

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

项目2 设计数据库

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

【能力目标】

- 学会将现实世界的事物和特性抽象为信息世界的实体与关系
- 会使用实体联系图（E-R图）描述实体、属性和实体间的关系
- 会将E-R图转化为关系模型
- 能根据开发需求，将关系模型规范化到一定程度
- 对数据完整性有清晰的认识

【项目描述】

设计学生管理系统的数据库，绘制E-R图，转换成关系模型，指出各表关键字。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

【项目分析】

设计数据库是一个把现实世界抽象化，把信息世界数据化的过程，本项目以学生管理系统的XS数据库设计过程为例，介绍必要的数据库基础知识，学习数据库应用开发技术，达到能够设计开发数据库应用系统的目的。熟悉XS数据库中三个表XSDA、XSCJ、KCXX及它们的关系，初步了解数据库。

【任务设置】

任务1 附加与分离数据库认知数据库结构

任务2 现实世界数据化

任务3 转换成关系模型

任务4 认知关键字和数据完整性

项目2 设计数据库

任务1

任务2

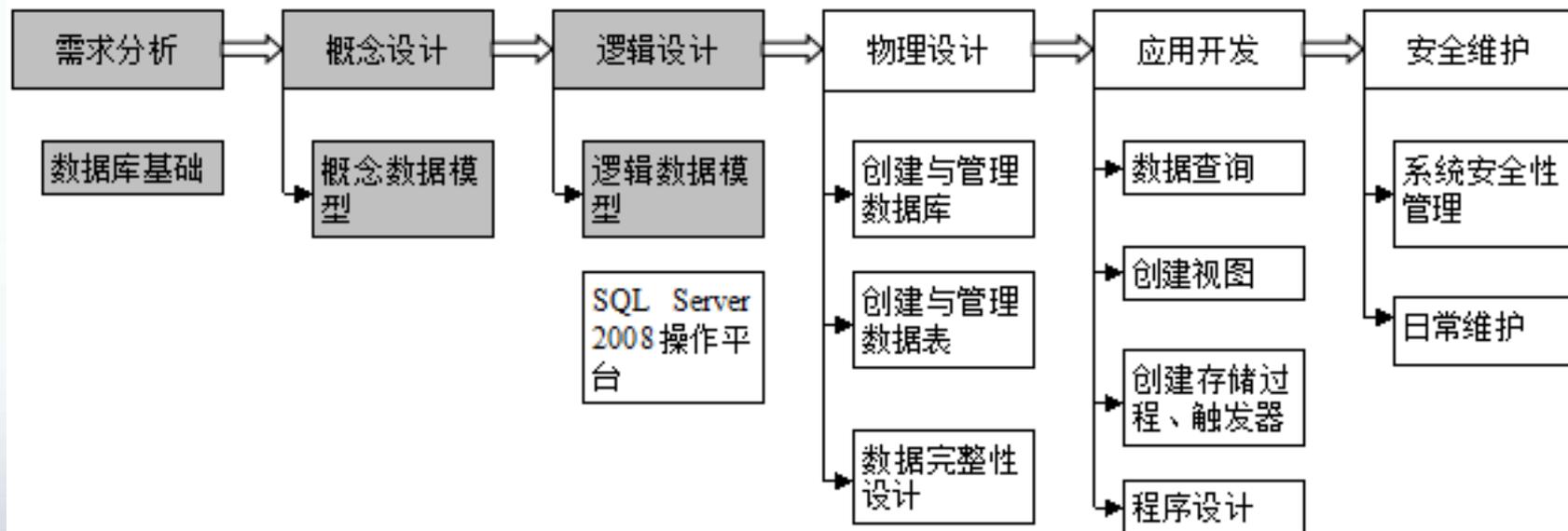
任务3

任务4

实训

【项目定位】

数据库系统开发



任务1 附加与分离数据库认知数据库结构

【任务目标】

- 熟练操作附加数据库的数据转移方法
- 熟练操作分离数据库的数据转移方法
- 理解SQL Server 2008数据库结构
- 熟悉本书示例数据库XS数据

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

【任务描述】

附加XS数据库，熟悉数据库数据，理解数据库结构。

【任务分析】

我们可以下载到XS数据库，但是数据如何加载到SQL Server 2008中呢？当然不需要重新建立数据库，只要附加数据库就可以了。本任务就介绍如何使用已有的数据库，并熟悉本书示例数据库XS。

任务1-1 附加数据库

用户在使用SQL Server 2008的过程中，可能会遇到下面的问题：将数据库从一个SQL Server服务器移到另一个SQL Server服务器上；数据库文件所在的磁盘空间用完。

对第一个问题一般的处理方法是创建一个新数据库，然后通过备份和还原移动数据库；对第二个问题的解决办法是在另一个磁盘上增加一个辅助数据文件。显然这两种办法都比较复杂，而在SQL Server 2008中为用户准备了一种简单得多的办法，**分离和附加数据库**。SQL Server 2008允许分离数据库的数据和事务日志文件，然后将其重新附加到另一台服务器，甚至同一台服务器上。

在SQL Server 2008中，除了**系统数据库**外，其余的数据库都可以从服务器中分离出来，脱离当前服务器的管理。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

1 使用SQL Server Management Studio附加数据库

【例2-1】使用SQL Server Management Studio将E:\databeifen文件夹中的数据库附加到当前的SQL Server实例上。

- (1) 启动SQL Server Management Studio，在【对象资源管理器】中右键单击【数据库】，选择【附加】，如图2-1所示。
- (2) 打开【附加数据库】对话框，进行相关设置，如图2-2所示。
- (3) 单击【添加】按钮，打开【定位数据库文件】对话框，选择要附加的主数据文件。
- (4) 附加数据库准备就绪后，单击【确定】按钮，完成数据库的附加操作。数据库附加成功后，在【数据库】节点中将会出现“XS”数据库节点。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

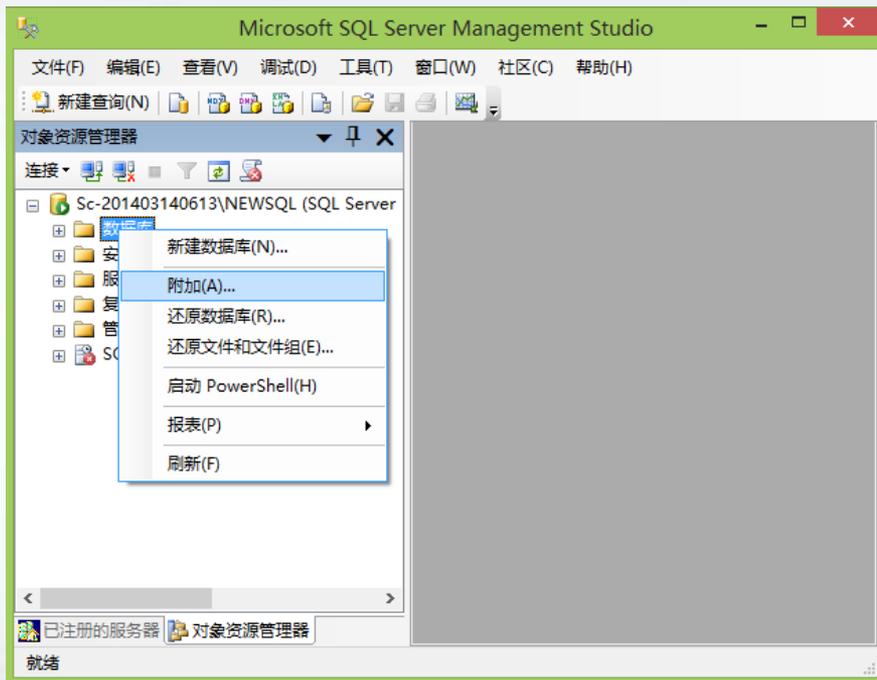


图2-1数据库“附加”

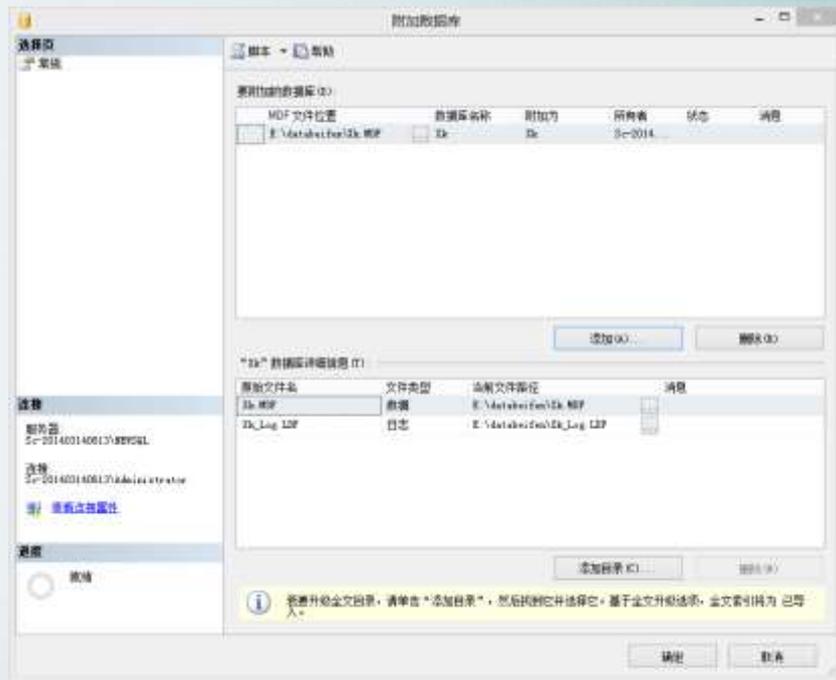


图2-2 “附加数据库”设置

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

2 使用T-SQL附加数据库

在SQL Server 2008中，使用存储过程EXEC sp_attach_db可以实现数据库的附加。

语法格式：

```
sp_attach_db 数据库名, @filename=文件名 [, ...16]
```

【例2-2】使用T-SQL语句将E:\data文件夹中的数据库附加到当前的SQL Server实例上。

```
exec sp_attach_db XS, 'I:\sql\xk\xk.mdf', 'I:\sql\xk\xk_Log.ldf'
```

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

任务1-2 认识系统数据库结构

XS数据库由包含选课系统所需的数据库信息组成，如图2-3所示。

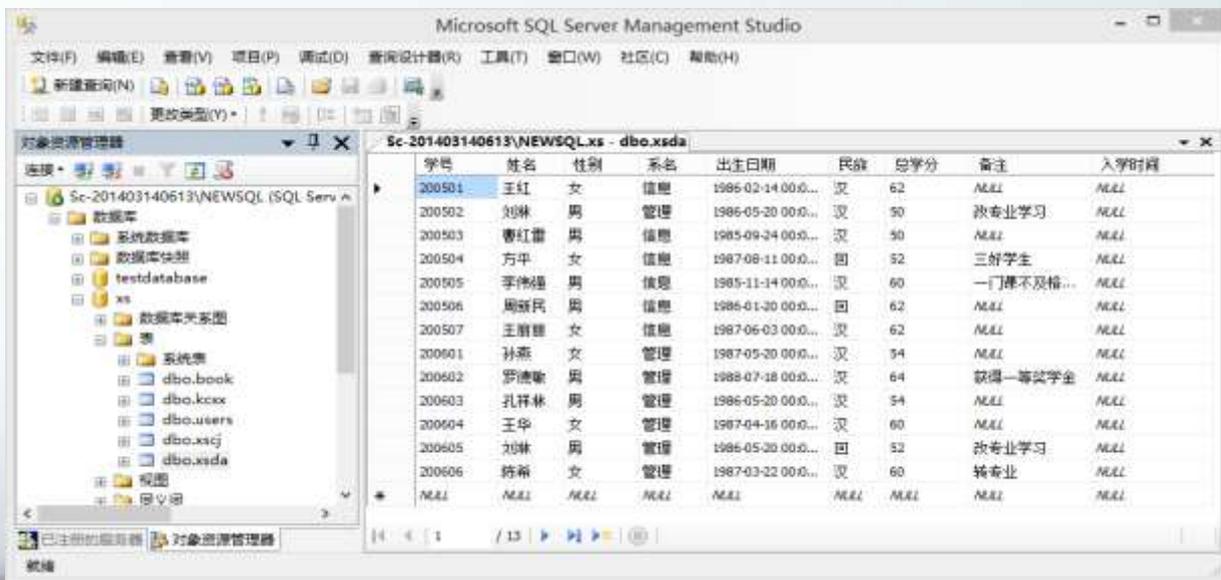


图2-3 “XS数据库”结构

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

服务器上数据库结构如图2-4所示。

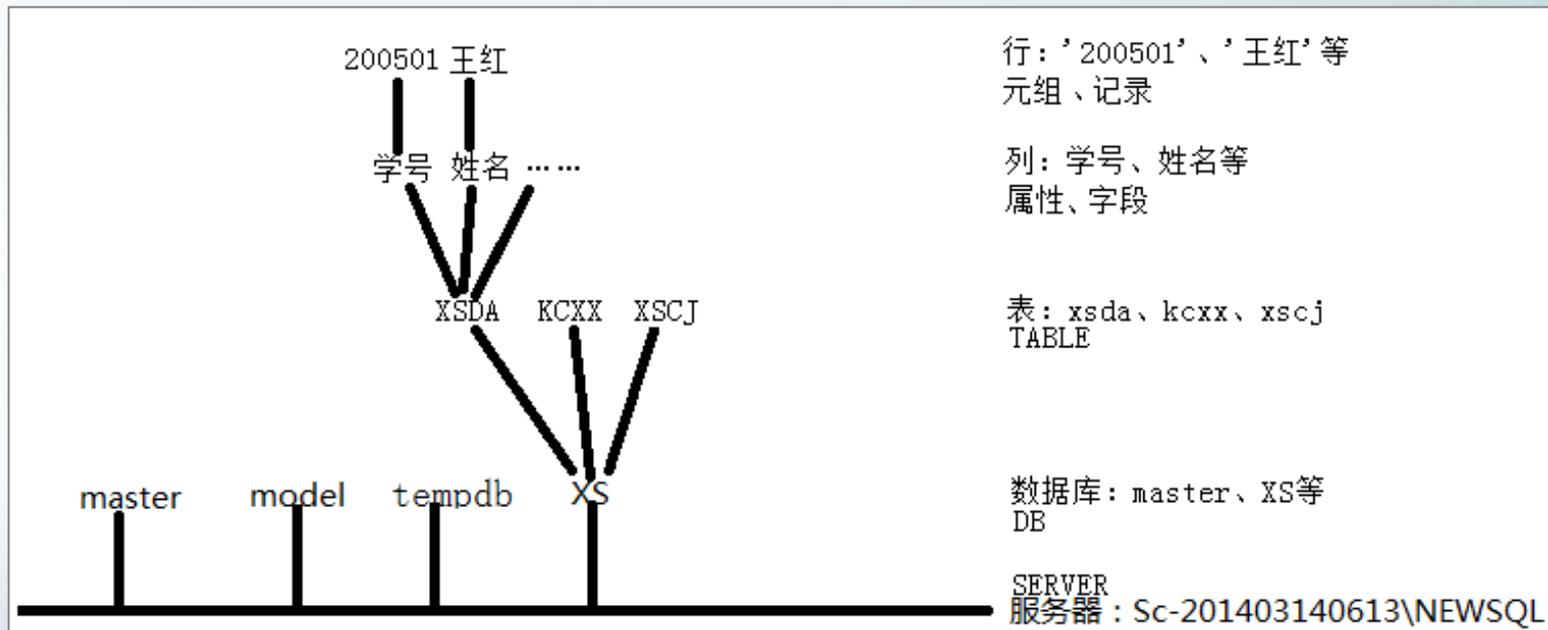


图2-4 服务器上数据库结构

任务1-3 分离数据库

1 使用SQL Server Management Studio分离数据库

【例2-3】使用SQL Server Management Studio实现xs数据库的分离，并将数据库对应的文件复制到E:\databeifen中。

(1) 启动SQL Server Management Studio，在【对象资源管理器】中展开【数据库】节点。

(2) 右键单击【XS】，选择【任务】中的【分离】，如图2-5所示。

(3) 打开【分离数据库】对话框，选择要分离的数据库，并进行相关设置，如图2-6所示。

任务1-3 分离数据库

1 使用SQL Server Management Studio分离数据库

(4) 分离数据库准备就绪后，单击【确定】按钮，完成数据库的分离操作。数据库分离成功后，在【数据库】节点中“XS”将不复存在。

(5) 将“E:\data”文件夹中XS数据库对应的两个文件复制到“E:\databeifen”文件夹中（如果该文件夹不存在，请首先创建该文件夹）。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

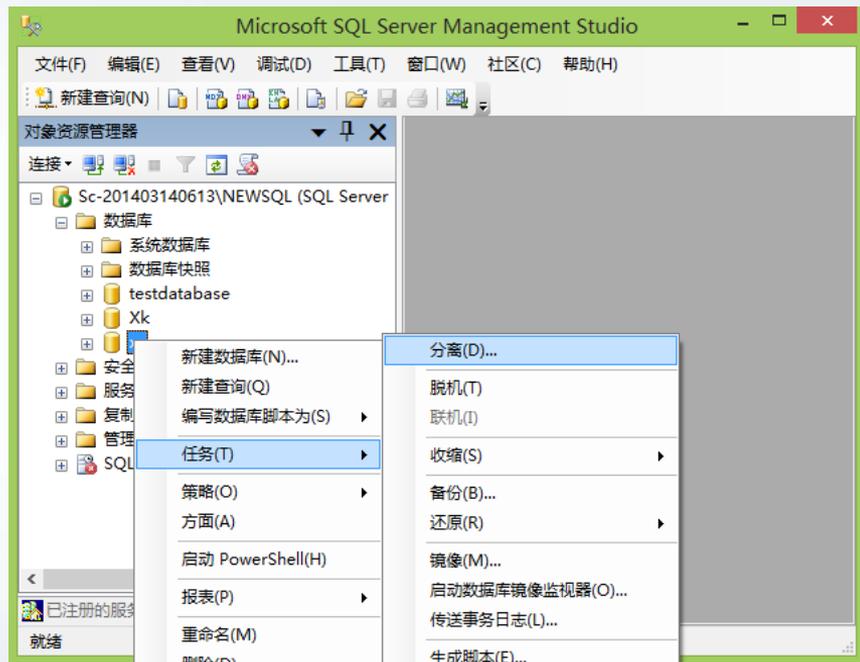


图2-5 数据库分离操作

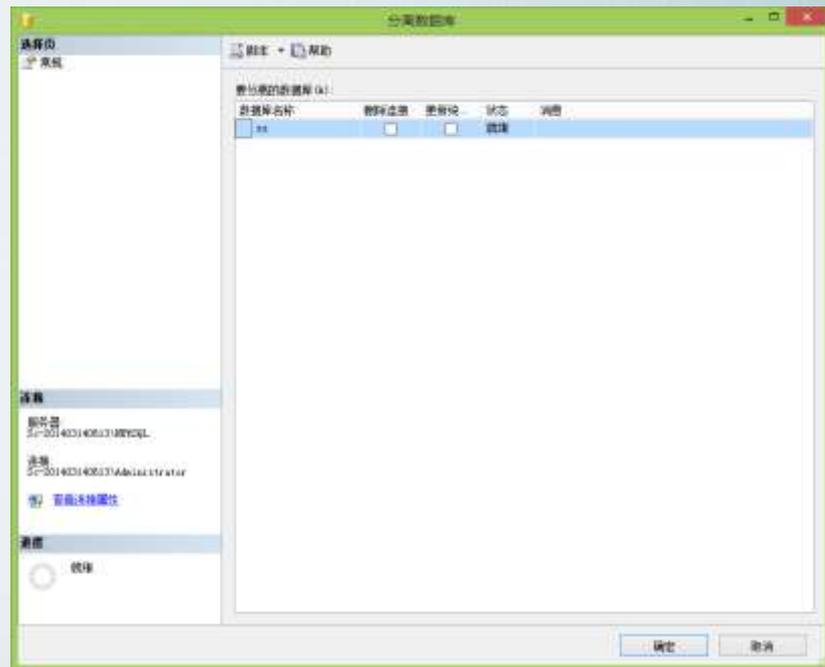


图2-6 “分离数据库”设置

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

2 使用T-SQL分离数据库

在SQL Server 2008中，使用存储过程sp_detach_db可以实现数据库的分离。

语法格式：

```
sp_detach_db 数据库名
```

【例2-4】使用T-SQL语句实现XS数据库的分离。

```
EXEC sp_detach_db 'XS'
```

任务2 现实世界数据化

【任务目标】

- 学会将现实世界的事物和特性抽象为信息世界的实体与关系
- 会使用实体联系图（E-R图）描述实体、属性和实体间的关系

【任务描述】

把学生选修课程抽象出来，绘制出E-R图。

【任务分析】

不能将现实世界中存在的客观事物直接输入到计算机中进行处理，必须将它们进行数据化后才能在计算机中进行处理。本任务以学生选课为具体应用，介绍将显示世界的客观事物进行数据化的过程。

任务2-1 现实世界数据化过程

1 数据

数据（Data）是描述事物的符号记录，用类型和数值来表示。随着计算机技术的发展，数据的含义更加广泛了，不仅包括数字，还包含文字、图像、声音和视频等多种数据。在数据库技术中，**数据是数据库中存储的基本对象。**例如，学生的档案管理记录、货物的运输情况等都是数据。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

信息不同于数据，**数据是信息的载体，信息是数据的含义，是一种已经被加工为特定形式的数据**，这种数据形式对接受者来说是有意义的。即只有有价值的数据才是信息。根据这个定义，那些能表达某种含义的信号、密码、情报、消息都可概括为信息。例如，一个“会议通知”，可以用文字（字符）写成，也可用广播方式（声音）传送，还可用闭路电视（图像）来通知，不管用哪种形式，含义都是通知，它们所表达的信息都是“会议通知”，所以“会议通知”就是信息。

数据和信息二者密不可分，**信息是客观事物性质或特征在人脑中的反映，信息只有通过数据形式表示出来才能被人理解和接受**，对信息的记载和描述产生了数据；反之，对众多相关数据加以分析和处理又将产生新的信息。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

人们从客观世界中提取所需数据，根据客观需要对数据处理得出相应信息，该信息将对现实世界的行为和决策产生影响，它对决策者能增加知识，具有现实的或潜在的价值，信息是经过加工处理以后的数据，两者的转换过程如图2-7所示。



图2-7 数据与信息的转换过程

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

2 数据处理

数据处理是指将数据转换成信息的过程。它是由人、计算机等组成的能进行信息的收集、传递、存储、加工、维护、分析、计划、控制、决策和使用的系统。经过处理，信息被加工成特定形式的数据。

在数据处理过程中，数据计算相对简单，但是处理的数据量大，并且数据之间存在着复杂的联系，因此，数据处理的关键是数据管理。

数据管理是指对数据收集、整理、组织、存储和检索等操作。这部分操作是数据处理业务的基本环节，是任何数据处理业务中必不可少的共有部分。因此必须学习和掌握数据管理的技术，对数据处理提供有利的支持。有效的数据管理可以提高数据的使用效率，减轻程序开发人员的负担。数据库技术就是针对数据管理的计算机软件技术。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

3 数据库

数据库（DataBase，DB）是指长期存储在计算机内，按一定数据模型组织存储、可共享的数据集合。它可以供各种用户共享，具有最小冗余度和较高的数据独立性。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

4 数据库管理系统

数据库管理系统（DataBase Management System, DBMS）是用户和操作系统之间的数据管理软件。它使用户方便地定义数据和操纵数据、并能够保证数据的安全性、完整性以及多用户对数据的并发使用及发生故障后的数据回复。其功能如下：

- (1) 数据定义功能
- (2) 数据操纵功能
- (3) 数据库运行管理功能
- (4) 数据库的维护功能

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

总之，**数据库管理系统是用户和数据库之间的交互界面**，在各种计算机软件中，数据库管理系统软件占有极其重要的位置。用户只需通过它就可以实现对数据库的各种操作与管理。数据库管理系统在计算机层次结构中的地位如图2-8。

目前，广泛应用的大型网络数据库管理系统有SQL Server、DB2、Oracle、Mysql、Sybase等。常用的桌面数据库管理系统有Visual FoxPro、Access等。



图2-8 DBMS在计算机层次结构中的地位

5 数据库系统

数据库系统 (Database System, DBS) 是指在计算机系统中引入数据库后的系统。一般由数据库、数据库管理系统及其开发工具、应用系统、数据库管理员和用户构成。数据库系统可用图2-9表示。

其中数据库管理员和用户主要是指存储、维护和检索数据各类使用者，主要有三类：

- (1) 最终用户 (End User, EU)
- (2) 应用程序员 (Application Programmer, AP)
- (3) 数据库管理员 (Database Administrator, DBA)

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

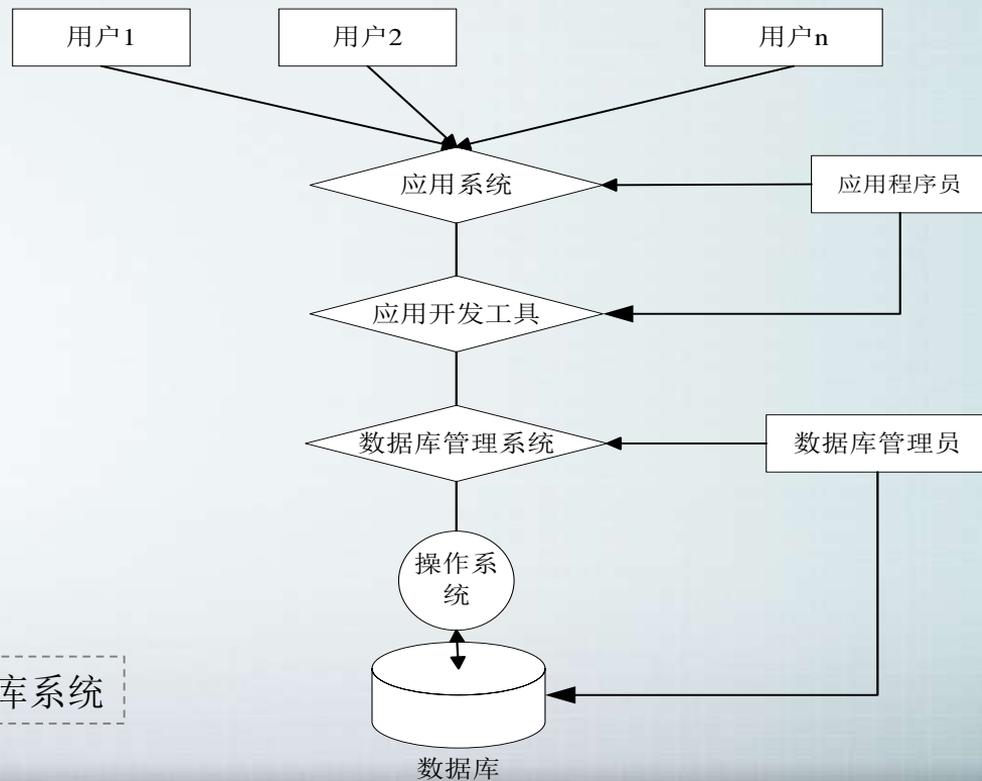


图2-9 数据库系统

项目2 设计数据库

任务1

任务2

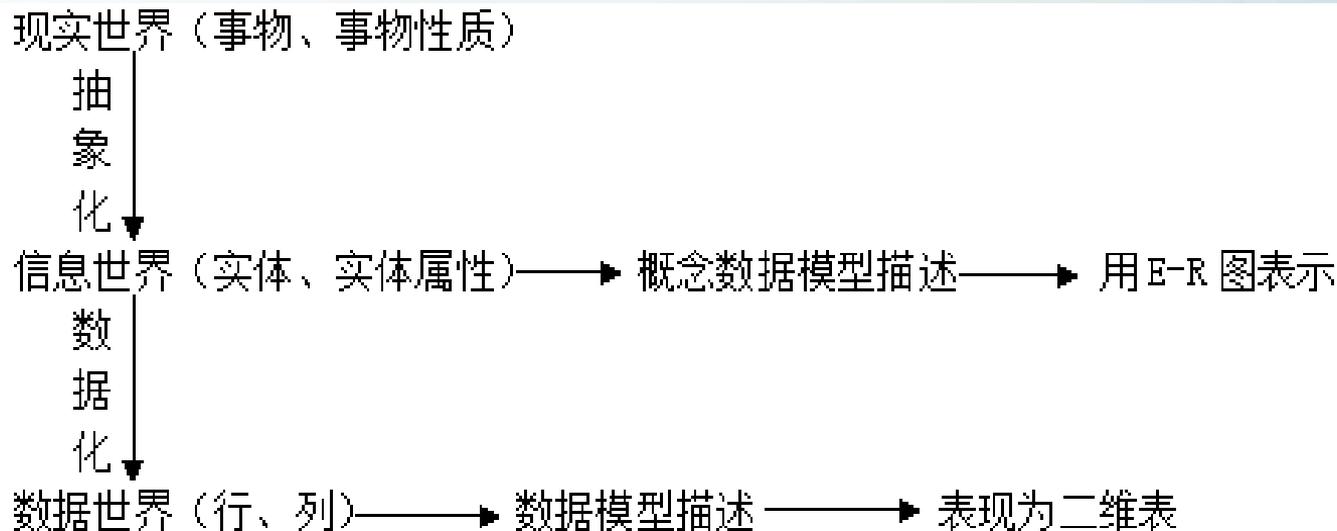
任务3

任务4

实训

5 现实世界数据化过程

将现实世界存在的客观事物进行数据化，要经历从**现实世界**到**信息世界**，再从信息世界到**数据世界**3个阶段。现实世界、信息世界和数据世界三者之间的关系如下。



项目2 设计数据库

任务1

任务2

任务3

任务4

实训

首先将现实世界中的客观存在的事物及它们所具有的特性抽象为信息世界的实体和属性。然后使用实体联系（Entity Relationship, E-R）图表示实体、属性、实体之间的联系（即概念数据模型），最后将E-R图转化为数据世界中的关系。

任务2-2 数据模型的概念

数据库是某个企业、组织或部门所涉及的数据的综合，它不仅要反映数据本身的内容，而且要反映数据间的联系。由于计算机不可能直接处理现实世界中的具体事物，所以人们必须实现把具体事物转换成计算机能够处理的数据，**在数据库中用数据模型这个工具来抽象、表示和处理现实世界中的数据和信息。通俗地讲，数据模型就是现实世界的模拟。**现有的数据库系统均是基于某种数据模型的。

数据库管理系统是按照一定的数据模型组织数据的，所谓的**数据模型是指数据结构、数据操作和完整性约束**，这三方面称为数据模型的三要素。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

1 数据结构

一组规定的用以构造数据库的基本数据结构类型。这是数据模型中最基本的部分，它规定如何把基本数据项组织成更大的数据单位，并通过这种结构来表达数据项之间的关系。由于数据模型是现实世界与机器世界的中介，因此，它的基本数据结构类型应是简单且易于理解的；同时，这种基本数据结构类型还应有很强的表达能力，可以有效地表达数据之间各种复杂的关系。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

2 数据操作

这些操作能实现对上述中数据结构按任意方式组合起来所得数据库的任何部分进行检索、推导和修改等。实际上，上述中的结构只规定了数据的静态结构，而操作的定义则说明了数据的动态特性。同样的静态结构，由于定义在其上的操作的不同，可以形成不同的数据模型。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

3 完整性约束

它用于给出不破坏数据库完整性、数据相容性等数据关系的限定。为了避免对数据执行某些操作时破坏数据的正常关系，常将那些有普遍性的问题归纳起来，形成一组通用的约束规则，只允许在满足该组规则的条件对数据库进行插入、删除和更新等操作。

综上所述，一个数据模型实际上给出了一个通用的在计算机上可实现的现实世界的信息结构，并且可以动态地模拟这种结构的变化。因此它是一种抽象方法，为在计算机上实现这种方法，研究者开发和研制了相应的软件——数据库管理系统（Data Base Management System, DBMS），DBMS是数据库系统的主要组成部分。

数据模型大体上分为两种类型：一种是独立于计算机系统的数据模型，即**概念模型**；另一种则是涉及到计算机系统和数据库管理系统的数据模型（**逻辑模型**）。

任务2-3 概念模型

信息是对客观事物及其相互关系的表征，同时数据是信息的具体化、形象化，是表示信息的物理符号。在管理信息系统中，要对大量的数据进行处理，首先要弄清楚现实世界中事物及事物间的联系是怎样的，然后再逐步分析、变换，得到系统可以处理的形式。因此对客观世界的认识、描述是一个逐步的过程，有层次之分，可将它们分成三个层次：

1 现实世界

现实世界是客观存在的事物及其相互联系，客观存在的事物分为“对象”和“性质”两个方面，同时事物之间有广泛的联系。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

2 信息世界

信息世界是客观存在的现实世界在人们头脑中的反映。人们对客观世界经过一定的认识过程，进入到信息世界形成关于客观事物及其相互联系的信息模型，在信息模型中，客观对象用实体表示，而客观对象的性质用属性表示。

3 数据世界

对信息世界中的有关信息经过加工、编码、格式化等具体处理，便进入了数据世界。数据世界中的数据既能代表和体现信息模型，同时又向机器世界前进了一步，便于用机器进行处理。在这里，每一实体用记录表示，相应于实体的属性用数据项（或称字段）表示，现实世界中的事物及其联系就用数据模型来表示。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

三个领域间的关系如图2-10表示。

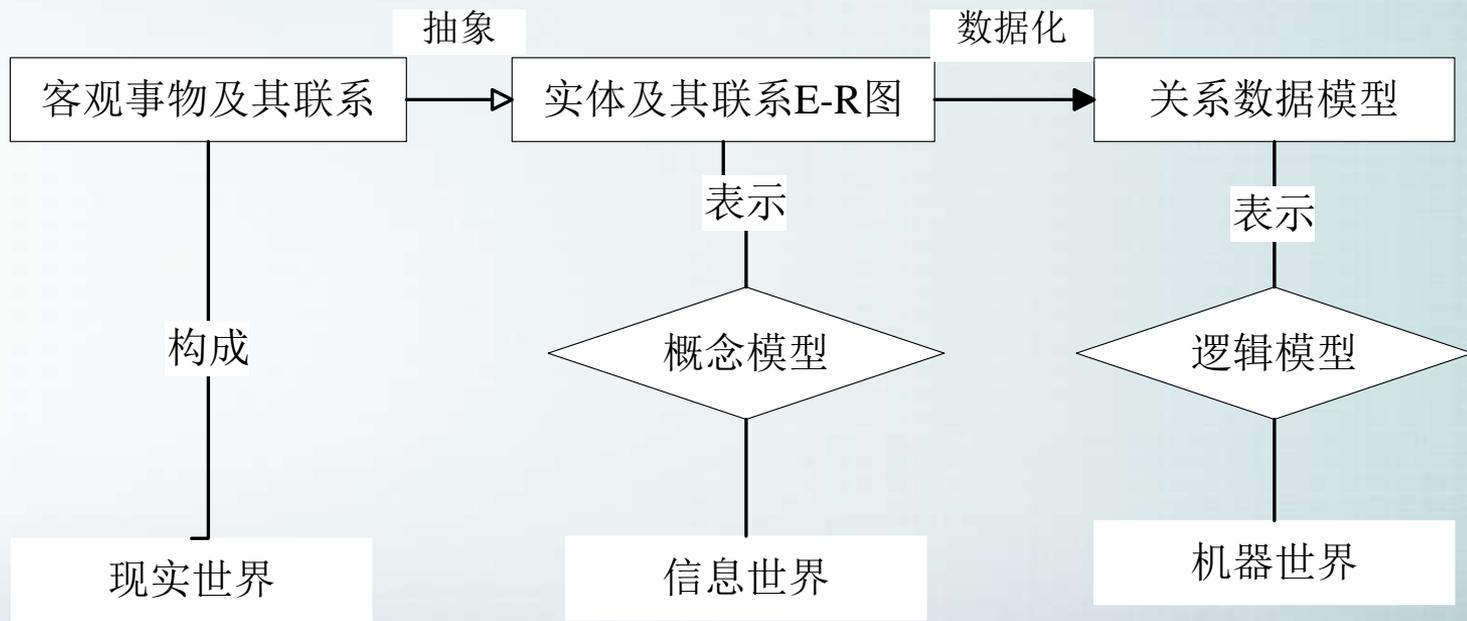


图2-10 客观描述的层次

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

由此可以看出，客观事物及其联系是信息之源，是组织和管理数据的出发点，同时也是使用数据库的归宿。为了把现实世界中的具体事物进行抽象，人们常常首先把现实世界抽象成为信息世界，然后再把信息世界转化为计算机世界。把现实世界抽象为信息世界，实际上是抽象出现实系统中有应用价值的元素及其联系，这时所形成的信息结构是概念模型。在抽象出概念模型后，再把概念模型转换为计算机上某一DBMS所支持的数据模型。**概念模型是现实世界到真实机器的一个中间层次，是按照用户的观点对数据和信息建模，是数据库设计人员与用户之间进行交流的语言。**

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

目前描述**概念模型的最常用的方法是实体—联系**（Entity-Relationship, E-R）方法。这种方法简单、实用，它所使用的工具称为E-R图。E-R图中包括实体、属性和联系三种图素。**实体用矩形框表示，属性用椭圆形框表示，联系用菱形框表示**，框内填入相应的实体名，实体与属性或者实体与联系之间用无向直线连接，多值属性用双椭圆形框表示，派生属性用虚椭圆形框表示。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

E-R模型中使用的基本符号如图2-11所示。

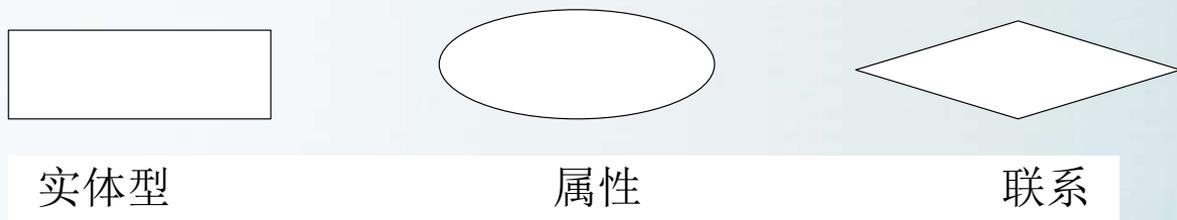


图2-11 E-R图基本符号表示

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

(1) 实体

客观存在并且可以相互区别的事物称为实体。实体可以是具体的事物，也可以是抽象的事件。例如，学生、图书等属于实际具体事物，订货、借阅图书等活动是抽象的事件。

(2) 实体集

同一类实体的集合。由于实体集中的个体成千上万，人们不可能也没有必要一一指出每一个属性，因此引入实体型。

(3) 实体型

对同类实体的共有特征的抽象定义。即把那个实体名及其属性名集合来抽象和描述。例如，学生（学号，姓名，年龄，性别，成绩）是一个实体型。

(4) 属性

描述实体的特性称为属性。例如，学生实体用学号、姓名、性别、年龄等属性来描述。不同的实体用不同的属性区分。

(5) 联系

实体之间的相互关系称为联系。它反映现实世界事物之间的相互关联。实体之间的联系可以归纳为三种类型：

- ① 一对一联系 (1:1)**：设A、B为两个实体集，如果A中的每个实体至多和B中的一个实体有联系，反过来，B中的每个实体至多和A中的一个实体有联系，称A对B或者B对A是一对一联系。例如，班级和班长这两个实体之间就是一对一的联系，如图2-12(a)所示。
- ② 一对多联系 (1:n)**：设A、B为两个实体集，如果A中的每个实体可以和B中的多个实体有联系，而B中的每个实体至多和A中的一个实体有联系，称A对B是一对多联系。例如，班级和学生这两个实体之间就是一对多联系，如图2-12(b)所示。
- ③ 多对多联系 (m:n)**：设A、B为两个实体集，如果A中的每个实体可以和B中的多个实体有联系，而B中的每个实体也可以和A中的多个实体有联系，称A对B或B对A是多对多联系。例如，学生和课程这两个实体之间就是多对多联系，如图2-12(c)所示。

项目2 设计数据库

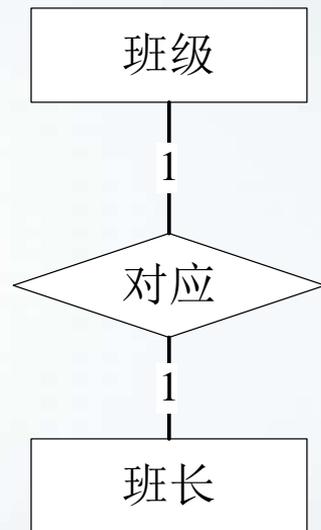
任务1

任务2

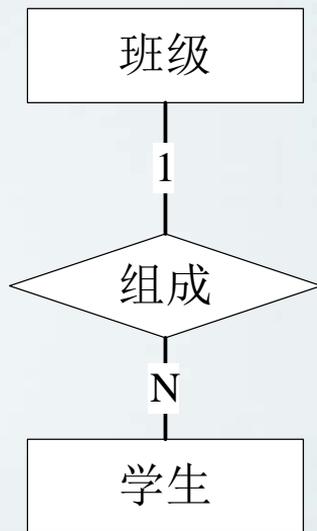
任务3

任务4

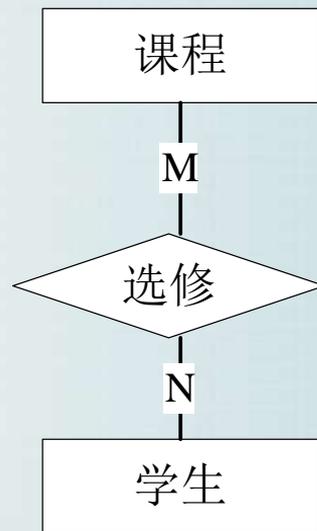
实训



(a)



(b)



(c)

图2-12 实体间的联系

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

值得注意的是：**联系也可以有属性**，例如，学生选修课程，则“选修”这个联系就有“成绩”属性，如图2-13所示。

由于实体集中的个体成千上万，人们不可能也没有必要一一指出个体间的对应关系，只需指出实体“型”间的联系，注明联系方式，这样既简单又能表达清楚概念。**具体画法是：把有联系的实体（方框）通过联系（菱形框）连接起来，注明联系方式，再把实体的属性（椭圆框）连到相应实体上。**

一般为了简洁起见，在E-R图中可略去属性，着重表示实体联系情况，属性可单独以表格形式列出。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

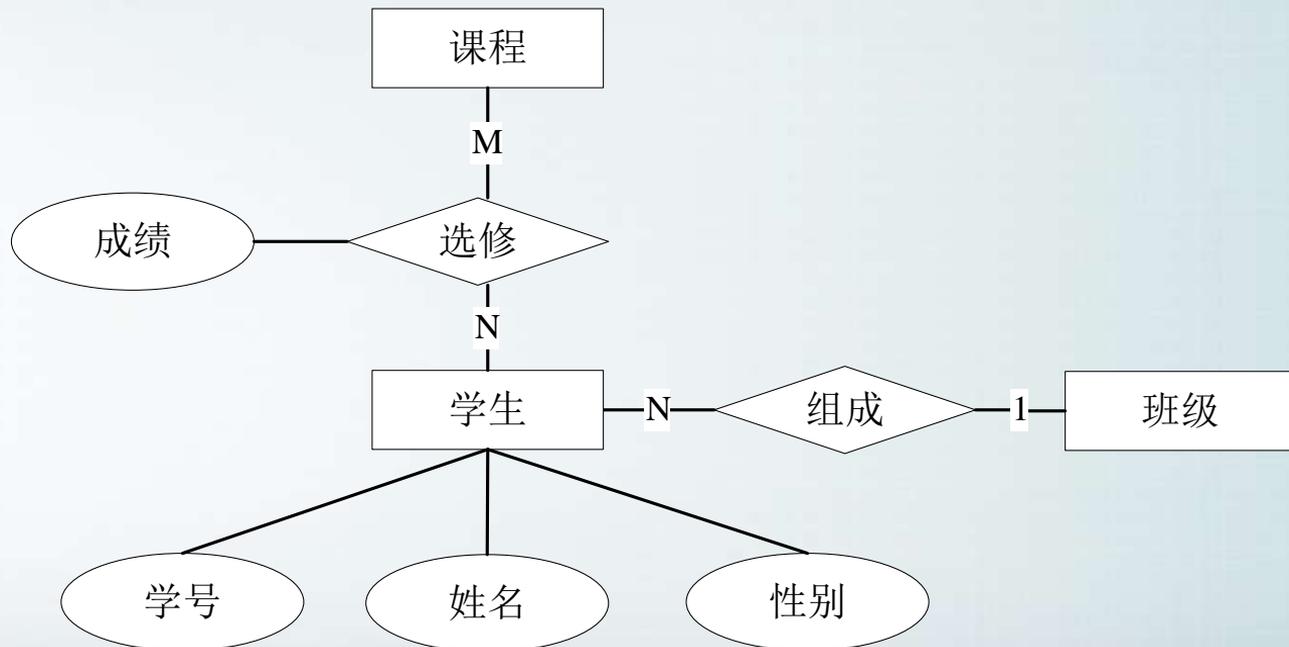


图2-13 实体联系属性图

任务2-4 绘制学生选修E-R图

1 需求分析

进行数据库设计首先必须准确了解与分析用户需求（包括数据与处理）。需求分析是整个设计过程的基础，是最困难、最耗费时间的一步，需求分析的结果是否准确地反映了用户的实际要求，将直接影响后续各个设计阶段。最终将影响到设计结果是否合理和实用。它的目的是分析系统的需求。该过程的主要任务是从数据库的所有用户那里收集对数据的需求和对数据处理的要求，主要涉及应用环境分析、数据流程分析、数据需求的收集与分析等，并把这些需求写成用户和设计人员都能接受的说明书。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

确定用户的最终需求其实是一件很困难的事，这是因为一方面用户缺少计算机知识，开始时无法确定计算机究竟能为自己做什么，不能做什么，因此无法马上准确地表达自己的需求，他们所提出的需求往往不断地变化。另一方面设计人员缺少用户的专业知识，不易理解用户的真正需求。此外新的硬件、软件技术的出现也会使用户需求发生变化。因此设计人员必须与用户不断深入地进行交流，才能逐步确定用户的实际需求。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

本书以“学生信息管理系统”的开发为例，简单描述数据库的开发流程。通过与某校学生处及教务处的学生管理流程为基准收集到其所需的**基本需求**包括：**学生档案管理、教学课程管理、学生成绩管理、系统管理**等内容。在学生档案管理中能够查询、修改、添加学生的基本档案信息；在教学课程管理中能针对每门所开设课程进行修改、添加新开设课程、删除淘汰的课程；在学生成绩管理中能针对学生每门课程的学习情况，记录其成绩并提供查询和修改功能。在系统管理中可以提供用户登陆验证、用户修改密码等功能。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

2 概念结构设计形成E-R图

针对“学生信息管理系统”的需求，我们抽取出各实体极其所需属性并形成局部E-R图。学生实体E-R图，如图2-14所示。

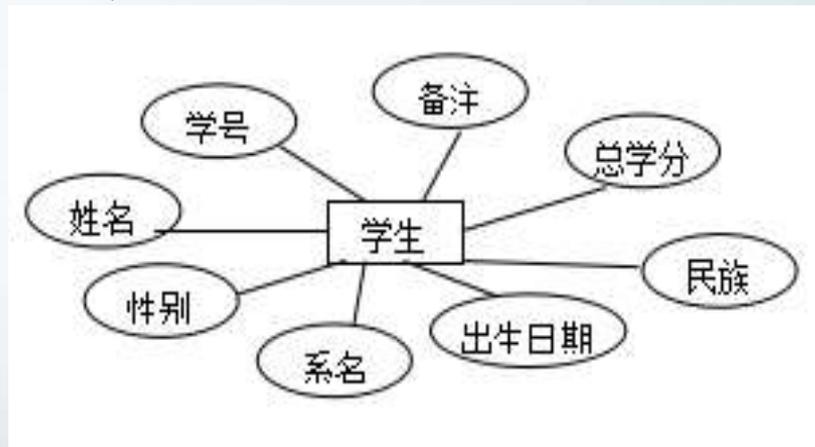


图2-14 学生实体E-R图

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

课程实体E-R图如图2-15所示。

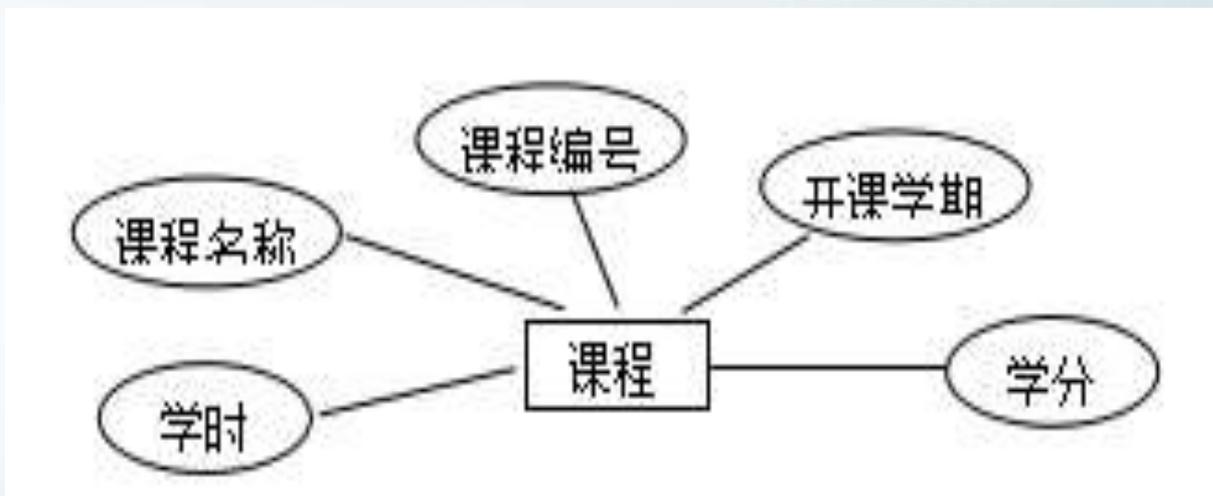


图2-15 课程实体E-R图

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

学生实体与课程实体之间的关系用成绩E-R图表示，如图2-16所示。



图2-16 实体联系成绩E-R图

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

用户实体E-R图如图2-17所示。



图2-17 实体用户E-R图

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

对局部E-R图综合整理后得到全局E-R图，如图2-18所示。

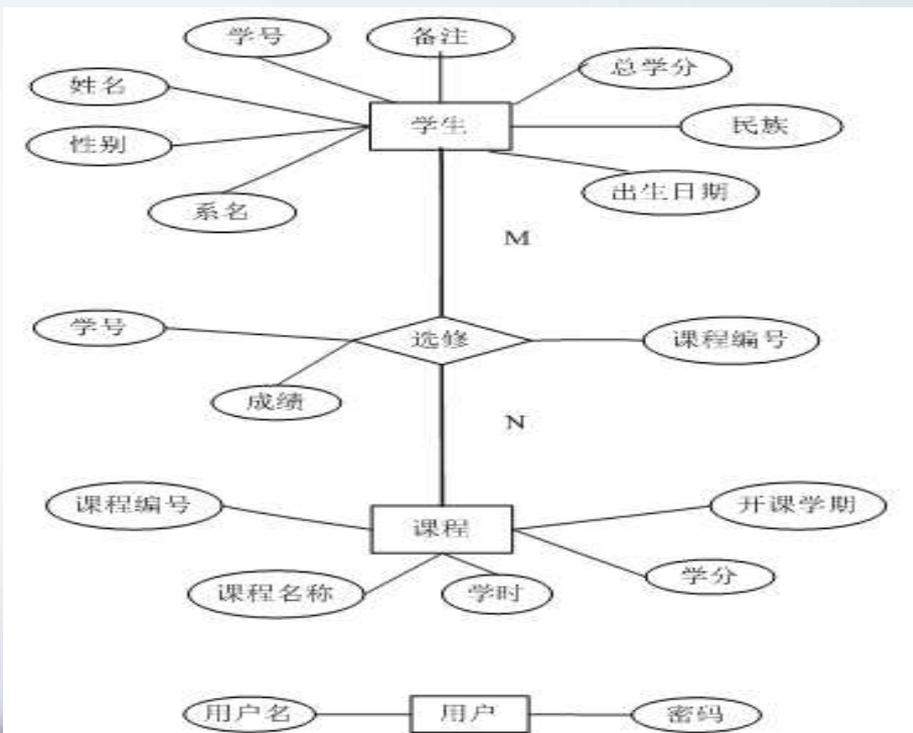


图2-18 全局E-R图

任务3 转换成关系模型

【任务目标】

会将E-R图转换成关系模型

能根据开发需求，将关系模型规范化到一定程度

【任务描述】

将学生选修E-R图转化成关系模型。

【任务分析】

关系模型是目前数据库系统普遍采用的数据模型，也是应用最广泛的数据类型。关系模型通过二维表来表示实体以及实体之间的联系。本任务就详细介绍关系模型和二维表。

任务3-1 逻辑模型

逻辑数据模型指数据库中数据的组织形式和联系方式，简称数据模型。数据库中的数据是按照一定的逻辑结构存储的，这种结构是用数据模型来表示的。现有的数据库管理系统都是基于某种数据模型的。按照数据库中数据采取的不同联系方式，数据模型可分为三种：层次模型、网状模型和关系模型。

1 层次模型

用树形结构表示实体及其之间联系的模型称为层次模型。在这种模型中，数据被组织成由根开始的倒置的一棵树，每个实体由根开始沿着不同的分支放在不同的层次上。

层次模型的优点是结构简单，层次清晰，易于实现。适合描述类似家族关系、行政编制及目录结构等信息载体的数据结构。

其基本结构有两个限制：

- 此模型中有且仅有一个结点没有双亲结点，称之为根节点。其层次最高。
- 根节点以外的其他结点有且仅有一个双亲结点。

所以使用层次模型可以非常直接、方便地表示1：1和1：n联系，但不能直接表示m：n联系，难以实现对复杂数据关系的描述。一个层次模型的简单例子如图2-19所示。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

该层次数据库PS具有6个记录类型。记录类型P（学院）是**根结点**，由字段P#（学院编号）、PN（学院名称）、PL（办公地点）组成，它有两个**子女结点**：D（系）和DP（部）。记录类型R（教研室）和S（学生）是记录类型D的两个子女结点。T（教师）是R的子女结点。其中，记录类型D由字段D#（系编号）和DN（系名）组成，记录类型DP由字段DP#（部门编号）和DPN（部门名称）组成，记录类型R由R#（教研室编号）、RN（教研室名称）和RC（教研室人数）组成，记录类型S由S#（学号），SN（学生姓名）和SS（学生成绩）组成。记录类型T由T#（教师编号）、TN（教师姓名）和TD（研究方向）组成。

在该层次结构中，DP、S、T是**叶子结点**，它们没有子女结点。由P到D、由P到DP、由D到R、由D到S、由R到T均是一对多的联系。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

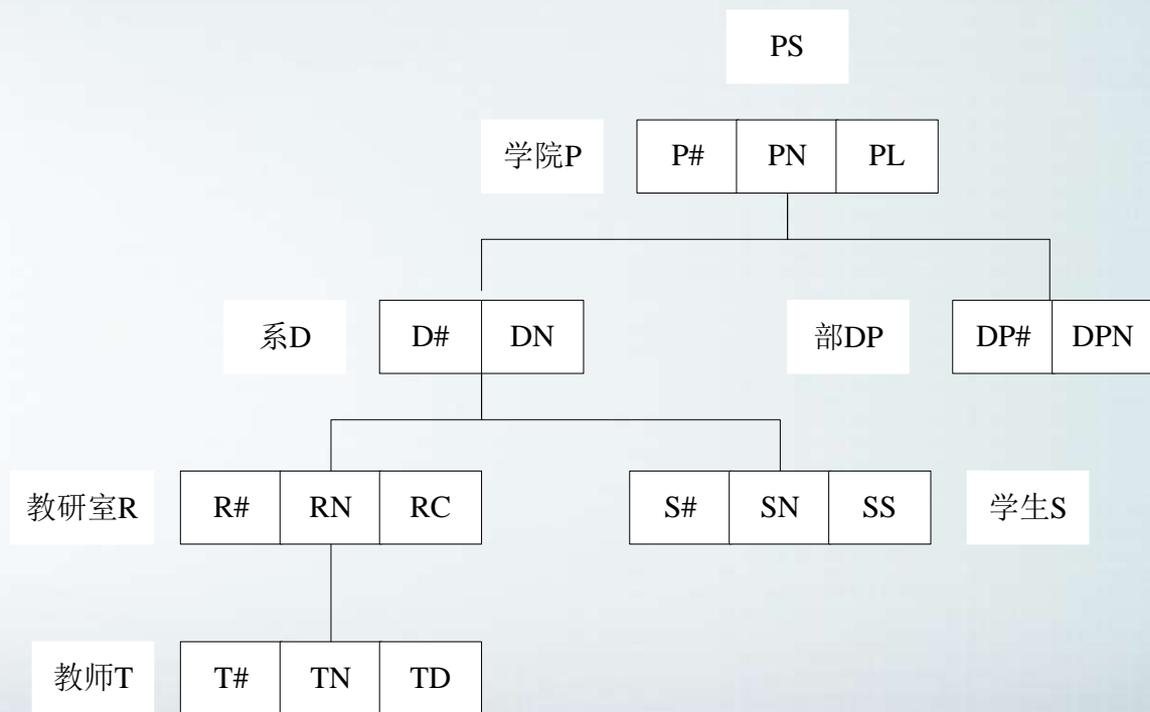


图2-19 PS数据库层次模型

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

2 网状模型

网状模型是一种比层次模型更具有普遍性的结构，它去掉了层次模型的两个限制，允许多个节点没有双亲结点，允许结点有多个双亲结点，此外它还允许两个结点之间有多种联系。因此网状模型可以更直接地描述现实世界。而层次模型实际上是网状模型的一个特例。

网状模型的主要优点是在表示数据之间的多对多联系时具有很大的灵活性，但是这种灵活性则是以数据结构的复杂化为代价的。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

网状结构可以有很多种，如图2-20所示。其中（a）是一个简单的网状结构，其记录类型之间都是1：n的联系。（b）是一个复杂的网状结构，学生与课程之间是m：n的联系，一个学生可以选修多门课程，一门课程可以被多个学生选修。（c）是一个简单环形网状结构，每个父亲可以有多个已为人父的儿子，而这些已为人父的儿子却只有一个父亲。（d）是一个复杂环形网状结构，每个子女都可以有多个子女，而这多个子女中的每一个都可以再有多多个子女（m：n）。（e）中人和树的联系有多种。（f）中既有父母到子女的联系，又有子女到父母的联系。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

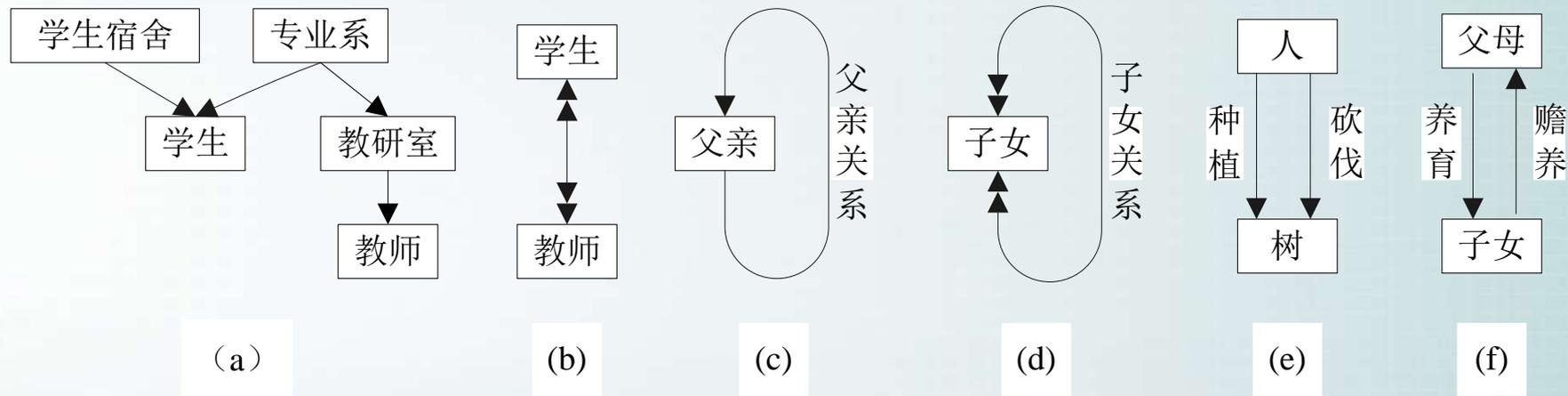


图2-20 网状模型

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

3 关系模型

关系模型是目前最重要的一种模型。美国IBM公司的研究员E. F. Codd于1970年发表题为“大型共享系统的关系数据库的关系模型”的论文，文中首次提出了数据库系统的关系模型。20世纪80年代以来，计算机厂商推出的数据库管理系统（DBMS）几乎都支持关系模型。

关系模型与层次和网状模型的理论 and 风格截然不同，如果说层次和网状模型是用“图”表示实体及其联系的话，那么关系模型则是用“二维表”来表示的。从现实世界中抽象出的实体及其联系都使用关系这种二维表表示。而**关系模型就是用若干个二维表来表示实体及其联系的，这是关系模型的本质。**关系模型如图2-21所示。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

学生登
记表

学号	姓名	性别	系别	年龄	籍贯
95001	李勇	男	计算机科学	20	江苏
95002	刘晨	女	信息	19	山东
95003	王名	女	数学	18	北京
95004	张立	男	计算机科学	19	北京
.....
95700	杨晓冬	男	物理	20	山西

图2-21 关系模型

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

由于关系型数据库采用了人们习惯使用的表格形式作为存储结构，易学易用，因而成为使用最广泛的数据库模型。

任务3-2 认知关系模型的基本概念

二维表并不一定是关系模型，只有具有下面**特点**二维表才是关系模型。

- (1) 表格中的每一列都是不可再分的基础数据项。
- (2) 每列的名字不同，数据类型相同或者兼容。
- (3) 行的顺序无关紧要。
- (4) 列的顺序无关紧要。
- (5) 关系中不能存在完全相同的两行。

通常将关系模型称为关系或者表；将关系中的行称为元祖或记录；将关系中的列称为属性或字段。

2 关系的约束

关系表现为二维表，但不是所有的二维表都是关系。成为关系的二维表有如下约束。

- (1) 不允许“表中套表”，即表中每个属性必须是不可分割的数据单元。或者说每个字段不能再分为若干个字段，即表中不能再包含表。
- (2) 在同一个关系中不能出现相同的属性名。
- (3) 列的次序可以任意交换，不改变关系的实际意义。
- (4) 表中的行叫元组，代表一个实体，因此表中不允许出现相同的两行。
- (5) 行的次序可以任意交换，不会改变关系的意义。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

1 关系术语

在关系数据库中，有以下几个常见的**关系术语**：

- (1) 关系
- (2) 记录（元组）
- (3) 字段（属性）
- (4) 域
- (5) 关键字
- (6) 关系模式

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

注意事项：

上述各种名词，在实际的关系型DBMS中往往叫法不一样，如在SQL Sever大型数据库中把关系称为“数据库文件”或“表”，把属性称为“字段”，把元组称为“记录”。

任务3-3 转换学生选修E-R图成关系模型

关系模型的逻辑结构是一组关系模式的集合。而E-R图则是实体、实体的属性和实体之间的联系三个要素组成的。所以**将E-R图转换为关系模型实际上就是要将实体、实体的属性和实体之间的联系转化为关系模式**，这种转换一般遵循如下原则：

- (1) 一个实体转化为一个关系模式，实体的属性即为关系的属性，实体的关键字就是关系的关键字。
- (2) 若是一个1:1的联系，可在联系两端的实体关系中的任意一个关系的属性中加入另一个关系的关键字。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

(3) 若是一个1:n的联系，可在n端实体转换成的关系中加入1端实体关系中的关键字。

(4) 若是一个n:m的联系，可转化为一个关系。联系两端各实体关系的关键字组合构成该关系的关键字，组成关系的属性中除关键字外，还有联系自有的属性。

(5) 三个或三个以上实体间的一个多元联系转换为一个关系模式。

(6) 实体集的实体间的联系，即自联系，也可按上述1:1、1:n、m:n三种情况分别处理。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

(7) 具有相同关键字的关系可以合并。

按照上述转换原则，将图2-14所示的学生信息管理系统全局E-R图转换为如下的关系模型：

学生档案（学号，姓名，性别，系名，出生日期，民族，总学分，备注）

课程信息（课程编号，课程名称，开课学期，学时，学分）

学生成绩（学号，课程编号，成绩）

用户信息（用户名，密码）

任务3-4 关系规范化

关系规范化的目的是为了**消除存储异常、减少数据冗余**（即重复），以保证数据完整性（即数据的正确性、一致性）和存储效率，一般将关系规范到III范式即可。

表2-3、表2-4都满足关系模型的5个特点，它们都是关系，但还存在以下几个问题：

- (1) 数据冗余：学号、姓名在2个表中重复出现，数据冗余（重复）。
- (2) 数据可能会不一致：学号、姓名的重复出现，在修改数据时容易出现数据不一致的情况。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

(3) 数据维护困难：数据在多个表中的重复出现，造成对数据库的维护困难。例如，某个学生因故更改姓名，则需要在学生表和学生选课表中更改两次，才能保证数据的一致性。数据维护工作量大。

关系数据库中的关系要满足一定的规范化要求，对于不同的规范化程度，可以使用“范式”进行衡量，记作NF。满足最低要求的为I范式，简称1NF。在I范式的基础上，进一步满足一些要求的为II范式，简称2NF。同理，还可以进一步规范为III范式。（参考P 57—59）

1. I 范式
2. II 范式
3. III 范式

任务3-5 关系运算

数学中的算术运算是大家熟悉的，例如 $2+3$ 是一个数学运算，该运算的运算对象是数值2和3，运算法则是加法，运算结果也是数值（5）。与此类似，关系运算的运算对象是关系（二维表），关系运算的法则包括选择、投影、连接，关系运算结果也是关系（二维表）。下面我们分别介绍这三种关系运算法则。

1. 选择 (Selection)

选择运算是在一个关系中找到满足给定条件的**记录**。选择是**从行的角度**对二维表格的内容进行筛选，形成新的关系。如关系学生，如表2-9所示。说明：性别一列中0表示女，1表示男。

2. 投影 (Projection)

投影是从一个关系中找到符合条件的**属性列**组成新的关系。投影是**从列的角度**对二维表的内容进行的筛选或重组，形成新的关系。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

表2-9 学生表

学号	姓名	性别	系名	出生日期	民族	总学分	备注
200801	王红	女	信息	1986-02-14	汉	60	无
200802	刘林	男	信息	1986-05-20	汉	54	无
200803	曹红雷	男	信息	1985-09-24	汉	50	无
200804	方平	女	信息	1987-08-11	回	52	三好学生
200805	李伟强	男	信息	1985-11-14	汉	60	一门课不及格

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

3. 连接 (Join)

连接是从两个关系的笛卡儿积中选取属性之间满足一定条件的元组形成的新关系。在此我们不给出笛卡尔积的定义，仅仅通过一个举例希望读者对笛卡尔积作一个初步了解，对于笛卡尔积的定义请大家参考离散数学。

设关系R和关系S分别如表2-13和1-14所示。

表2-13 关系R

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

表2-14 关系S

A	D
a1	c4
a3	c5

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

说明：关系R中的属性A和关系S中的属性A取自相同的域。

对关系R和关系S做笛卡尔积的结果如表2-15所示。

R.A	B	C	S.A	D
a1	b1	c1	a1	c4
a1	b1	c1	a3	c5
a2	b2	c2	a1	c4
a2	b2	c2	a3	c5
a3	b3	c3	a1	c4
a3	b3	c3	a3	c5

表2-15 R和S的笛卡尔积

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

连接运算中有两种常用的连接，一种是等值连接（Equi-join），另一种是自然连接（Natural join）。

等值连接是选取R与S的笛卡儿积的属性值相等的那些元组。（P 61）

自然连接要求两个关系中进行比较的分量必须是相同属性组，且要在结果中把重复的属性去掉。（P 61）

任务3-6 关系数据库

基于关系模型建立的数据库称为关系型数据库，它具有通用的数据管理功能、数据表示能力较强、易于理解、使用方便。20世纪80年代以来，关系数据库理论日益完善，并在数据库系统中得到了广泛的应用，是各类数据库中使用最为重要、最为广泛、最为流行的数据库。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

关系数据库是一些相关的表和其他数据库对象的集合。这里有三层**含义**：

1 在关系型数据库中，数据存储在二维表结构的数据表中，**一个表是一个关系**，又称为实体集。

- 一个表包含若干行，每一行称为一条记录，表示一个实体。
- 每一行数据有多列组成，每一列称为一个字段，反映了该实体某一方面的属性。
- 实体的属性中，能唯一标识实体集中每个实体的某个或某几个属性，称为实体的**关键字**，在关系型数据库中，**关键字被称为主键**。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

2 数据库所包含的表之间是有联系的，联系由表的**主键**和**外键**所体现的参照关系实现。

- 关系表现为表。关系数据库一般由多个关系（表）组成。
- 表之间由某些字段的相关性而产生联系。在关系数据库中，表既能反映实体，又能表示实体之间的联系。
- 用表的主键和外键反映实体间的联系。在关系数据库中，**外键**（Foreign Key）是指表中含有的与另一个表的主键相对应的字段，它用来与其他表建立联系。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

3 数据库不仅包含表，还包含其他的数据库对象，如视图、存储过程和索引等。

- 数据库是存储数据的容器，数据主要保存在数据库的表中，所以**数据表是数据库的基本对象**。除此之外，在数据库中还有其他对象，常用的有：
- **视图**：是一个虚拟表，可用于从实际表中检索数据，并按指定的结构形式浏览。
- **存储过程**：是一个预编译的语句和指令的集合，可执行查询或者数据维护工作。
- **触发器**：是特殊的存储过程，可设计在对数据进行插入、修改或删除时自动调用。
- **索引**：用于实现快速对数据表中数据的检索访问，以及增强数据完整性。
- **规则**：通过绑定操作，可用于限定数据表中数据的有效值或数据类型。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

目前使用的数据库系统大都是关系型数据库。现在关系型数据库以其完备的理论基础、简单的模型和使用的便捷性等优点获得了最广泛的应用。本教材中阐述的数据库模型都是关系型模型。

任务4 认知关键字和数据完整性

【任务目标】

- 对数据完整性有清晰的认识
- 对关键字有清晰的认识

【任务描述】

指出各表的关键字，举例说明如何保证学生成绩表的数据完整性。

【任务分析】

数据完整性是数据库设计，日常维护的关键技术，本任务就介绍关键字和数据完整性。

任务4-1 认知关键字

1 关键字

关键字是用来唯一标识表中每一行的属性或属性的组合，通常也被称为**关键码、码或键**。

2 候选关键字与主关键字

候选关键字是那些可以用来作关键字的属性或属性的组合。将选中的那个关键字称**为主关键字**（**Primary Key**，或称**为主码、主键**）。在一个表中能指定一个主关键字，它的值必须是唯一的，并且不允许为空值（NULL值，未输入值的未知值）。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

3 公共关键字

公共关键字就是连接两个表的公共属性。

4 外关键字

外关键字（Foreign Key）亦称外键或外码，它由一个表中的一个属性或多个属性所组成，是另一个表的主关键字。实际上外关键字本身只是主关键字的副本，它的值允许为空（NULL）。外关键字是一个公共关键字。使用主关键字和外关键字建立起表和表之间的联系。

5 主表与从表

将主关键字所在的表称为主表（亦被称为父表），将外关键字所在的表称为从表（亦被称为子表）。

任务4-2 认知数据完整性

数据的完整性就是数据的正确性和一致性，它反映了现实世界中实体的本来面貌。如一个人身高为15米、年龄为300岁等就是完整性受到破坏的例子，因为这样的数据是无意义的，也是不正确的数据。

数据的完整性分为列完整性、表完整性和参照完整性。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

1 列完整性

列完整性也可称为域完整性或用户定义完整性。**列完整性**是指表中任一列的数据类型必须符合用户的定义，或数据必须在规则的有效范围之内。

如表学生档案表2-8的学号，已定义长度为6，数据类型为字符型。如果输入“0000001”（长度为7），该数据不符合对学号属性的定义，则说学号列完整性遭到了破坏。

再如，表2-5学生成绩表中的成绩属性，已定义数据的有效范围为0-100，如果输入“124”，则破坏了成绩属性的列完整性。

2 表完整性

表完整性也可称为实体完整性。**所谓表完整性**，是指表中必须有一个主关键字，且主键值不能为空（NULL）。

例如，表学生档案表2-8的学号为主关键字，它的值不允许为空并且要唯一，从而保证学生档案表的完整性。

表2-5学生成绩表以（学号，课程编号）为主关键字，它的值不允许为空，意味着学号、课程编号的值都不能为空；并且主键的值要唯一，才能保证学生成绩表的完整性。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

3 参照完整性

参照完整性也称为引用完整性，它指对外键值进行插入或修改时一定要参照主键的值是否存在。对主键进行修改或删除时，也必须参照外键的值是否存在。这样才能使得通过公共关键字连接的两个表保证参照完整性，则说两个表的主关键字、外关键字是一致的。

例如，表2-2课程信息的课程编号为主关键字，表2-5学生成绩的课程编为外关键字。学生成绩表的课程编号属性值一定要在主表（课程信息表）中存在，如果它在主表中不存在，或者在课程信息表中删除了一个在学生成绩表中存在的课程标号，就破坏了参照完整性。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

SQL Server 2008提供了一系列的技术来保证数据的完整性。例如，定义数据类型、CHECK约束、DEFAULT约束、唯一标识、规则、默认值保证了列数据完整性；唯一索引、主关键字等保证了表数据完整性；主关键字与外关键字、触发器可以保证表与表之间的参照完整性。

实训二 设计数据库练习

1. 绘制sale销售数据库E-R图，要求包括客户表、产品表、入库表、销售表，可参看实训四。
2. 指出sale数据库中各表的主关键字、公共关键字、外关键字、数据完整性关系。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

实训二 设计数据库练习

小结

本项目主要介绍了数据库的基本概念，数据库设计方法、E-R图绘制方法、常见的数据模型、关系数据库中的基本术语、关系运算、数据库设计的基本步骤等知识。本项目内容是本书的基础，有助于以后理解和掌握以后章节的内容。

数据库（DataBase，DB）是指长期存储在计算机内的，按一定数据模型组织的、可共享的数据集合。它可以供各种用户共享，具有最小冗余度和较高的数据独立性。

数据库系统（DataBase System，DBS）是指在计算机系统中引入数据库后的系统。一般有数据库、数据库管理系统及其开发工具、数据库管理人员和用户构成的。数据库管理系统是整个数据库系统的核心。数据库系统的主要特点包括数据结构化、数据共享、数据独立性以及统一的数据控制功能。

项目2 设计数据库

任务1

任务2

任务3

任务4

实训

实训二 设计数据库练习

数据库管理系统是按照一定的数据模型组织数据的，所谓的数据模型是指数据结构、数据操作和完整性约束，这三方面成为数据模型的三要素。数据模型大体上分为两种类型：一种是独立于计算机系统的数据模型，即概念模型；另一种则是涉及到计算机系统和数据库管理系统的数据模型，现有的数据库管理系统都是基于某种数据模型的。按照数据库中数据采取的不同联系方式，数据模型可分为三种：层次模型、网状模型和关系模型。关系数据库主要支持三种关系运算：选择、投影和连接。从数据库应用系统开发的全过程来考虑，将数据库的设计归纳为六大步骤：需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库实施和数据库运行与维护。