关系数据库设计概述

在数据库领域内，通常把使用数据库的各类应用系统称为数据库应用系统，如办公自动化系统、电子政务系统、电子商务系统。数据库设计是设计数据库的逻辑模式和物理结构并建立数据库，这是构建数据库应用系统的基础部分。通常，一个好的数据库设计才能构建出好的数据库应用系统。关系数据库设计是指为基于关系数据库管理系统构建的应用系统设计关系模式。

设计步骤

与第三章中介绍的文档数据库设计一样，关系数据库设计分为需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库实施以及数据库运维六个基本步骤。在数据库设计过程中，需求分析和概念结构设计独立于任何数据管理系统进行，逻辑结构设计和物理结构设计与选用的数据管理系统密切相关。

本章重点介绍关系数据库设计中的概念结构设计和逻辑结构设计。

- 概念结构设计，解决应用系统中"存什么"的问题，即通过需求分析确定数据库中应用存储哪些数据对象（数据实体），每个数据实体包含哪些信息，数据实体和实体之间具有怎样的联系，然后形成一个具体的概念模型来描述应用；
- 逻辑结构设计，解决应用系统中数据"怎么存"的问题，即基于概念模型设计关系数据库的关系模式，用二维表描述数据实体和实体之间的联系。一旦关系模式定义之后，就可以创建关系数据库以及构建应用程序系统。

关系数据库的概念结构设计与文档数据库的概念结构设计一样，可以利用E-R图来描述现实世界的概念模型。但是，二者在逻辑结构设计时遵循的规则是不一样的。文档模式设计的常用规则为：

- 一个数据实体可以转换为一个文档集；数据实体之间的一对一联系合并到任意一端的文档集；数据实体之间的一对多联系合并到多端的文档集；数据实体之间的多对多联系单独转换为一个文档集。
- 多个数据实体也可以转换为一个文档集，以文档嵌套的方式来描述数据实体之间的联系。

文档数据库支持以数组、文档嵌套、文档数组的形式组织信息，所以多个数据实体可以转换为一个文档集。但是，关系数据库不支持数组和表嵌套的形式，无法实现将多个数据实体转换成一个关系表和将多对多联系合并到任意一端的关系表中。文档数据库的模式设计规则并不适用于关系数据库。关系数据库的模式设计规则是怎样的呢？读者朋友们可以先思考一下。

概念结构设计

关系数据库概念结构设计常使用E-R模型来描述现实世界的概念模型。E-R模型是由实体、属性、实体间的联系三个要素组成。第3章3.2小节介绍了三要素的基本概念以及E-R图的表示方法。

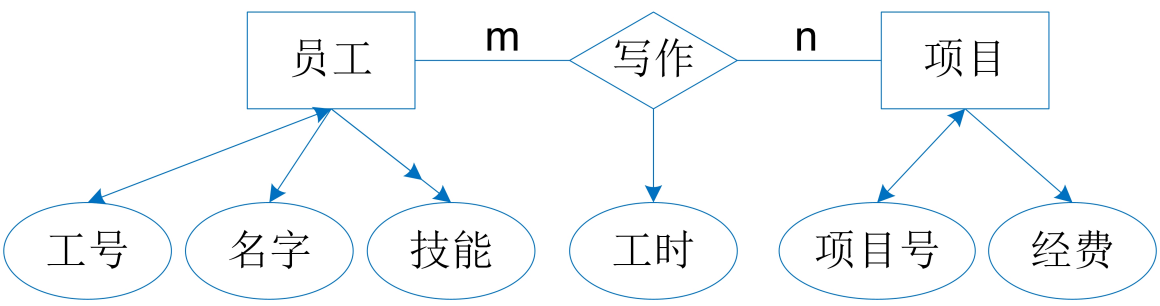


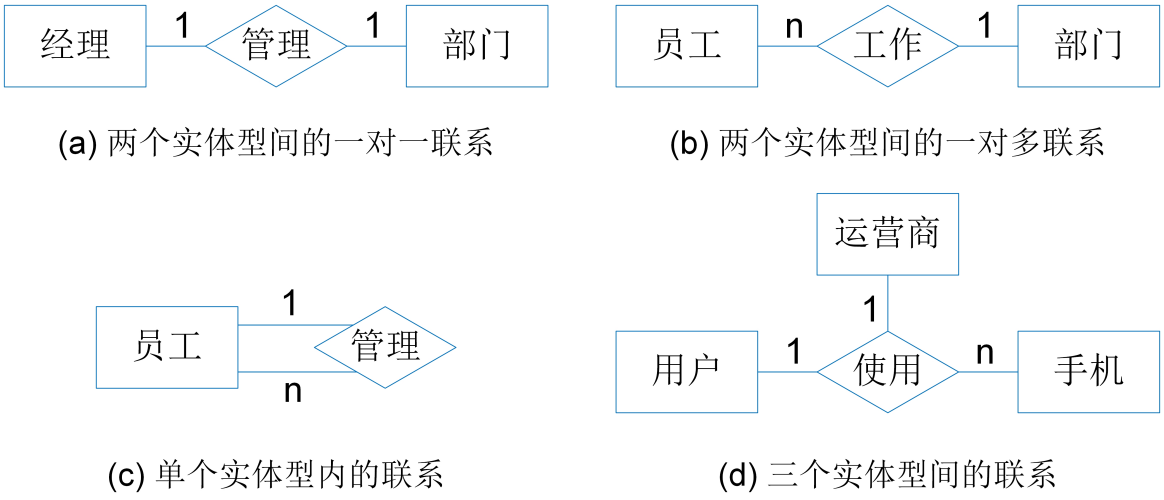
图 6.1 员工与项目E-R图

图6.1的E-R图描述了员工、项目两类实体、属性以及它们之间的联系。E-R图中的矩形表示实体型、椭圆表示属性、菱形表示联系。员工实体包含工号、名字、技能三个属性，项目实体包含项目号、经费两个属性，参与联系包含工时一个属性。实体的属性又可以分为唯一属性、单值属性和多值属性。

- 唯一属性是指能够唯一识别一个实体的属性，如员工的工号，项目的项目号，通过工号可以唯一确定一名员工，通过项目号可以唯一确定一个项目；
- 单值属性是指只有一个值的属性，如员工的名字，项目的经费和参与联系的工时，一个员工只能有一个名字，一个项目只能有一个经费值，员工参与某个项目的工时也只会有一个；
- 多值属性是指存在多个值的属性，如员工的技能，一个员工可以有多项技能。

在E-R图中唯一属性用双向箭头表示，单值属性用一个单向的箭头表示，多值属性用一个单向的双箭头表示。

实体与实体之间的联系有一对一联系（1:1）、一对多联系（1:n）和多对多联系(m:n)三种。员工与项目之间的参与联系是多对多的联系，即一个员工可以参与多个项目，一个项目可以由多个员工参与。在现实世界中，单个实体内部，两个实体之间，两个以上的实体之间都可能存在一对一、一对多和多对多联系。



(a) 两个实体型间的一对一联系　(b) 两个实体型间的一对多联系

(c) 单个实体型内的联系　(d) 三个实体型间的联系

图 6.2 E-R示例图

图6.2给出了多种联系的示例。图6.2(a)中经理实体和部门实体之间具有一对一的管理联系，即一名经理只负责管理一个部门，而一个部门只能被一名经理管理；图6.2(b)中员工实体和部门实体之间具有多对一的从属联系，即某一部门中包含若干名员工，而一名员工只能属于一个部门；图6.2(c)中员工实体内部具有管理和被管理的一对多联系，即某一员工（干部）管理若干名员工，而一名员工只能被一名干部员工直接领导；图6.2(d)描述了用户实体、手机实体和运营商实体之间的联系，即一个用户可以使用若干个手机，一个运营商可以服务若干个手机，但是一个手机只属于某一用户，只能选择某一个运营商。

一般地，把参与联系的实体的数目称为联系的度。单个实体之间的联系度为1，也成一元联系；两个实体之间的联系度为2，称为二元联系；三个实体之间的联系度为3，称为三元联系；N个实体之间的联系度为N，称为N元联系。

此外，E-R图还可以描述一些更加复杂的联系。比如，实体与实体之间的父类与子类联系，实体之间的依附联系。在网上购物系统中，商品可以分为电子产品、日用品、化妆品等。商品是父类实体，电子产品、日用品和化妆品是商品的子类实体，这种父类-子类联系称为ISA联系。在现实世界中，城市的存在依附于国家，每座楼的房间依附于楼房。所以，城市与国家，房间与楼房之间具有依附联系。对这两种联系感兴趣的读者可以参考其他教材资料。

逻辑结构设计

关系数据库的逻辑结构设计是将E-R图转换成关系模型，也就是将E-R图中的实体、实体间的联系转换为关系模式，并确定这些关系模式的属性和主码。关系模式的设计规则有：

- 实体转换规则：一个实体转换为一个关系模式（二维表）；
- 属性转换规则一：一个实体的唯一属性和单值属性构成该关系模式的属性，其中唯一属性是关系模式的主码；
- 属性转换规则二：一个实体的多值属性转换成一个关系模式，实体的唯一属性和多值属性是该关系模式的属性，也是关系模式的主码；
- 联系转换规则：（1）实体之间的1:1联系与任意一端的关系模式合并，合并端的关系模式中需要加入联系自身的属性和另一端关系模式的主码属性；（2）实体之间的1:n联系与n端的关系模式合并，n端的关系模式中加入联系自身属性和另一端模式的主码属性；（3）实体之间的m:n联系转换为一个关系模式，联系自身属性以及联系两端实体的唯一属性组成该关系模式的属性。

根据上述规则，图6.1中员工和项目的E-R图可以转换成以下四个关系表。

```
员工与项目的关系模式设计：
员工表：Employee(Id, Name), Id是主码;
项目表：Project(Pid, Budget), Pid是主码;
技能表：Skills(Id, Skill), Id和Skill是主码;
工作表：Work(Id, Pid, Time), Id和Pid是主码;
```

首先，根据实体转换规则和属性转换规则一将员工实体转换为员工表Employee(Id, Name)，Id是员工表的主码，项目实体转换为项目表Project(Pid, Budget)，Pid是项目表的主码。两个表中只包含了实体的唯一性属性和单值属性；然后根据属性转换规则二将员工的多值属性“技能”转换为技能表Skills(Id, Skill)，Id和Skill一起构成技能表的主码；最后根据联系转换规则将员工与项目之间的多对多联系转换为工作表Work(Id, Pid, Time)，Id和Pid一起构成工作表的主码。

博客网站的关系模式设计

回顾第三章3.2小节中提到的博客网站实例。在博客网站中，网站博主们可以相互关注，形成博主与粉丝之间的关系。博主登录网站之后进入个人主页，可以查看自己的个人信息，粉丝数量，浏览已关注博主最新发布的博客简介；点击博客简介之后可以查看博客的具体内容和博客的评论内容，也可以发表评论；此外，博主可以编辑新的博客并发表。

通过需求分析，博客网站包含博主、博客和评论三类实体，博主与博主之间存在多对多的关注联系，博主与博客之间存在一对多的写作联系，博主与评论之间存在一对多的写作联系，博客与评论之间存在一对多的评价联系。博客网站的E-R图如图6.3所示。

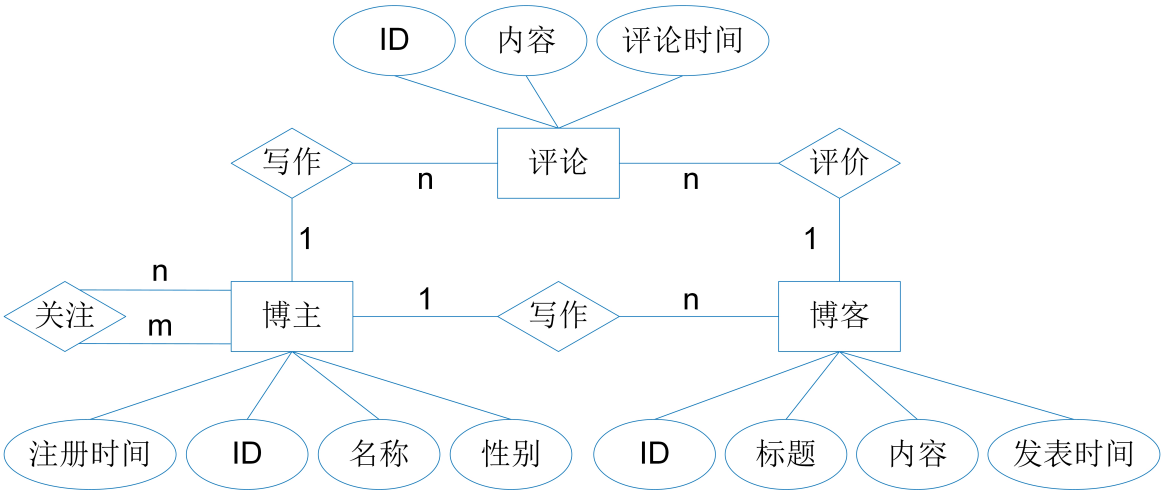


图 6.3 博客网站的E-R图

基于文档模式设计规则，博客网站E-R图可以转换为博主User和博客Doc两个文档集。评论实体comment以文档数组的形式嵌入博客文档集中，这种嵌套方式也自然地转换了博客与评论之间的一对多联系。博主与博主之间的关注联系通过博主文档集的粉丝数组fans[u_id]和关注者数组followee[u_id]进行刻画，博主与博客之间的写作联系通过在博客文档集中加入博主号u_id进行刻画，博主与评论之间的写作联系通过在评论文档集中加入博主号u_id进行刻画。

```
博客网站的文档模式设计
User { u_id, name, gender, reg_date, fans[u_id], followee[u_id] }
Doc  { d_id, title, d_content, pub_date, u_id, name,
       comment [ { c_id, c_content, com_date, u_id, name } ] }
```

基于关系模式设计规则，博客网站E-R图可以转换为博主User、博客Doc、点评Comment和关注Follow四张表。博主、博客、点评三个实体分别转换成三个关系模式，博主与博主之间的多对多关注联系转换成关系模式Follow，博主与博客之间的一对多写作联系通过在博客表中加入博主号u_id进行刻画，博主与评论之间的一对多写作联系和博客与评论之间的一对多评价联系通过在评论表中加入博主号u_id，博客号d_id进行刻画。

```
博客网站的关系模式设计
博主表：User(u_id, name, gender, reg_date),u_id是主码;
博客表：Doc(d_id,title, d_content, pub_date, u_id), d_id是主码;
评论表：Comment(c_id, c_content, com_date, u_id, d_id),  c_id是主码;
关注表：Follow(followee_uid, follower_uid), followee_uid和follower_uid是主码;
```