北京航空航天大學
BEIHANG UNIVERSITY

数据管理技术

北京航空航天大学

周号益

2024年

第十一章 数据库设计

1 概述

2 需求分析

3 概念结构设计

4 逻辑结构设计

数据库设计的任务

- 数据库设计是指根据用户需求研制数据库结构的过程，具体地说，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，使之能有效的存储数据，满足用户的信息要求和处理要求。
- 也就是把现实世界中的数据，根据各种应用处理的要求，加以合理地组织，满足硬件和操作系统的特性，利用已有的DBMS来建立能够实现系统目标的数据库。

数据库设计的内容

- 数据库设计包括静态的数据库结构设计和动态的数据库行为设计（即数据库应用程序设计）两方面的内容
- 数据库的结构设计
  - 数据库的结构设计是指根据给定的应用环境，进行数据库的模式的设计。
  - 它包括数据库的概念设计、逻辑设计和物理设计。
  - 数据库模式是各应用程序共享的结构，是静态的、稳定的，一经形成后通常情况下是不容易改变的，所以结构设计又称为静态模型设计。

数据库设计方法简述

- 数据库设计方法主要有直观设计法和规范设计法
- 直观设计法也叫手工试凑法，它是最早使用的数据库设计方法。这种方法依赖于设计者的经验和技巧，缺乏科学理论和工程原则的支持，设计的质量很难保证，常常是数据库运行一段时间后又发现各种问题，这样再重新进行修改，增加了系统维护的代价。因此这种方法越来越不适应信息管理发展的需要。

数据库设计方法简述

- 为了改变这种情况，1978年10月，来自三十多个国家的数据库专家在美国新奥尔良（New Orleans）市专门讨论了数据库设计问题，他们运用软件工程的思想和方法，提出了数据库设计的规范，这就是著名的新奥尔良法，它是比较完整和权威的一种规范设计法。新奥尔良法将数据库设计分成需求分析（分析用户需求）、概念设计（信息分析和定义）、逻辑设计（设计实现）和物理设计（物理数据库设计）。目前，常用的规范设计方法大多起源于新奥尔良法，并在设计的每一阶段采用一些辅助方法来具体实现。
- 下面简单介绍几种常用的规范设计方法。

数据库设计方法简述

- 基于E-R模型的数据库设计方法
  - 基于E-R模型的数据库设计方法是由陈品山于1976年提出的数据库设计方法，其基本思想是在需求分析的基础上，用E-R（实体—联系）图构造一个反映现实世界实体之间联系的企业模式，然后再将此企业模式转换成基于某一特定的DBMS的概念模式。
- 基于3NF的数据库设计方法
  - 基于3NF的数据库设计方法是由S·Atre提出的结构化设计方法，其基本思想是在需求分析的基础上，确定数据库模式中的全部属性和属性间的依赖关系，将它们组织在一个单一的关系模式中，然后再分析模式中不符合3NF的约束条件，将其进行投影分解，规范成若干个3NF关系模式的集合

数据库设计方法简述

- 基于视图的数据库设计方法
  - 此方法先从分析各个应用的数据着手，其基本思想是为每个应用建立自己的视图，然后再把这些视图汇总起来合并成整个数据库的概念模式。视图合并过程中要解决诸如命名冲突、冗余实体/联系等问题

- 新奥尔良规范化设计方法采用过程迭代，逐步求精的思想，在不同阶段对上述方法进行融合应用

数据库设计的过程

- 1 需求分析阶段
  - 准确了解与分析用户需求（包括数据与处理）
  - 是整个设计过程的基础，是最困难、最耗费时间的一步
- 2 概念结构设计阶段
  - 是整个数据库设计的关键
  - 通过对用户需求进行综合、归纳与抽象，形成一个独立于具体DBMS的概念模型

数据库设计的过程

- 3 逻辑结构设计阶段
  - 将概念结构转换为某个DBMS所支持的数据模型
  - 对其进行优化
- 4.数据库物理设计阶段
  - 为逻辑数据模型选取一个最适合应用环境的物理结构（包括存储结构和存取方法）

数据库设计的过程

- 5 数据库实施阶段
  - 运用DBMS提供的数据语言、工具及宿主语言，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.数据库运行和维护阶段
  - 数据库应用系统经过试运行后即可投入正式运行
  - 在数据库系统运行过程中必须不断地对其进行评价、调整与修改

数据库设计各阶段的任务

- 需求分析阶段
  - 综合各个用户的应用需求

- 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(E-R图)

数据库设计各阶段的任务

- 逻辑设计阶段
  - 首先将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

- 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，建立索引，形成数据库内模式

第十一章 数据库设计

1 概述

2 需求分析

3 概念结构设计

4 逻辑结构设计

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

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

- 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。

需求分析的重点

- 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
- 处理要求
  - 对处理功能的要求
  - 对处理的响应时间的要求
  - 对处理方式的要求(批处理 / 联机处理)

需求分析过程

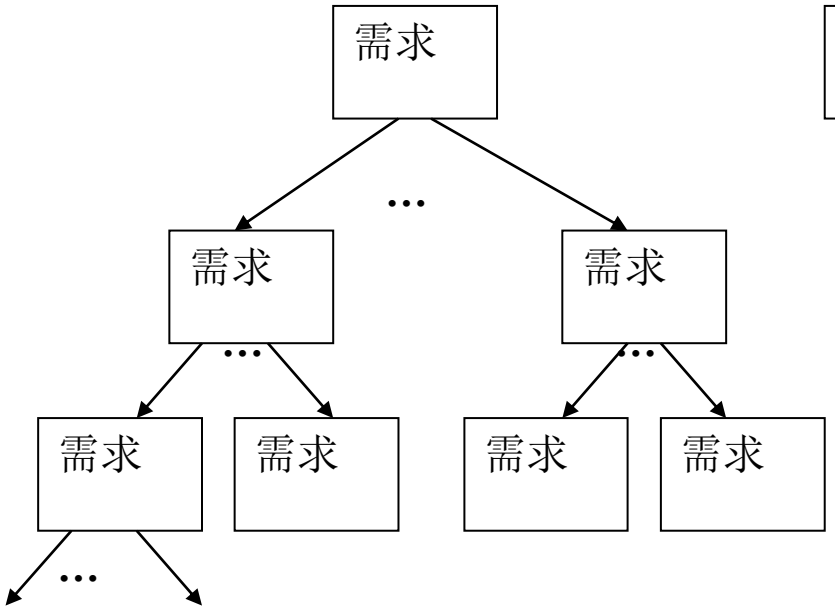
- 需求获取
  - 与客户交流
    - ✓ 询问；跟班调查；查阅单据；开调研会；请领域专家座谈等
  - 形成需求文档
    - ✓ 描述组织机构；部门与岗位职责；部门业务活动等
    - ✓ 重点关注信息要求，处理要求和完全性与完整性要求
    - ✓ 确定系统边界
- 需求分析与表达

需求分析与表达

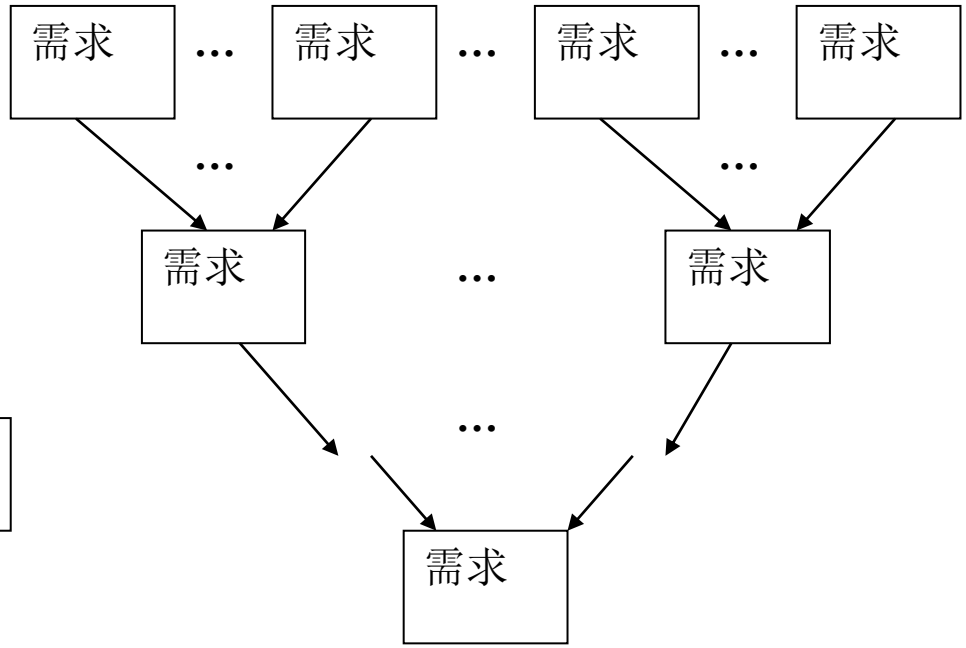
- 分析和表达用户的需求的常用方法
  - 自顶向下的结构化分析方法（Structured Analysis，简称SA方法）
  - 面向对象的分析方法

- SA方法从最上层的系统组织机构入手，采用逐层分解的方式分析系统，并用数据流图和数据字典描述系统。

需求分析与表达



(a)　自顶向下的需求分析



(b)　自底向上的需求分析

我们通常采用自顶向下的需求分析方法

即：将总体需求组层分解，在细化的需求上进行详细描述和表达

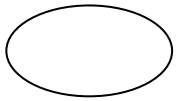
数据流图

■ 数据流图（Data Flow Diagram），是从实际系统抽象出来的、用特定的符号反映系统的数据传递和变换过程的图。它是系统的逻辑模型，与实际系统中具体的处理人员、处理工具和处理方式等无关，只反映数据处理和传送的过程和方向。

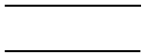
数据流图的组成成分

- 数据流：沿箭头方向传递数据的通道，描述数据的流向。

- 处理：对流入的数据进行的操作。数据流图的核心。



- 数据存储：与处理有关的数据集合。



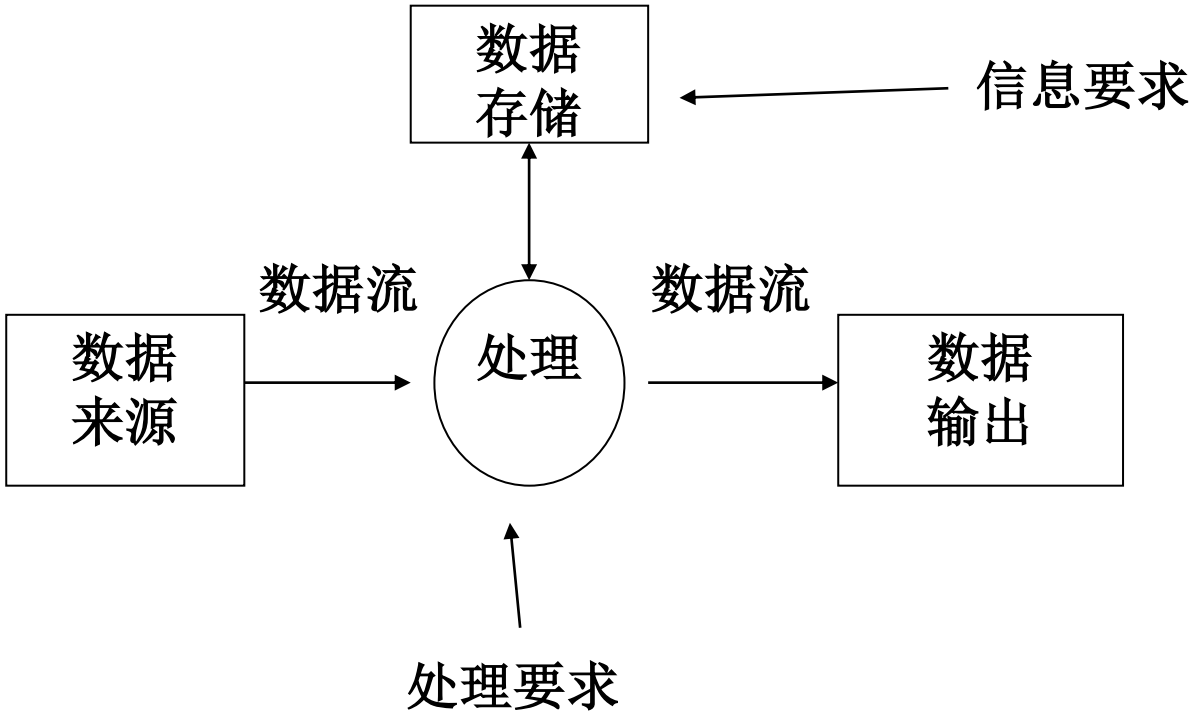
- 实体：描述数据流的起点和终点。

数据流图的画法

- 自顶向下，逐步求精的方法
  - 顶层图：描述系统的范围和边界
  - 底层图：描述一个简单的独立功能
  - 中间图：描述上一层的某个处理，分解成几个独立的功能
- 由外向里的原则：在绘制顶层图时先考虑整个系统的输入和输出数据流，然后再考虑系统内部的其他元素。
- 注意事项：恰当地命名、处理框编号

数据流图的画法

- 1．首先把任何一个系统都抽象为：



数据流图的画法

- 2．分解处理功能和数据
  - 分解处理功能
    - ✓ 将处理功能的具体内容分解为若干子功能，再将每个子功能继续分解，直到把系统的工作过程表达清楚为止。
  - 分解数据
    - ✓ 在处理功能逐步分解的同时，其所用的数据也逐级分解，形成若干层次的数据流图
    - ✓ 数据流图表达了数据和处理过程的关系

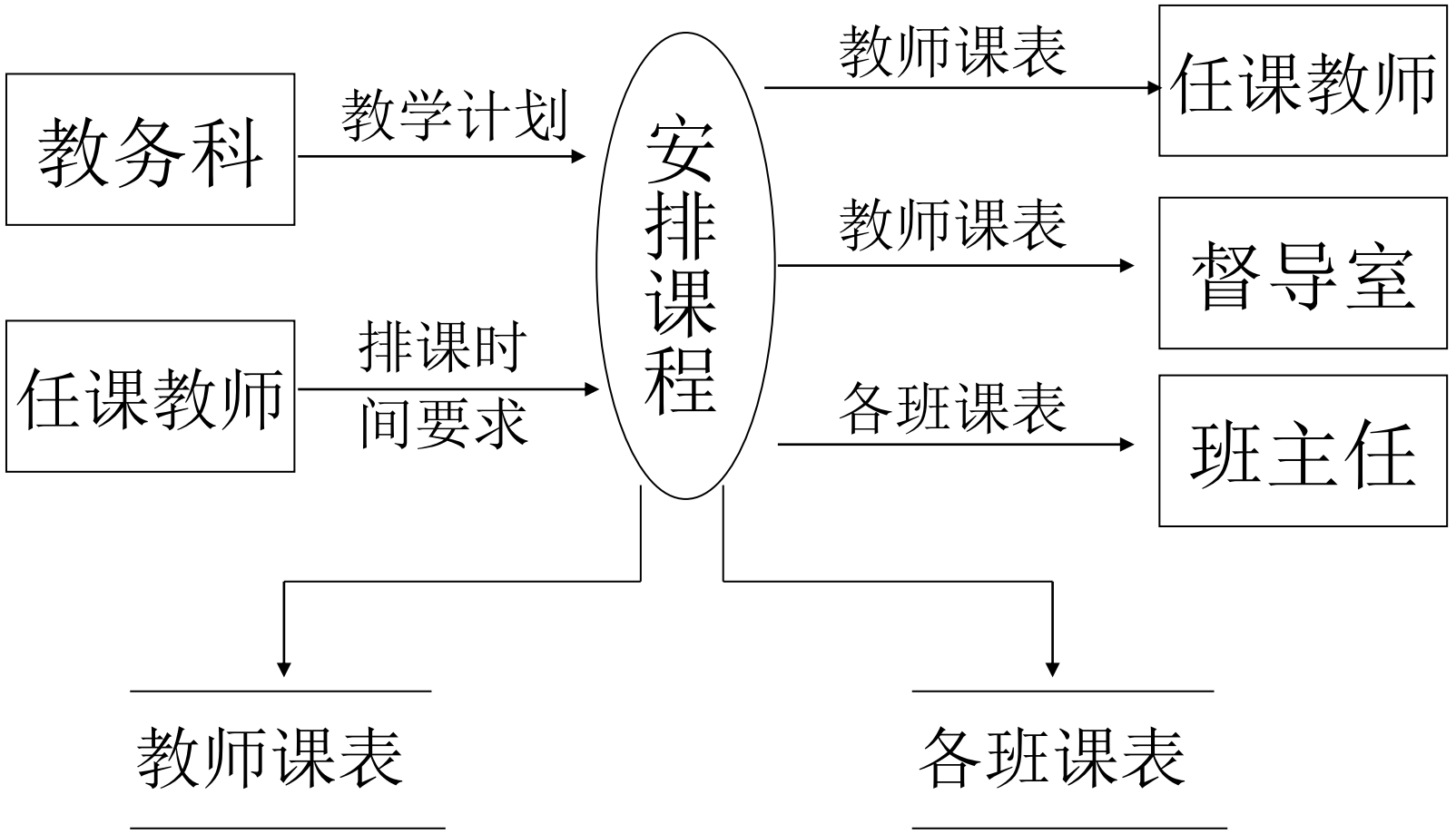
- 3．将分析结果再次提交给用户，征得用户的认可

数据流图的示例

- 例：某学校排课业务流程调研如下：
  - 教务科根据各专业教学计划统计出各班级下学期所开课程情况，并将该表交各专业教研室主任。
  - 各教研室主任为本室教师安排课程，每位教师写出自己的上课要求，一并交教务科。
  - 教务科安排排课人员排课。
  - 将班级课表交班主任，教师课表交任课教师和督导室。
  - 各班课表和教师课表都要留一份存档。

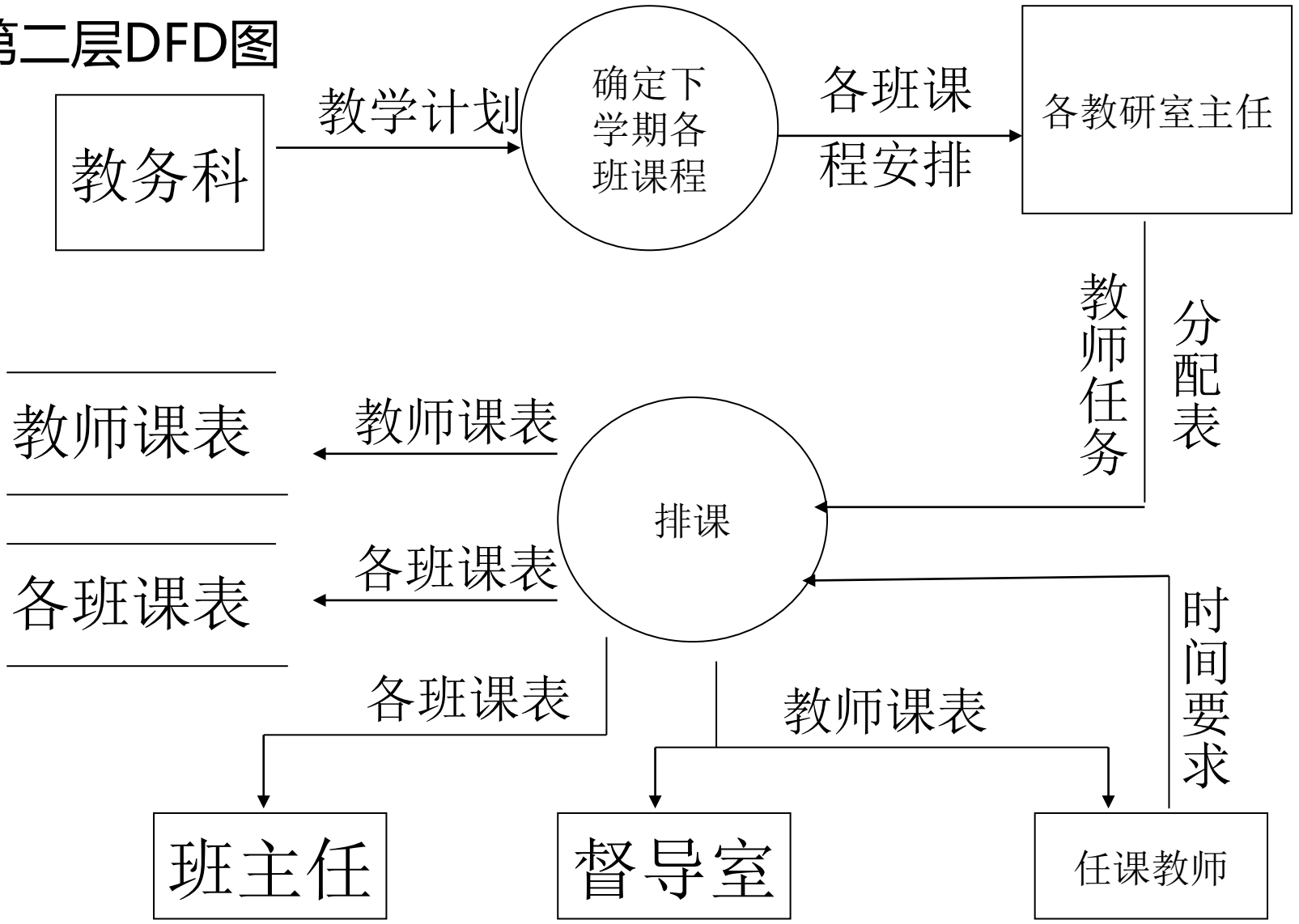
数据流图的示例

- 第一层DFD图



数据流图的示例

- 第二层DFD图



数据字典

- 数据字典(Data Dictionary)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

数据字典

- 数据字典中有五类条目：
  - 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。

数据字典中常用的符号

- +表示“与”；
- [|]表示“或”，即选择括号中的某一项；
- {}表示“重复”，即括号中的项要重复若干次；
- []表示“可选”，即括号中的项可有可无

数据字典的内容

- 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

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}

数据字典的内容

- 1 数据项

数据项名称：CourseName

说明：课程的名称

别名：课程名

数据类型：String

数据长度：30

数据字典的内容

- 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} }

数据字典的内容

- 2 数据结构

数据结构名称：教师课表

说明：记录教师上课时间

组成：{教师名称＋课程名称＋周次＋星期几＋时间}

数据字典的内容

- 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

数据字典的内容

3. 数据流

数据流名称：教师课表

说明：下学期教师上课时间表

数据流来源：排课

数据流去向：任课教师、督导室、 D1 教师课表

数据流组成：{教师姓名+课程名+星期 +节次}

数据流的流通量：300份/学期

数据字典的内容

- 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

需求分析结果

- 需求分析最终结果是一份需求规格说明书
- 需求规约是概念设计的最主要依据
- 需求规约模板
  - (1) 系统概况，系统的目标、范围、背景、历史和现状；
  - (2) 系统的原理和技术，对原系统的改善；
  - (3) 系统总体结构与子系统结构说明；
  - (4) 系统功能说明；
  - (5) 数据处理概要、工程体制和设计阶段划分；
  - (6) 系统方案及技术、经济、功能和操作上的可行性

第十一章 数据库设计

1 概述

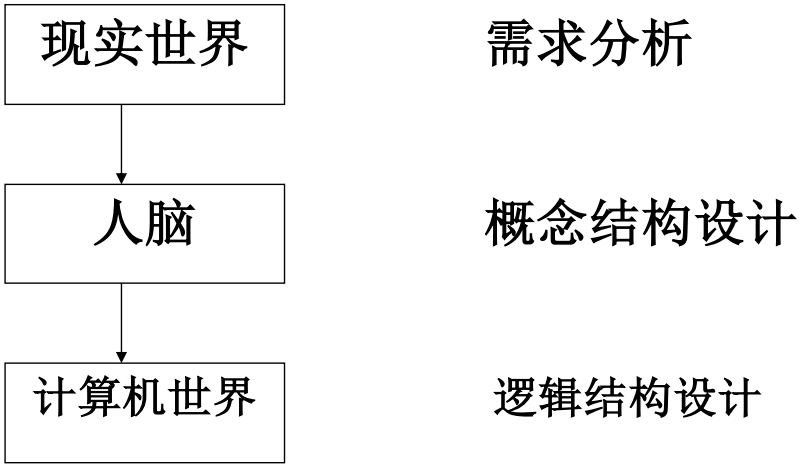
2 需求分析

3 概念结构设计

4 逻辑结构设计

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
- 概念结构是各种逻辑模型的共同基础，它比逻辑模型更独立于机器、更抽象，从而更加稳定。



概念结构设计的特点

- 能真实、充分地反映现实世界，包括事物和事物之间的联系，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。
- 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。
- 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。

概念结构设计的方法和步骤

- 自底向上概念结构设计
  - 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
    - ✓ 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

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

概念结构设计内容

- 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
  - 这些概念组成了概念数据模型
  - 常用抽象方法：分类，聚集，概括
- 相关描述参见ER模型章节

概念结构设计内容

- 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

概念结构设计内容

- 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。
  - 通常以中层数据流图作为设计分E-R图的依据。原因：
    - ✓ 高层数据流图只能反映系统的概貌，而低层数据流图过细
    - ✓ 中层数据流图能较好地反映系统中各局部应用的子系统组成

概念结构设计内容

- 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）
    - ✓ 注意区分实体和属性 (准则参见ER模型一章)

概念结构设计内容

- 2 逐一设计分E-R图
  - 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。

概念结构设计内容

- 2 逐一设计分E-R图
  - 设计分E-R图的步骤
    - ✓ 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：

1). 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。

2). 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。

概念结构设计内容

- 2 逐一设计分E-R图
  - 设计分E-R图的步骤
    - ✓ 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。

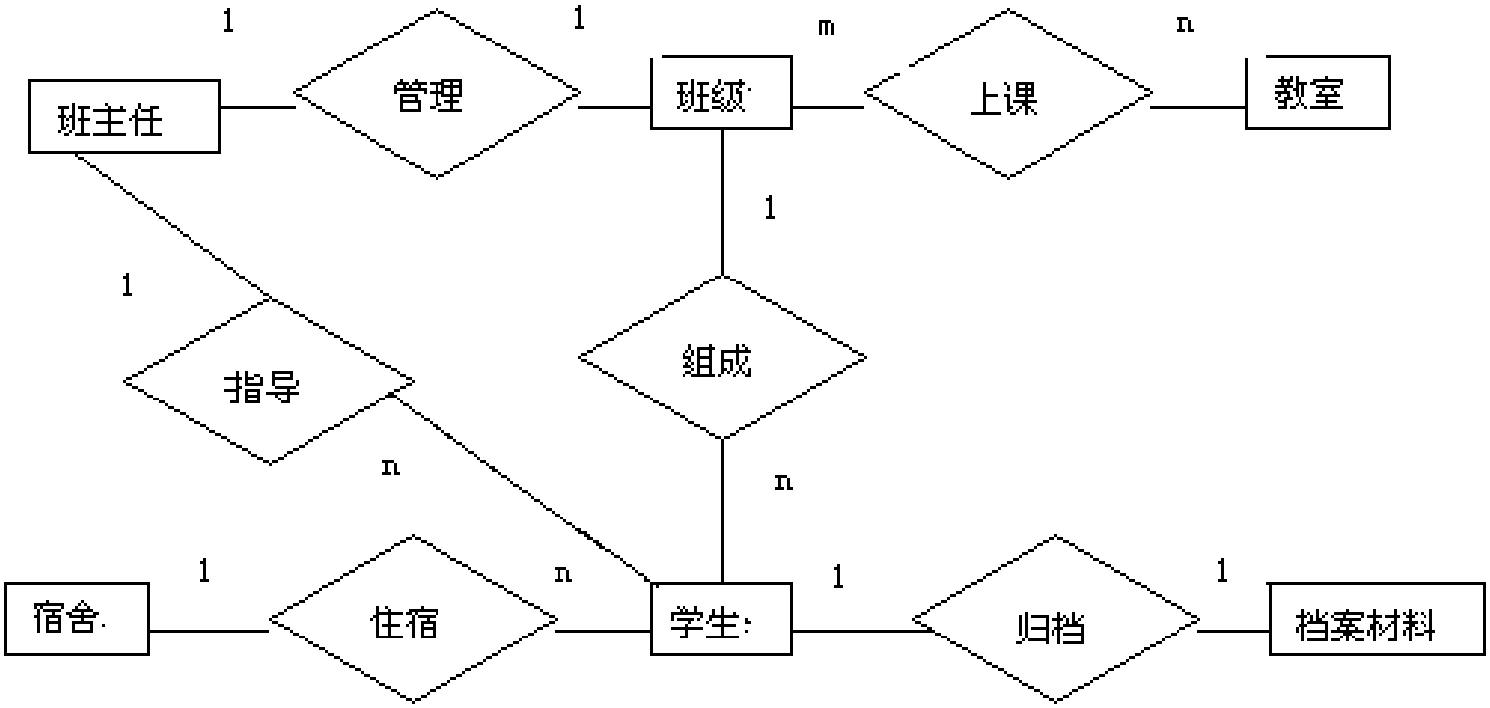
实体之间的联系：

3). 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。

4). 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。

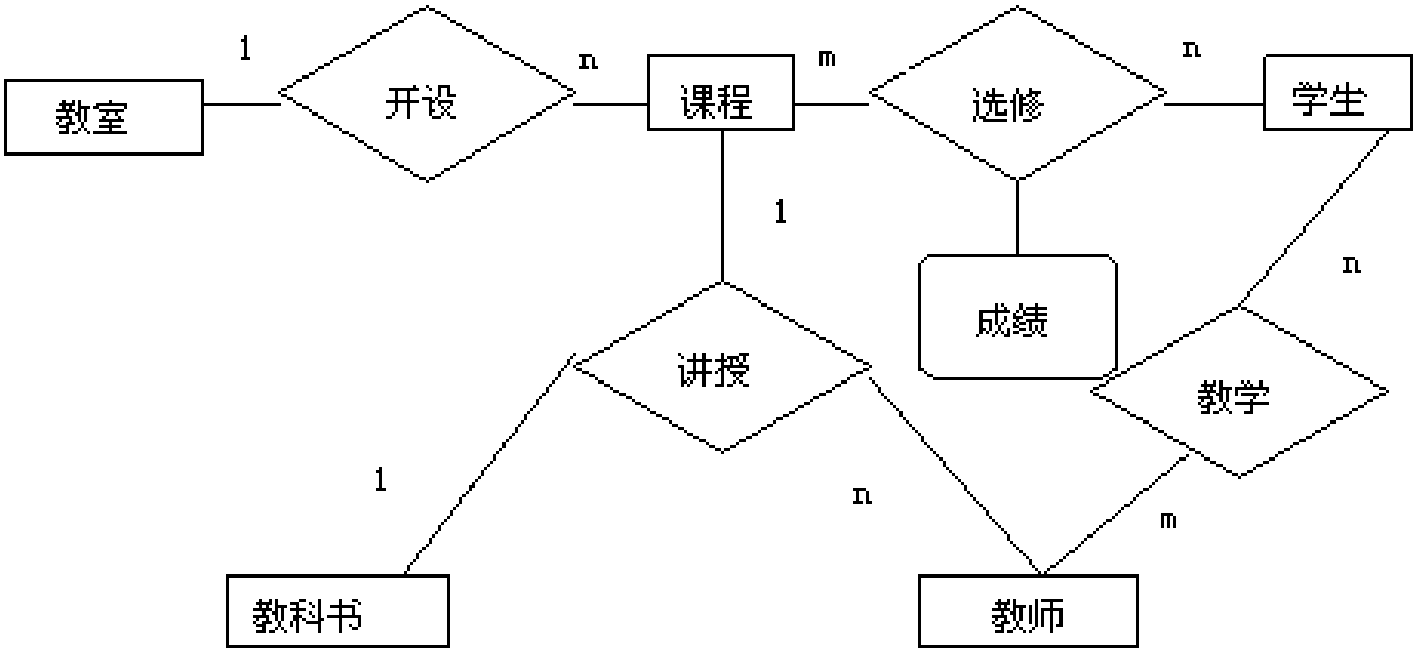
概念结构设计内容

最后得到学籍管理局部应用的分E-R图



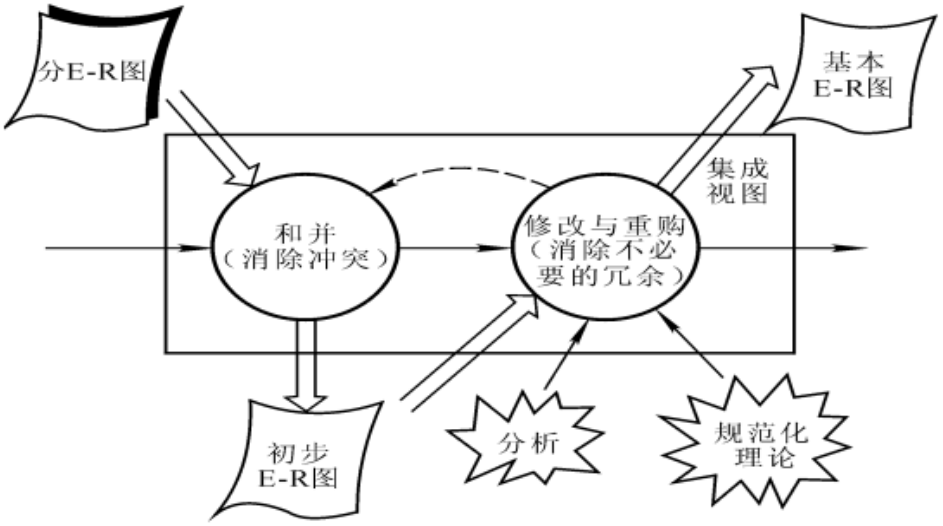
概念结构设计内容

同样方法可以得到课程管理局部应用的分E-R图



概念结构设计内容

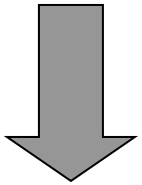
- 3. 集成局部视图，得到全局概念结构
  - 各个局部视图即分E-R图建立好后，还需要对它们进行合并，集成为一个整体的数据概念结构即总E-R图。
  - 集成的步骤：
    - ✓ 1. 合并
    - ✓ 2. 修改与重构



概念结构设计内容

- 1. 集成局部视图，得到全局概念结构
  - 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

概念结构设计内容

- 冲突的种类
  - 1. 属性冲突---讨论、协商解决
    - ✓ 属性域冲突：属性值的类型、取值范围或取值集合不同。

例：由于学号是数字，因此某些部门（即局部应用）将学号定义为整数形式，而由于学号不用参与运算，因此另一些部门（即局部应用）将学号定义为字符型形式。

- ✓ 属性取值单位冲突

例：学生的身高，有的以米为单位，有的以厘米为单位，有的以尺为单位

概念结构设计内容

- 冲突的种类
  - 2. 命名冲突---讨论、协商解决
    - ✓ 同名异义：不同意义的对象在不同的局部应用中具有相同的名字

    例：局部应用A中将教室称为房间，局部应用B中将学生宿舍称为房间

    - ✓ 异名同义（一义多名）：同一意义的对象在不同的局部应用中具有不同的名字

    例：有的部门把教科书称为课本，有的部门则把教科书称为教材

概念结构设计内容

- 冲突的种类
  - 3. 结构冲突
    - ✓ 同一对象在不同应用中具有不同的抽象

      例，“课程”在某一局部应用中被当作实体，在另一局部应用中则被当作属性

      解决方法：通常是把属性变换为实体或把实体变换为属性，使同一对象具有相同的抽象。

概念结构设计内容

- 冲突的种类
  - 3. 结构冲突
    - ✓ 同一实体在不同局部视图中所包含的属性不完全相同，或者属性的排列次序不完全相同

    产生原因：不同的局部应用关心的是该实体的不同侧面。

    解决方法：使该实体的属性取各分E-R图中属性的并集，再适当设计属性的次序

概念结构设计内容

- 冲突的种类
  - 3. 结构冲突
    - ✓ 实体之间的联系在不同局部视图中呈现不同的类型

例1， 实体E1与E2在局部应用A中是多对多联系，而在局部应用B中是一对多联系

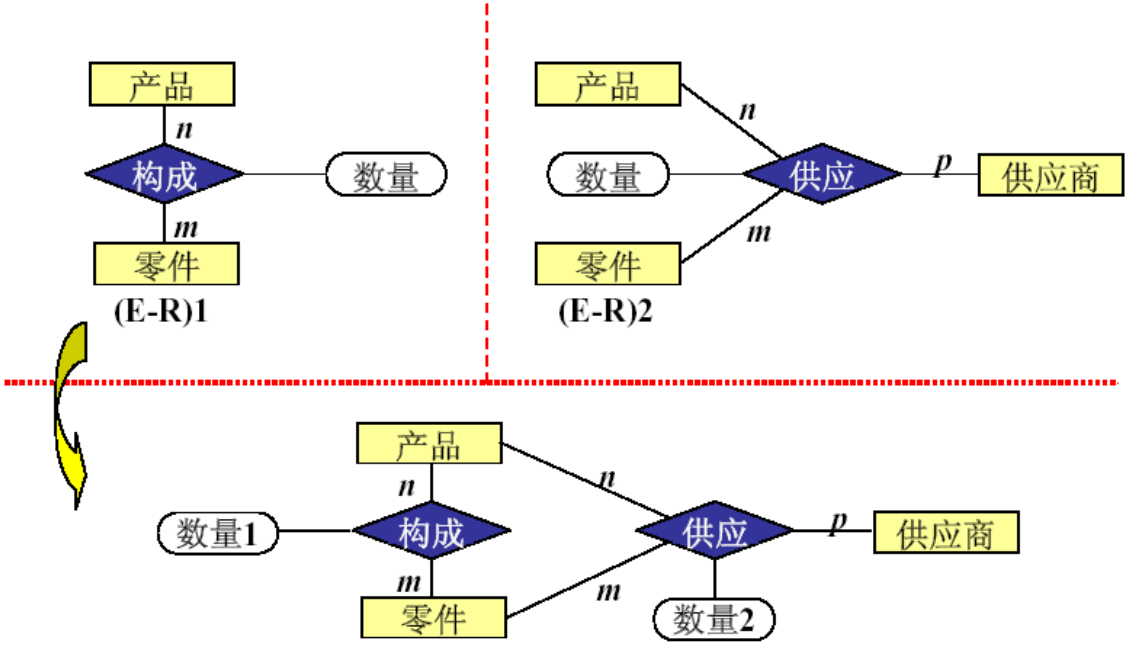
例2， 在局部应用X中E1与E2发生联系，而在局部应用Y中E1、E2、E3三者之间有联系。

解决方法：根据应用语义对实体联系的类型进行综合或调整。

概念结构设计内容

- 冲突的种类
  - 3. 结构冲突
    - ✓ 实体之间的联系在不同局部视图中呈现不同的类型

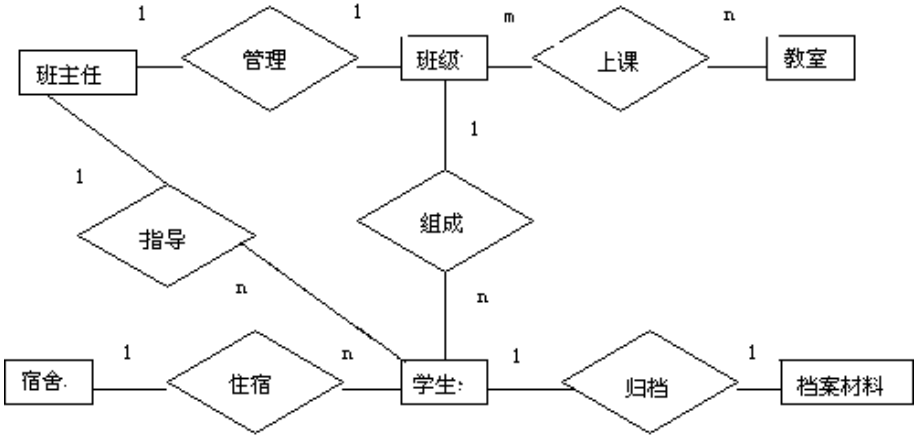
例：



概念结构设计内容

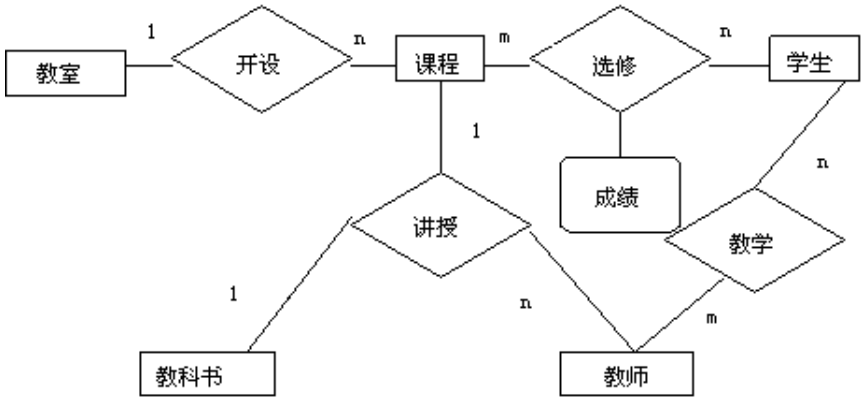
- 合并分E-R图，生成初步E-R图实例

  合并前例：学籍管理局部视图, 课程管理局部视图



学籍管理局部视图

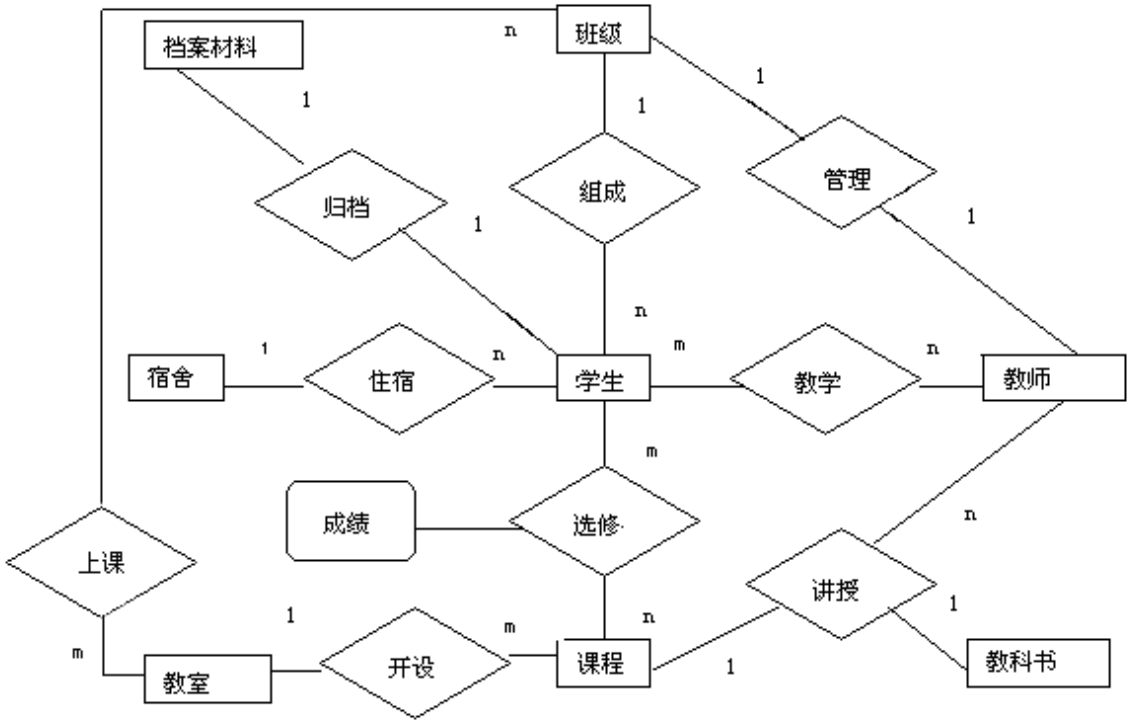
课程管理局部视图



概念结构设计内容

- 合并分E-R图，生成初步E-R图实例

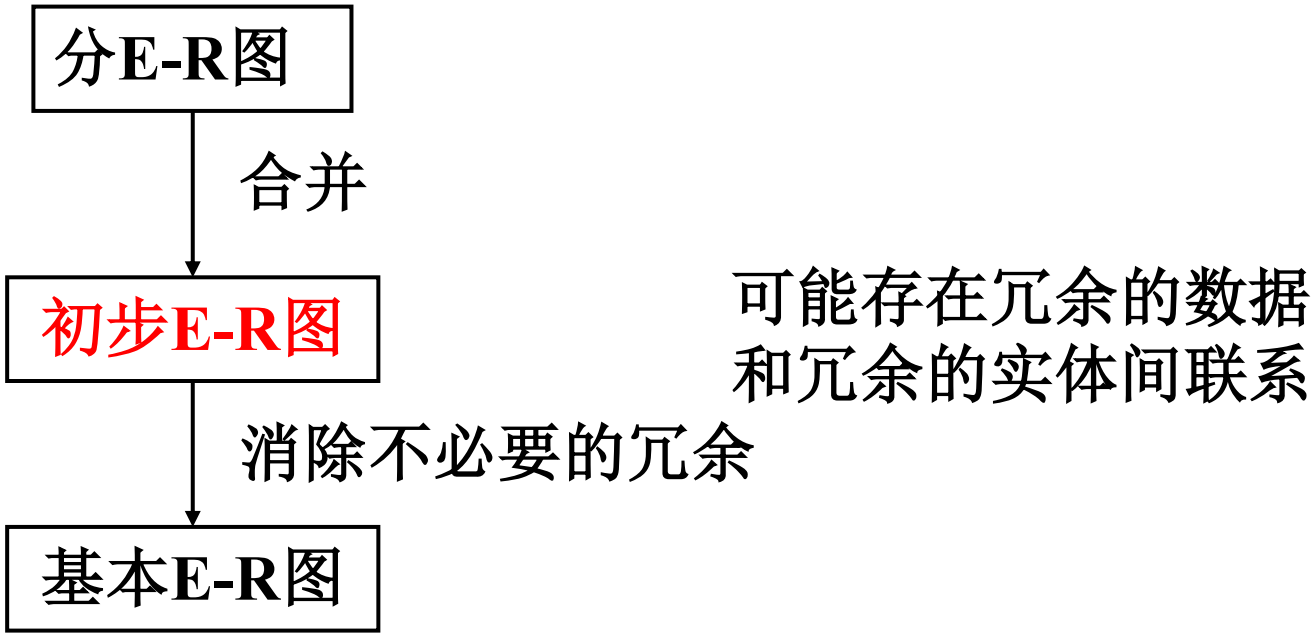
  合并后获得学生管理系统初步ER图



学生管理系统初步ER图

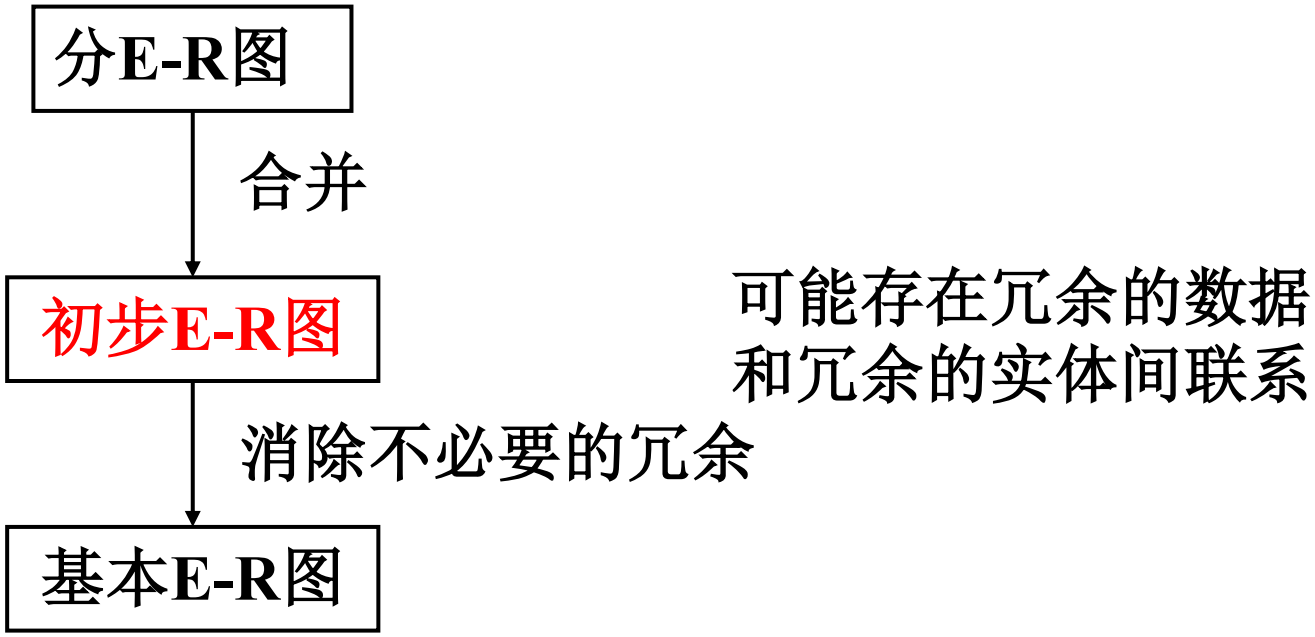
概念结构设计内容

- 2. 修改与重构
  - 在初步ER图基础上，消除不必要的冗余，设计生成基本E-R图



概念结构设计内容

- 2. 修改与重构
  - 在初步ER图基础上，消除不必要的冗余，设计生成基本E-R图



概念结构设计内容

- 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。
  - 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。

概念结构设计内容

- 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。
    - ✓ 例：教师工资单中包括该教师的基本工资、各种补贴、应扣除的房租水电费以及实发工资。由于实发工资可以由前面各项推算出来，因此可以去掉。
  - 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

概念结构设计内容

- 消除冗余的方法---
  - 规范化理论方法
    - ✓ 1. 确定分E-R图实体之间的数据依赖F_L。实体之间一对一、一对多、多对多的联系可以用实体键之间的函数依赖来表示

例:

班级和学生之间一对多的联系:

学号→班级号

学生和课程之间多对多的联系:

(学号，课程号) →成绩

概念结构设计内容

- 消除冗余的方法---
  - 规范化理论方法
    - ✓ 2. 求FL的最小覆盖G_L，差集为

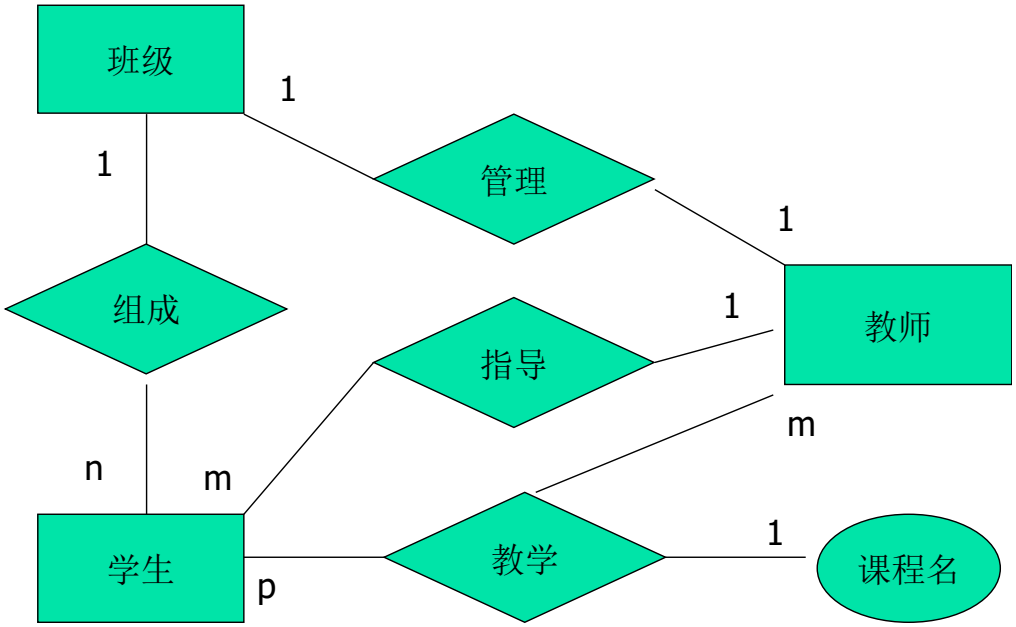
      $D = F_L - G_L$。

      逐一考察D中的函数依赖，确定是否是冗余的联系，若是，就把它去掉。
    - ✓ 注意：冗余的联系一定在D中，而D中的联系不一定是冗余的，需要根据需求定义加以鉴别

概念结构设计内容

- 消除冗余的方法---
  - 规范化理论方法

    发现和消除冗余联系



概念结构设计内容

- 消除冗余的方法---
  - 规范化理论方法
    - ✓ 找到描述实体间联系的函数依赖集F：

      学生->班级; 班级<->教师;

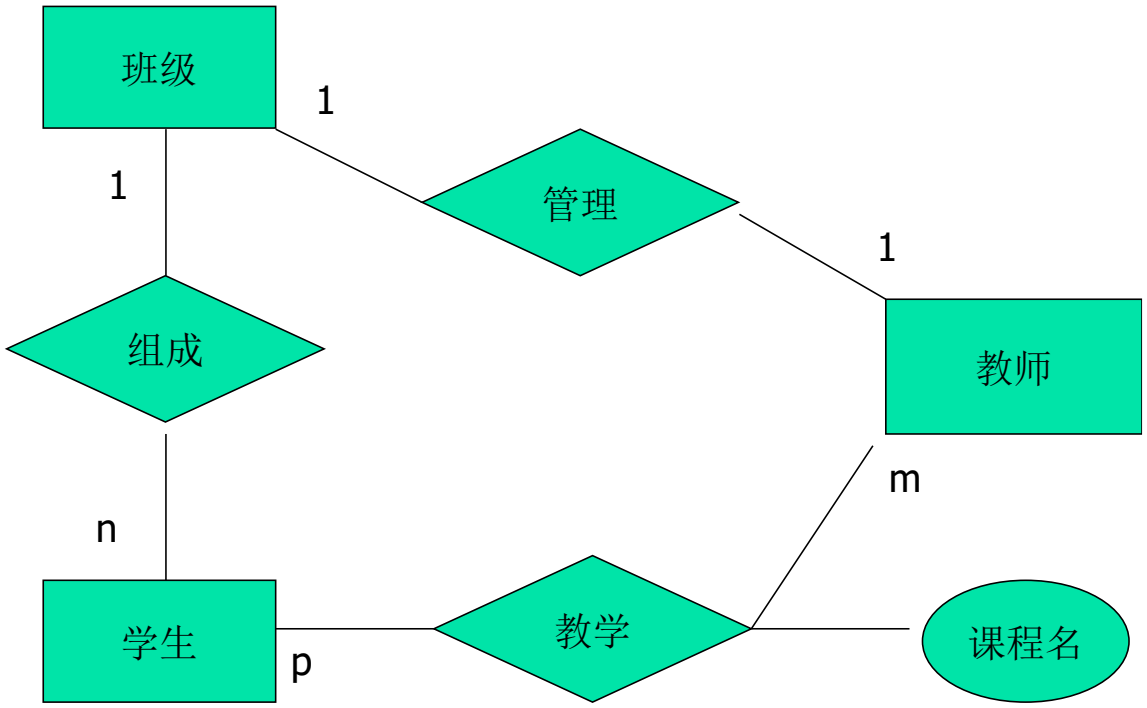
      (学生,教师)->课程名; 学生->教师
    - ✓ 计算F的最小函数依赖集F`：

      学生->班级; 班级<->教师;

      (学生,教师)->课程名;
    - ✓ F-F`={学生->教师}，因此学生与教师的指导联系是冗余联系

概念结构设计内容

- 消除冗余的方法---
  - 规范化理论方法
    - ✓ 消除冗余联系后的基本ER图



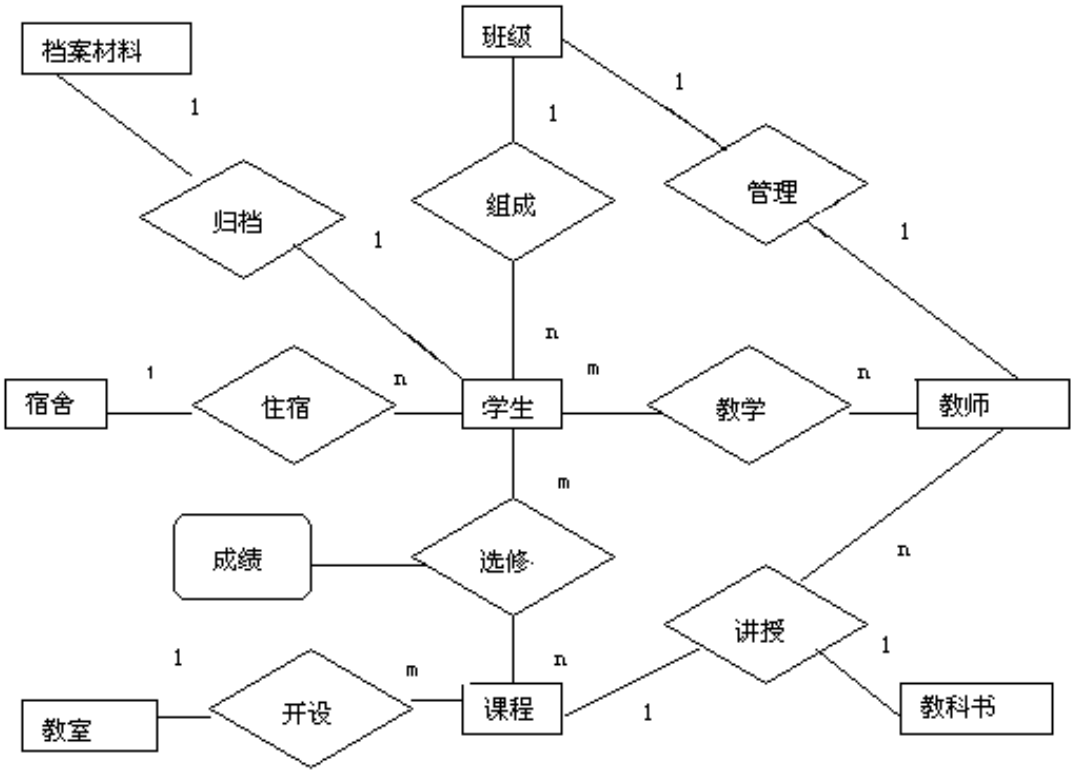
概念结构设计内容

- 3. 数据模型的优化
  - 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。

- 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（是否需要用到非规范化设计手段？）。

概念结构设计内容

对学生管理系统分ER图进行修改和重构后生成的基本E-R图



概念结构设计内容

- 整体概念结构最终还应该提交给用户，征求用户和有关人员的意见，进行评审、修改和优化，然后把它确定下来，作为数据库的概念结构，作为进一步设计数据库的依据

第十一章 数据库设计

1 概述

2 需求分析

3 概念结构设计

4 逻辑结构设计

逻辑结构设计的任务

- 概念结构是各种数据模型的共同基础
- 为了能够用某一DBMS实现用户需求，还必须将概念结构进一步转化为相应的数据模型，这正是数据库逻辑结构设计所要完成的任务。
- 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

逻辑结构设计的内容

- 1. E-R图向关系模型的转换
- 2. 数据模型的优化
- 3. 设计用户子模式

逻辑结构设计的内容

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

逻辑结构设计的内容

- 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1:1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
  - 4. 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
  - 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. 具有相同键的关系模式可合并

逻辑结构设计的内容

- 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

逻辑结构设计的内容

- 数据模型优化的方法
  - 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。
  - 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. 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。

逻辑结构设计的内容

- 设计用户子模式（定义视图）
  - 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：
    - ✓ 使用更符合用户习惯的别名
    - ✓ 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
    - ✓ 简化用户对系统的使用
    - ✓ 具体含义和实现方法参见视图一节

作业

- 课本241页第1，3，4，5，6，7，8，9，10，11，12
- 下次上课前交