据文件 books.mdf 和一个日志文件 books_log.ldf,同时显示文件组、自动增长和路径的默认值,用户可以根据需要修改这些默认设置,也可以单击右下角的“添加”按钮添加数据文件。这里均采用默认值。

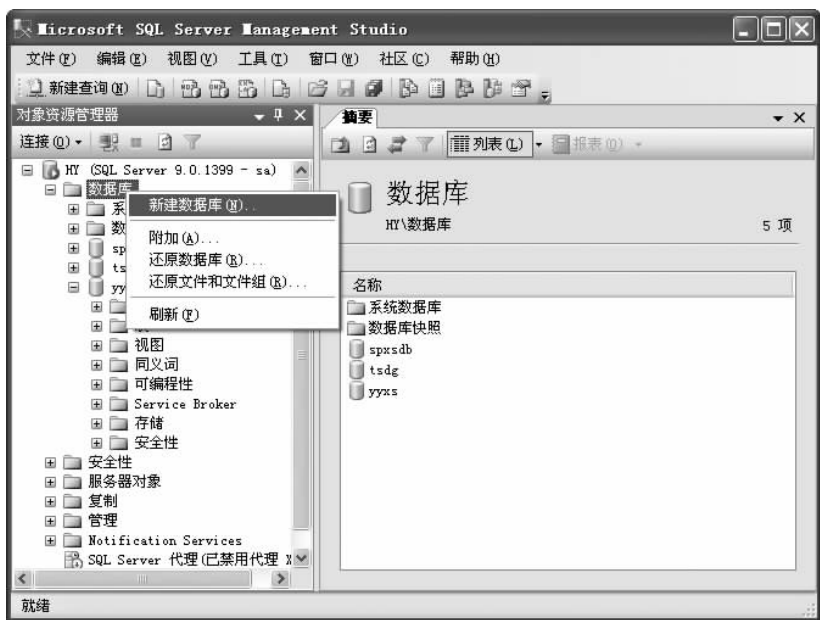


图 6-38 选择“新建数据库”命令

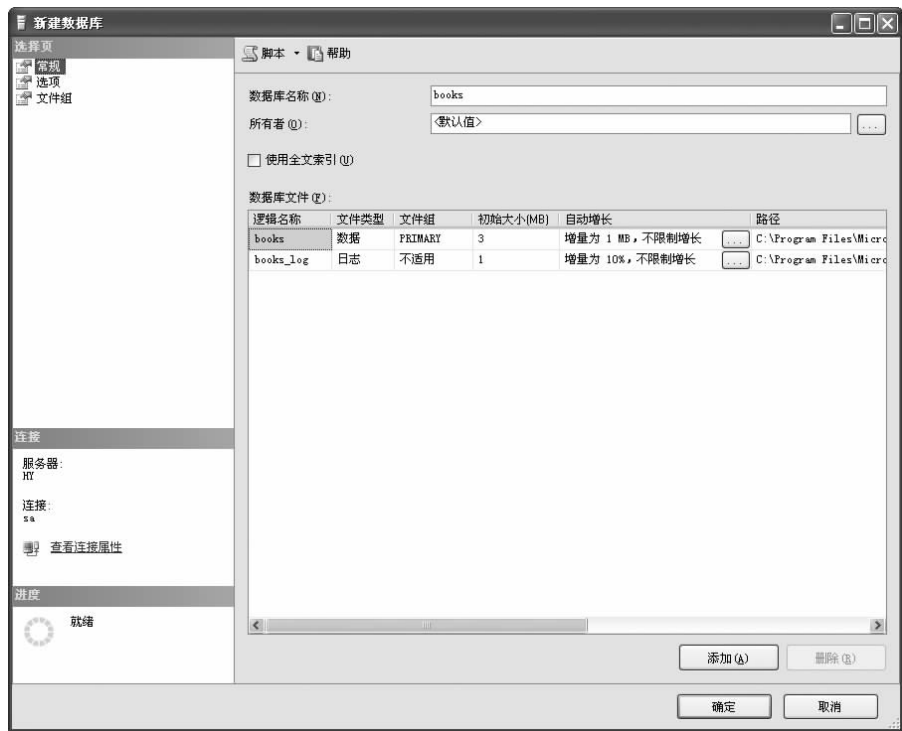


图 6-39 “新建数据库”窗口

(3) 单击“选项”选项，设置数据库的排序规则、恢复模式、兼容级别和其他选项，如图 6-40 所示，这里采用默认设置。

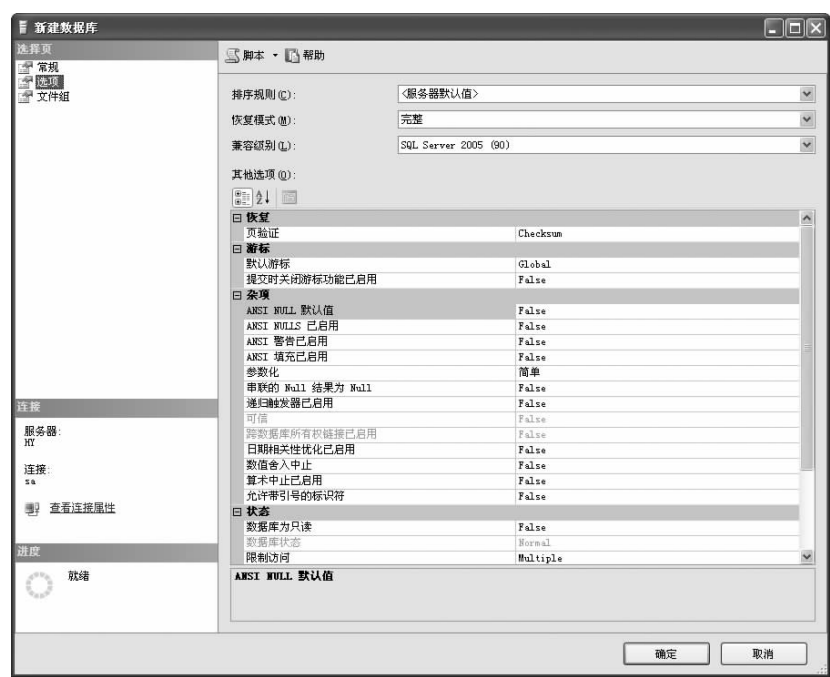


图 6-40 “选项”选项

(4) 在“新建数据库”窗口中单击“文件组”选项，设置或添加文件组的属性，如是否只读、是否为默认文件组，单击“添加”或“删除”按钮即可添加或删除用户自定义的文件组。

本例为 books 数据库添加文件组 booksfg。先在“新建数据库”窗口中单击“文件组”选项，然后单击“添加”按钮，在文件组列表框中输入一个新文件组名 booksfg，如图 6-41 所示。

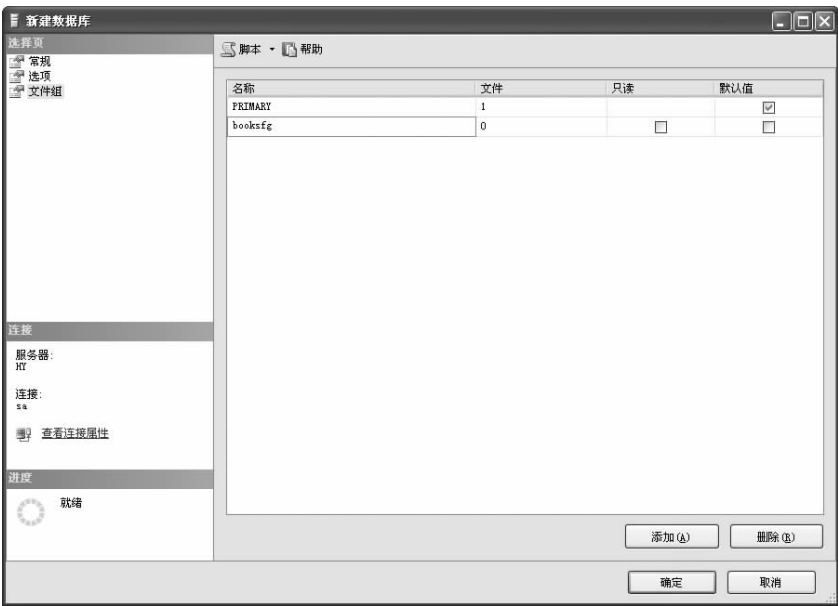


图 6-41 设置“文件组”选项

(5) 在“新建数据库”窗口中选择“常规”选项,分别为各个文件指定文件组,如图 6-42 所示。

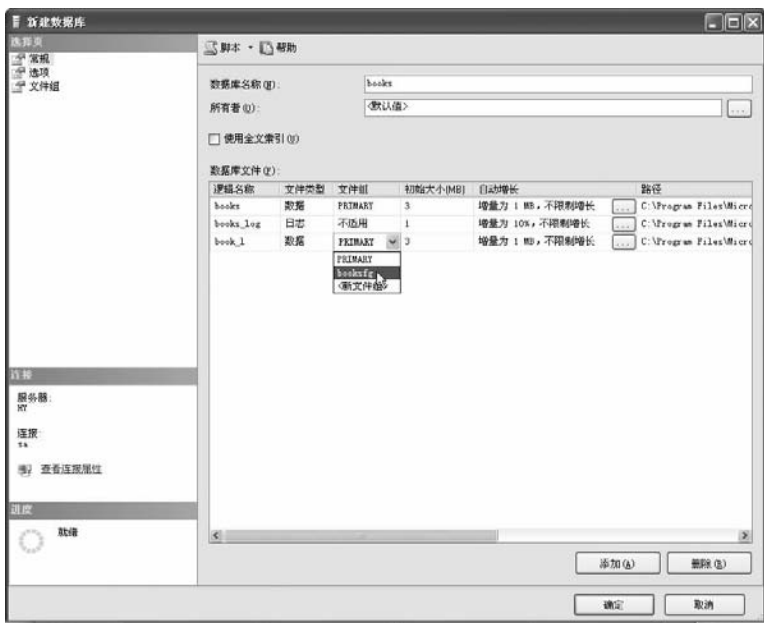


图 6-42 为文件指定文件组

(6) 单击“确定”按钮,系统开始创建数据库。创建成功后,刷新“对象资源管理器”窗格的“数据库”节点的内容,再展开“数据库”节点,则会显示出新创建的数据库 books。

三、数据库的管理

随着数据库的增长或变化,用户需要用手动方式对数据库进行管理,数据库的管理包括修改数据库的配置、删除数据库、查看数据库的属性、更改名称、扩充或收缩数据库与日志文件等。

(一) 查看和修改数据库参数

对于已有的数据库,可以查看数据库的属性,在数据库创建完成后,又可能因为种种原因,需要修改数据库的设置。修改数据库的主要内容包括:扩充数据库的数据或事务日志的存储空间,增加或减少数据文件和事务日志文件,更改数据库的名称、数据库的配置,收缩数据库或事务日志空间等。

查看和修改数据库参数可以在 SQL Server Management Studio 中进行,也可以用 T-SQL 语句完成。在此介绍第一种方法。

在 SQL Server Management Studio 中查看和修改数据库的步骤如下:

(1) 打开 SQL Server Management Studio,在“对象资源管理器”窗格中展开服务器,定位到要查看的数据库,如 books。

(2) 右击目标数据库 books,在弹出的快捷菜单中选择“属性”命令,打开如图 6-43 所示的数据库属性窗口。

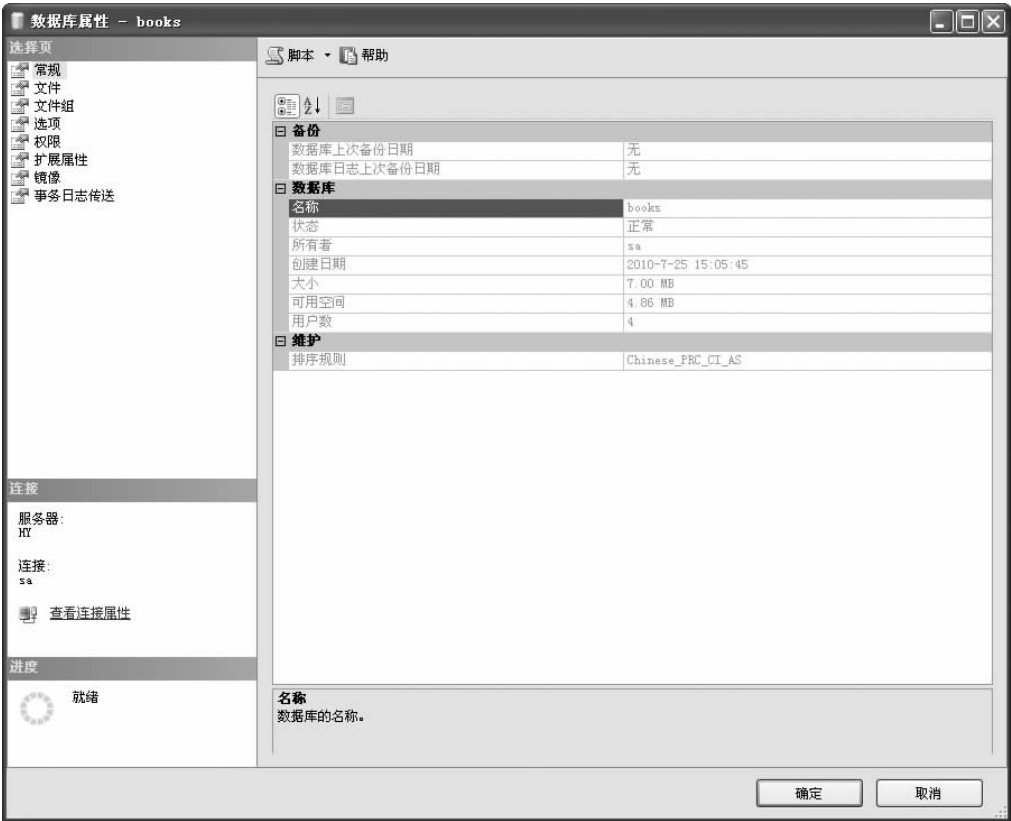


图 6-43 数据库属性窗口

(3) 在“选择页”列表框中选择“常规”选项,可看到该数据库的基本信息,如数据库的名称、状态、所有者、创建数据库的时间、大小、数据文件和日志文件剩余的可用空间、数据库的备份、连接数据库的用户数等信息。

(4) 选择“文件”选项,可以添加、删除数据库文件以及修改数据库文件的相关属性。

(5) 选择“文件组”选项,可以添加、删除文件组以及修改文件组的相关属性。

(6) 选择“选项”选项,可以控制数据库是单用户使用模式还是多用户使用模式、数据库是否仅可读,设置此数据库是否自动关闭、自动收缩和数据库的兼容级别等。这里对数据库 books 进行手动收缩,将“自动收缩”选项设置为 True,如图 6-44 所示。

(7) 设置好后,单击“确定”按钮即可。

(二) 重命名数据库

在 SQL Server 2005 中,用户可以根据需要修改数据库的名称,其名称可以包含任何符合标识符规则的字符。

更改数据库的名称可以在 SQL Server Management Studio 的“对象资源管理器”中修改,也可以用 T-SQL 语句进行更改。

在 SQL Server Management Studio 中修改数据库名称的步骤如下:

(1) 在“对象资源管理器”中展开“数据库”节点。

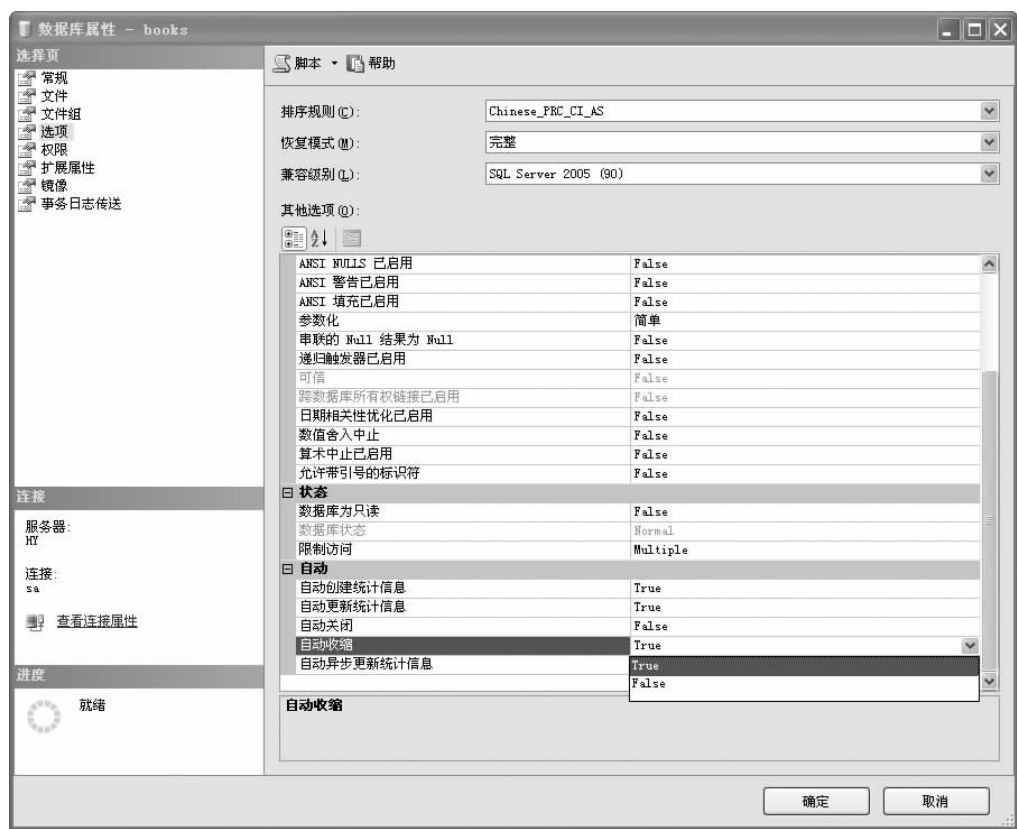


图 6-44 设置自动收缩

(2) 右击要重命名的数据库 books,在弹出的快捷菜单中选择“重命名”命令。此时数据库名处于可修改的状态,直接输入修改后的数据库名称,如 book,然后按 Enter 键进行确认即可。

需要注意的是,在重命名数据库之前,应确保没有用户在使用该数据库,而且该数据库已被设置为单用户模式。

(三) 删除数据库

删除数据库也是数据库管理中的重要操作之一,如果用户不再需要某一数据库,就可以将其从 SQL Server 服务器上删除,以释放该数据库所占用的磁盘空间。

删除数据库一定要慎重,因为数据库的删除是彻底地将相应的数据库文件从物理磁盘上删除,是永久性的。一旦删除,与此数据库有关联的数据库文件和事务日志文件都会被删除,存储在系统数据库中有关该数据库的所有信息也会被删除。如果数据库正在被用户使用,则无法将其删除。删除数据库仅限于 dbo(database owner)和 sa(super administrator)用户。

在 SQL Server Management Studio 中删除数据库的步骤如下:

- (1) 在 SQL Server Management Studio 中的“对象资源管理器”中展开“数据库”节点。
- (2) 右击要删除的数据库 book,在弹出的快捷菜单中选择“删除”命令,弹出“删除对象”

窗口。

(3) 单击“确定”按钮,即可删除数据库。

如果在“删除对象”窗口的下方选中了“删除数据库备份和还原历史记录信息”复选框,那么在删除数据库的同时,也将从 msdb 数据库中删除该数据库的备份和还原历史记录。如果选中了“关闭现有连接”复选框,在删除数据库前,SQL Server 会自动将所有与该数据库相连的连接全部关闭后,再删除数据库。如果没有选中该复选框,而且有其他活动连接在要删除的数据库时,将会出现错误信息。

需要注意的是,系统数据库 msdb、model、master、tempdb 无法删除,删除数据库后应立即备份 master 数据库,因为删除数据库将更新 master 数据库中的信息。

第五节 SQL Server 2005 数据库基本对象操作

数据库创建好后,就可投入使用,用户可以根据需要在数据库中建立相关的对象,如表、视图等。其中,表是 SQL Server 数据库系统的基本信息存储结构。本节主要介绍如何利用 SQL Server Management Studio 的“对象资源管理器”创建各种数据库对象。

一、表的创建和维护

SQL Server 数据库中的表是一个非常重要的数据库对象,是数据存放的地方,也称基本表。用户所关心的数据都存储在表中,对数据的访问、维护都是通过对表的操作实现的,一个数据库管理员在数据库中打交道最多的也是表。因此掌握数据库表的操作就显得尤为重要。

表的创建一般要经过定义表的结构、设置约束和添加数据 3 个步骤,其中设置约束可以在定义表结构时进行,也可以在表结构定义完成后,在已创建好的表结构上通过修改表的方式添加约束。

(一) SQL Server 2005 的数据类型

在创建表时,需要确定表的结构,也就是确定表中的各列及各列的数据类型,例如,是字符型、货币型还是数值型。只有设计好表的结构,系统才会在磁盘上开辟相应的空间,用户才能在表中填写数据。因此在创建表之前,首先介绍 SQL Server 2005 的数据类型。

数据类型是指用于存储、检索以及解释数据值类型的预先定义的命名方法,它决定了数据在计算机中的存储方式,代表不同的信息类型。在 SQL Server 2005 中,为表中的列选择合适的数据类型尤为重要,因为它影响着系统的空间利用、性能和是否易于管理等特性。数据分为系统数据和用户自定义数据两大类。

1. 系统数据类型

SQL Server 2005 提供了 20 多种系统数据类型,可以支持大部分数据库应用系统。按系统数据的表现形式和存储方式分为整数型、二进制型、字符型、日期型、精确数值型和货币型等。具体介绍如表 6-1 所示。

表 6-1 SQL Server 2005 的基本数据类型

类型名称	说 明	取值范围
bit	整数型	0、1,常用于表示真假逻辑关系,如 yes/no,占 1 字节
int		可存储 $-2^{31} \sim 2^{31}-1$ 范围的整数数据,占 4 字节
smallint		可存储 $-2^{15} \sim 2^{15}-1$ 范围的整数数据,占 2 字节
tinyint		可存储 0~125 的所有整数,占 1 字节
bigint		可存储 $-2^{63} \sim 2^{63}-1$ 范围的整数数据,占 8 字节
char	字符型	存放 1~8 000 固定长度的字符
varchar		存放 1~8 000 可变长的字符
text		存放最大长度为 $2^{31}-1$ 的字符数据
nchar	Unicode	存放 1~4 000 固定长度的字符
nvarchar		存放 1~4 000 可变长的字符
ntext		存放最大长度为 $2^{30}-1$ 的字符数据
datetime	日期型	日期为 1/1/1753 至 12/31/9999,占 8 字节
smalldatetime		日期为 1/1/1900 至 6/6/2079,占 4 字节
money	货币型	可存储 $-2^{63} \sim 2^{63}-1$ 范围的货币型数据,精确到 0.01,占 4 字节
smallmoney		可存储-214 748.346 8~214 748.3 467 范围的货币型数据
decimal	精确数值型	可存储 $-10^{38}+1 \sim 10^{38}-1$ 范围的数据,占 2~17 字节
numeric		可存储 $-10^{38}+1 \sim 10^{38}-1$ 范围的数据,占 2~17 字节
float	浮点型	可存储 $-1.79\text{E}+308 \sim 1.79\text{E}+308$ 范围的数据,占 8 字节
real		可存储 $-3.40\text{E}+38 \sim 3.40\text{E}+38$ 范围的数据,占 4 字节
binary	二进制型	存储 1~8 000 字节的定长二进制数据
varbinary		存储 1~8 000 字节的变长二进制数据
image		存储最大长度为 $2^{31}-1$ 的二进制数据(2 GB)

2. 自定义数据类型

SQL Server 2005 除了使用系统数据类型外,还允许用户根据需要自己定义数据类型,并且可以用自定义数据类型来声明变量或字段。但用户自定义数据类型是基于 SQL Server 2005 系统提供的数据类型,由系统已有的数据类型来派生,而不是定义一个具有新的存储及检索特性的类型。

当多个表中的列要存储相同类型的数据,且想确保这些列具有完全相同的数据类型、长度和是否为空属性时,可以使用用户自定义数据类型。例如,在一个数据库中,有许多表都

需要用到 `varchar(10)` 的数据类型,那么用户就可以自己定义一个数据类型,如用 `vc10` 代表 `varchar(10)` 的数据类型,在表需要用到 `varchar(10)` 的列时,都可以将其设为 `vc10`。

创建用户自定义数据类型时必须提供名称、新数据类型所依据的系统数据类型、是否允许为空值等。可以使用 SQL Server Management Studio 或 T-SQL 语句来创建,用户自定义数据类型一旦创建成功,就可以像使用系统数据类型一样使用。

下面利用 SQL Server Management Studio 为数据库 `book` 创建基于 `nvarchar` 型的用户自定义数据类型 `address_code`,长度为 20,允许为空。其操作步骤如下:

(1) 在 SQL Server Management Studio 中的“对象资源管理器”中展开“`book` 数据库”→“可编程性”→“类型”节点。

(2) 右击“类型”节点,从弹出的快捷菜单中选择“新建”→“用户定义数据类型”命令,如图 6-45 所示。

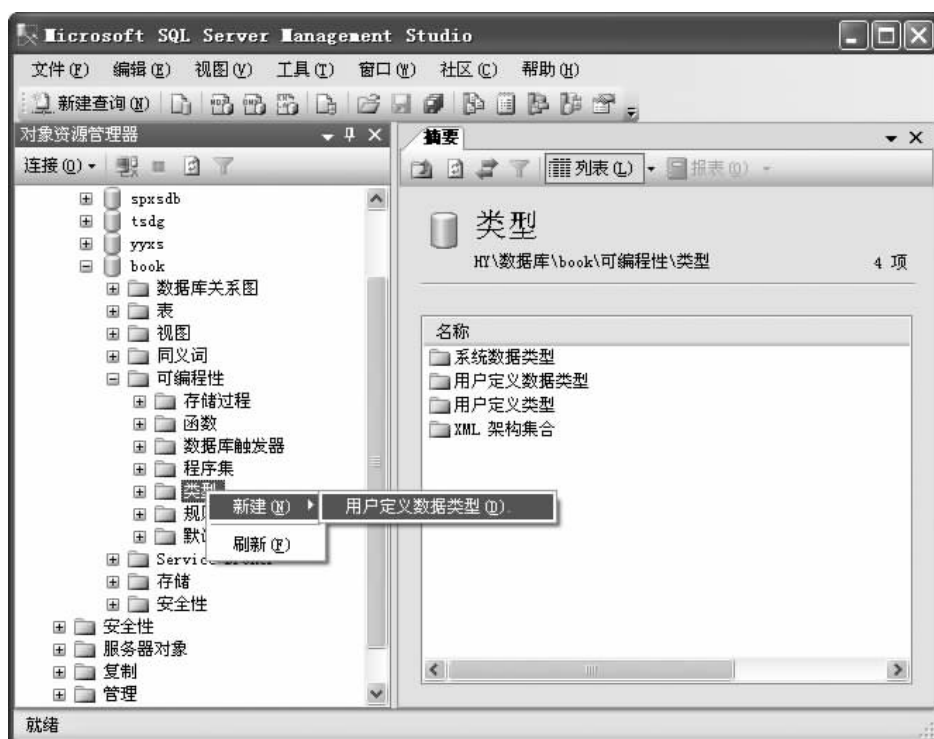


图 6-45 选择“用户定义数据类型”命令

(3) 在弹出的“新建用户定义数据类型”对话框中进行如下设置(如图 6-46 所示):

- 在“名称”文本框中输入类型名称 `address_code`。
- 在“数据类型”下拉列表框中选择 `nvarchar` 数据类型。
- 在“长度”数值框中输入 20。
- 选中“允许空值”复选框。

(4) 设置完毕后,单击“确定”按钮,即创建了自定义的数据类型 `address_code`。

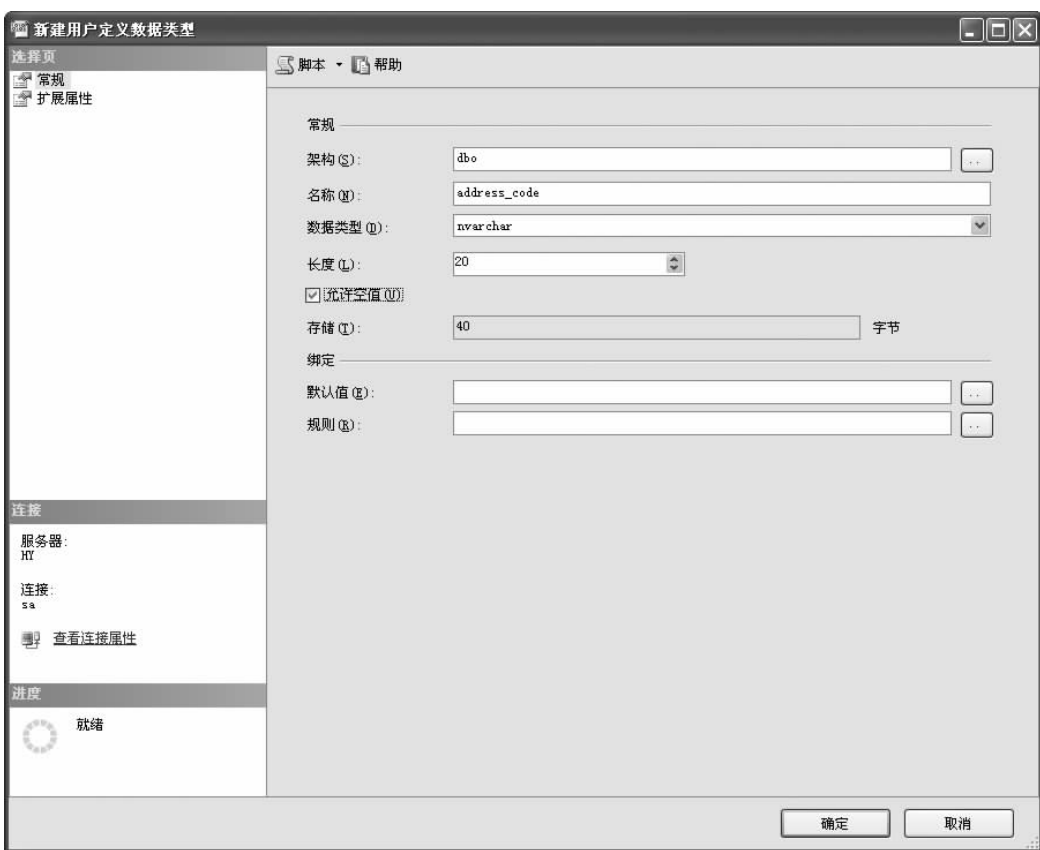


图 6-46 “新建用户定义数据类型”对话框

（二）表结构的创建、修改和删除

表是包含数据库中所有数据的数据库对象，与平常所说的表类似，但必须遵循关系的性质。表也是由行和列组成，用于组织和存储数据。表的创建首先要设计和定义表的结构，即录入数据的表的框架，由多少列组成，每列的数据类型，有无约束等。表结构创建好后，还可以根据实际需要进行表结构的修改或删除，如新增某些列或约束。

1. 创建表结构

表定义为列的集合，创建表结构也就是定义表中列的过程，如添加字段、设置字段的主键等属性。

使用 SQL Server Management Studio 可以非常方便地创建表结构，下面以客户信息表为例，介绍创建表结构的具体步骤。

（1）在 SQL Server Management Studio 的“对象资源管理器”中依次展开“数据库”→book→“表”节点。右击“表”节点，从弹出的快捷菜单中选择“新建表”命令，调出表设计器，如图 6-47 所示。通过表设计器可以进行表结构的创建、修改。

表设计器的上部分用来定义表的列（字段），包括列名、数据类型、允许空属性；下部分用来设置列的其他属性，如是否使用默认值、是否为“标识列”等。



图 6-47 表设计器

用户可用鼠标、Tab 键或方向键在单元格间移动和选择，完成“列名”、“数据类型”、“长度”、“允许空”栏中相关数据的输入。需要注意的是，有些数据类型的长度是固定的，不能修改。

(2) 各字段设置完成后，右击“客户编号”字段，在弹出的快捷菜单中选择“设置主键”命令或单击工具栏上的“设置主键/取消主键”按钮, 将“客户编号”设置为客户表的主键，设置好后，客户编号左侧显示一个小钥匙图标，如图 6-48 所示。



图 6-48 设置表的主键

(3) 执行“文件”→“保存”命令或者单击工具栏的“保存”按钮, 打开“选择名称”对话框，如图 6-49 所示。

(4) 在文本框中输入要保存的数据表名称“图书表”，单击“确定”按钮即可完成表的创建。关闭表设计器后，可以在 SQL Server Management Studio 看到创建的表。

使用同样的方法，可以创建 book 数据库中的其他表，如客户表、订购表，结果如图 6-50 所示。



图 6-49 “选择名称”对话框



图 6-50 book 数据库中的 3 张表

2. 修改表结构

在使用数据表的过程中可以根据需要随时对表的列、约束等属性进行修改。修改操作包括增加或删除列，修改列的名称、数据类型、长度，修改表的名称等。在 SQL Server Management Studio 中修改表结构的操作步骤如下：

(1) 在 SQL Server Management Studio 的“对象资源管理器”中依次展开“数据库”→book→“表”节点。右击需要修改的表，从弹出的快捷菜单中选择“修改”命令，打开表设计器。

(2) 如果要增加一列，首先选择新增加列的位置，然后右击，在弹出的快捷菜单中选择“插入列”命令，这时在选定列的上面会出现一个空行，在空行中输入新增加列的信息。

(3) 如果要删除一列，右击要删除的列，在弹出的快捷菜单中选择“删除列”命令。

(4) 如果要修改某一列的名称、数据类型、数据长度以及是否为空值，可以直接在表设计器中修改。

(5) 修改完后，单击工具栏上的“保存”按钮即可。

3. 删除表结构

对于数据库中不需要的数据表可以将其删除，以释放存储空间。删除表时，表的结构定义、数据、约束等都将永久地从数据库中删除。删除基本表同样可以在 SQL Server Management Studio 的“对象资源管理器”中删除，其操作方法是：右击需要删除的表，从弹出的快捷菜单选择“删除”命令，打开“删除对象”对话框，单击“确定”按钮即可。

(三) SQL Server 的表约束

为了减少输入错误、防止出现非法数据，在创建表结构时可以在列字段上设置约束，以

保证数据库中数据的一致性和完整性。SQL Server 2005 中的约束包括主键约束、唯一性约束、空值约束、默认值约束、外键约束、检查约束 6 种。

约束可以实施在列级和表级两个层次上：列级约束是指约束只对表中的一列起作用；表级约束是指对表中的多列起作用。

1. 主键约束

主键约束是能够唯一地标识表中的每一条记录的列或列的组合，即主键的列不能为空值，也不能为重复的值。例如，客户表的“客户编号”列上可以设置主键约束，以保证该列没有空值和重复值。主键约束实现了实体完整性规则，每一个表只能定义一个主键，并且系统自动为其创建索引。

2. 唯一性约束

唯一性约束用于保证列中不会出现重复的数据。在一个表中可以定义多个唯一性约束，定义了唯一性约束的列可以取空值。

唯一性约束与主键约束的区别有两点：一是一个表可以定义多个唯一性约束，但只能定义一个主键约束；二是定义了唯一性约束的列可以输入空值，而定义了主键约束的列不能取空值。

3. 空值约束

空值约束限制字段允许为空或不允许为空，其值为 null，表示该列允许为空，其值为 not null，表示不能为空，必须填入内容，否则不能被数据库接收。

4. 默认值约束

在 SQL Server 2005 中，可以给列设置默认值。如果某列已设置了默认值，当用户在表中插入记录时，如果没有给该列输入数据，系统会自动为该列输入默认值。每列只能有一个默认约束。

5. 外键约束

外键约束是指用于建立两个表之间连接的一列或多个列，如果某列是表的外键，则需指定该外键要关联到哪一个表的主键字段上。外键约束实现了参照完整性规则。

6. 检查约束

检查约束即 CHECK 约束，用于指定某列的取值范围。通过在列上设置逻辑表达式来限制列上可以输入的数据值，以此判断输入数据的合法性。检查约束实现域完整性规则。

可以在 SQL Server Management Studio 的“对象资源管理器”中设置表的约束。在表设计器的空白处右击，在弹出的快捷菜单中选择相应的选项进行设置，如图 6-51 所示。这里选择“CHECK 约束”命令，在弹出的“CHECK 约束”对话框中单击“添加”按钮，在弹出的对话框中单击“表达式”右边的按钮，在打开的“CHECK 约束表达式”对话框的表达式文本框中输入约束表达式，即可进行 CHECK 约束设置。例如，对性别进行 CHECK 约束，则约束表达式为“性别 in (‘男’,‘女’)”，表示性别只能取“男”和“女”两个值，也可以使用 T-SQL 语句进行约束设置。



图 6-51 选择“CHECK 约束”命令

(四) 管理表中的数据

创建表结构的目的是利用表来存储和管理数据,没有数据的表只是一个空的表结构,没有实际意义。因此完成数据表结构的创建后,就可以在表中添加数据,也可以根据用户的需要进行记录的更新和删除操作。

1. 插入数据

使用 SQL Server Management Studio 插入数据的具体步骤如下:

- (1) 在 SQL Server Management Studio 中的“对象资源管理器”中依次展开“数据库”→book→“表”节点,定位到“客户表”节点上。
- (2) 右击“客户表”节点,在弹出的快捷菜单中单击“打开表”命令,右窗格显示如图 6-52 所示的数据表编辑窗口,该窗口与电子表格类似。



图 6-52 数据表编辑窗口

(3) 输入各记录的字段值后,只要将光标定位到其他记录上,或关闭数据表编辑窗口,新记录就会自动保存。

在输入新记录时,需要注意以下几点:

- 输入字段的数据类型要和字段定义的数据类型一致,否则会出现警告提示框。
- 字段不能为空,必须输入内容。
- 有约束的字段,输入的内容必须满足这些约束;有默认的字段,可以不输入任何数据,因为在保存记录时,系统会自动填入默认值。

2. 修改数据

打开数据表后,找到要修改的记录,然后可以在记录上直接修改字段内容,修改完毕后,只需将光标从该记录上移开,定位到其他记录上,SQL Server 会自动保存修改的记录。

同样,需要注意修改后的数据记录的合法性,否则系统也会报错。

3. 删除数据

删除数据时,选中要删除的一行或多行连续的记录,然后右击,在弹出的快捷菜单中选择“删除”命令,弹出删除确认对话框,如图 6-53 所示,单击“是”按钮确认删除,单击“否”按钮则取消删除。

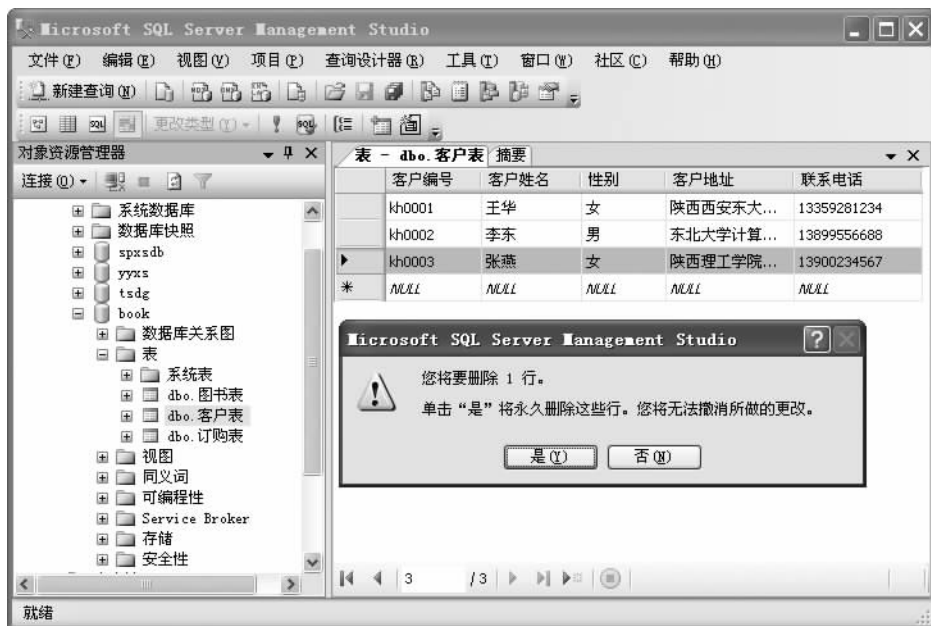


图 6-53 删除确认对话框

需要注意的是,记录删除后不能再恢复,所以删除前一定要确认。也可以按住 Shift 键或 Ctrl 键一次选中多条记录再删除。

二、视图的创建和维护

视图是从一张或多张表中导出的表,是用户查看数据库中数据的一种方式,其结构和数

据是建立在对表的查询基础上,由查询语句执行后所得到的查询结果所构成。与表一样,视图也是由字段和记录组成,只是这些字段和记录来源于其他被引用的表或视图,所以视图并不是真实存在的,而是一个虚拟的表。

(一) 视图的用途

1. 简化用户的操作、提高数据操作效率

视图大大简化了用户对数据库的操作,因为定义视图的本身就是一个复杂的查询结果集,这样在每一次执行相同的查询时,不必重写这些复杂的查询语句,只用一条简单的查询视图语句即可,隐藏了表与表之间的复杂连接操作;通过定义视图,用户只需要关注自己感兴趣的数据,从而提高数据的查询效率。

2. 使用户以不同的方式看待同一数据

当多个不同水平的用户共享数据库的数据时,视图能够让不同用户以不同的方式看到不同的或相同的数据集合。

3. 在一定程度上保证数据的逻辑独立性

对于视图的操作,例如,查询只依赖于视图的定义,当创建视图的基本表需要修改时,只需要修改视图定义中的子查询部分,而基于视图的查询不用改变,从而使外模式保持不变,原有的应用程序通过视图来重载数据,保证数据的逻辑独立性。

4. 可以对机密的数据提供安全保护

视图可以作为一种安全机制。在设计数据库应用系统时,针对不同用户定义不同的视图,使用户只能看到与自己有关的数据,其他数据库或表既不可见也不可访问,视图所引用表的访问权限与视图权限的设置互不影响。

(二) 视图的创建、删除和修改

1. 创建视图

用户创建的视图可以基于表,也可以基于其他视图。在 SQL Server 2005 中,只能在当前数据库中创建视图,但是创建的视图所引用的表可以在不同的数据库中。

通过 SQL Server Management Studio 创建视图的步骤如下:

(1) 在 SQL Server Management Studio 的“对象资源管理器”中展开目标数据库 book 节点。

(2) 右击 book 下的“视图”节点,在弹出的快捷菜单中单击“新建视图”命令,打开如图 6-54

所示的“添加表”对话框。在列表框中选择“客户表”,单击“添加”按钮,然后关闭该对话框。

(3) 进入视图设计器,在表选择区中通过字段左边的复选框选择需要的字段,如图 6-55 所示。



图 6-54 “添加表”对话框

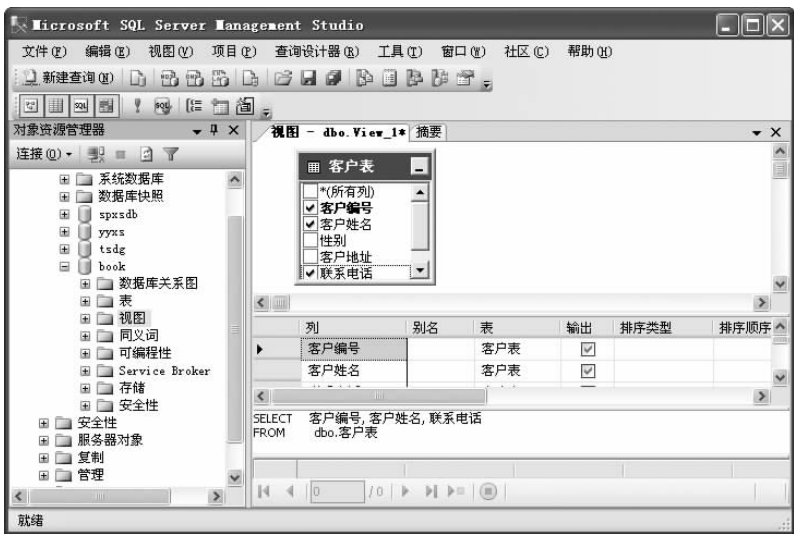


图 6-55 选择视图字段

(4) 单击工具栏中的“执行”按钮,或者右击视图设计器的空白区域,在弹出的快捷菜单中选择“执行 SQL”命令,运行视图,在窗口下显示查询结果,如图 6-56 所示。

(5) 单击工具栏中的“保存”按钮,打开“选择名称”对话框,输入视图名 v_kh,并单击“确定”按钮,完成视图的创建。随后,在 SQL Server Management Studio 中可以看到新创建的视图,如图 6-57 所示。



图 6-56 视图查询结果

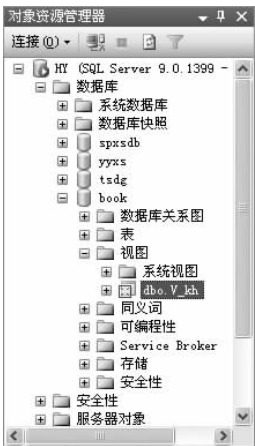


图 6-57 新创建的视图

2. 修改视图

创建好的视图也可以根据实际需要进行修改,以满足新的需求。如果基本表发生变化,或者要通过视图查询更多的信息,就可以根据需要来修改视图的定义。

修改视图的方法和创建视图类似,右击需要修改的视图,在弹出的快捷菜单中选择“修改”命令,打开视图设计器,即可以像创建视图一样进行修改。

3. 删除视图

创建好的视图,如果不再需要,或想清除视图定义及与之相关联的权限,则可以删除该视图。删除视图后,表和视图所基于的数据并不受影响,删除的是一个对象,但任何基于已删除视图的查询将会失败。

同创建视图一样,删除视图也可以在 SQL Server Management Studio 的“对象资源管理器”中右击需要删除的视图,在弹出的快捷菜单中选择“删除”命令,在弹出的“删除对象”对话框中单击“确定”按钮即可。还可以在该对话框中单击“显示依赖关系”按钮查看删除该视图对数据库的影响。

(三) 通过视图修改数据

可以利用视图对创建它的数据源进行一定的修改,如插入新的数据记录、删除记录和更新记录,但通过视图添加、更新和删除数据时,与表相比有一些限制,要求所建的视图必须满足一定的条件。有兴趣的读者可以阅读相关的书籍。

通过视图修改数据和创建视图类似,右击要修改的视图,在弹出的快捷菜单中选择“打开视图”命令,在打开的视图中直接添加、删除、修改数据。

三、索引的创建和维护

索引是以“表列”为基础的数据库对象,它保存着表中排序的索引列,并且记录了索引列在数据表中的物理存储位置,实现了表中数据的逻辑排序,其主要目的是提高 SQL Server 系统的性能、加快数据的查询速度和减少系统的响应时间。在实际的数据库应用中,在表上创建和维护索引是一项重要的工作。数据库中的索引与书籍中的目录类似,在一本书中,利用目录可以快速查找到所需要的信息,无须阅读整本书,在数据库中,索引使数据库程序无须对整个表进行扫描,就可以在其中找到所需要的数据。当创建数据库并优化其性能时,应该为数据查询所使用表的“列”创建索引。总之,使用索引有以下优点:

- 使用索引可以明显地加快数据查询的速度。
- 使用索引可以保证数据的唯一性。
- 使用索引可以加快数据库中基本表的连接速度。

下面简单介绍使用 SQL Server Management Studio 创建索引的步骤:

(1) 在 SQL Server Management Studio 的“对象资源管理器”中展开要创建索引的“表”节点。

(2) 右击“表”节点下面的“索引”节点,在弹出的快捷菜单中选择“新建索引”命令,打开“新建索引”窗口。

(3) 在“新建索引”窗口中,输入新建索引的名称、类型,添加索引字段以及字段的排列顺序,如图 6-58 所示,单击“确定”按钮,完成索引的创建。

索引的建立有利有弊,可以提高查询速度,但建立的索引过多会占用很多的磁盘空间,给维护带来麻烦。所以,建立索引时数据库管理员要权衡利弊。这部分内容可参照第五章第五节的内容。

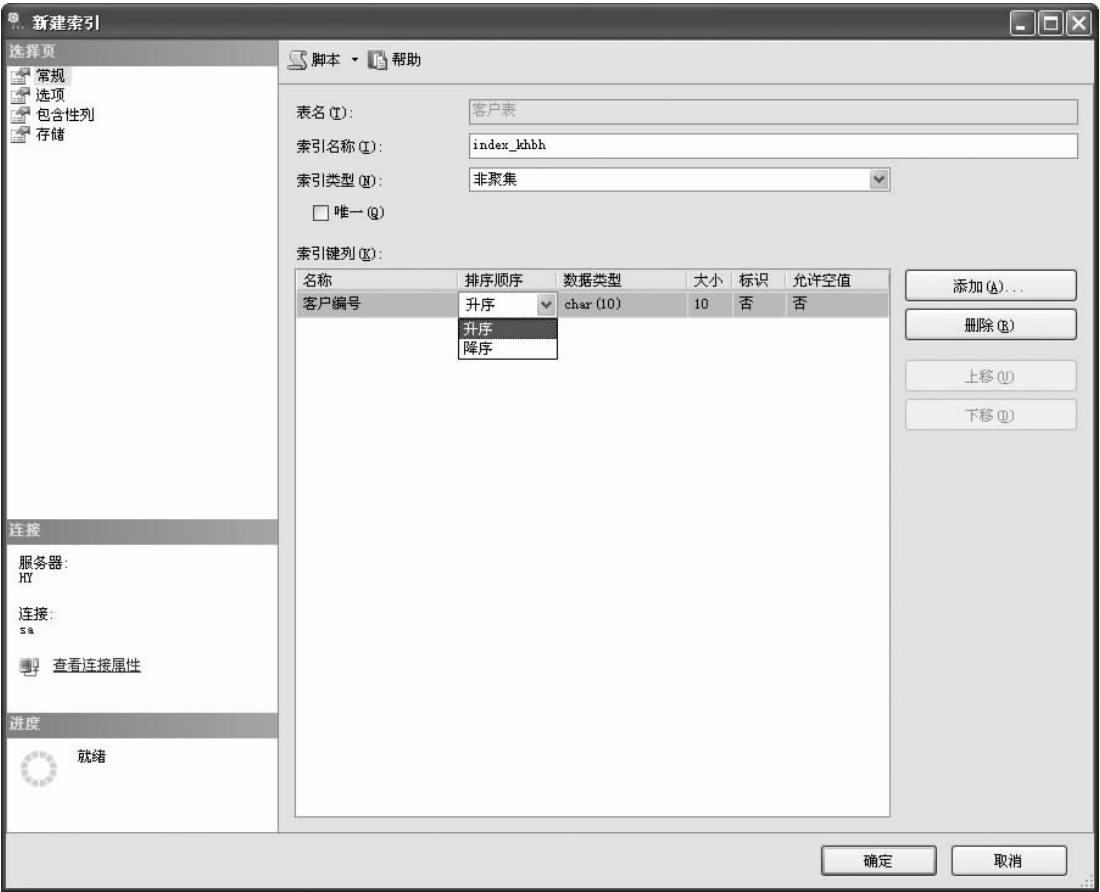


图 6-58 “新建索引”窗口

四、存储过程的创建

存储过程是数据库中运用十分广泛的一种数据对象,已成为数据库管理人员的重要工具。它是一组预先编译好的、为了实现某种特定功能的 T-SQL 语句集合,包括查询、插入、删除和更新等操作,经编译后以名称的形式存储在 SQL Server 服务器端的数据库中,由用户通过指定存储过程的名称来执行。对于需要经常执行的操作,可以将所需的 SQL 语句都编写在一个存储过程中,然后由前台应用程序执行即可。

存储过程与存储在客户机的本地 T-SQL 语句相比,具有以下优点。

1. 模块化的程序设计

一个存储过程就是一个模块,可用来封装并实现特定的功能。一旦创建,就可在程序中调用多次,还可改进应用程序的可维护性,并允许应用程序统一访问数据库。

2. 改善系统性能,提高执行效率

系统在创建存储过程时会对其进行分析和优化,并在首次执行该存储过程后将其驻留于高速缓冲存储器中,以加速存储过程的后续执行。

3. 提供了一种安全机制

通过对执行某一存储过程的权限设置,实现对相应表、视图等对象的数据访问权限限制,避免非授权用户访问在定义存储过程中被引用对象的数据。

4. 减少了网络流量

创建存储过程后,网络中传送的只是调用存储过程的语句,而不是从客户端发送的数百行 SQL 语句,只需一条执行语句就可实现相同的功能,减少了网络的流量。

(一) 创建自定义存储过程

在 SQL Server 2005 系统中,存储过程主要分为系统存储过程和用户定义的存储过程两种。系统存储过程由 SQL Server 2005 提供,用户可以直接使用,用户定义存储过程用于实现用户需要实现的操作。下面主要介绍用户定义的存储过程。

使用 SQL Server Management Studio 创建存储过程的步骤如下:

(1) 在 SQL Server Management Studio 的“对象资源管理器”中依次展开要创建存储过程的 book 数据库中的“可编程性”→“存储过程”节点。右击“存储过程”节点,从弹出的快捷菜单中选择“新建存储过程”命令。

(2) 打开创建存储过程的查询编辑器,如图 6-59 所示。在该窗格中,系统自动生成带有提示的 CREATE PROCEDURE 语句的存储过程模板,用户可以根据模板提示补充输入存储过程包含的相应语句即可,也可以直接输入 T-SQL 语句。



图 6-59 创建存储过程的查询编辑器

(3) 在此创建名为 pro_khdh 的无参数的存储过程(相关 SQL 语句在第七章介绍)。

单击查询编辑器工具栏上的“分析”按钮, 检查输入的 T-SQL 语句是否有语法错误, 确认无错误后, 单击“执行”按钮, 完成存储过程的创建。

(二) 执行存储过程

在 SQL Server 中, 使用 EXECUTE 语句来执行存储过程。例如, 执行 pro_khdh 存储过程的语句为 exec pro_khdh, 执行的结果如图 6-60 所示。



图 6-60 存储过程执行结果

五、触发器的创建

触发器(trigger)和存储过程一样, 都是由 SQL 语句和流程控制语句组成的。就本质而言, 触发器也是一种特殊的存储过程, 一般的存储过程通过过程名直接调用, 而触发器的特殊性表现在: 它是在执行某些特定的 T-SQL 语句时通过事件进行触发而被执行。它与表紧密相连, 当用户对表中的数据进行插入、删除和修改时, 触发器就会自动激活执行所定义的 SQL 语句, 以保证数据的完整性和一致性。

在 SQL Server 2005 中, 可以用约束和触发器两种方法来强制执行业务规则和保证数据的完整性。但触发器可以实现比约束更为复杂的数据完整性约束, 实现由主键和外键所不能保证的复杂的参照完整性和数据的一致性。

触发器具有以下优点。

1. 可以强制保证数据的完整性

在数据库中要实现数据的完整性约束, 可以使用 CHECK 约束或触发器来实现。触发器包含了使用 T-SQL 语句的复杂处理逻辑, 不仅支持约束的所有功能, 还可以实现更为复杂的数据完整性约束, 例如, 在 CHECK 约束中不允许引用其他表中的列来完成检查工作, 而触发器可以引用其他表中的列来完成数据的完整性约束。

2. 自动执行

触发器不需要用户调用,是通过事件触发来自动执行的,只要对表中的数据进行了修改,触发器就会立即被激活。

3. 实现数据库中多张表的级联更改

触发器虽然是基于一个表建立的,但可以针对多个表进行操作,用户可以通过触发器对数据库中的相关表进行级联修改。而且可以评估数据修改前后表的状态,并根据其差异采取对策。

4. 维护非规范化数据

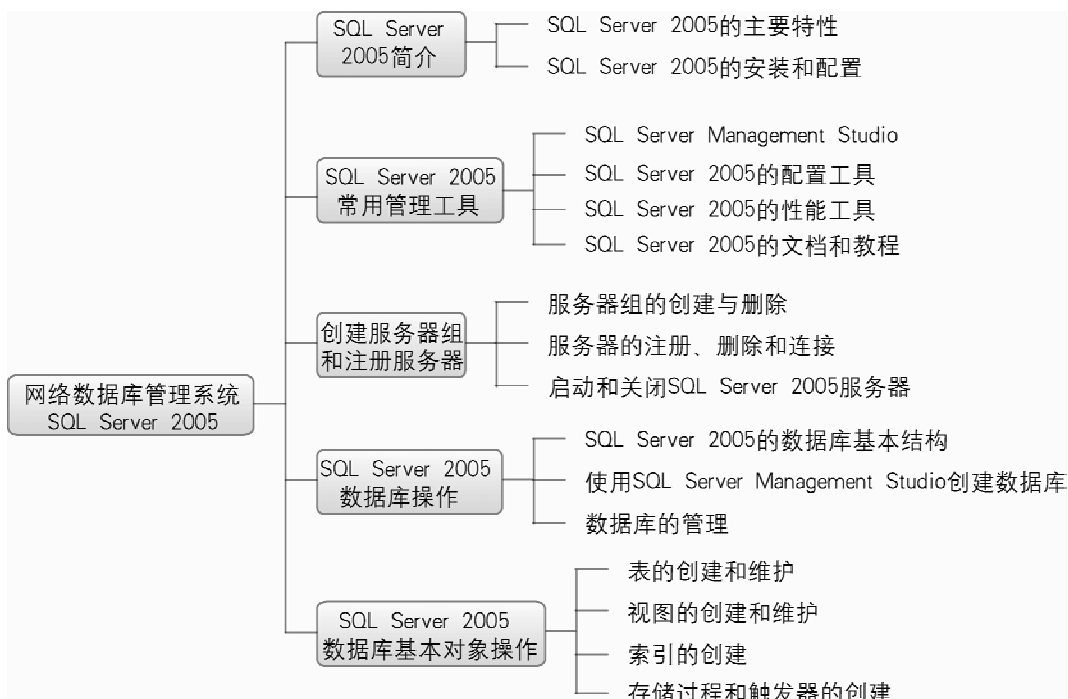
用户可以使用触发器来保证非规范数据库中低级数据的完整性。维护非规范化数据与表的级联是不同的。表的级联是指不同表之间的为主外键关系,维护表的级联可以通过设置表的主键与外键的关系来实现。而非规范化数据通常是指在表中派生的、冗余的数据值,维护非规范化数据应该通过触发器来实现。

在 SQL Server Management Studio 中,触发器的创建和存储过程的创建类似,在“对象资源管理器”中依次展开“数据库”→book→“表”→要创建触发器的表(这里是 dbo. 订购表)→“触发器”节点,右击“触发器”节点,在弹出的快捷菜单中选择“新建触发器”命令,如图 6-61 所示,打开创建触发器的查询编辑器。在查询编辑器中显示触发器模板,用户根据模板输入触发器的 T-SQL 语句,当语句执行成功后,触发器便创建成功。



图 6-61 创建触发器的查询编辑器

本章小结



综合训练

一、简答题

1. 简述组成 SQL Server 2005 数据库的 3 种类型的文件。
2. SQL Server 2005 有哪些系统数据库？它们的作用是什么？

二、上机题

1. 创建数据库

使用 SQL Server Management Studio 创建一个名为 SPXS 的数据库(商品销售管理系统),要求包含 3 个数据文件。其中主数据文件为 20 MB,最大容量为 50 MB,每次增长 2 MB;次要数据文件为 10 MB,最大容量不受限制,每次增长 20%;事务日志文件为 20 MB,最大容量为 50 MB,每次增长 20 MB。

2. 创建表

在 Server Management Studio 的“对象资源管理器”中为 SPXS 数据库创建 3 张表,分别为客户表、商品表、商品销售表,并建立 3 张表的关系。3 张表中各列的要求分别如表 6-2~表 6-4 所示。

表 6-2 客户表的各列要求

字段名称	字段类型	大 小	说 明
客户编号	char	10	主键
客户名	char	8	
性别	char	2	
地址	char	30	
联系电话	char	12	

表 6-3 商品表的各列要求

字段名称	字段类型	大 小	说 明
商品编号	char	10	主键
商品名称	char	8	
类别	char	10	
单位	char	4	
单价	money		

表 6-4 商品销售表的各列要求

字段名称	字段类型	大 小	取值范围	说 明
商品编号	char	10	数据来自商品表	主键
客户编号	char	10	数据来自客户表	主键
销售日期	datetime			
销售数量	smallint			

3. 创建视图

为 SPXS 数据库中的各表录入相关数据后,创建一个名为 XSL 的视图,用来查看每一个商品的销售信息,只显示商品编号和销售数量。