

# 数据库系统概论

## An Introduction to Database System

### 第七章 数据库设计





# 第7章 数据库设计

---

- 7.1 数据库设计概述
  - 7.2 需求分析
  - 7.3 概念结构设计
  - 7.4 逻辑结构设计
  - 7.5 数据库的物理设计
  - 7.6 数据库实施
  - 7.7 数据库运行与维护
-



## 7.1 数据库设计概述

---

### 7.1.1 数据库设计

- **定义：** 是指对一个给定的应用环境，构造最优的、最有效的数据库模式，建立数据库及其应用系统，使之能够高效率地存取数据，满足各种用户的应用需求。包括信息处理要求和数据操作要求。
  - **特点：**
    - 1) 三分技术、七分管理、十二分基础数据
    - 2) 数据库设计要和应用系统设计相结合，即数据库结构设计和对数据的处理设计密切结合。
-



## 7.1 数据库设计概述

---

- **7.1.2 数据库设计方法**
  - 大型数据库设计是涉及多学科的综合性的技术。要求从事数据库设计的专业人员具备多方面的技术和知识。主要包括：
    - ✓ 计算机的基础知识
    - ✓ 软件工程的原理和方法
    - ✓ 程序设计的方法和技巧
    - ✓ 数据库的基本知识
    - ✓ 数据库设计技术
    - ✓ 应用领域的知识
-



## 7.1 数据库设计概述

---

- **7.1.2 数据库设计方法（规范化设计法）**
  - 基于**E-R**模型的数据库设计方法。用**E-R**模型来设计数据库的概念模型，是数据库概念设计阶段广泛采用的方法
  - **3NF**（第三范式）的设计方法。用关系数据理论为指导来设计数据库的逻辑模型，是设计关系数据库时在**逻辑设计阶段**可以采用的一种有效的方法。
  - **ODL**（**Object Definition Language**）方法。是面向对象的数据库设计方法。用面向对象的概念和术语来说明数据结构。
-



## 7.1 数据库设计概述

---

- 数据库设计之前，首先必须选定参加设计的人员，包括：

- ✓ 系统分析人员和数据库设计人员

数据库设计的核心人员，自始至终参与数据库设计，水平决定了数据库系统的质量。

- ✓ 应用开发人员（程序员和操作员）

编制程序和准备软硬件环境

- ✓ 数据库管理员和用户代表

需求分析和数据库的运行和维护

---



# 数据库的设计分为6个阶段

---

- （1）需求分析。收集和分析用户对系统的信息需求和处理需求，得到设计系统所必须的需求信息，建立系统说明文档。
- （2）概念结构设计。概念结构设计是整个数据库设计的关键。它通过对用户的需求进行综合、归纳与抽象，形成一个独立于具体DBMS的概念模型。
- （3）逻辑结构设计。在概念模型的基础上导出一种DBMS支持的逻辑数据库模型（如关系型、网络型或层次型），该模型应满足数据库存取、一致性及运行等各方面的用户需求。



# 数据库的设计分为6个阶段


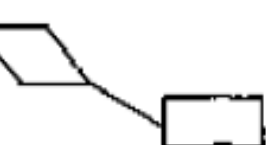
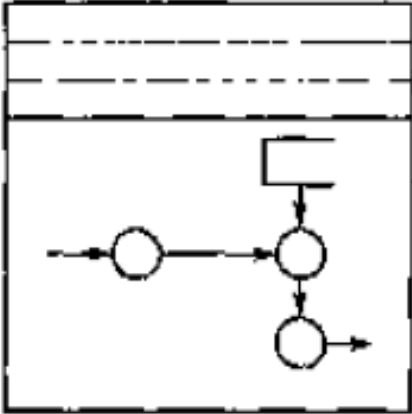

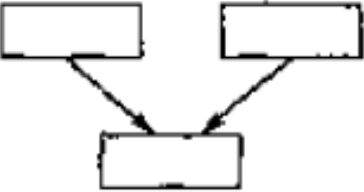
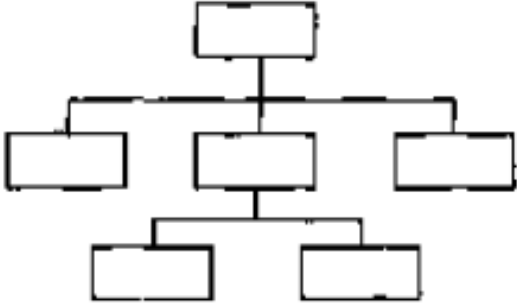
---

- （4）物理结构设计。从一个满足用户需求的已确定的逻辑模型出发，在限定的软、硬件环境下，利用DBMS提供的各种手段设计数据库的内模式，即设计数据的存储结构和存取方法。
  - （5）数据库实施。运用DBMS提供的数据语言及宿主语言，根据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，并进行试运行。
  - （6）数据库运行和维护。
-





# 数据库设计各个阶段的设计描述

设计阶段	设计描述	
	数据	处理
需求分析	数据字典、全系统中数据项、数据流、数据存储的描述	数据流图和判定表（判定树）、数据字典中处理过程的描述
概念结构设计	<p>概念模型 (F-R图)</p>  <p>数据字典</p> 	<p>系统说明书包括:</p> <ul style="list-style-type: none"><li>① 新系统要求、方案和概图</li><li>② 反映新系统信息流的数据流图</li></ul> 
逻辑结构设计	<p>某种数据模型</p> <p>关系</p>  <p>非关系</p> 	<p>系统结构图</p> <p>(模块结构)</p> 



设计阶段	设计描述	
	数 据	处 理
物理设计	<div>存储安排</div> <div>方法选择</div> <div>存取路径建立</div> <div><div>分区 1</div><div>分区 2</div></div>	<div>模块设计</div> <div>IPO 表</div> <div><div>IPO 表...</div><div>输入:</div><div>输出:</div><div>处理:</div></div>
数据库实施阶段	<div>编写模式</div> <div>装入数据</div> <div>数据库试运行</div> <div><div>Creat .....</div><div>Load . . .</div></div>	<div>程序编码、</div> <div>编译联结、</div> <div>测试</div> <div><div>Main( )</div><div>.....</div><div>if... ..</div><div>then</div><div>.....</div><div>end</div></div>
数据库运行和维护	<div>性能监测、转储 /恢复</div> <div>数据库重组和重构</div>	<div>新旧系统转换、运行、维护 (修正性、适应性、改善性维护)</div>



## 7.2 需求分析

---

- 7.2.1 需求分析的任务
  - 7.2.2 需求分析的基本步骤
  - 7.2.3 需求分析应用实例
-



## 7.2.1 需求分析的任务

---

根据需求分析的目标，需求分析这一阶段的任务主要有两项：

- （1）确定设计范围。通过详细调查现实世界要处理的对象（组织、部门和企业等），弄清现行系统（手工系统或计算机系统）的功能划分、总体工作流程，明确用户的各种需求。
  - （2）数据收集与分析。需求分析的重点是在调查研究的基础上，获得数据库设计所必须的数据信息。
-



## 7.2.2 需求分析的基本步骤

---

- **1. 调查与初步分析用户的需求，确定系统的边界**
  - **2. 分析和表达用户的需求**
-



# 1. 调查与初步分析用户的需求，确定系统的边界

- (1) 首先调查组织机构情况。
- (2) 然后调查各部门的业务活动情况。
- (3) 在熟悉了业务活动的基础上，协助用户明确对新系统的各种要求，包括信息要求、处理要求、安全性与完整性要求，这是调查的又一个重点。
- (4) 最后对前面调查的结果进行初步分析，确定新系统的边界，确定哪些功能由计算机完成或将来由计算机完成，哪些活动由人工完成。

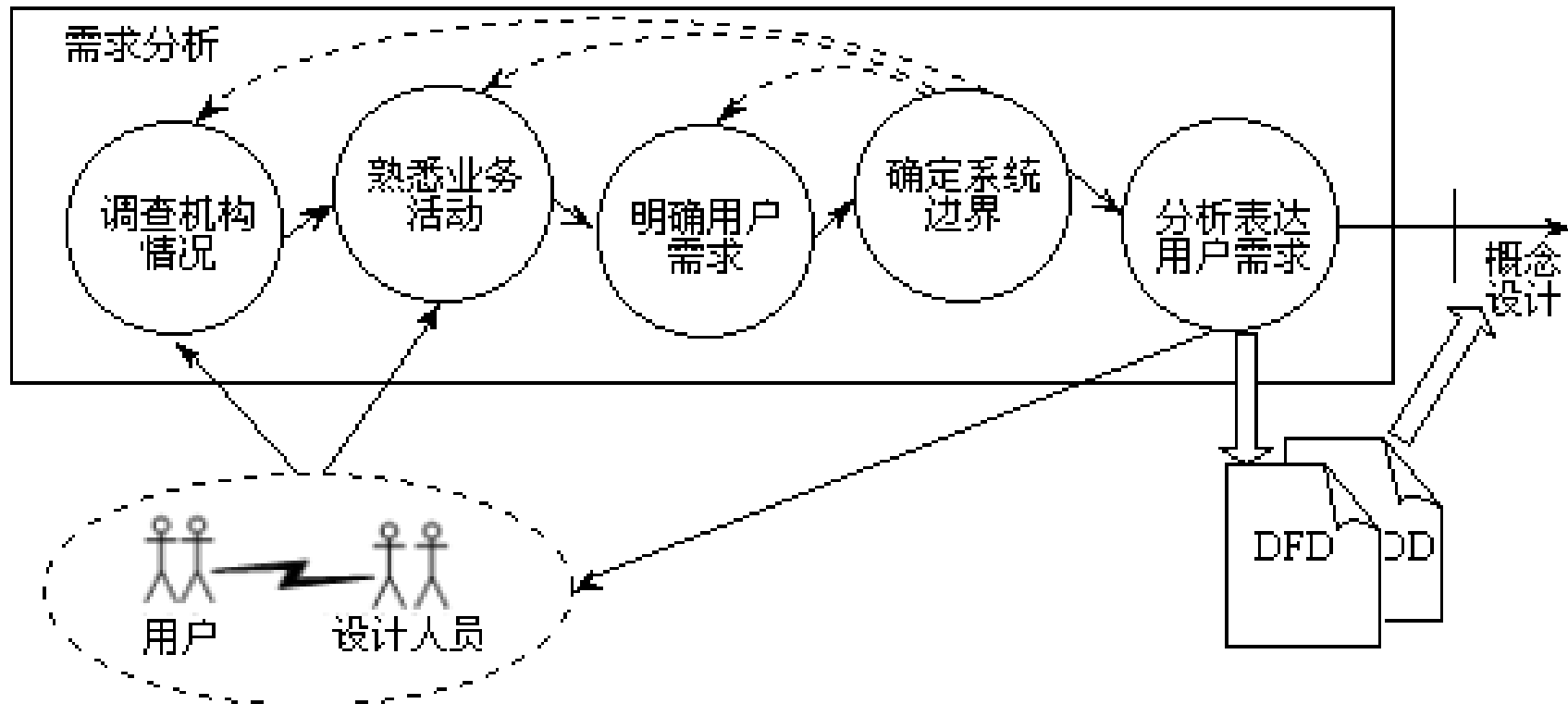


图 8-3 需求分析过程



## 2. 分析和表达用户的需求

- (1) 数据流图
- (2) 数据字典

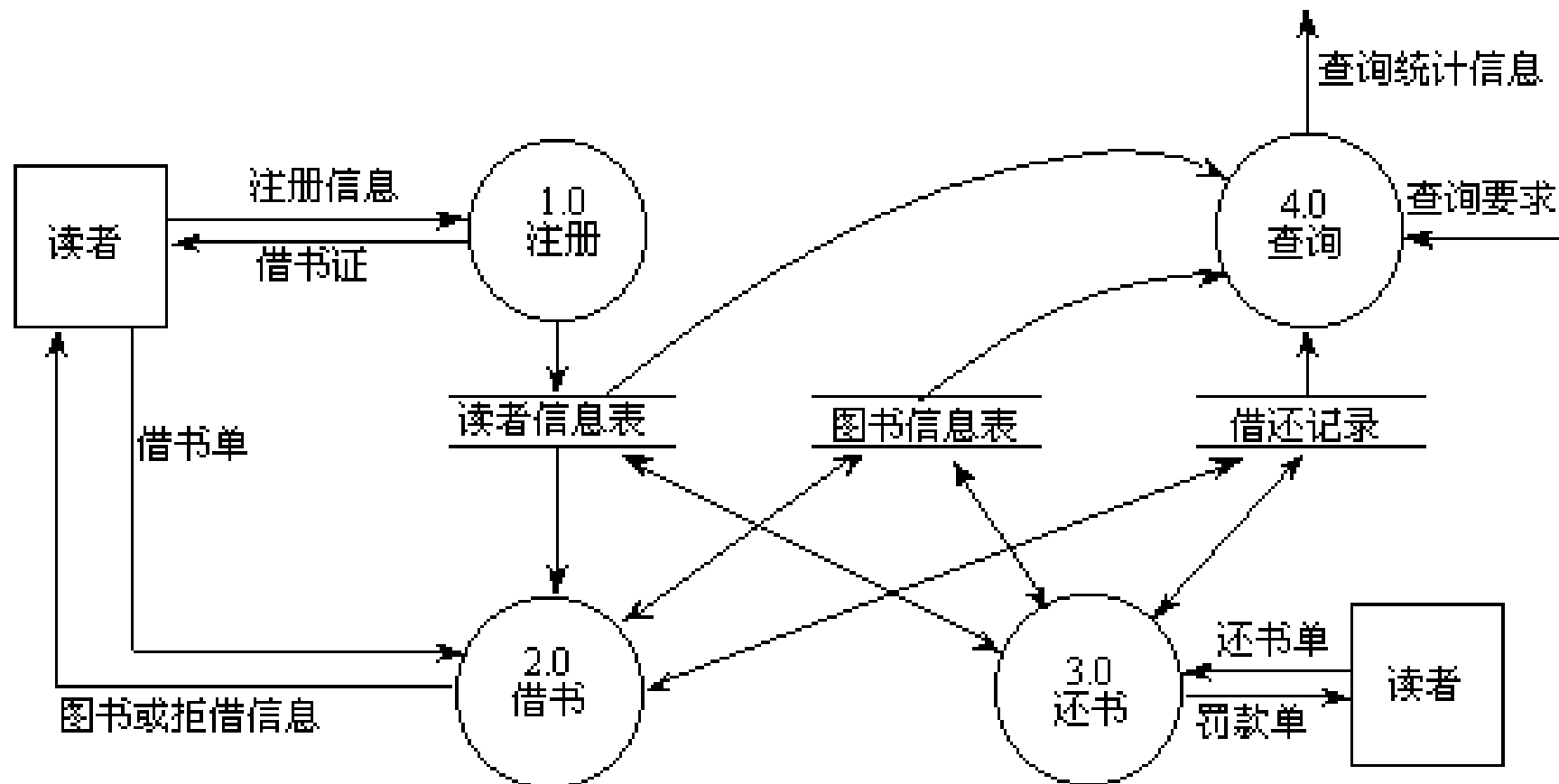


图 8-6 图书管理系统第 0 层数据流图





## 2. 数据字典

---

- (1) 数据项描述。

数据项名称：借书证号

别名：卡号

含义说明：惟一标识一个借书证

类型：字符型

长度：20



## • (2) 数据结构描述。

名称：读者类别

含义说明：定义了一个读者类别的有关信息

组成结构：类别代码+类别名称+可借阅数量+借阅天数+超期罚款额

名称：读者

含义说明：定义了一个读者的有关信息

组成结构：姓名+性别+所在部门+读者类型

名称：图书

含义说明：定义了一本图书的有关信息

组成结构：图书编号+图书名称+作者+出版社+价格



### • (3) 数据流（非数据项）说明。

数据流名称：借书单

含义：读者借书时填写的单据

来源：读者

去向：审核借书

数据流量：250份/天

组成：借书证编号+借阅日期+图书编号

数据流名称：还书单

含义：读者还书时填写的单据

来源：读者

去向：审核还书

数据流量：250份/天

组成：借书证编号+还书日期+图书编号



## (4) 数据存储说明。

数据存储名称：图书信息表

含义说明：存放图书有关信息

组成结构：图书+库存数量

说明：数量用来说明图书在仓库中的存放数

数据存储名称：读者信息表

含义说明：存放读者的注册信息

组成结构：读者+卡号+卡状态+办卡日期

说明：卡状态是指借书证当前被锁定还是正常使用

数据存储名称：借书记录

含义说明：存放读者的借书、还书信息

组成结构：卡号+书号+借书日期+还书日期

说明：要求能立即查询并修改



## (5) 处理过程说明。

---

处理过程名称：审核借书证

输入：借书证

输出：认定合格的借书证

加工逻辑：根据读者信息表和读者借书证，如果借书证在读者信息表中存在并且没有被锁定，那么借书证是有效的借书证，否则是无效的借书证。



## 7.3 概念结构设计

---

- 7.3.1 概念结构设计的方法和步骤
  - 7.3.2 局部视图设计
  - 7.3.3 视图的集成
  - 7.3.4 概念结构设计实例
-



## 7.3.2 概念结构设计的方法和步骤

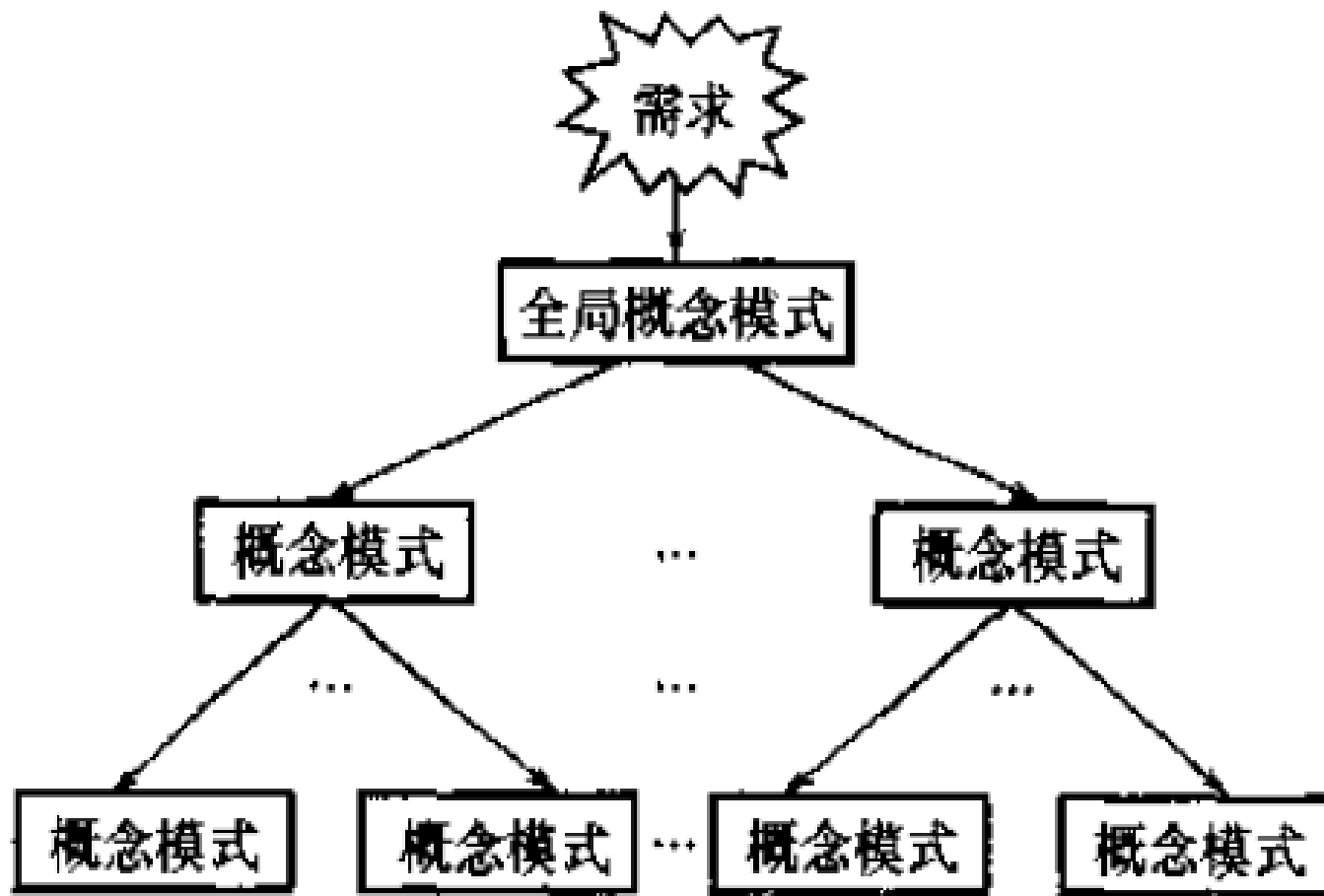
---

- 概念结构的主要特点：
  - 能真实、充分地反映现实世界，包括事物和事物之间的联系。
  - 易于理解，可以与不熟悉计算机的用户交换意见
  - 易于更改，当应用环境和应用要求改变时，容易对概念模型修改和扩充。
  - 易于向关系、网状、层次等各种数据模型转换。
-



## 7.3.2 概念结构设计的方法和步骤

- 1. 自顶向下：首先定义全局概念结构的框架，再逐步细化



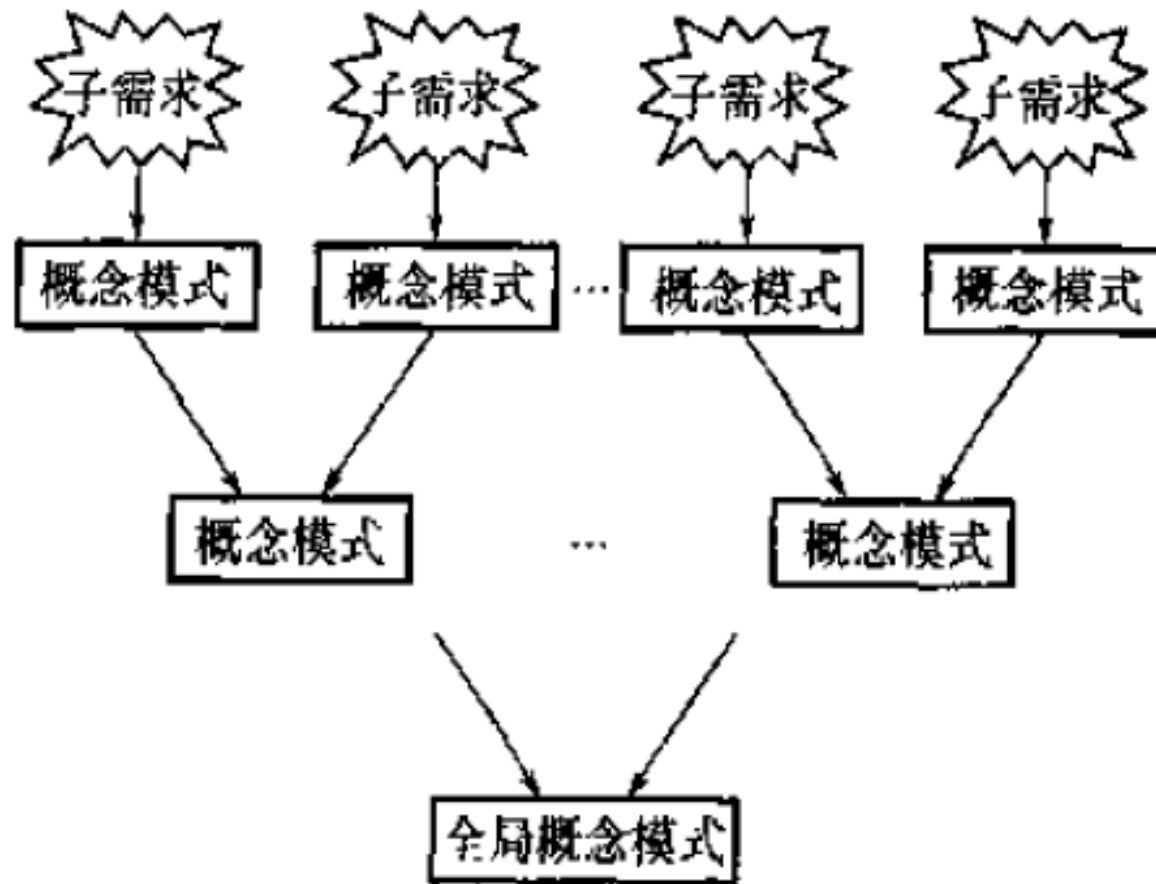
(a) 自顶向下策略





## 7.3.2 概念结构设计的方法和步骤

- 2. 自底向上：首先定义各局部应用的概念结构，然后将它们集成起来，得到全局概念结构。

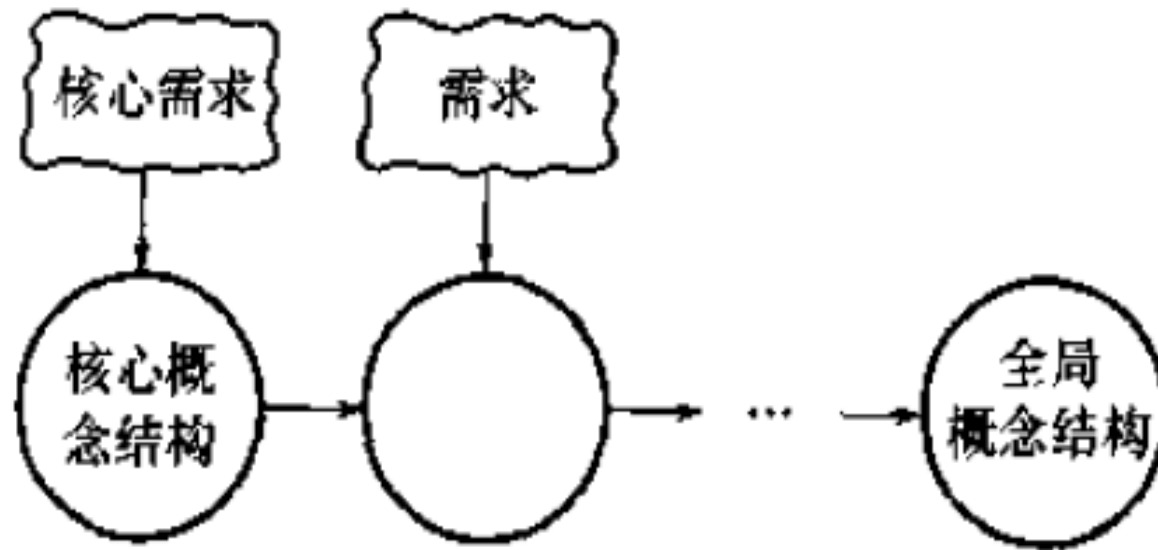


(b) 自底向上策略



## 7.3.2 概念结构设计的方法和步骤

- 3. 逐步扩张：首先定义最重要的核心概念结构，然后向外扩充，以滚雪球的方式逐步生成其他概念结构，直至总体概念结构。



(c) 逐步扩张策略

图 7.7 设计概念结构的策略



## 7.3.2 概念结构设计的方法和步骤

---

- 4. 混合策略：即将自顶向下和自底向上相结合。用自顶向下策略设计一个全局概念结构的框架，以它为骨架集成由自底向上策略中设计的各局部概念结构。
-



### 7.3.3 数据抽象

概念结构是对现实世界的一种抽象。一般分为三种：

1 分类（Classification）：定义某一类概念作为现实世界中一组对象的类型。这些对象具有共同的特性和行为。抽象了对象值和型之间的“is member of”的语义，对应E-R模型中的实体型。

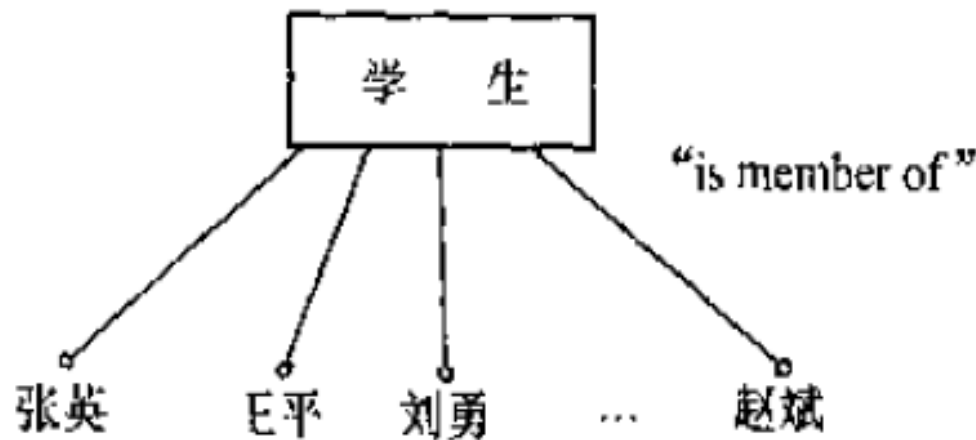


图 7.10 分类



### 7.3.3 数据抽象

2 聚集（Aggregation）：定义某一类型的组成成分，抽象了对象内部类型和成分“is part of”的语义。对应E-R模型中若干属性的聚集组成了实体型。

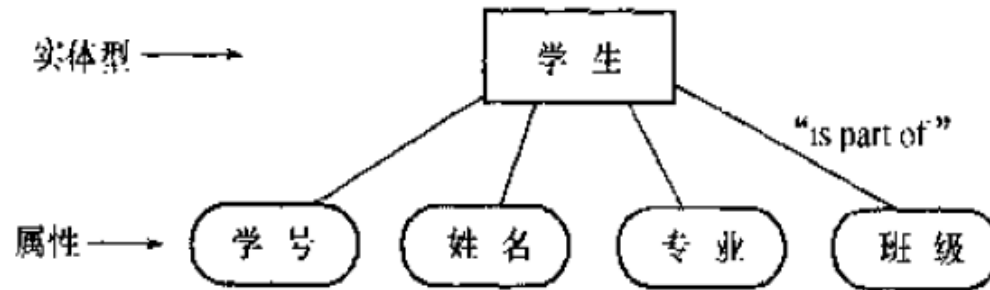


图 7.11 聚集

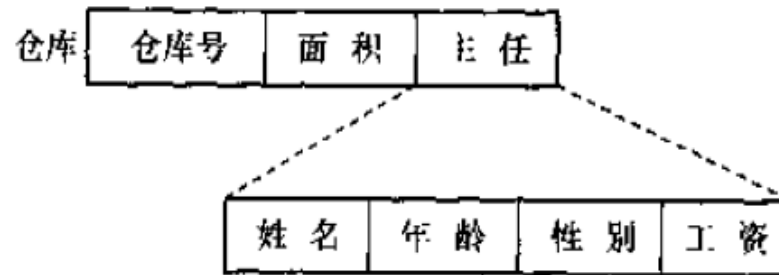


图 7.12 更复杂的聚集



### 7.3.3 数据抽象

3 概括（Generalization）：定义类型之间的一种子集联系，抽象了类型之间的“is subset of”的语义。

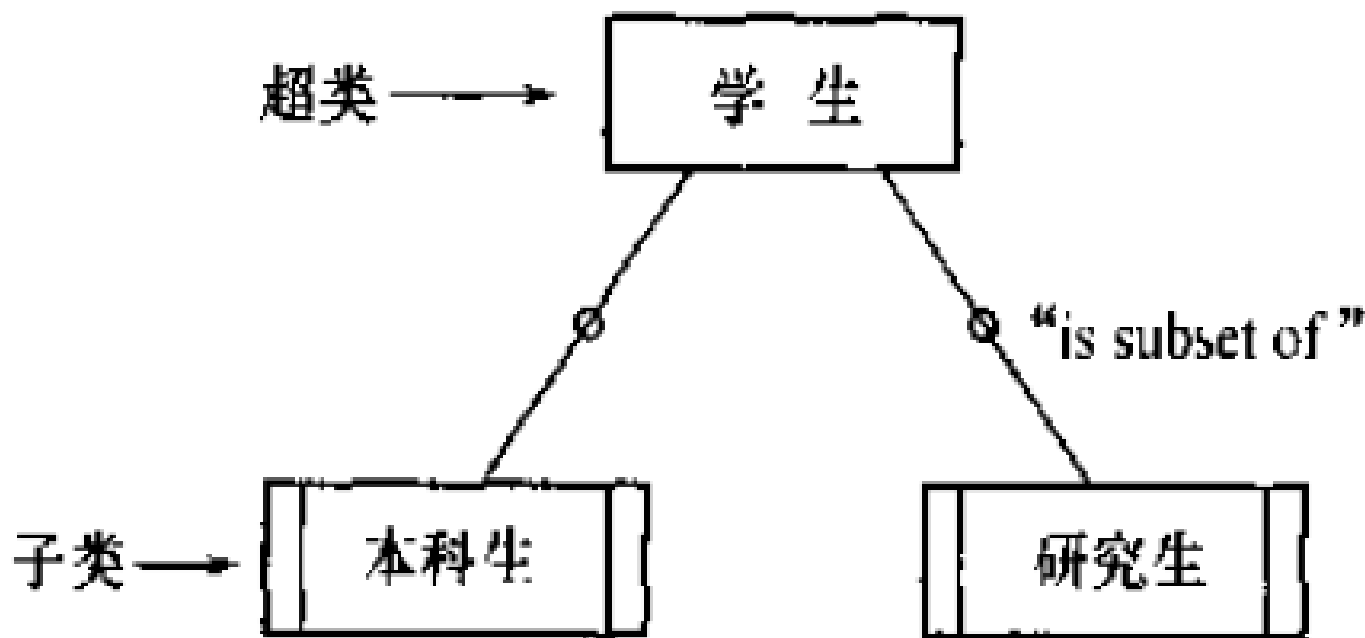


图 7.13 概括



## 7.4.1 逻辑结构设计任务和步骤

---

- 逻辑结构设计的主要目标是将概念结构转换为一个特定的**DBMS**可处理的数据模型和数据库模式。该模型必须满足数据库的存取、一致性及运行等各方面的用户需求。
-



## 7.4.1 逻辑结构设计任务和步骤

- (1) 将概念结构转换为一般的关系、网状、层次模型；
- (2) 将转换来的关系、网状、层次模型向特定 DBMS 支持下的数据模型转换；
- (3) 对数据模型进行优化。

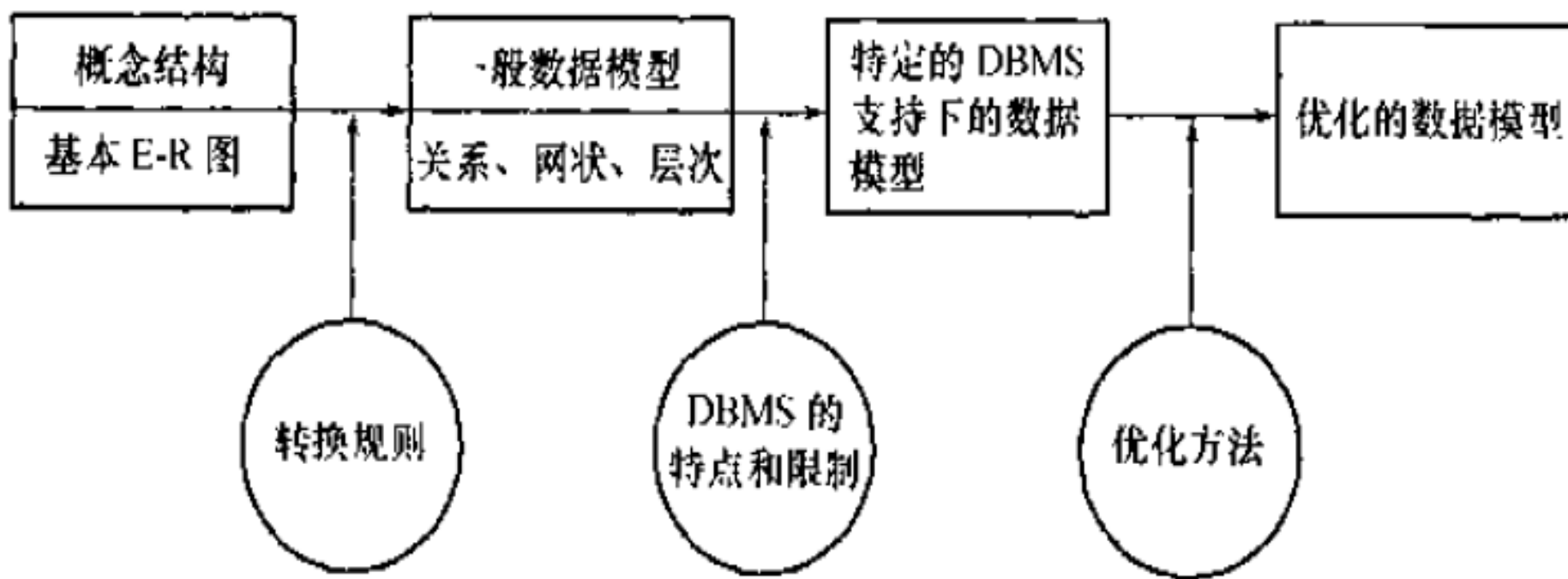


图 7.31 逻辑结构设计时的 3 个步骤





## 7.4.1 E-R图向关系模型的转换

---

- E-R图向关系模型的转换要解决的问题是如何将实体型和实体之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码。这种转换一般遵循下面的原则：**一个实体型转换为一个关系模式。实体的属性就是关系的属性，实体的码就是关系的码。**
-



# 实体间联系的转换规则

(1) 一个1:1联系可以:

✓ 转换为一个独立的关系模式

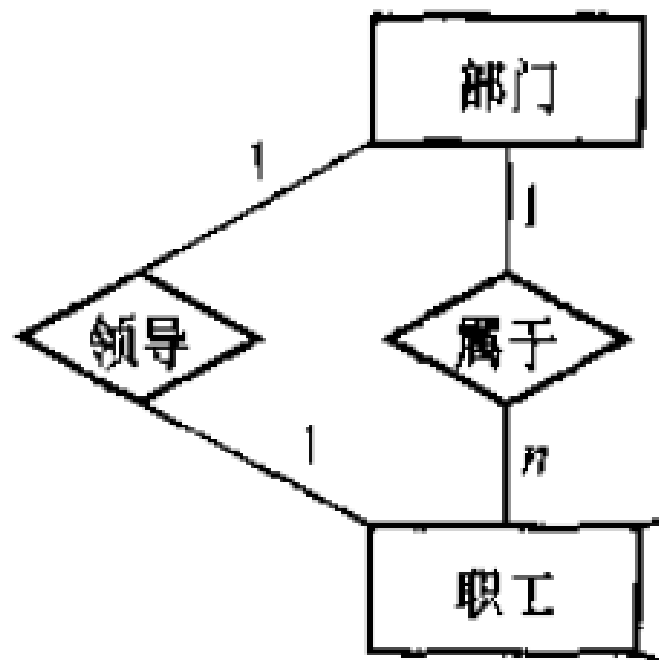
与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，每个实体的码均是该关系的候选码。

领导（部门号，经理的职工号，。。。）

✓ 也可以与任意一端所对应的关系模式合并。

在该关系模式的属性中加入另一个关系模式的码和联系本身的属性。

部门（部门号，部门名，经理的职工号，。。。）





# 实体间联系的转换规则

(2) 一个1:n联系可以:

✓ 转换为一个独立的关系模式

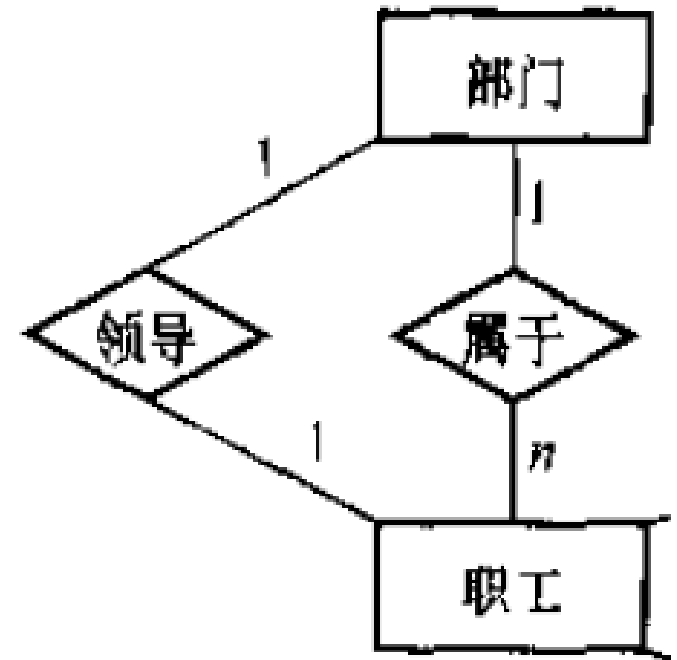
与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，关系的码为n端实体的码。

隶属（职工号，部门号，。。。）

✓ 也可以与n端所对应的关系模式合并。

在该关系模式的属性中加入n端关系模式的码和联系本身的属性。

职工（职工号，部门号，职工名，职务。。。）





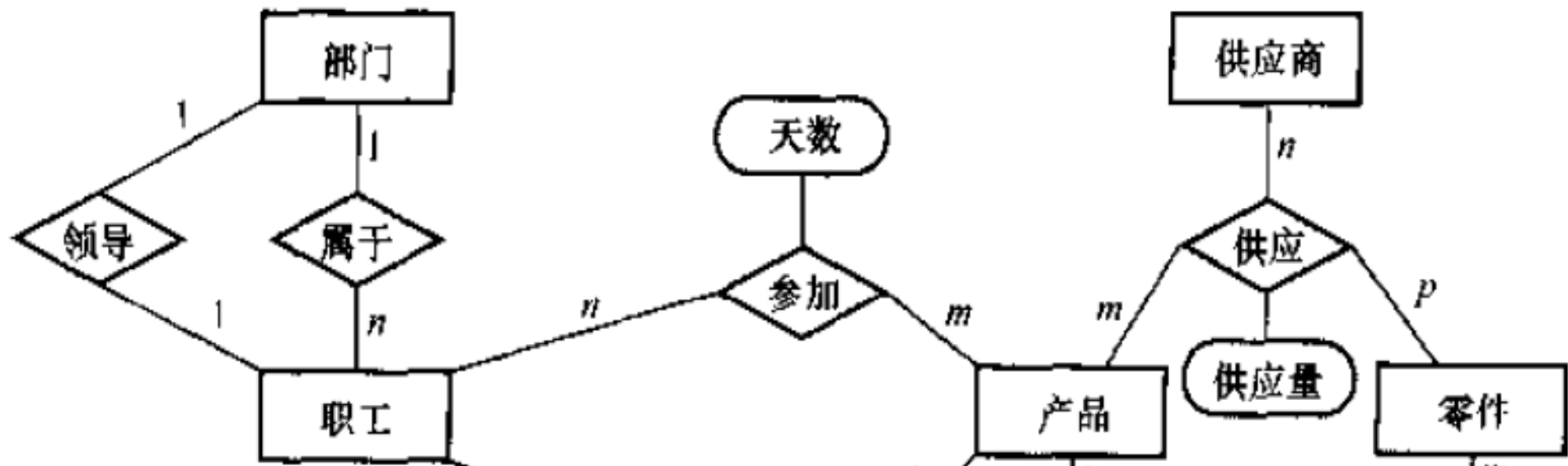
# 实体间联系的转换规则

(3) 一个 $m:n$ 联系转换为一个关系模式。

转换的方法为：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，各实体的码组成关系的码或者关系码的一部分。

职工工作（职工号，产品号，工作天数，。。。）

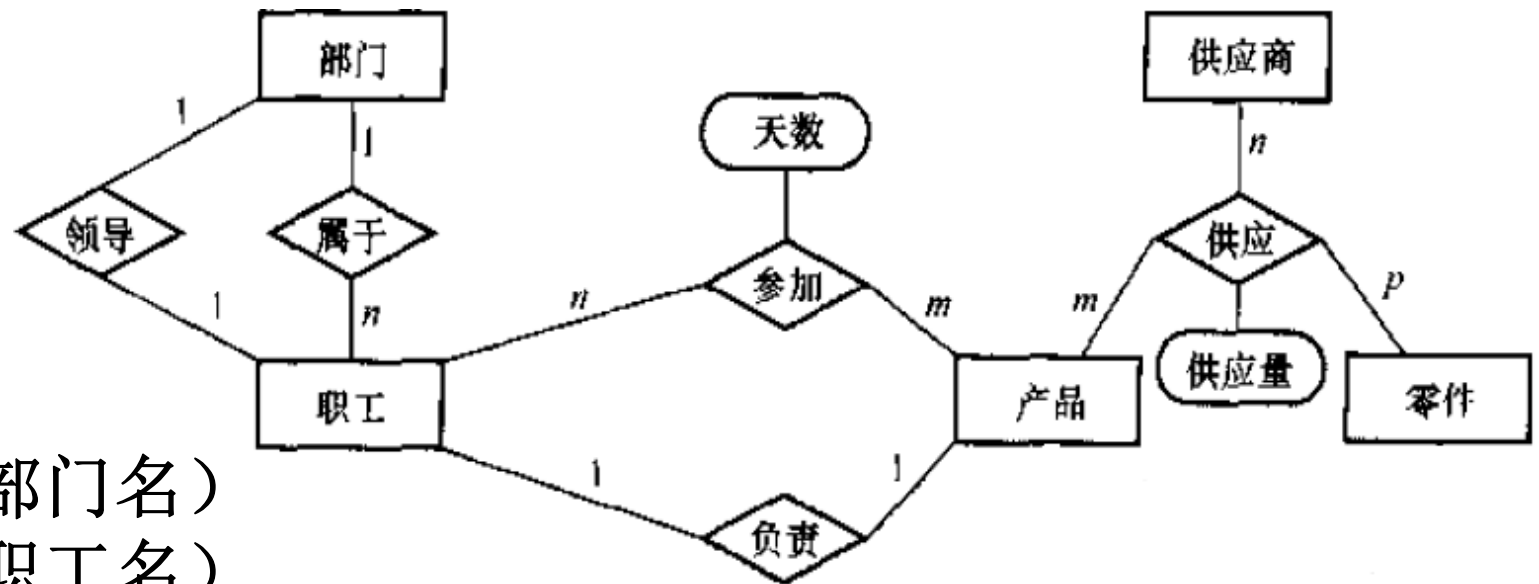
供应（产品号，供应商号，零件号，供应量，。。。）





# 实体间联系的转换规则

(4) 三个或三个以上实体间的多元联系转换为一个关系模式。：与该多元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，各实体的码组成关系的码或关系码的一部分。



部门（部门号，部门名）

职工（职工号，职工名）

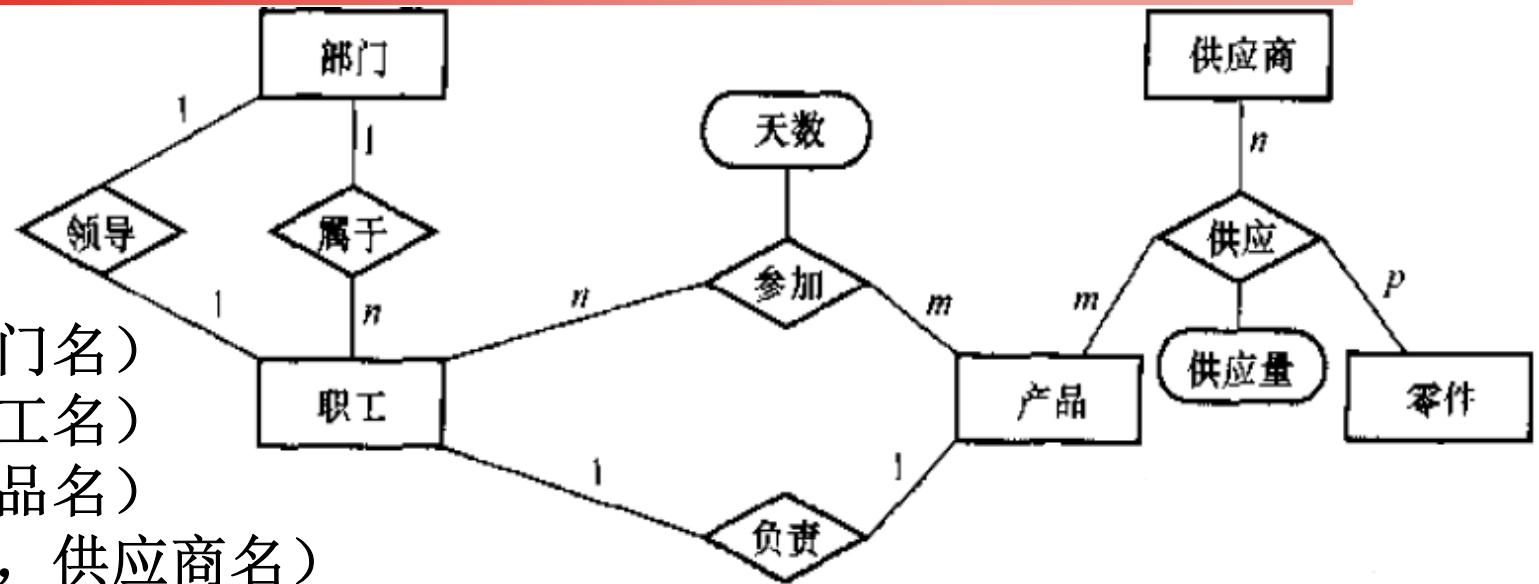
产品（产品号，产品名）

供应商（供应商号，供应商名）

零件（零件号，零件名）



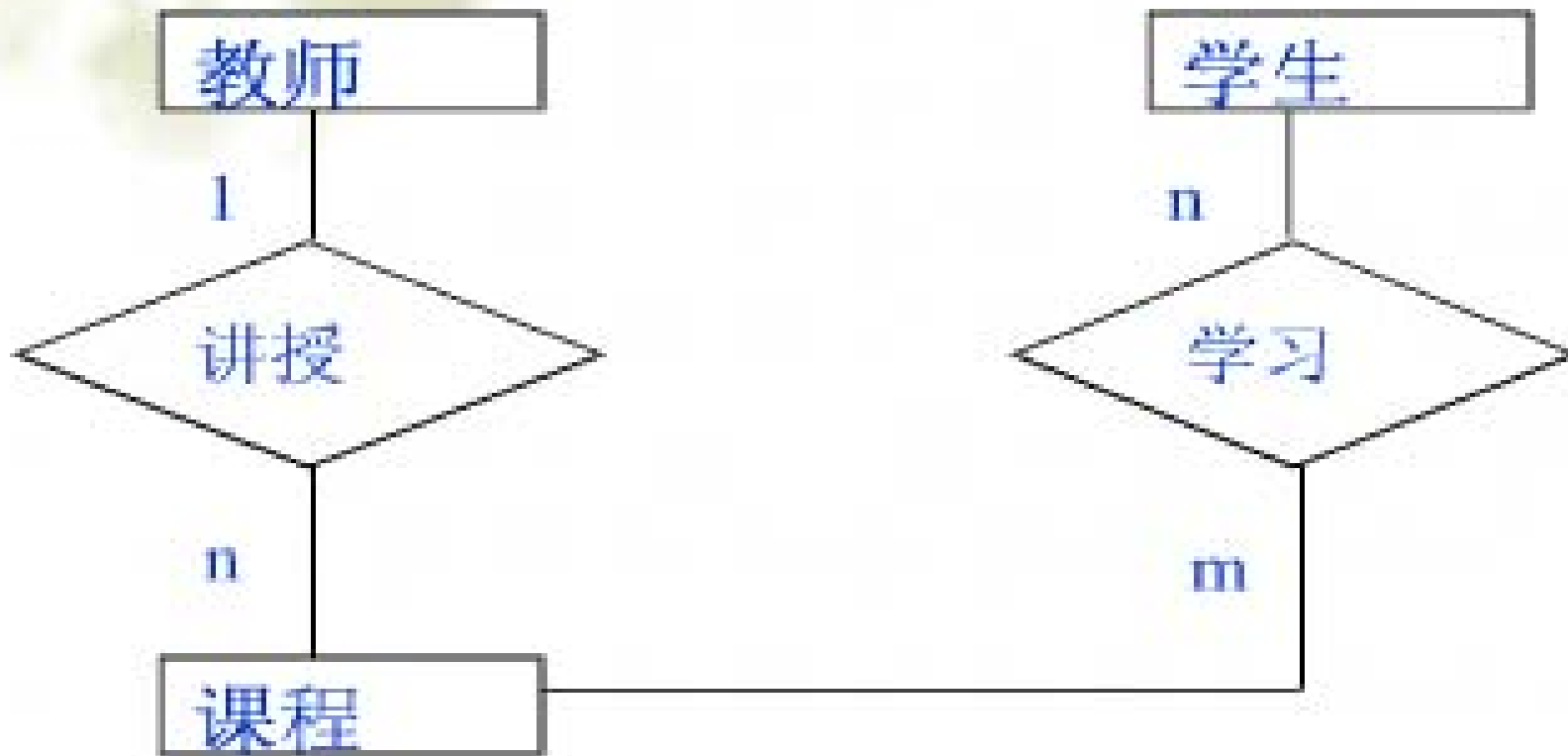
# 实体间联系的转换规则



1. 部门 (部门号, 部门名)
2. 职工 (职工号, 职工名)
3. 产品 (产品号, 产品名)
4. 供应商 (供应商号, 供应商名)
5. 零件 (零件号, 零件名)



1. 部门 (部门号, 部门名, **经理的职工号**)
2. 职工 (职工号, 职工名, **部门号**)
3. **职工工作** (职工号, 产品号, **工作天数**)
4. 产品 (产品号, 产品名, **产品组长的职工号**)
5. **供应** (产品号, 供应商号, 零件号, **供应量**)
6. 供应商 (供应商号, 供应商名)
7. 零件 (零件号, 零件名)

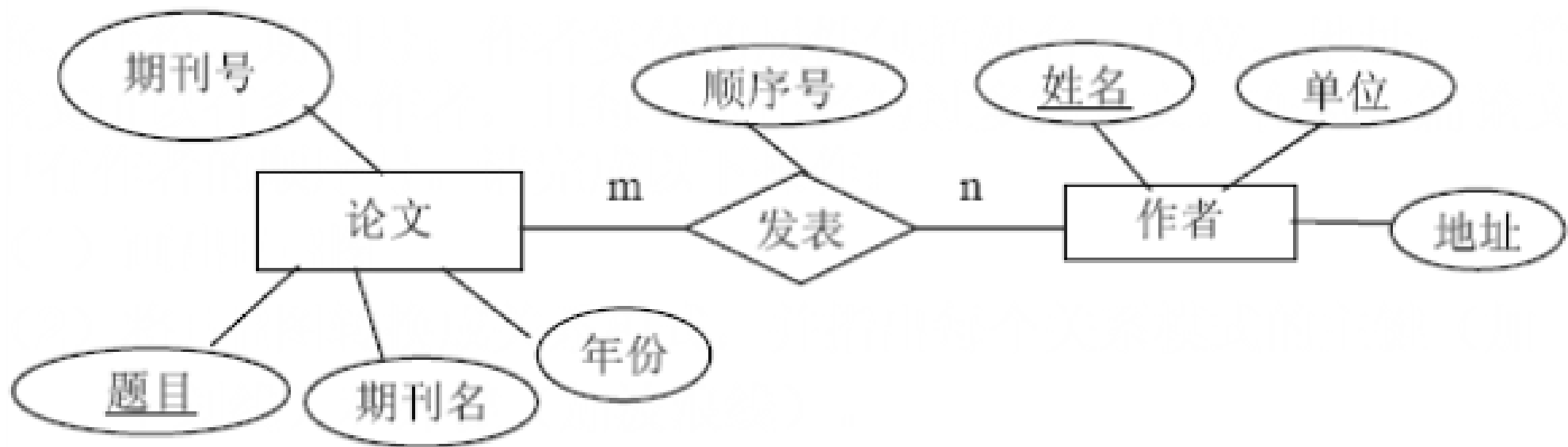


教师 (职工号#, 姓名, 年龄, 职称, 学历, 工资)

学生 (学号#, 姓名, 生日, 专业, 班级)

课程 (课程编号#, 名称, 学分, 职工号\*)

学习 (学号#, 课程编号#, 成绩)



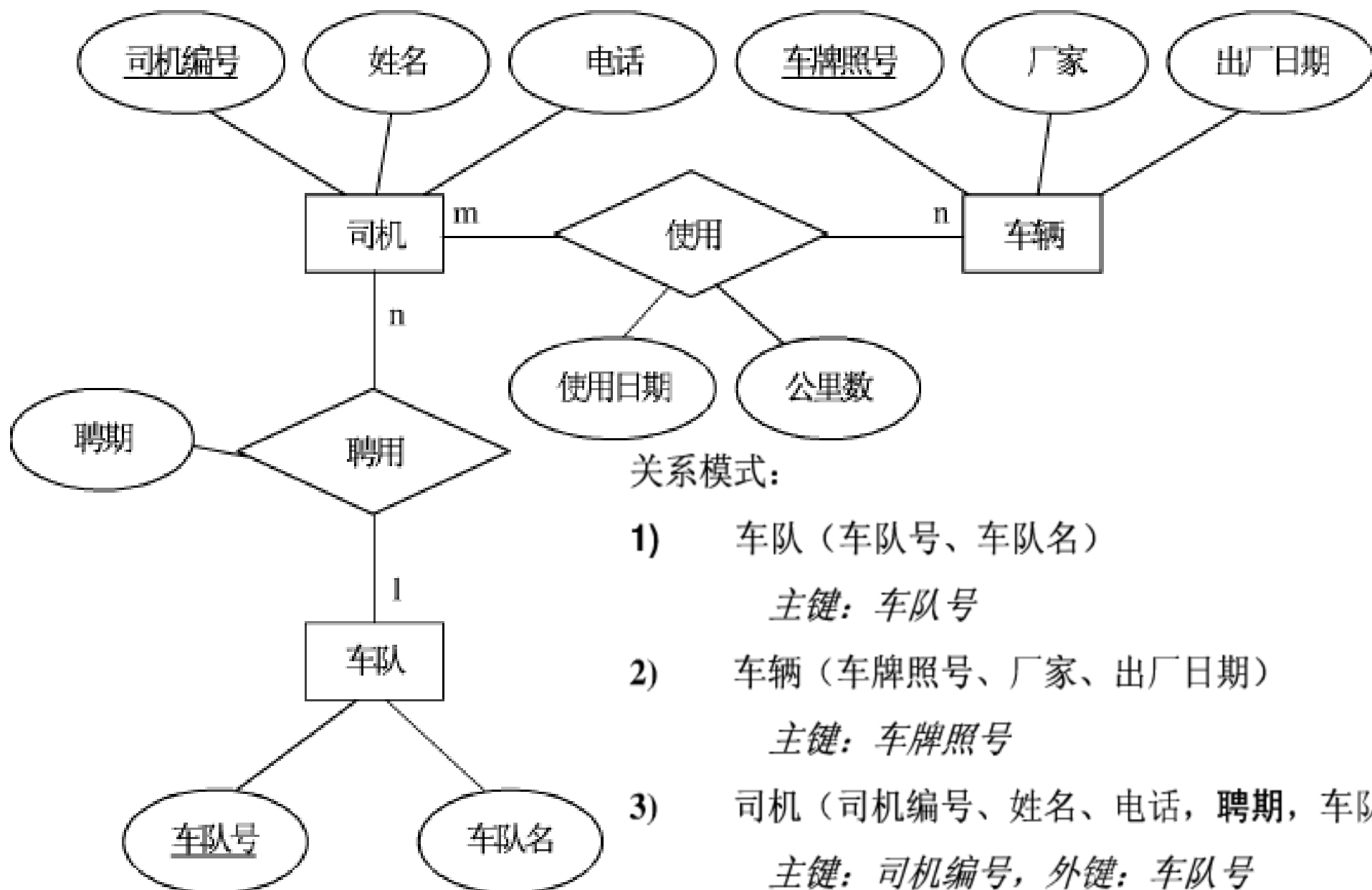
关系模式:

论文 (题目, 期刊名称, 年份, 期刊号)

作者 (姓名, 单位, 地址)

发表 (姓名, 题目, 顺序号)





关系模式:

- 1) 车队 (车队号、车队名)  
主键: 车队号
- 2) 车辆 (车牌照号、厂家、出厂日期)  
主键: 车牌照号
- 3) 司机 (司机编号、姓名、电话, 聘期, 车队号)  
主键: 司机编号, 外键: 车队号
- 4) 使用 (司机编号, 车牌照号, 使用日期, 公里数)  
主键: (司机编号, 车牌照号, 使用日期);  
外键1: 司机编号, 外键2: 车牌照号



## 7.4.2 数据模型的优化

---

- 数据库逻辑设计的结果不唯一。关系数据模型的优化通常以规范化理论为指导。
  - 1 确定数据依赖。
  - 2 根据各关系模式之间的数据依赖进行极小化处理，消除冗余的联系。
  - 3 根据数据依赖的理论对关系模式逐一进行分析，考察是否存在部分函数依赖、传递函数依赖等，确定各关系模式分别属于第几范式。
  - 4 按照需求分析阶段得到的处理要求，分析是否对某些模式进行合并或分解。
  - 5 进行必要的分解，提高数据操作的效率和存储空间的利用率。
-



## 7.4.3 设计用户外模式

---

在定义外模式时可以考虑以下因素：

- （1）使用更符合用户习惯的别名。
  - （2）对不同级别的用户定义不同的外模式，以保证数据的安全。
  - （3）简化用户对系统的使用。
-



## 7.5 数据库的物理设计

---

- 数据库在物理设备上的存储结构与存取方法称为数据库的物理结构，依赖于选定的数据库管理系统。为一个给定的逻辑数据模型选取一个最适合应用要求的物理结构的过程，即为数据库的物理设计。
  - 通常分两步：
    - 1. 确定数据库的物理结构
    - 2. 评价物理结构
-



# 1. 确定数据库的物理结构

---

- (1) 存储结构的设计。
    - 1) 顺序存储。
    - 2) 散列存储。
    - 3) 索引存储。
  - (2) 存取方法设计。
  - (3) 存放位置的设计。
-



## 2. 评价物理结构

---

- 评价物理数据库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、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。
-



## 7.6 数据库的实施与维护

---

- (1) 数据库实施阶段：数据的载入与应用程序编码和调试。
  - (2) 数据库的试运行
  - (3) 数据库的运行与维护
    - ✓ 数据库的转储和恢复。
    - ✓ 数据库安全性、完整性控制。
    - ✓ 数据库性能的监督、分析和改进。
    - ✓ 数据库的重组和重构造。
-



## 小结

---

- 本章介绍了数据库设计的全过程。设计一个数据库应用系统需要经历需求分析、概念设计、逻辑结构设计、物理设计、实施和运行维护六个阶段。
  - 概念结构设计用于设计某个企业或组织所关心的信息结构，是对现实世界的第一层抽象，它独立于机器特点，独立于具体的数据库管理系统，它用**E-R**模型来描述。逻辑设计是将概念设计的结果**E-R**模型转换为数据的组织模型，对于关系数据库来说，是转换为关系表。
-