



计算机科学与技术学院

数据库原理

王兴梅

计算机科学与技术学院

Email: wangxingmei@hrbeu.edu.cn





第七章 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
- 7.8 小结



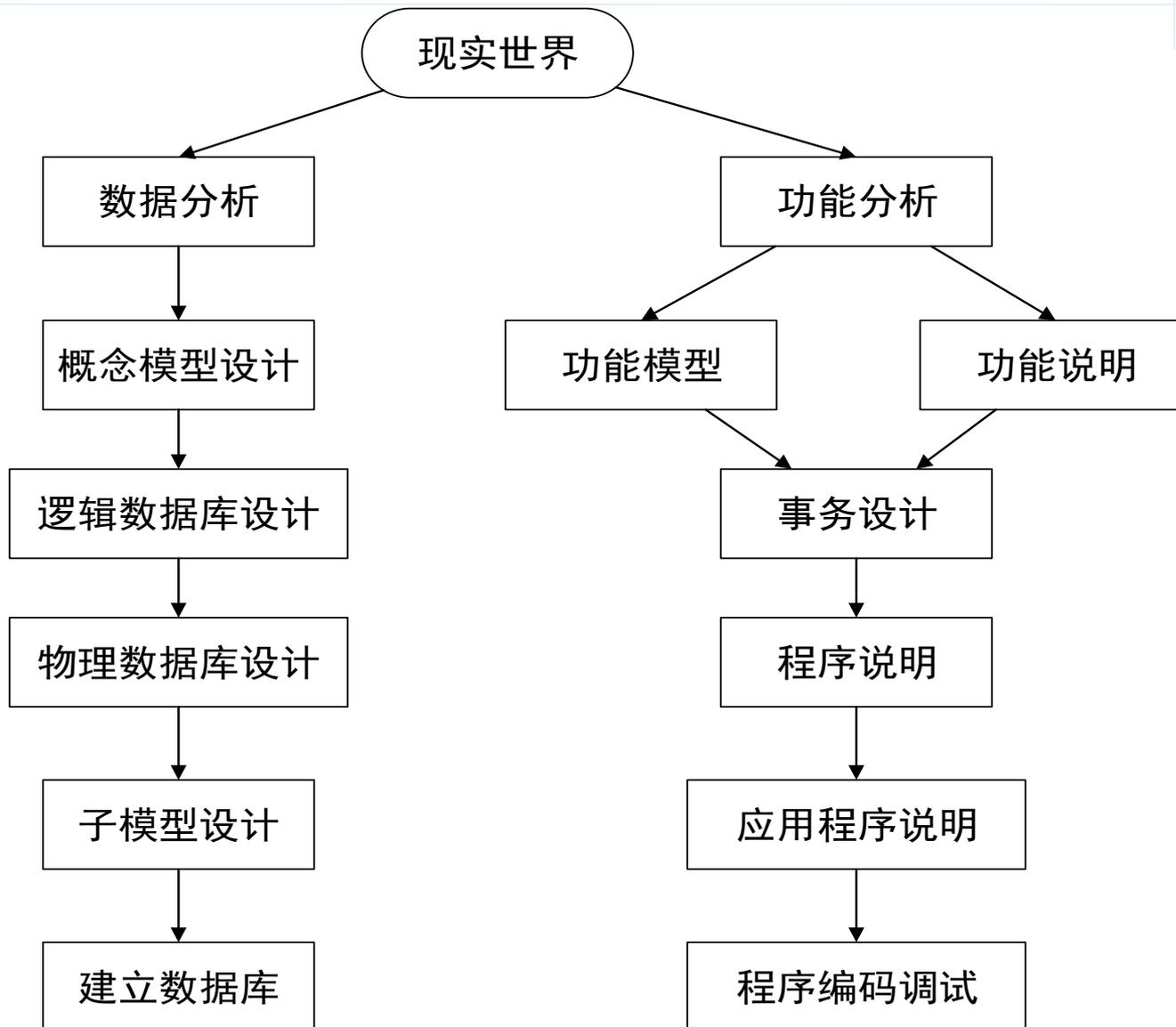
数据库设计概述（续）

- 什么是数据库设计
 - 数据库设计是指对于一个给定的应用环境，构造最优的数据库模式，建立数据库及其应用系统，使之能够有效地存储数据，满足各种用户的应用需求（信息要求和处理要求）



数据库设计概述（续）

- 数据库是信息系统的核心和基础
 - 把信息系统中大量的数据按一定的模型组织起来
 - 提供存储、维护、检索数据的功能
 - 使信息系统可以方便、及时、准确地从数据库中获得所需的信息
- 数据库是信息系统的各个部分能否紧密地结合在一起以及如何结合的关键所在
- 数据库设计是信息系统开发和建设的重要组成部分





数据库设计的基本步骤

一、数据库设计的准备工作

选定参加设计的人员

- 1.数据库分析设计人员
- 2.用户
- 3.程序员
- 4.操作员



数据库设计的基本步骤（续）

二、数据库设计的过程(六个阶段)

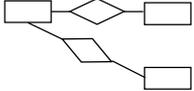
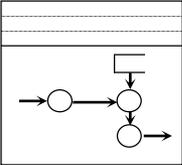
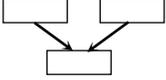
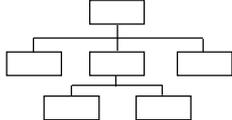
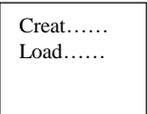
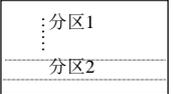
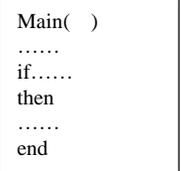
- 1.需求分析阶段
- 2.概念结构设计阶段
- 3.逻辑结构设计阶段
- 4.数据库物理设计阶段
- 5.数据库实施阶段
- 6.数据库运行和维护阶段



数据库设计的基本步骤（续）

设计一个完善的数据库应用系统往往是上述六个阶段的不断反复。



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	概念模型（E-R图）  数据字典	系统说明书包括： ①新系统要求、方案和概图 ②反映新系统信息流的数据流图 
逻辑结构设计	某种数据模型 关系  非关系 	系统结构图（模块结构） 
物理设计	存储安排 方法选择 存取路径建立 	模块设计 IPO表 
实施阶段	编写模式 装入数据 数据库试运行 	程序编码、编译连接、测试 
运行、维护	性能监测、转储/恢复 数据库重组和重构	新旧系统转换、运行、维护（修正性、适应性、改善性维护）



第七章 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
- 7.8 小结



需求分析

- 需求分析就是分析用户的需要与要求
 - 需求分析是设计数据库的起点
 - 需求分析的结果是否准确地反映了用户的实际要求，将直接影响到后面各个阶段的设计，并影响到设计结果是否合理和实用

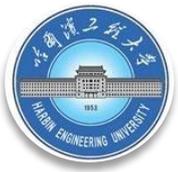


7.2 需求分析

7.2.1 需求分析的任务

7.2.2 需求分析的方法

7.2.3 数据字典



一、需求分析的任务

- 通过详细调查现实世界要处理的对象（组织、部门、企业等），充分了解原系统（手工系统或计算机系统）工作概况，明确用户的各种需求
- 在此基础上确定新系统的功能。新系统必须充分考虑今后可能的扩充和改变，不能仅仅按当前应用需求来设计数据库



二、需求分析的重点

- 需求分析的重点是调查、收集与分析用户在数据管理中的信息要求、处理要求、安全性与完整性要求。
- 信息要求
 - 用户需要从数据库中获得信息的内容与性质
 - 由用户的信息要求可以导出数据要求，即在数据库中需要存储哪些数据



三、需求分析的难点

- 确定用户最终需求的难点
 - 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，因此无法一下子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。
 - 设计人员缺少用户的专业知识，不易理解用户的真正需求，甚至误解用户的需求。
 - 新的硬件、软件技术的出现也会使用户需求发生变化。



7.2.2 需求分析的方法

- 调查清楚用户的实际需求并进行初步分析
- 与用户达成共识
- 进一步分析与表达这些需求



7.2 需求分析

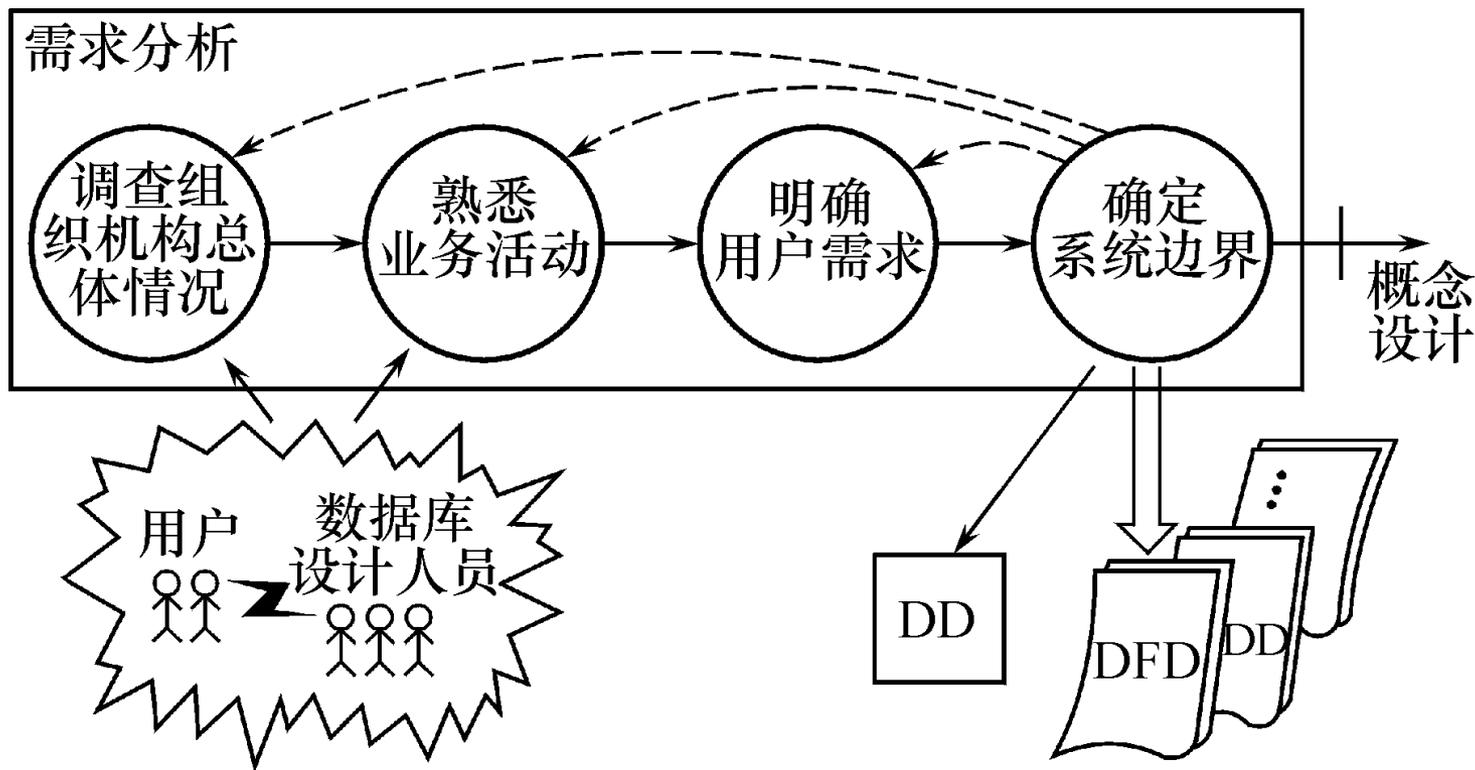
7.2.1 需求分析的任务

7.2.2 需求分析的方法

7.2.3 数据字典



需求分析





需求分析

实例：假设我们要开发一个学校管理系统。

1. 经过可行性分析和初步需求调查，抽象出该系统最高层数据流图，该系统由教师管理子系统、学生管理子系统、后勤管理子系统组成，每个子系统分别配备一个开发小组。
2. 进一步细化各个子系统。

其中学生管理子系统开发小组通过进行进一步的需求调查，明确了该子系统的主要功能是进行学籍管理和课程管理，包括学生报到、入学、毕业的管理，学生上课情况的管理。通过详细的信息流程分析和数据收集后，他们生成了该子系统的数据流图。



7.2 需求分析

7.2.1 需求分析的任务

7.2.2 需求分析的方法

7.2.3 数据字典



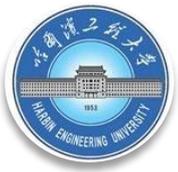
一、数据字典的用途

- 数据字典是各类数据描述的集合
- 数据字典是进行详细的数据收集和数据分析所获得的主要结果
- 数据字典在数据库设计中占有很重要的地位



二、数据字典的内容

- 数据字典的内容
 - 数据项
 - 数据结构
 - 数据流
 - 数据存储
 - 处理过程
- 数据项是数据的最小组成单位
- 若干个数据项可以组成一个数据结构
- 数据字典通过对数据项和数据结构的定义来描述数据流、数据存储的逻辑内容。



1. 数据项

- 数据项是不可再分的数据单位
- 对数据项的描述

数据项描述 = { 数据项名, 数据项含义说明,
别名, 数据类型, 长度, 取值范围,
取值含义, 与其他数据项的逻辑关系 }

- 取值范围、与其他数据项的逻辑关系定义了数据的完整性约束条件



2. 数据结构

- 数据结构反映了数据之间的组合关系。
- 一个数据结构可以由若干个数据项组成，也可以由若干个数据结构组成，或由若干个数据项和数据结构混合组成。
- 对数据结构的描述
数据结构描述 = { 数据结构名, 含义说明,
组成: { 数据项或数据结构 } }



3. 数据流

- 数据流是数据结构在系统内传输的路径。
- 对数据流的描述

数据流描述 = { 数据流名, 说明, 数据流来源, 数据流去向, 组成: { 数据结构 }, 平均流量, 高峰期流量 }

- 数据流来源是说明该数据流来自哪个过程
- 数据流去向是说明该数据流将到哪个过程去
- 平均流量是指在单位时间（每天、每周、每月等）里的传输次数
- 高峰期流量则是指在高峰时期的数据流量

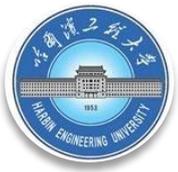


4. 数据存储

- 数据存储是数据结构停留或保存的地方，也是数据流的来源和去向之一。
- 对数据存储的描述

数据存储描述 = { 数据存储名, 说明, 编号,
流入的数据流, 流出的数据流,
组成: { 数据结构 }, 数据量, 存取方式 }

- 流入的数据流: 指出数据来源
- 流出的数据流: 指出数据去向
- 数据量: 每次存取多少数据, 每天 (或每小时、每周等) 存取几次等信息
- 存取方法: 批处理 / 联机处理; 检索 / 更新; 顺序检索 / 随机检索



5. 处理过程

- 处理过程的具体处理逻辑一般用判定表或判定树来描述。数据字典中只需要描述处理过程的说明性信息
- 处理过程说明性信息的描述
处理过程描述 = { 处理过程名, 说明,
 输入: { 数据流 }, 输出: { 数据流 },
 处理: { 简要说明 } }
- 简要说明: 主要说明该处理过程的功能及处理要求
 - 功能: 该处理过程用来做什么
 - 处理要求: 处理频度要求 (如单位时间里处理多少事务, 多少数据量); 响应时间要求等
 - 处理要求是后面物理设计的输入及性能评价的标准



第七章 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
- 7.8 小结



7.3 概念结构设计

7.3.1 概念结构设计概述

7.3.2 概念结构设计的方法与步骤

7.3.3 数据抽象与局部视图设计

7.3.4 视图的集成



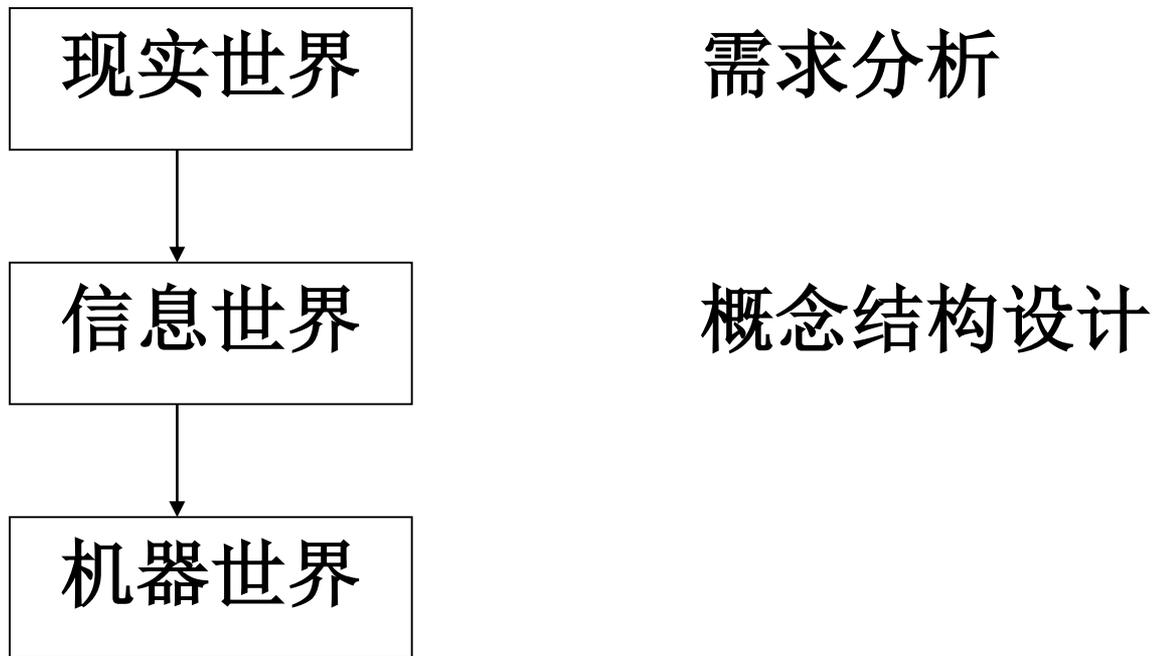
7.3.1 概念结构

■ 什么是概念结构设计

- 需求分析阶段描述的用户应用需求是现实世界的
具体需求
- 将需求分析得到的用户需求抽象为信息结构即概
念模型的过程就是概念结构设计
- 概念结构是各种数据模型的共同基础，它比数据
模型更独立于机器、更抽象，从而更加稳定。
- 概念结构设计是整个数据库设计的关键



概念结构（续）





概念结构（续）

- 概念结构设计的特点
 - (1) 能真实、充分地反映现实世界，包括事物和事物之间的联系，能满足用户对数据的处理要求。是对现实世界的一个真实模型。
 - (2) 易于理解，从而可以用它和不熟悉计算机的用户交换意见，用户的积极参与是数据库的设计成功的关键。
 - (3) 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。
 - (4) 易于向关系、网状、层次等各种数据模型转换。



概念结构（续）

- 描述概念模型的工具
 - E-R模型



7.3 概念结构设计

7.3.1 概念结构设计概述

7.3.2 概念结构设计的方法与步骤

7.3.3 数据抽象与局部视图设计

7.3.4 视图的集成



7.3.2 概念结构设计的方法与步骤

■ 设计概念结构的四类方法

➤ 自顶向下

首先定义全局概念结构的框架，然后逐步细化

➤ 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



概念结构设计的方法与步骤（续）

➤ 逐步扩张

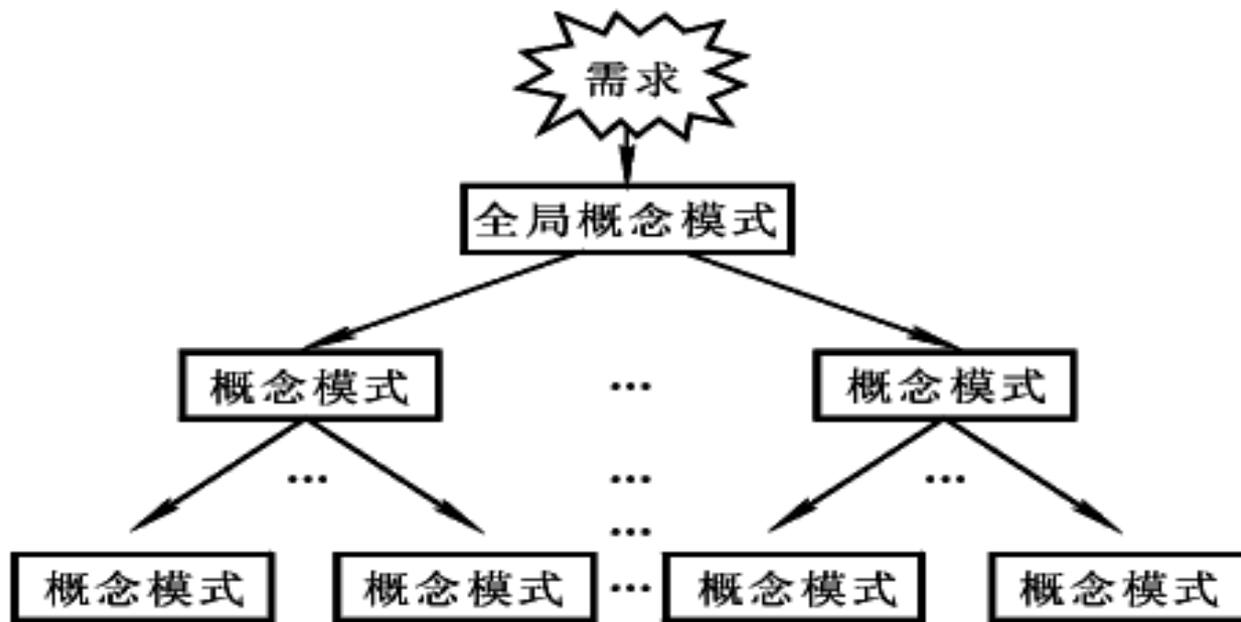
首先定义最重要的核心概念结构，然后向外扩充，以滚雪球的方式逐步生成其他概念结构，直至总体概念结构

➤ 混合策略

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。



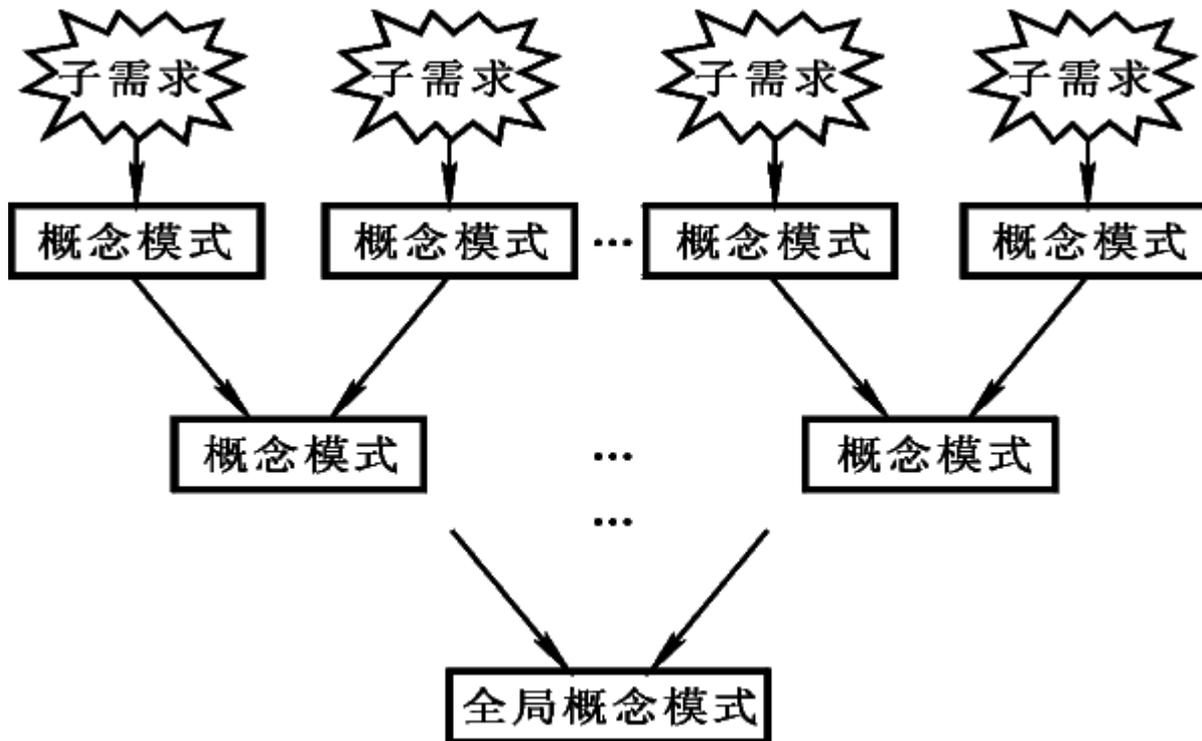
概念结构设计的方法与步骤（续）



自顶向下策略



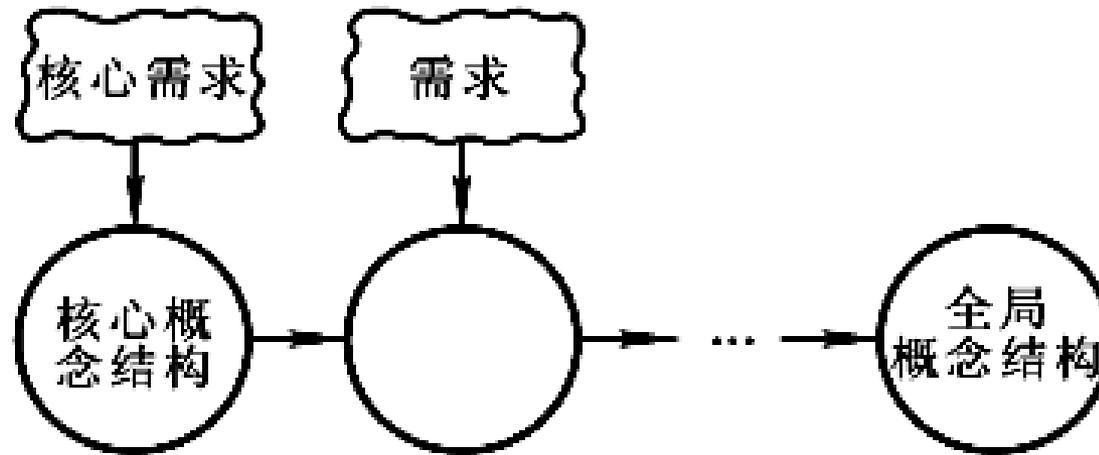
概念结构设计的方法与步骤（续）



自底向上策略



概念结构设计的方法与步骤（续）



逐步扩张



概念结构设计的方法与步骤（续）

- 常用策略
 - 自顶向下地进行需求分析
 - 自底向上地设计概念结构
- 自底向上设计概念结构的步骤
 - 第1步：抽象数据并设计局部视图
 - 第2步：集成局部视图，得到全局概念结构



7.3 概念结构设计

7.3.1 概念结构设计概述

7.3.2 概念结构设计的方法与步骤

7.3.3 数据抽象与局部视图设计

7.3.4 视图的集成



7.3.3 数据抽象与局部视图设计

- 数据抽象
- 局部视图设计



一、数据抽象

- 概念结构是对现实世界的一种抽象
 - 从实际的人、物、事和概念中抽取所关心的共同特性，忽略非本质的细节
 - 把这些特性用各种概念精确地加以描述
 - 这些概念组成了某种模型



数据抽象（续）

■ 三种常用抽象

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
- 这些对象具有某些共同的特性和行为
- 它抽象了对象 **值和型** 之间的 “is member of” 的语义
- 在 **E-R** 模型中，实体型就是这种抽象



数据抽象（续）

2. 聚集（Aggregation）

- 定义某一类型的组成成分
- 它抽象了对象内部类型和成分之间“is part of”的语义
- 在E-R模型中若干属性的聚集组成了实体型，就是这种抽象



数据抽象（续）

3. 概括（Generalization）

- 定义类型之间的一种子集联系
- 它抽象了类型之间的“is subset of”的语义
- 概括有一个很重要的性质：继承性。子类继承超类上定义的所有抽象。



数据抽象（续）

■ 数据抽象的用途

- 对需求分析阶段收集到的数据进行分类、组织（聚集），形成
 - 实体
 - 实体的属性，标识实体的码
 - 确定实体之间的联系类型(1:1, 1:n, m:n)

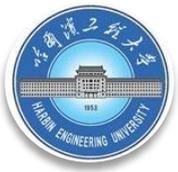


二、局部视图设计

设计分E-R图的步骤:

1.选择局部应用

2.逐一设计分E-R图



1. 选择局部应用

- 需求分析阶段，已用多层数据流图和数据字典描述了整个系统。
- 设计分E-R图首先需要根据系统的具体情况，在多层的数据流图中选择一个适当层次的数据流图，让这组图中每一部分对应一个局部应用，然后以这一层次的数据流图为出发点，设计分E-R图。



2. 逐一设计分E-R图

■ 任务

- ▶ 标定局部应用中的实体、属性、码，实体间的联系

将各局部应用涉及的数据分别从数据字典中抽取出来，参照数据流图，标定各局部应用中的实体、实体的属性、标识实体的码，确定实体之间的联系及其类型（1:1，1:n，m:n）



逐一设计分E-R图（续）

- 如何抽象实体和属性

- **实体**：现实世界中一组具有某些共同特性和行为的对象就可以抽象为一个实体。对象和实体之间是“is member of”的关系。

例：在学校环境中，可把张三、李四等对象抽象为学生实体。



逐一设计分E-R图（续）

- **属性**：对象类型的组成成分可以抽象为实体的属性。组成成分与对象类型之间是“is part of”的关系。

例：学号、姓名、专业、年级等可以抽象为学生实体的属性。其中学号为标识学生实体的码。



逐一设计分E-R图（续）

■ 如何区分实体和属性

- ▶ **实体与属性是相对而言的。**同一事物，在一种应用环境中作为“属性”，在另一种应用环境中就必须作为“实体”。

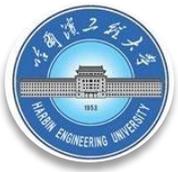
例：学校中的系，在某种应用环境中，它只是作为“学生”实体的一个属性，表明一个学生属于哪个系；而在另一种环境中，由于需要考虑一个系的系主任、教师人数、学生人数、办公地点等，这时它就需要作为实体了。



逐一设计分E-R图（续）

➤ 一般原则

- 属性不能再具有需要描述的性质。即属性必须是不可分的数据项，不能再由另一些属性组成。
- 属性不能与其他实体具有联系。联系只发生在实体之间。
- 符合上述两条特性的事物一般作为属性对待。
- 为了简化E-R图的处置，现实世界中的事物凡能够作为属性对待的，应尽量作为属性。



逐一设计分E-R图（续）

■ 设计分E-R图的步骤

- （1）以数据字典为出发点定义E-R图。
 - 数据字典中的“数据结构”、“数据流”和“数据存储”等已是若干属性的有意义的聚合
- （2）按上面给出的准则进行必要的调整。



逐一设计分E-R图（续）

例：学籍管理局部应用中主要涉及的实体包括学生、宿舍、档案材料、班级、班主任。

实体之间的联系：

- 由于一个宿舍可以住多个学生，而一个学生只能住在某一个宿舍中，因此宿舍与学生之间是1:n的联系。
- 由于一个班级往往有若干名学生，而一个学生只能属于一个班级，因此班级与学生之间也是1:n的联系。



逐一设计分E-R图（续）

- 由于班主任同时还要教课，因此班主任与学生之间存在指导联系，一个班主任要教多名学生，而一个学生只对应一个班主任，因此班主任与学生之间也是1:n的联系。
- 而学生和他自己的档案材料之间，班级与班主任之间都是1:1的联系。



逐一设计分E-R图（续）

接下来需要进一步斟酌该E-R图，做适当调整。

- (1) 在一般情况下，性别通常作为学生实体的属性，但在本局部应用中，由于宿舍分配与学生性别有关，根据准则 2，应该把性别作为实体对待。
- (2) 数据存储“学生登记表”，由于是手工填写，供存档使用，其中有用的部分已转入学生档案材料中，因此这里就不必作为实体了。



逐一设计分E-R图（续）

该E-R图中各个实体的属性描述：

学生： { 学号，姓名，出生日期 }

性别： { 性别 }

档案材料： { 档案号， }

班级： { 班级号，学生人数 }

班主任： { 职工号，姓名，性别，
是否为优秀班主任 }

宿舍： { 宿舍编号，地址，人数 }

其中有下列划线的属性为实体的码。



7.3 概念结构设计

7.3.1 概念结构

7.3.2 概念结构设计的方法与步骤

7.3.3 数据抽象与局部视图设计

7.3.4 视图的集成



7.3.4 视图的集成

- 各个局部视图即分E-R图建立好后，还需要对它们进行合并，集成为一个整体的数据概念结构即总E-R图。



视图的集成（续）

- 视图集成的两种方式
 - 一次集成
 - 一次集成多个分E-R图
 - 通常用于局部视图比较简单时
 - 逐步累积式
 - 首先集成两个局部视图（通常是比较关键的两个局部视图）
 - 以后每次将一个的新的局部视图集成进来

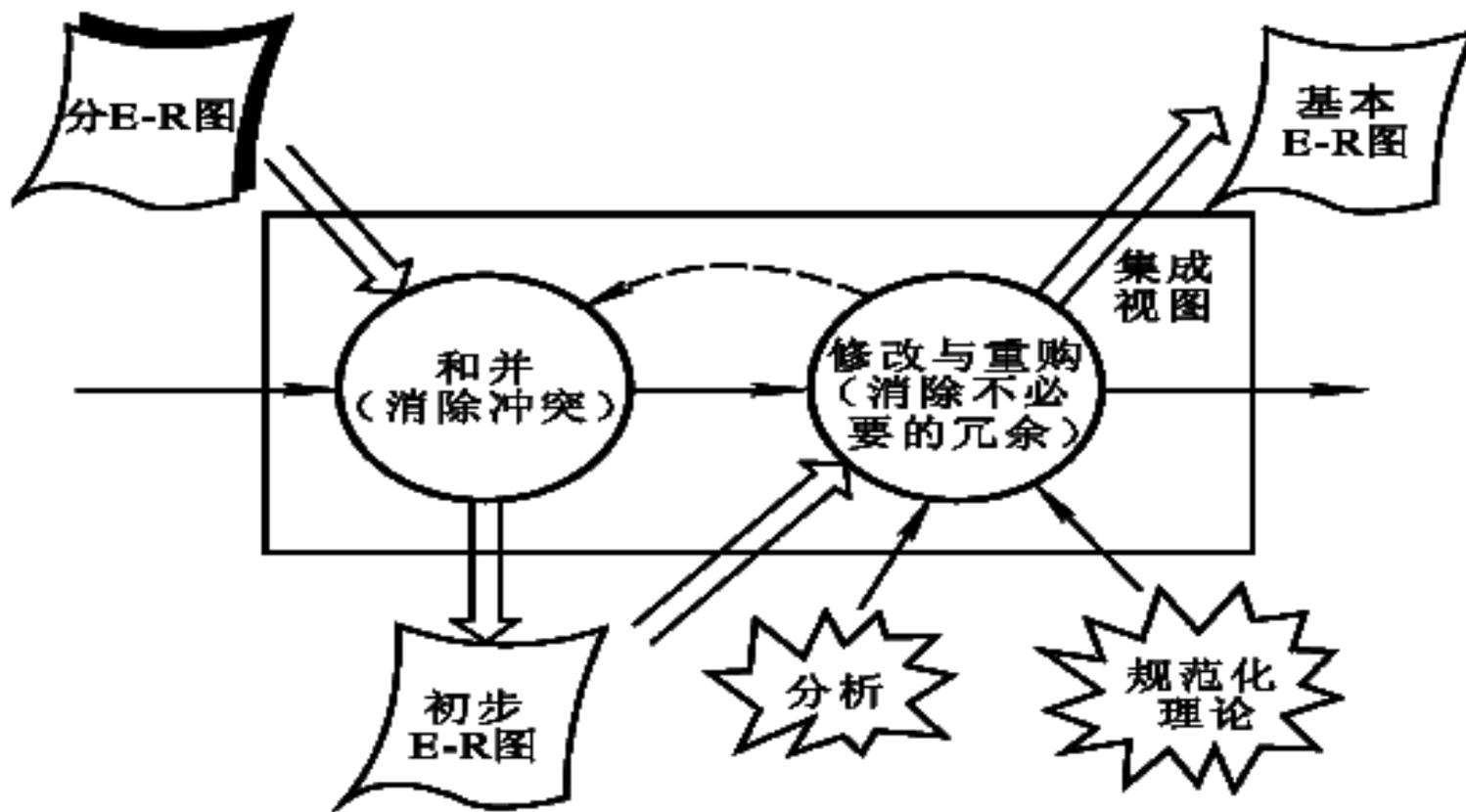


视图的集成（续）

- 集成局部E-R图的步骤
 1. 合并
 2. 修改与重构



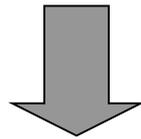
视图的集成 (续)





一、合并分E-R图，生成初步E-R图

- 各分E-R图存在冲突
 - 各个局部应用所面向的问题不同
 - 由不同的设计人员进行设计



各个分E-R图之间必定会存在许多不一致的地方

- 合并分E-R图的主要工作与关键所在：合理消除各分E-R图的冲突



合并分E-R图，生成初步E-R图（续）

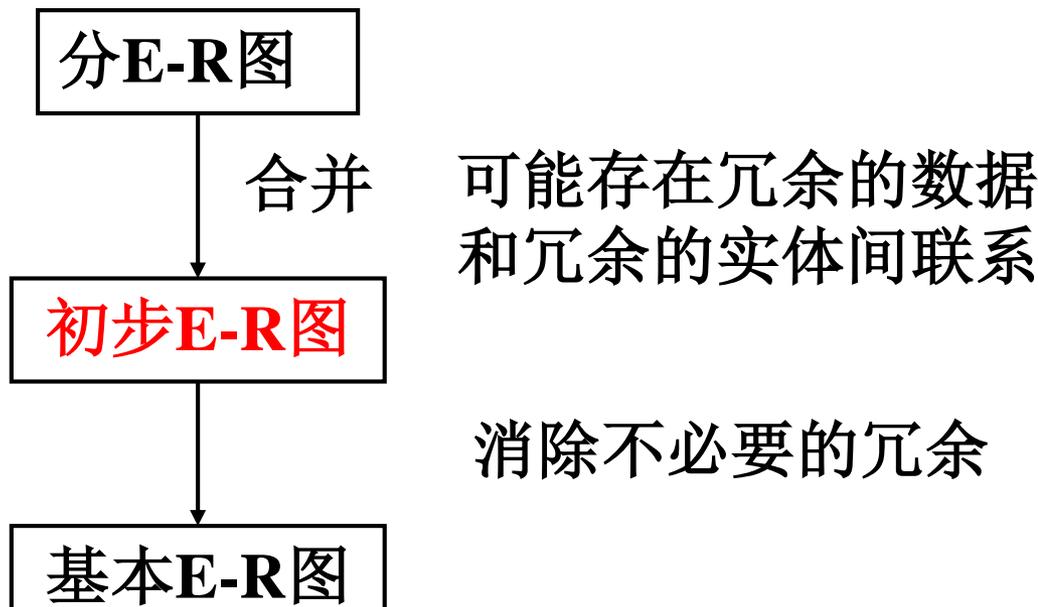
- 冲突的种类
 - 属性冲突
 - 命名冲突
 - 结构冲突



二、修改与重构

■ 基本任务

- 消除不必要的冗余，设计生成基本E-R图





修改与重构（续）

1. 冗余
2. 消除冗余的方法



1. 冗余

- 冗余的数据是指可由基本数据导出的数据，冗余的联系是指可由其他联系导出的联系。
- 冗余数据和冗余联系容易破坏数据库的完整性，给数据库维护增加困难
- 并不是所有的冗余数据与冗余联系都必须加以消除，有时为了提高某些应用的效率，不得不以冗余信息作为代价。



冗余（续）

- 设计数据库概念结构时，哪些冗余信息必须消除，哪些冗余信息允许存在，需要根据用户的整体需求来确定。
- 消除不必要的冗余后的初步E-R图称为基本E-R图。



2. 消除冗余的方法

■ 分析方法

- 以数据字典和数据流图为依据，根据数据字典中关于数据项之间逻辑关系的说明来消除冗余。



消除冗余的方法（续）

- 如果是为了提高效率，人为地保留了一些冗余数据，则应把数据字典中数据关联的说明作为完整性约束条件。
- 一种更好的方法是把冗余数据定义在视图中



消除冗余的方法（续）

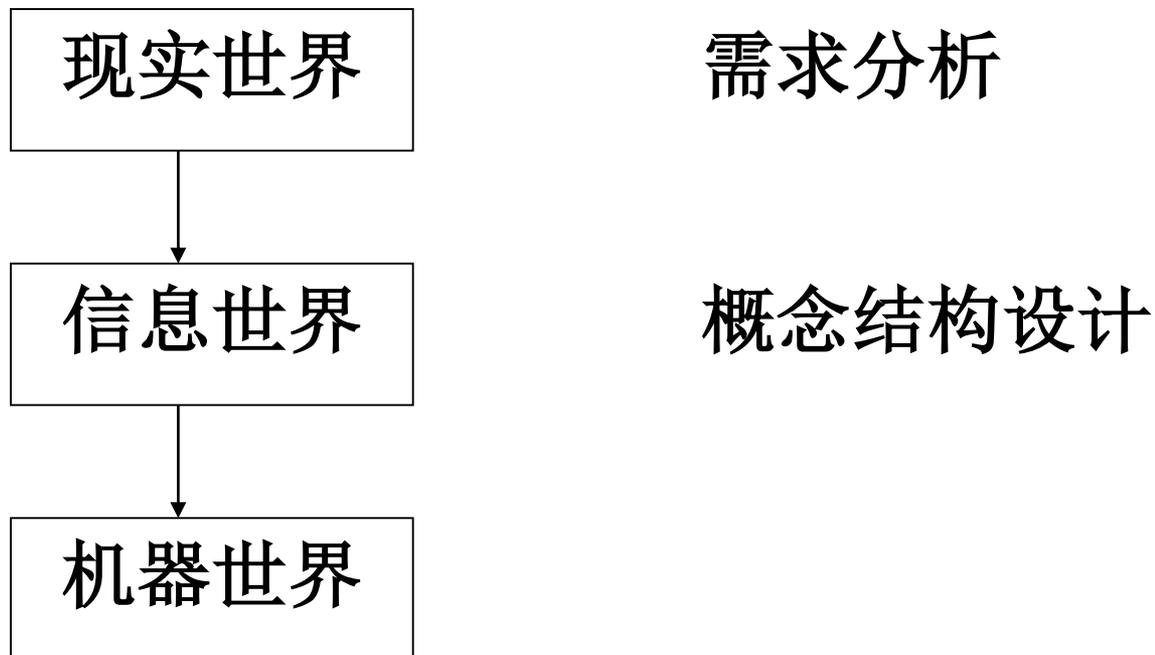
■ 规范化理论

- 函数依赖的概念提供了消除冗余联系的形式化工具



概念结构设计小结

- 什么是概念结构设计





概念结构设计小结

- 概念结构设计的步骤
 - 抽象数据并设计局部视图
 - 集成局部视图，得到全局概念结构
 - 验证整体概念结构



概念结构设计小结

- 数据抽象
 - 分类
 - 聚集
 - 概括



概念结构设计小结

- 设计局部视图
 - 1. 选择局部应用
 - 2. 逐一设计分E-R图
 - 标定局部应用中的实体、属性、码，实体间的联系
 - 用E-R图描述出来



概念结构设计小结

- 集成局部视图
 - 1. 合并分E-R图，生成初步E-R图
 - 消除冲突
 - 属性冲突
 - 命名冲突
 - 结构冲突
 - 2. 修改与重构
 - 消除不必要的冗余，设计生成基本E-R图
 - 分析方法
 - 规范化理论



第七章 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
- 7.8 小结



7.4 逻辑结构设计

■ 逻辑结构设计的任务

- 把概念结构设计阶段设计好的基本E-R图转换为特定数据库管理系统产品所支持的数据类型相符合的逻辑结构
- 目前的数据库应用系统大多采用关系数据模型，即关系数据库管理系统RDBMS，支持关系数据模型。



7.4 逻辑结构设计

■ 逻辑结构设计的步骤

- 将概念结构转化为一般的关系模型
- 将转化来的关系模型向特定DBMS支持下的数据模型转换
- 对数据模型进行优化



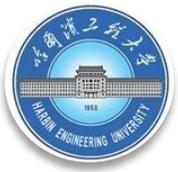
7.4 逻辑结构设计

7.4.1 E-R图向关系模型的转换

7.4.2 向特定DBMS规定的模型进行转换

7.4.3 数据模型的优化

7.4.4 设计用户子模式



7.4.1 E-R图向关系模型的转换

- 转换内容
- 转换原则



E-R图向关系模型的转换（续）

■ 转换内容

- E-R图由实体、实体的属性和实体之间的联系三个要素组成
- 关系模型的逻辑结构是一组关系模式的集合
- 将E-R图转换为关系模型：将实体、实体的属性和实体之间的联系转化为关系模式 $R(U)$ 。



E-R图向关系模型的转换（续）

■ 转换原则

1. 一个实体型转换为一个关系模式。

➤ *关系的属性*: 实体型的属性

➤ *关系的码*: 实体型的码

例，学生实体可以转换为如下关系模式：

学生（学号，姓名，出生日期，所在系，
 年级，平均成绩）



E-R图向关系模型的转换（续）

2. 一个m:n联系转换为一个关系模式。

➤ *关系的属性*: 与该联系相连的各实体的码以及联系本身的属性

➤ *关系的码或关系码的一部分*: 各实体的码组成

例，“选修”联系是一个m:n联系，可以将它转换为如下关系模式，其中学号与课程号为关系的组合码：

选修（学号，课程号，成绩）



E-R图向关系模型的转换（续）

3. 一个1:n联系可以转换为一个独立的关系模式，也可以与n端对应的关系模式合并。

1) 转换为一个独立的关系模式

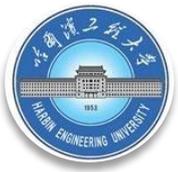
- *关系的属性*: 与该联系相连的各实体的码以及联系本身的属性
- *关系的码*: n端实体的码



E-R图向关系模型的转换（续）

2) 与n端对应的关系模式合并

- 合并后关系的属性: 在n端关系中加入1端关系的码和联系本身的属性
 - 合并后关系的码: 不变
- 可以减少系统中的关系个数，一般情况下更倾向于采用这种方法



E-R图向关系模型的转换（续）

例，“组成”联系为1:n联系。

将其转换为关系模式的两种方法：

1)使其成为一个独立的关系模式：

组成（学号，班级号）

2)将其学生关系模式合并：

学生（学号，姓名，出生日期，所在系，
年级，**班级号**，平均成绩）



E-R图向关系模型的转换（续）

4. 一个1:1联系可以转换为一个独立的关系模式，也可以与任意一端对应的关系模式合并。

1) 转换为一个独立的关系模式

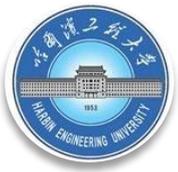
- **关系的属性:** 与该联系相连的各实体的码以及联系本身的属性
- **关系的候选码:** 每个实体的码均是该关系的候选码



E-R图向关系模型的转换（续）

2) 与某一端对应的关系模式合并

- 合并后关系的属性：加入对应关系的码和联系本身的属性
- 合并后关系的码：不变



E-R图向关系模型的转换（续）

例，“管理”联系为1:1联系，可以有三种转换方法：

(1) 转换为一个独立的关系模式：

管理（职工号，班级号）

或

管理（职工号，班级号）

(2) “管理”联系与班级关系模式合并，则只需在班级关系中加入教师关系的码，即职工号：

班级：（班级号，学生人数，**职工号**）

(3) “管理”联系与教师关系模式合并，则只需在教师关系中加入班级关系的码，即班级号：

教师：（职工号，姓名，性别，职称，**班级号**，
是否为优秀班主任）



E-R图向关系模型的转换（续）

注意：

从理论上讲，1:1联系可以与任意一端对应的关系模式合并。

但在一些情况下，与不同的关系模式合并效率会大不一样。因此究竟应该与哪端的关系模式合并需要依应用的具体情况而定。

由于连接操作是最费时的操作，所以一般应以尽量减少连接操作为目标。

例如，如果经常要查询某个班级的班主任姓名，则将管理联系与教师关系合并更好些。



E-R图向关系模型的转换（续）

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。

➤ **关系的属性:** 与该多元联系相连的各实体的码以及联系本身的属性

➤ **关系的码:** 各实体码的组合

例，“讲授”联系是一个三元联系，可以将它转换为如下关系模式，其中课程号、职工号和书号为关系的组合码：

讲授（课程号，职工号，书号）



E-R图向关系模型的转换（续）

6. 同一实体集的实体间的联系，即自联系，也可按上述1:1、1:n和m:n三种情况分别处理。

例，如果教师实体集内部存在领导与被领导的1:n自联系，我们可以将该联系与教师实体合并，这时主码职工号将多次出现，但作用不同，可用不同的属性名加以区分：

教师：{ 职工号，姓名，性别，职称，系主任 }



E-R图向关系模型的转换（续）

7. 具有相同码的关系模式可合并。

➤ 目的：减少系统中的关系个数。

➤ 合并方法：将其中一个关系模式的全部属性加入到另一个关系模式中，然后去掉其中的同义属性（可能同名也可能不同名），并适当调整属性的次序。



E-R图向关系模型的转换（续）

例，“拥有”关系模式：

拥有（学号，性别）

与学生关系模式：

学生（学号，姓名，出生日期，所在系，年级，
班级号，平均成绩）

都以学号为码，可以将它们合并为一个关系模式：

学生（学号，姓名，性别，出生日期，所在系，
年级，班级号，平均成绩）



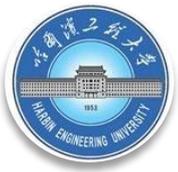
7.4 逻辑结构设计

7.4.1 E-R图向关系模型的转换

7.4.2 向特定DBMS规定的模型进行转换

7.4.3 数据模型的优化

7.4.4 设计用户子模式



7.4.2 向特定DBMS规定的模型进行转换

- 一般的数据模型还需要向特定DBMS规定的模型进行转换。
- 转换的主要依据是所选用的DBMS的功能及限制。没有通用规则。
- 对于关系模型来说，这种转换通常都比较简单。



7.4 逻辑结构设计

7.4.1 E-R图向关系模型的转换

7.4.2 向特定DBMS规定的模型进行转换

7.4.3 数据模型的优化

7.4.4 设计用户子模式



7.4.3 数据模型的优化

- 数据库逻辑设计的结果不是唯一的。
- 得到初步数据模型后，还应该适当地修改、调整数据模型的结构，以进一步提高数据库应用系统的性能，这就是数据模型的优化。
- 关系数据模型的优化通常以规范化理论为指导。



数据模型的优化（续）

■ 优化数据模型的方法

1. 确定数据依赖

- 按需求分析阶段所得到的语义，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数据依赖。



数据模型的优化（续）

2. 对于各个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。



数据模型的优化（续）

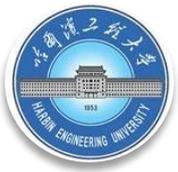
3. 按照数据依赖的理论对关系模式逐一进行分析，考查是否存在部分函数依赖、传递函数依赖、多值依赖等，确定各关系模式分别属于第几范式。

例如经过分析可知，课程关系模式属于**BC**范式。



数据模型的优化（续）

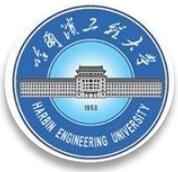
4. 按照需求分析阶段得到的各种应用对数据处理的要求，分析对于这样的应用环境这些模式是否合适，确定是否要对它们进行合并或分解。



数据模型的优化（续）

非BCNF的关系模式虽然从理论上分析会存在不同程度的更新异常，但如果在实际应用中对此关系模式只是查询，并不执行更新操作，则就不会产生实际影响。

对于一个具体应用来说，到底规范化进行到什么程度，需要权衡响应时间和潜在问题两者的利弊才能决定。一般说来，第三范式就足够了。



数据模型的优化（续）

例：在关系模式

学生成绩单(学号,英语,数学,语文,平均成绩)

中存在下列函数依赖：

学号→英语

学号→数学

学号→语文

学号→平均成绩

(英语, 数学, 语文)→平均成绩



数据模型的优化（续）

显然有：

学号 \rightarrow (英语,数学,语文)

因此该关系模式中存在传递函数依赖，是2NF关系。

虽然平均成绩可以由其他属性推算出来，但如果应用中需要经常查询学生的平均成绩，为提高效率，我们仍然可保留该冗余数据，对关系模式不再做进一步分解。



数据模型的优化（续）

5. 按照需求分析阶段得到的各种应用对数据处理的要求，对关系模式进行必要的分解或合并，以提高数据操作的效率和存储空间的利用率

➤ 常用分解方法

- 水平分解
- 垂直分解



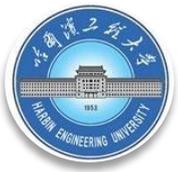
7.4 逻辑结构设计

7.4.1 E-R图向关系模型的转换

7.4.2 向特定DBMS规定的模型进行转换

7.4.3 数据模型的优化

7.4.4 设计用户子模式



7.4.4 设计用户子模式

- 定义数据库模式主要是从系统的时间效率、空间效率、易维护等角度出发。
- 定义用户外模式时应该更注重考虑用户的习惯与方便。包括三个方面：



设计用户子模式（续）

(1) 使用更符合用户习惯的别名

- 合并各分E-R图曾做了消除命名冲突的工作，以使数据库系统中同一关系和属性具有唯一的名字。这在设计数据库整体结构时是非常必要的。
- 但对于某些局部应用，由于改用了不符合用户习惯的属性名，可能会使他们感到不方便，



设计用户子模式（续）

(1) 使用更符合用户习惯的别名(续)

- 因此在设计用户的子模式时可以重新定义某些属性名，使其与用户习惯一致。
- 当然，为了应用的规范化，我们也不应该一味地迁就用户。

例：负责学籍管理的用户习惯于称教师模式的职工号为教师编号。因此可以定义视图，在视图职工号重定义为教师编号



设计用户子模式（续）

(2) 针对不同级别的用户定义不同的外模式，以满足系统对安全性的要求。



设计用户子模式（续）

(3) 简化用户对系统的使用

- 如果某些局部应用中经常要使用某些很复杂的查询，为了方便用户，可以将这些复杂查询定义为视图。



逻辑结构设计小结

■ 任务

- 将概念结构转化为具体的数据模型

■ 逻辑结构设计的步骤

- 将概念结构转化为一般的关系、网状、层次模型
- 将转化来的关系、网状、层次模型向特定DBMS支持下的数据模型转换
- 对数据模型进行优化
- 设计用户子模式



逻辑结构设计小结

- E-R图向关系模型的转换内容
 - 将E-R图转换为关系模型：将实体、实体的属性和实体之间的联系转化为关系模式。



逻辑结构设计小结

■ E-R图向关系模型的转换原则

1. 一个**实体型**转换为一个关系模式。
2. 一个**m:n联系**转换为一个关系模式。
3. 一个**1:n联系**可以转换为一个独立的关系模式，也可以与n端对应的关系模式合并。
4. 一个**1:1联系**可以转换为一个独立的关系模式，也可以与任意一端对应的关系模式合并。



逻辑结构设计小结

■ E-R图向关系模型的转换原则

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. 同一实体集的实体间的联系，即**自联系**，也可按上述1:1、1:n和m:n三种情况分别处理。
7. 具有**相同码**的关系模式可合并。



逻辑结构设计小结

■ 优化数据模型的方法

1. 确定数据依赖
2. 对于各个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. 确定各关系模式分别属于第几范式。
4. 分析对于应用环境这些模式是否合适，确定是否要对它们进行合并或分解。
5. 对关系模式进行必要的分解或合并



逻辑结构设计小结

■ 设计用户子模式

1. 使用更符合用户习惯的别名
2. 针对不同级别的用户定义不同的外模式，以满足系统对安全性的要求。
3. 简化用户对系统的使用



第七章 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
- 7.8 小结



7.5 数据库的物理设计

- 什么是数据库的物理设计
 - 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，它依赖于给定的计算机系统。
 - 为一个给定的逻辑数据模型选取一个最适合应用环境的物理结构的过程，就是数据库的物理设计。



7.5 数据库的物理设计

■ 数据库物理设计的步骤

- 确定数据库的物理结构
- 对物理结构进行评价，评价的重点是时间和空间效率
- 如果评价结果满足原设计要求则可进入到物理实施阶段，否则，就需要重新设计或修改物理结构，有时甚至要返回逻辑设计阶段修改数据模型。



7.5 数据库的物理设计

7.5.1 数据库的物理设计的内容和方法

7.5.2 关系模式存取方法选择

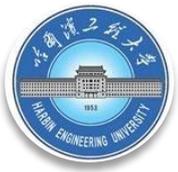
7.5.3 确定数据库的存储结构

7.5.4 评价物理结构



7.5.1 数据库的物理设计的内容和方法

- 设计物理数据库结构的准备工作
 - 1. 充分了解应用环境，详细分析要运行的事务，以获得选择物理数据库设计所需参数
 - 2. 充分了解所用**DBMS**的内部特征，特别是系统提供的存取方法和存储结构



数据库的物理设计的内容和方法（续）

- 选择物理数据库设计所需参数
 - 数据库查询事务
 - 查询的关系
 - 查询条件所涉及的属性
 - 连接条件所涉及的属性
 - 查询的投影属性



数据库的物理设计的内容和方法（续）

■ 选择物理数据库设计所需参数(续)

➤ 数据更新事务

■ 被更新的关系

■ 每个关系上的更新操作条件所涉及的属性

■ 修改操作要改变的属性值

➤ 每个事务在各关系上运行的频率和性能要求



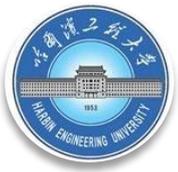
7.5 数据库的物理设计

7.5.1 数据库的物理设计的内容和方法

7.5.2 关系模式存取方法选择

7.5.3 确定数据库的存储结构

7.5.4 评价物理结构



7.5.2 关系模式存取方法选择

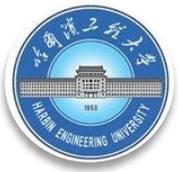
- 数据库系统是多用户共享的系统，对同一个关系要建立多条存取路径才能满足多用户的多种应用要求。
- 物理设计的第一个任务就是要确定选择哪些存取方法，即建立哪些存取路径。



关系模式存取方法选择（续）

■ DBMS常用存取方法

- 索引方法，目前主要是**B+**树索引方法
- 聚簇（**Cluster**）方法
- **HASH**方法



7.5 数据库的物理设计

7.5.1 数据库的物理设计的内容和方法

7.5.2 关系模式存取方法选择

7.5.3 确定数据库的存储结构

7.5.4 评价物理结构



7.5.3 确定数据库的存储结构

- 确定数据库物理结构的内容
 - 1. 确定数据的存放位置和存储结构
 - 关系
 - 索引
 - 聚簇
 - 日志
 - 备份
 - 2. 确定系统配置



1. 确定数据的存放位置

- 影响数据存放位置和存储结构的因素
 - 硬件环境
 - 应用需求
 - 存取时间
 - 存储空间利用率
 - 维护代价

这三个方面常常是相互矛盾的

例：消除一切冗余数据虽能够节约存储空间和减少维护代价，但往往会导致检索代价的增加

必须进行权衡，选择一个折中方案。



确定数据的存放位置（续）

■ 基本原则

➤ 根据应用情况将

- 易变部分与稳定部分

- 存取频率较高部分与存取频率较低部分
分开存放，以提高系统性能



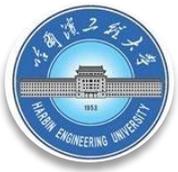
2. 确定系统配置

- DBMS产品一般都提供了一些存储分配参数
 - 同时使用数据库的用户数
 - 同时打开的数据库对象数
 - 使用的缓冲区长度、个数
 - 时间片大小
 - 数据库的大小
 - 装填因子
 - 锁的数目
 - 等等



确定系统配置（续）

- 系统都为这些变量赋予了合理的缺省值。但是这些值不一定适合每一种应用环境，在进行物理设计时，需要根据应用环境确定这些参数值，以使系统性能最优。
- 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做进一步的调整，以期切实改进系统性能。



7.5 数据库的物理设计

7.5.1 数据库的物理设计的内容和方法

7.5.2 关系模式存取方法选择

7.5.3 确定数据库的存储结构

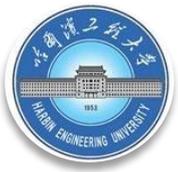
7.5.4 评价物理结构



7.5.4 评价物理结构

■ 评价内容

- 对数据库物理设计过程中产生的多种方案进行细致的评价，从中选择一个较优的方案作为数据库的物理结构



7.5.4 评价物理结构

■ 评价方法

➤ 定量估算各种方案

- 存储空间
- 存取时间
- 维护代价

➤ 对估算结果进行权衡、比较，选择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



第七章 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
- 7.8 小结



7.6 数据库的实施

- 数据库实施的工作内容
 - 用DDL定义数据库结构
 - 组织数据入库
 - 编制与调试应用程序
 - 数据库试运行



一、定义数据库结构

- 确定了数据库的逻辑结构与物理结构后，就可以用所选用的**DBMS**提供的**数据定义语言（DDL）**来严格描述数据库结构。



定义数据库结构（续）

例，对于前面的例子，可以用SQL语句如下定义表结构：

```
CREATE TABLE 学生
    (学号 CHAR(8),
    .....
    );
CREATE TABLE 课程
    (
    .....
    );
.....
```



二、数据装载

- 数据库结构建立好后，就可以向数据库中装载数据了。组织数据入库是数据库实施阶段最主要的工作。
- 数据装载方法
 - 人工方法
 - 计算机辅助数据入库



数据装载（续）

- 人工方法：适用于小型系统
 - 步骤
 - 1) **筛选数据**。需要装入数据库中的数据通常都分散在各个部门的数据文件或原始凭证中，所以首先必须把需要入库的数据筛选出来。
 - 2) **转换数据格式**。筛选出来的需要入库的数据，其格式往往不符合数据库要求，还需要进行转换。这种转换有时可能很复杂。
 - 3) **输入数据**。将转换好的数据输入计算机中。
 - 4) **校验数据**。检查输入的数据是否有误。



数据装载（续）

- 计算机辅助数据入库：适用于中大型系统
 - 步骤
 - 1) 筛选数据
 - 2) 输入数据。由录入员将原始数据直接输入计算机中。数据输入子系统应提供输入界面。
 - 3) 校验数据。数据输入子系统采用多种检验技术检查输入数据的正确性。



数据装载（续）

- 4) **转换数据**。数据输入子系统根据数据库系统的要求，从录入的数据中**抽取**有用成分，对其进行**分类**，然后**转换**数据格式。抽取、分类和转换数据是数据输入子系统的主要工作，也是数据输入子系统的复杂性所在。
- 5) **综合数据**。数据输入子系统对转换好的数据根据系统的要求进一步综合成最终数据。



三、编制与调试应用程序

- 数据库应用程序的设计应该与数据设计并行进行。
- 在数据库实施阶段，当数据库结构建立好后，就可以开始编制与调试数据库的应用程序。调试应用程序时由于数据入库尚未完成，可先使用模拟数据。



四、数据库试运行

- 应用程序调试完成，并且已有一小部分数据入库后，就可以开始数据库的试运行。
- 数据库试运行也称为联合调试，其主要工作包括：
 - 1) **功能测试**：实际运行应用程序，执行对数据库的各种操作，测试应用程序的各种功能。
 - 2) **性能测试**：测量系统的性能指标，分析是否符合设计目标。



数据库试运行（续）

■ 数据库性能指标的测量

- 数据库物理设计阶段在评价数据库结构估算时间、空间指标时，作了许多简化和假设，忽略了许多次要因素，因此结果必然很粗糙。
- 数据库试运行则是要实际测量系统的各种性能指标（不仅是时间、空间指标），如果结果不符合设计目标，则需要返回物理设计阶段，调整物理结构，修改参数；有时甚至需要返回逻辑设计阶段，调整逻辑结构。



数据库试运行（续）

- 数据库的转储和恢复
 - 在数据库试运行阶段，系统还不稳定，硬、软件故障随时都可能发生
 - 系统的操作人员对新系统还不熟悉，误操作也不可避免
 - 因此必须做好数据库的转储和恢复工作，尽量减少对数据库的破坏。



第七章 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
- 7.8 小结



7.7 数据库运行与维护

- 数据库试运行结果符合设计目标后，数据库就可以真正投入运行了。
- 数据库投入运行标志着开发任务的基本完成和维护工作的开始
- 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。
 - 应用环境在不断变化
 - 数据库运行过程中物理存储会不断变化



数据库运行与维护（续）

- 在数据库运行阶段，对数据库经常性的维护工作主要是由DBA完成的，包括：

1.数据库的转储和恢复

- 转储和恢复是系统正式运行后最重要的维护工作之一。
- DBA要针对不同的应用要求制定不同的转储计划，定期对数据库和日志文件进行备份。
- 一旦发生介质故障，即利用数据库备份及日志文件备份，尽快将数据库恢复到某种一致性状态。



数据库运行与维护（续）

2.数据库的安全性、完整性控制

- **DBA**必须根据用户的实际需要授予不同的操作权限
- 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，**DBA**需要根据实际情况修改原有的安全性控制。
- 由于应用环境的变化，数据库的完整性约束条件也会变化，也需要**DBA**不断修正，以满足用户要求。



数据库运行与维护（续）

3.数据库性能的监督、分析和改进

- ▶ 在数据库运行过程中，**DBA**必须监督系统运行，对监测数据进行分析，找出改进系统性能的方法。
 - 利用监测工具获取系统运行过程中一系列性能参数的值
 - 通过仔细分析这些数据，判断当前系统是否处于最佳运行状态
 - 如果不是，则需要通过调整某些参数来进一步改进数据库性能



数据库运行与维护（续）

4.数据库的重组和重构造

1) 数据库的重组

➤ 为什么要重组数据库

- 数据库运行一段时间后，由于记录的不断增、删、改，会使数据库的物理存储变坏，从而降低数据库存储空间的利用率和数据的存取效率，使数据库的性能下降。



数据库运行与维护（续）

2) 数据库的重构造

- 为什么要进行数据库的重构造
 - 数据库应用环境发生变化，会导致实体及实体间的联系也发生相应的变化，使原有的数据库设计不能很好地满足新的需求
 - 增加新的应用或新的实体
 - 取消某些已有应用
 - 改变某些已有应用



第七章 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
- 7.8 小结



7.8 小结

■ 数据库的设计过程

- 需求分析
- 概念结构设计
- 逻辑结构设计
- 物理设计
- 实施
- 运行维护

设计过程中往往还会有许多反复。



小结（续）

■ 数据库各级模式的形成

- 数据库的各级模式是在设计过程中逐步形成的
- 需求分析阶段综合各个用户的应用需求（现实世界的需求）。
- 概念设计阶段形成独立于机器特点、独立于各个DBMS产品的**概念模式**（信息世界模型），用E-R图来描述。



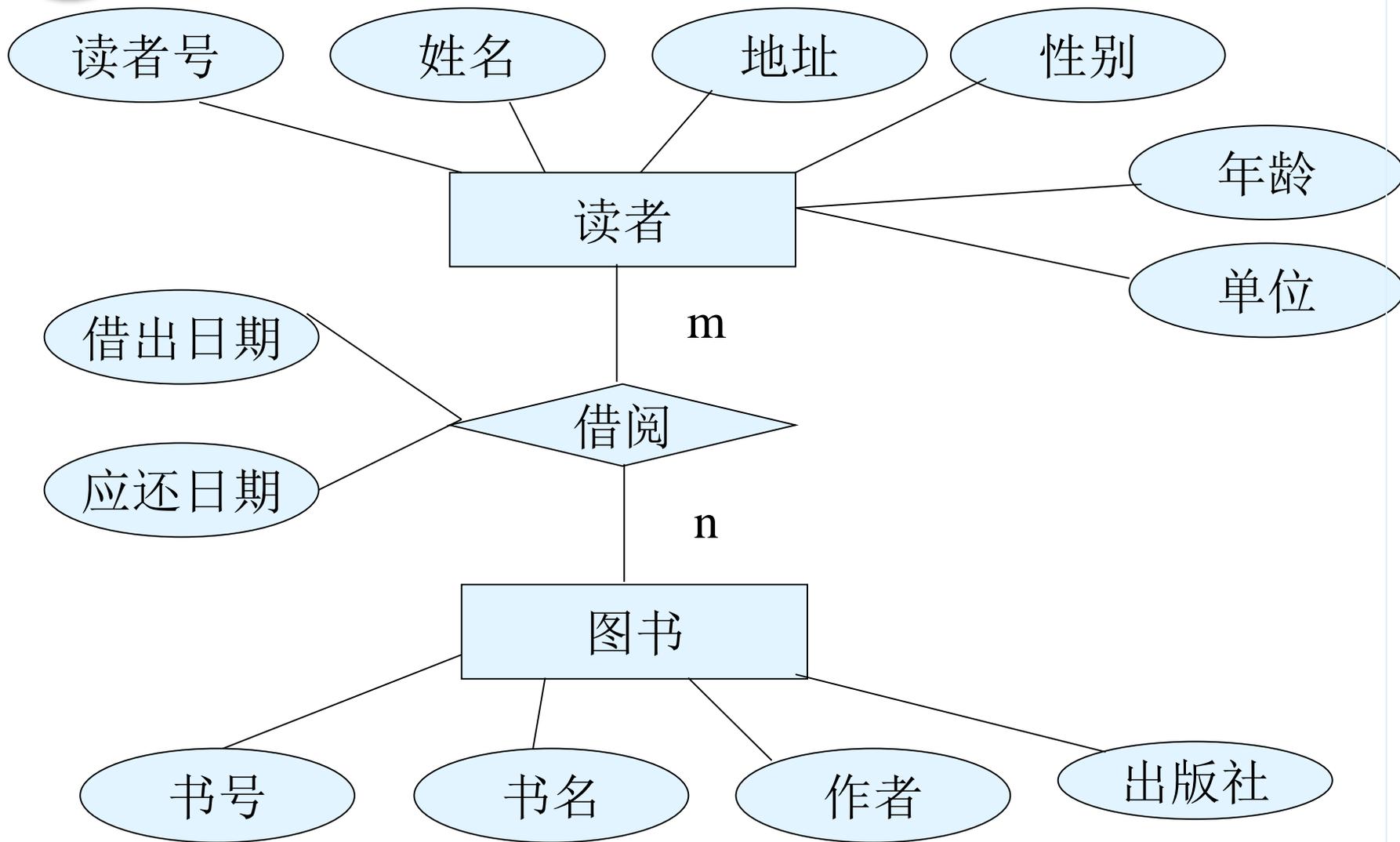
小结（续）

- 在逻辑设计阶段将E-R图转换成具体的数据库产品支持的数据模型如关系模型，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（VIEW）形成数据的**外模式**。
- 在物理设计阶段根据DBMS特点和处理的需要，进行物理存储安排，设计索引，形成数据库**内模式**。



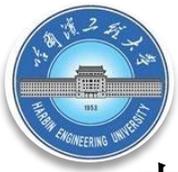
本章总结

- 主要内容：
- 数据库设计各阶段内容
- 重要知识点：
- 概念结构设计；逻辑结构设计
- 题型：
- 选择、填空、应用题

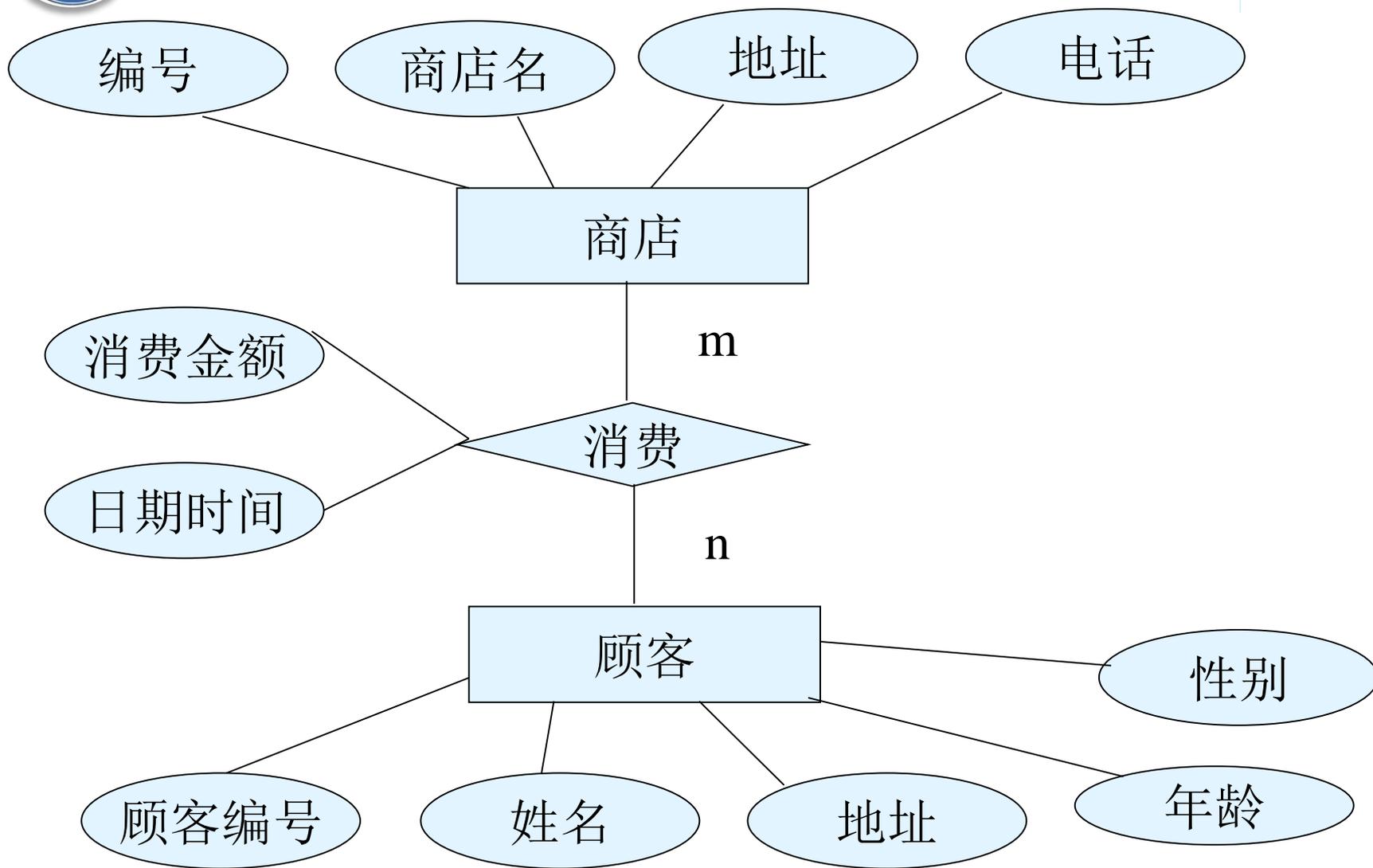




- 读者（读者号，姓名，地址，性别，年龄，单位）
- 书（书号，书名，作者，出版社）
- 借阅（读者号，书号，借出日期，应还日期）



- 某商业集团数据库中有两实体集。一是“商店”实体集，属性有商店编号、商店名、地址、电话等；另一个“顾客”，属性有顾客编号、姓名、地址、年龄、性别等。
- 顾客与商店间存在着消费关系。如果一个商店可以有許多顾客前来购买物品，而每位顾客可选多个商店前去消费。顾客每次去商店购买有一个消费金额和日期。规定，每个顾客在每个商店每天最多消费一次。





- 商店（编号，商店名，地址，电话）
- 顾客（顾客编号，姓名，地址，年龄，性别）
- 消费（商店编号，顾客编号，日期时间，消费金额）