

面向“十二五”教学课程改革立项教材
计算机类校企合作、任务驱动项目教材

SQL Server

数据库案例与实训教程

(项目化教学)

SQL Server

SHUJUKU ANLI

YU SHIXUN JIAOCHENG

◎主编 王丽晖



中国商业出版社

“二五”教学课程改革立项教材
校企合作、任务驱动项目教材

SQL Server

数据库案例与实训教程

(项目化教学)

SQL Server

SHUJUKU ANLI

YU SHIXUN JIAOCHENG

主 编 王丽晖

参 编 温 铂 阿 伦 勾智楠

刘文红 唐玉琦 刘志中

王 江 刘 扬

中国商业出版社

图书在版编目(CIP)数据

SQL Server 数据库案例与实训教程/王丽晖主编.
—北京:中国商业出版社, 2013. 2
ISBN 978 - 7 - 5044 - 7974 - 7

I. ①S… II. ①王… III. ①关系数据库系统 - 高等学校 - 教材 IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2013)第 019908 号

责任编辑:许延平

中国商业出版社出版发行
010 - 63180647 www. c - chook. com
(100053 北京广安门内报国寺 1 号)
新华书店总店北京发行所经销
北京市书林印刷有限公司印刷

* * * * *

787 × 1092 毫米 开本:1/16 印张:14.25 字数:300 千字
2013 年 2 月第 1 版 2013 年 2 月第 1 次印刷

定价:32.80 元

* * * * *

(如有印装质量问题可更换)

前 言

数据库应用技术是计算机相关专业的专业课程,也是计算机信息管理类专业的核心课程。SQL Server 是 Microsoft 公司开发的一种关系型数据库管理系统,是目前世界上最为流行的数据库管理系统之一,为 IT 从业者和信息工作者带来了强大的工具,能够为各种规模企业提供一套完整、高效、安全的数据库管理和数据分析解决方案。本书以一个完整项目实例“图书管理系统”为应用背景,系统介绍了 SQL Server 2005 数据库的设计、实施、管理、维护和开发过程。

根据学生的认知规律,本着“教学做一体”的理念,本书以应用为主线,理论够用为度,精心组织教材内容。全书采用项目驱动形式,一个实训应用项目贯穿全书,以学生实际动手操作为主体过程,所有知识都在实训项目中得以具体化和掌握,重点训练学生完成信息管理系统的设计、实施与维护等工作能力。整个项目又划分为若干任务,每章即是一个任务,下分多个子任务,每个任务都以实例驱动,有明确的实训目标,操作步骤清晰、详尽,易于掌握。每个任务(每章)后还安排有相应阅读材料,以使学生的知识体系更加完整和充实。

通过一个项目导向,必要理论支撑,多个任务和子任务驱动,实例讲解,实训强化,应用知识逐步深入展开的进阶式学习,教师可以指导学生完成一系列的实际工作任务,最终完成整个项目任务,达到面向就业,重点培养学生解决实际问题的训练目标,实现了能力训练项目化、课程结构模块化、理论与实践教学一体化。

本书主要包括:设计数据库、创建 SQL Server 2005 数据库和表、查询数据、管理和维护数据、使用索引和视图、数据库安全和事务管理、数据库备份与转换、开发存储过程、实现触发器以及利用数据库应用程序接口连接并访问数据库的初步知识。

本书可作为高职高专软件技术、网络技术、信息管理和电子商务等计算机相关专业的教材,也可作为相关培训学校的教材及 SQL Server 2005 数据库自学者的参考书。全书总课时数建议在 72 学时以上,实训课时建议在 36 学时以上,教师可以根据专业、学生、实际教学情况灵活安排。

本书由唐山职业技术学院的王丽晖老师主编,内蒙古职业技术学院温铂老师、阿伦老师,石家庄理工职业学院勾智楠老师、刘文红老师,唐山职业技术学院唐玉琦老师、刘志中老师、王江老师,石家庄工程职业学院刘扬老师参加了本书的编写。唐山企奥科技有限公司的赵二仲工程师对本书提出了宝贵的编写意见。

在课程内容、课程结构以及课程实践环节上，我们依据校企合作、工学结合的课程建设理念，做了一些改革尝试和探索，希望与各位读者朋友共同探讨，不当之处，恳请大家提出宝贵意见和建议。

编者

2013 年 2 月

目 录

任务 1 设计数据库	(1)
1.1 子任务:认识数据库	(1)
1.2 子任务:描述数据库	(3)
1.3 子任务:设计关系数据库结构	(8)
1.4 子任务:认识 SQL Server2005 数据库管理系统	(12)
任务 2 创建数据库	(26)
2.1 子任务:认识数据库存储结构	(26)
2.2 子任务:创建数据库结构	(28)
2.3 子任务:查看和修改数据库	(31)
2.4 删除数据库	(33)
任务 3 创建表	(41)
3.1 子任务:认识数据类型	(41)
3.2 子任务:创建表	(45)
3.3 子任务:实现数据完整性	(48)
3.4 子任务:查看和修改表	(56)
任务 4 查询数据	(67)
4.1 子任务:单表查询	(67)
4.2 子任务:多表连接和联合查询	(74)
4.3 子任务:使用子查询	(76)
任务 5 管理和维护数据	(82)
5.1 子任务:在表编辑器中添加、修改和删除数据	(82)
5.2 子任务:使用 T - SQL 命令插入数据	(85)
5.3 子任务:使用 T - SQL 命令更新数据	(86)
5.4 子任务:使用 T - SQL 命令删除数据	(86)
5.5 子任务:使用事务维护数据	(87)

任务 6 使用索引和视图	(102)
6.1 子任务:使用索引查询数据	(102)
6.2 子任务:使用视图查询数据	(108)
任务 7 数据库安全管理	(123)
7.1 子任务:登录管理	(123)
7.2 子任务:用户管理	(128)
7.3 子任务:角色管理	(131)
任务 8 数据备份与转换	(144)
8.1 任务:备份和还原数据库	(144)
8.2 子任务:分离和附加数据库	(152)
8.3 子任务:导入和导出数据	(157)
任务 9 开发存储过程	(167)
9.1 子任务:认识存储过程	(167)
9.2 子任务:创建和执行存储过程	(168)
9.3 子任务:管理存储过程	(175)
任务 10 实现触发器	(184)
10.1 子任务:认识触发器	(184)
10.2 子任务:创建触发器	(185)
10.3 子任务:管理触发器	(192)
任务 11 访问数据库	(200)
11.1 子任务:认识数据库应用程序接口	(200)
11.2 子任务:使用数据访问控件访问 SQL Server 数据库	(201)
11.3 子任务:使用 ADO 对象访问 SQL Server 数据库	(203)
11.4 子任务:使用 ODBC 连接访问 SQL Server 数据库	(204)

任务1 设计数据库

日常生活和工作中,人们对数据处理的需求无处不在,需要计算机处理的数据量越来越大,数据类型越来越多样化,处理数据的方法越来越复杂化,数据库技术应用越来越广泛。一个好的数据库,需要经过严格的设计,才能满足人们对数据处理的各种需要。

1.1 子任务:认识数据库

1.1.1 为什么学习数据库技术

现实生活中存在着海量信息,大多数信息是无序、杂乱无章的。比如:与我们密切相关的学籍管理信息、图书信息、教师信息、银行的用户信息等。如果不进行数据管理,信息的检索、更新、维护等工作量是相当大的,工作效率也会很低。

数据库技术主要用于科学的组织和管理数据,便于我们收集、存储、检索、维护和加工数据,使我们更加高效地使用数据,最大限度地提高工作效率。

数据库技术管理数据的功能主要体现在:

- (1)可以以一定组织结构存储大量数据信息,方便用户访问和检索;
- (2)可以保证数据信息的一致性、安全性和完整性,降低数据冗余;
- (3)可以满足多用户和多个应用程序的同时访问,具有共享性;
- (4)可以从中挖掘出规律性的信息,对相关工作决策具有指导作用。

1.1.2 数据库的几个概念

1. 什么是数据库

简单的说,数据库 DataBase, DB(**DataBase, DB**)是一个存储在存储介质上的数据集合。在这个集合中,相关联的数据以一定的组织方式存储在一起,统一管理,能够为多个用户共享,并且独立于各个应用程序。

在日常管理工作中,常常需要把某些相关的数据放进这样的“仓库”,并根据管理的需要进行相应的处理。例如,学校图书馆常常要把图书的基本情况(图书编号、ISBN、图书名称、图书类型、出版社、出版日期、作者、单价等)存放在表中,这张表就可以看成是一个数据库。有了这个“数据仓库”,我们就可以根据需要随时查询某本图书的基本情况,也可以查询某本图书的借阅情况等。这些工作如果都能在计算机上自动进行,那我们的管理工作就可以达到极高的水平。此外,在人事、财务、仓库、生产管理中也需要建立众多这样的“数据库”,使其可以利用计算机实现人事、财务、仓库、生产的自动化管理。

2. 数据库管理系统

为了更加科学地组织和管理数据库中的数据,我们需要一类系统管理软件——数据库管理系统(**DataBase Management System, DBMS**),数据库管理系统负责建立、使用和维护数据

库,对数据库进行统一的管理、控制,以保证数据库的安全性和完整性。可以同时允许多个应用程序和用户用不同的方法在同时或不同时刻访问数据,修改、维护和查询数据库。

3. 数据库应用系统

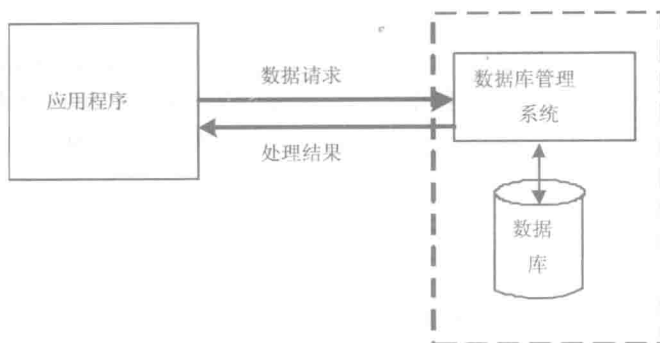


图 1-1 数据库应用系统各组成部分之间的关系

如图 1-1 所示,数据库应用系统(**DataBase Application System, DBAS**)是在数据库管理系统(DBMS)支持下建立的计算机应用(软件)系统,主要由数据库、数据库管理系统和应用程序三部分组成。

例如,以数据库为基础的学生管理系统、人事管理系统、图书管理系统等,都是以数据库为基础和核心的计算机应用系统。

4. 数据库系统

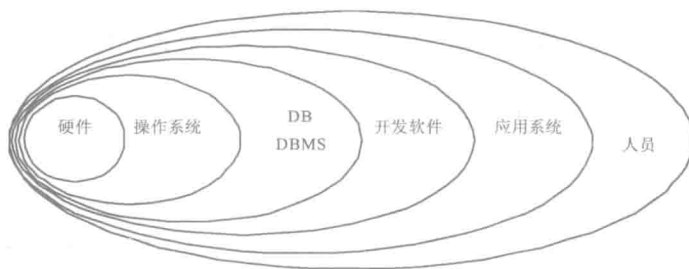


图 1-2 数据库系统(DBS)中各组成部分层次结构关系示意图

图 1-2 反映了数据库系统(**DataBase System, DBS**)的层次结构关系, DBS 是由计算机硬件、操作系统(如 Windows)、数据库、数据库管理系统(如 SQL Server)、开发软件(如 Vb. net)、数据库应用系统及用户(普通用户或者数据库管理员)构成的一个完整系统。一般,在不引起混淆的情况下,常常把数据库系统简称为数据库。

1.2 子任务:描述数据库

作为存储在存储介质(如磁盘等)上的相关联数据集合,为便于统一管理和用户共享,数据库中数据的组织方式需要统一的描述方法,我们称为**模型**。

1.2.1 实体-联系模型

1. 实体

在数据库中,我们把现实生活中客观存在的、可以被描述的事物称为**实体**。例如,一台计算机、一本图书、一个学生,学生的一次选课,读者的一次借阅等,都可以称为实体。

为了清晰地描述一个实体,需要描述它所具有的主要特征。例如“学生”这个实体,可以用学号、姓名、性别、年龄、系别、联系电话等特征来描述;“课程”这个实体,可以用课程编号、课程名、课程类型编号、学时、学分等特征来描述;而“选课”这个实体,可以用学号、课程编号、成绩、学期等特征描述。虽然描述具有统一格式,但是,具体到每一个学生,每一门课程,每一次选课,所描述的特征数据是不一样的。也就是说,不同的特征数据对应不同的实体个体。如表1-1所示,表中每行对应一个学生实体。

表1-1 学生基本情况表

学号	姓名	性别	年龄	系别	联系电话
20110803001	李丽	女	19	信息工程系	13150652313
20110803002	余姚远	男	18	信息工程系	13986453208
20110803003	张雪峰	男	19	信息工程系	13877658799
20110803004	杜子娟	女	19	信息工程系	13006677882

2. 属性

实体所具有的特征称为属性。例如表1-1“学生基本情况表”中,学生实体就具有学号、姓名、性别、年龄、系别、联系电话等属性。

3. 域

实体属性的取值范围称为该属性的域。例如,学生实体属性“性别”的域为(男,女)。

4. 实体联系

两个不同类型实体集之间常有一定的联系,如图1-3所示,常见实体之间的联系分为三种类型:

- (1) 一对一联系(1:1);
- (2) 一对多联系(1:n);
- (3) 多对多联系(m:n)。

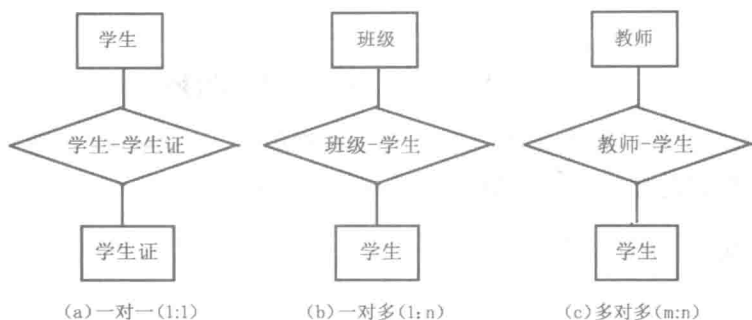


图 1-3 常见的实体联系

5. 实体-联系模型

利用实体内部的联系和实体间的联系来描述客观事物及其联系,称**实体-联系模型**,简称**E-R模型**,也称为E-R图。实体-联系模型是一种**概念模型**。

与机械行业需要机械制图,建筑行业需要施工图纸一样,数据库设计描述也需要图形化的表达式,即E-R图。例如,图1-2中三种不同类型的实体联系对应的E-R模型图如图1-4、1-5、1-6所示。

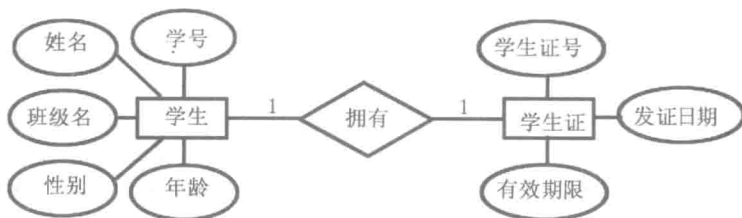


图 1-4 一对一(1:1)实体联系模型(E-R图)

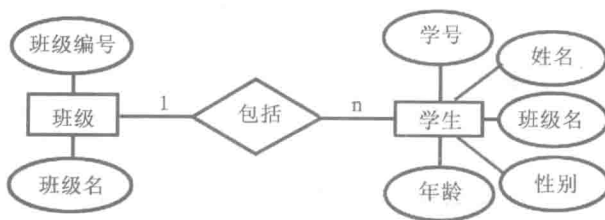


图 1-5 一对多(1:n)实体联系模型(E-R图)

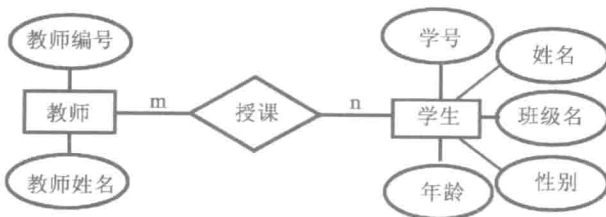


图 1-6 多对多(m:n)实体联系模型(E-R图)

可见, E-R 模型是由实体集、属性和联系集构成, 其表示方法如下:

(1) 实体集用矩形框表示, 矩形框内写上实体名。

(2) 实体的属性用椭圆框表示, 框内写上属性名, 并用无向边与其实体集相连。

(3) 实体间的联系用菱形框表示, 联系以适当的含义命名, 名字写在菱形框中, 用无向连线将参加联系的实体矩形框分别与菱形框相连, 并在连线上标明联系的类型, 即 1:1、1:n 或 m:n。

【例 1-1】画出学生和课程实体通过选课关系发生联系的 E-R 模型图。

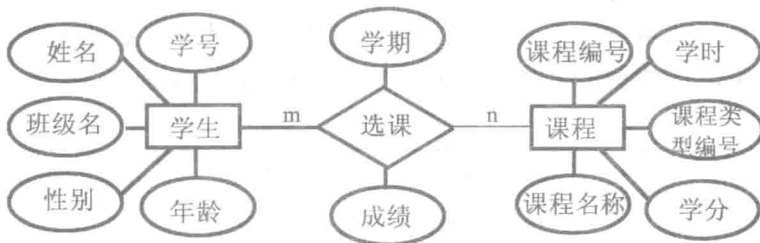


图 1-7 学生选课实体联系模型(E-R 图)

【注意】实体间一旦发生联系, 就有可能产生一些属性, 例如, 一旦学生开始选课, 就会产生选课成绩、选课学期等属性。

【例 1-2】画出“学生”、“课程”、“课程类型”和“选课”实体集的 E-R 图。

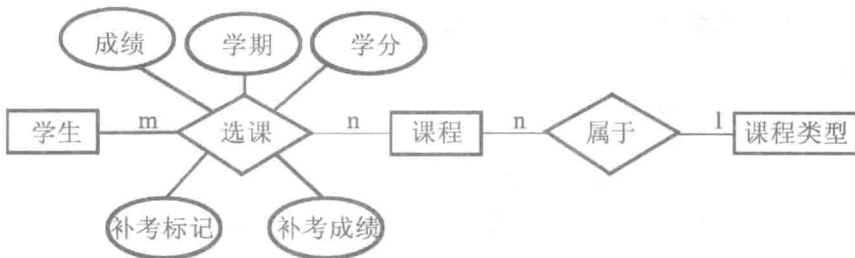


图 1-8 实体集联系模型(E-R 图)

1.2.2 数据模型

1. 数据模型

数据库中, 按照数据之间的逻辑关系来组织数据的结构称为**数据模型**。数据模型是一种数据存储的逻辑结构, 也称**逻辑模型**。

2. 数据模型分类

在数据库管理系统 DBMS 中, 数据模型可以分为四种: 层次模型、网状模型、关系模型和面向对象模型。四种数据模型中, 应用最广泛的就是关系模型。目前常用的数据库管理系统如 SQL Server、Oracle、DB2 等都是关系型 DBMS。

1.2.3 关系模型

1. 关系模型

如图 1-1 所示,用二维表形式来表示实体集和实体间联系的数据模型就是**关系模型**。关系模型中,实体和实体间的联系(可以是一对一,一对多或者多对多的联系)都用二维表表示,一张二维表称为一个关系,每个关系都有一个关系名。采用关系模型表示的数据库就是**关系型数据库**。一个关系型数据库可以包含多个二维表,简单地说,关系数据库就是表的集合。

例如,表 1-1 所示的关系名为“学生基本情况表”,描述了学生实体集合,表 1-2 所示的关系名为“选课表”,描述了学生实体和课程实体之间的联系,即选课关系。

表 1-2

选课表

学号	课程编号	成绩	学期
20110803001	080411051	89	2010-2011(1)
20110803002	080411051	94	2010-2011(1)
20110803003	080411051	73	2010-2011(1)
20110803004	080411051	90	2010-2011(1)

2. 关系的相关术语

(1) 关系模式

习惯上我们用**关系模式**的形式来描述和表示一个关系。

例如,对于表 1-1 所示名为“学生基本情况表”的关系,表 1-2 所示名为“选课表”的关系,用关系模式可以描述为:

学生(学号,姓名,性别,年龄,系别,联系电话)

选课(学号,课程编号,学期,成绩,学分,补考标记,补考成绩)

(2) 记录和字段

一个二维表(关系)中,每行对应一个实体,数据库中把二维表的每一行称为一条**记录**(元组),所有记录就构成了数据库中的一个表(Table)。每一列对应实体的一个属性特征,称为**字段**(属性,或者数据项),一条记录中每个字段的值称为**字段值**。

(3) 主关键字和候选关键字

主关键字是表中能够惟一区分一条记录的字段(或属性),简称**主键**。一个表也可以没有主键,但是一般情况下,一个表应当设置一个主键。

一个表中如果有多个字段都可以惟一标识一条记录,则把这多个字段都称为**候选关键字**,简称**候选键**。

我们可以根据实体描述的主要信息内容,选取其中一个候选键作为主键。例如,学生基本情况表中,若有“身份证号”这个字段,则“学号”和“身份证号”都可以惟一标识一条学生记录,都是候选键。可是学生基本情况表反映的主要是学生信息,而不是公民信息,所以可以选取“学号”作为主键。

【注意】主键不一定是—个字段，可能多个字段的组合才能惟一标识—条记录，这样的主键称为**组合键**。例如，表1-2所示的“选课表”，学号和课程编号组合才能惟一标识出某个学生某门课程的成绩。

(4) 外关键字

在表1-1“学生基本情况表”中，“学号”是主键，在表1-2“选课表”中，“学号”不是主键，主键是“学号”和“课程编号”的组合。“选课表”中“学号”的取值就是“学生基本情况表”中“学号”的取值。我们就称“学号”为“选课表”的**外关键字**，简称**外键**。

也就是说，假设存在两个表A和B，如果表A的主键在表B中存在，但不是表B的主键，仅仅作为表B的一个字段，则称此字段为表B的外键。

(5) 数据完整性

所谓**数据完整性**，是指“存储在数据库中的所有数据字段值都是正确的”。如果数据库中存储有不正确的数据字段值，我们就称该数据库**不具有数据完整性**。

在设计数据库表的过程中，通过设计一些约束条件，可以实现数据完整性。

【例1-3】数据库不具有数据完整性的几个实例。

实例1：“学生基本情况表”中，如果对所输入的记录信息不加任何限制，就有可能出现两条，甚至多条“学号”（主关键字）相同的记录，这是错误的，不允许的。

实例2：“学生基本情况表”中，如果不对表中学生记录的“性别”字段值加以限制，则学生的性别可能会出现不只是“男”或“女”这两种取值情况，对应数据库**不具有数据完整性**。

实例3：“学生基本情况表”中，如果不对学生记录的“年龄”字段值加以限制，则有可能出现负数的年龄。对应数据库**不具有数据完整性**。

(6) 数据完整性约束条件

为了实现数据完整性，设计数据库时，需要对表中的行和列的数据进行检查，看它是否符合实际要求。为此，SQL Server2005提供了下列四种约束。

①**实体完整性约束**：**实体完整性**的约束对象是表中的记录（行），目的是使表中不存在两条完全相同的记录。

例如，“学生基本情况表”中，作为主关键字，每个学生的学号是惟一的，并且不能为空。学号为“0803001”的记录只能是惟一的。

可见，实体完整性约束是通过定义主关键字段的惟一性和不为空来实现的。

②**域完整性约束**：**域完整性**的约束对象是表中的字段（列），目的是使字段的取值在有效范围内。

例如，“选课表”中，成绩字段的取值范围应该大于等于0，小于等于100，这个取值范围以外的字段值，都不符合域完整性约束。

③**参照完整性约束**：**参照完整性**也称**引用（外键）完整性**，它的约束对象是表与表之间的关系，目的是确保主关键字和外关键字的值在所有表中是一致的。如果一个关键字值发生变化，则在整个数据库中，对该关键字的所有引用都要进行一致的修改。

例如，在“学生基本情况表”中，“学号”字段是主键，在“选课表”中，“学号”字段是外键。输入“选课表”中的“学号”字段时，必须检查验证“学生基本情况表”中是否存在此学号对应的学生记录；在“学生基本情况表”中删除—条学生记录时，还要删除“选课表”中相同学号的所有选课记录。

④用户自定义完整性约束:SQL Server 允许数据库用户自己根据应用的需要,编写规则、默认值约束条件、存储过程、触发器等,目的也是保证数据的完整性,这些就是所谓**用户自定义完整性**。规则、默认值、存储过程、触发器相关知识将在后续章节详细介绍。

1.3 子任务:设计关系数据库结构

实体-联系模型是反映用户需求的一种概念模型,与数据库的具体实现无关。而数据库的设计需要描述数据的逻辑结构,即数据库赖以实现的数据模型。我们现在常用的数据模型都是关系模型,因此,需要把实体-联系模型转换为关系模型。

把实体-联系模型转换为关系模型,实际上就是要将 E-R 图中的实体、实体的属性和实体之间的联系等转换为若干个关系,并确定这些关系的属性、关键字和约束条件。

1.3.1 实体-联系模型转换为关系模型

1. 实体的转换

E-R 图中的每个实体转换为关系模型中的一个关系模式。关系模式中的属性就是实体的属性,主关键字就是实体的标识符,用下划线表示。

通常,E-R 图中的实体和属性都使用汉字命名方式,但是,在转换得到的关系模式中,关系名和属性名一般采用英文或者拼音字母方式命名,以方便应用程序的编写。

【例 1-4】对于表 1-1 所示名为“学生基本情况表”的关系,关系模式可以表示为:

学生(学号,姓名,性别,年龄,系别,联系电话)

2. 联系的转换

(1) 1:1 联系的转换方法

【例 1-5】将图 1-4 中学生与学生证的 E-R 图转换为关系。

学生(学号,姓名,性别,年龄,系别,联系电话,学生证号)
学生证(学生证号,发证日期,有效期限)

方案一

1:1 联系的转换方法为:

第一步,把其中一个实体转换为关系,用一个关系模式表示。

第二步,把另一个实体也转换为关系,用一个关系模式表示。

第三步,把两个实体之间联系的属性合并到其中一个关系中。如果联系本身没有属性,只需要在一个实体关系中增加另一个实体关系的主键即可。

图 1-4 中,联系“拥有”本身没有属性,因此,例 1-3 实际有两种方案。第二种方案如下:

学生(学号,姓名,性别,年龄,系别,联系电话)
学生证(学生证号,发证日期,有效期限,学号)

方案二

(2) 1:n 联系的转换方法

【例 1-6】将图 1-5 中学生与班级的 E-R 图转换为关系。

学生(学号,姓名,性别,年龄,系别,联系电话,班级编号)
 班级(班级编号,班级名)

图 1-5 所示 E-R 图中,实体间是 1:n 联系,其中 1 端对应的实体是“班级”,n 端对应的实体是“学生”。联系“包括”本身没有属性。

1:n 联系的转换方法为:

第一步,把所有实体转换为关系,用关系模式表示。

第二步,把 1:n 联系中 n 端对应的实体与联系合并,即:在 n 端对应实体关系中加入联系本身的属性。

第三步,把 1:n 联系中 1 端对应实体的主键加入 n 端对应实体关系中,作为外键,n 端对应实体关系的主键不变。

两个实体之间联系的属性合并到其中一个关系中。如果联系本身没有属性,只需要在一个实体关系中增加另一个实体关系的主键即可。

(3) m:n 联系的转换方法

【例 1-7】将图 1-8 中实体集的 E-R 图转换为关系。

学生(学号,姓名,性别,年龄,系别,联系电话)
 课程(课程编号,课程名,学时,学分)
 选课(学号,课程编号,成绩,学期)
 课程类型(课程类型编号,课程类型)

图 1-8 所示 E-R 图中,实体间是 m:n 联系,其中联系“选课”本身有 5 个属性:“学期”、“成绩”、“学分”、“补考标记”和“补考成绩”。

m:n 联系的转换方法为:

第一步,把所有实体转换为关系,用关系模式表示。

第二步,把和 m:n 联系相连的各个实体的主键,以及联系本身的属性合并,形成一个新的关系。新关系的主键由相连各个实体的主键组成(多属性构成的组合键)。

1.3.2 确定表结构

实体-联系模型转换为关系模型之后,实际上就已经确定了数据库中各个表的结构。每个实体对应一张二维表,实体的属性对应二维表的字段。

【例 1-8】例 1-7 中的“学生”、“课程”、“选课”、“课程类型”四个关系模式对应的二维表结构见表 1-3、1-4、1-5 和表 1-6 所示:

表 1-3

student(学生表)

字段名	字段含义	数据类型	长度	备注
Stu_id	学号	char	8	主键
Stu_name	姓名	varchar	8	非空
Stu_sex	性别	char	2	默认值和检查约束
Stu_age	年龄	int		>=0, 检查约束
Stu_depart	系别	varchar	20	
Stu_tel	联系电话	varchar	20	

表 1-4

course(课程表)

字段名	字段含义	数据类型	长度	备注
Course_id	课程编号	char	8	主键
Course_name	课程名	varchar	30	非空
Course_typeid	课程类型编号	char	8	外键
Course_typeid	学时	Int		>=0, 检查约束
Course_credit	学分	int		>=0, 检查约束

表 1-5

coursechoice(选课表)

字段名	字段含义	数据类型	长度	备注
Stu_id	学号	char	8	主键
Course_id	课程编号	char	8	主键
Stu_mark	成绩	int		>=0, 检查约束
Stu_term	学期	int		>=0, 检查约束

表 1-6

coursetype(课程类型表)

字段名	字段含义	数据类型	长度	备注
Course_typeid	课程类型编号	char	8	主键
Course_type	课程类型	varchar	30	非空

需要注意的是,对于没有主键的表,需要添加 ID 编号列,它没有实际的含义,只是用来作主键或外键。此外,还要按照实际情况,考虑到对每张表的数据完整性约束条件。

1.3.3 规范化数据

一般来说,如果数据库系统的数据存储量比较大,涉及的表比较多,表与表之间的联系较为复杂,就需要进行数据库结构的规范化设计,以合理规划数据库中的各个数据对象和它们之间的联系。

1. 什么是“规范化数据”

【例 1-9】假设有如下关系模式:

学生(学号,姓名,性别,年龄,系别,课程编号,课程名,成绩,学时,学分)