

第五章 数据库设计



主要内容

- 需求分析
- 概念结构设计
- 逻辑结构设计
- 数据库的物理设计
- 数据库的实施
- 数据库运行和维护
- <http://39.102.55.155:10099>



5.1 数据库设计概述

● 数据库设计

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，使之能有效地存储数据，满足各种用户的应用需求。

- 信息需求：数据库内容及结构的要求，静态
- 处理需求：数据库要进行的数据处理，动态

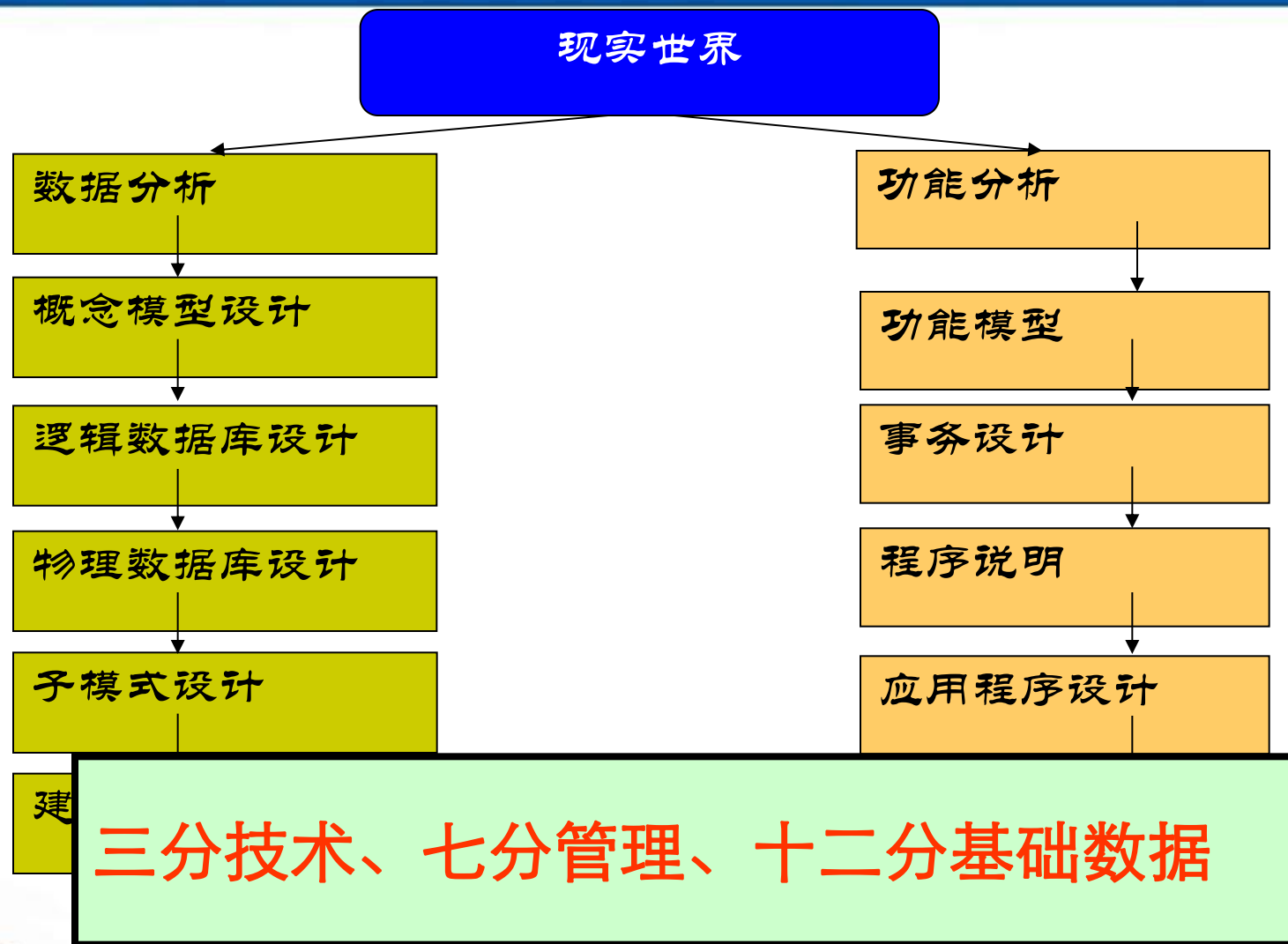


数据库设计人员应具备的技术和知识

- 1、计算机科学基础知识和程序设计技术
- 2、DB基本知识和DB设计技术
- 3、软件工程的原理和方法
- 4、应用领域的知识




5.1.2 数据库设计的内容



5.1.3、数据库的设计方法

- 直观设计法
- 规范设计法
- 计算机辅助设计法
- 自动化设计法。



基于实体联系的设计方法、基于3NF的数据库设计方法、基于视图概念的数据库设计方法等。



5.1.4、数据库设计的基本步骤

➤ **需求分析：**是整个设计过程的基础。

(1)信息需求分析 (2)操作需求分析

➤ **概念结构设计：**将需求分析的结果用一种工具进行形式化的定义和描述，是整个DB设计的关键。如：E-R图

➤ **逻辑结构设计：**概念模型→数据模型

根据需求分析和概念设计的结果，选择合适的数据模型，并选用某一设计方法构造一数据库模式。



5.1.4、数据库设计的基本步骤

➤ 4、物理设计：选择合适的物理结构。

包括存储结构和存取方法

➤ 5、实施阶段：根据逻辑设计和物理设计建立数据库；编制和调试应用程序；试运行；

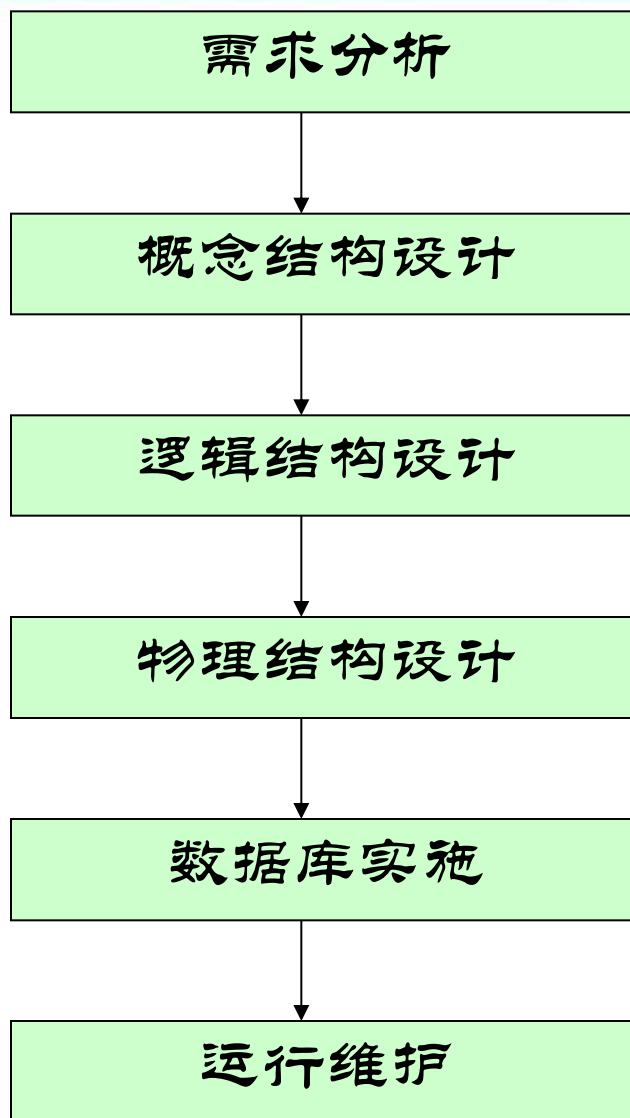
➤ 6、DB运行和维护阶段

根据运行记录对DB进行评价

根据评价对DB调整和修改



数据库设计的基本步骤



5.2 需求分析

需求分析的任务是通过详细调查现实世界要处理的对象，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，然后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善，不能仅仅按当前应用需求来设计数据库。



5.2.1、需求分析的任务

- 信息要求：DB中需存储的数据
- 处理要求：
 - 用户要求的处理功能
 - 对各处理的响应时间的要求
- 安全性与完整性要求

确定用户的最终需求是
非常困难的！



5.2.2、需求分析的方法和过程

1. 调查用户需求的具体步骤:

- 分析用户活动，产生业务流程图。
- 确定系统范围，产生业务范围图。
- 分析用户活动及所设计的数据，产生数据流图
- 分析系统数据，产生数据字典



2、需求分析的常用方法

- 跟班作业
- 开调查会
- 请专人介绍
- 询问
- 调查表
- 查阅记录

强调：

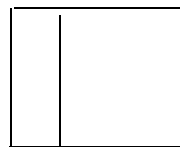
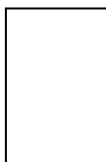
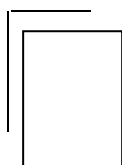
- 数据的收集要尽可能详尽，全面反映用户需求。
- 要考虑可能的扩充和改变。
- 必须强调用户的参与。



5.2.3、需求分析常用工具

1. 数据流图（DFD）

DFD由四种基本符号组成。如下图所示。



外部项 (S)

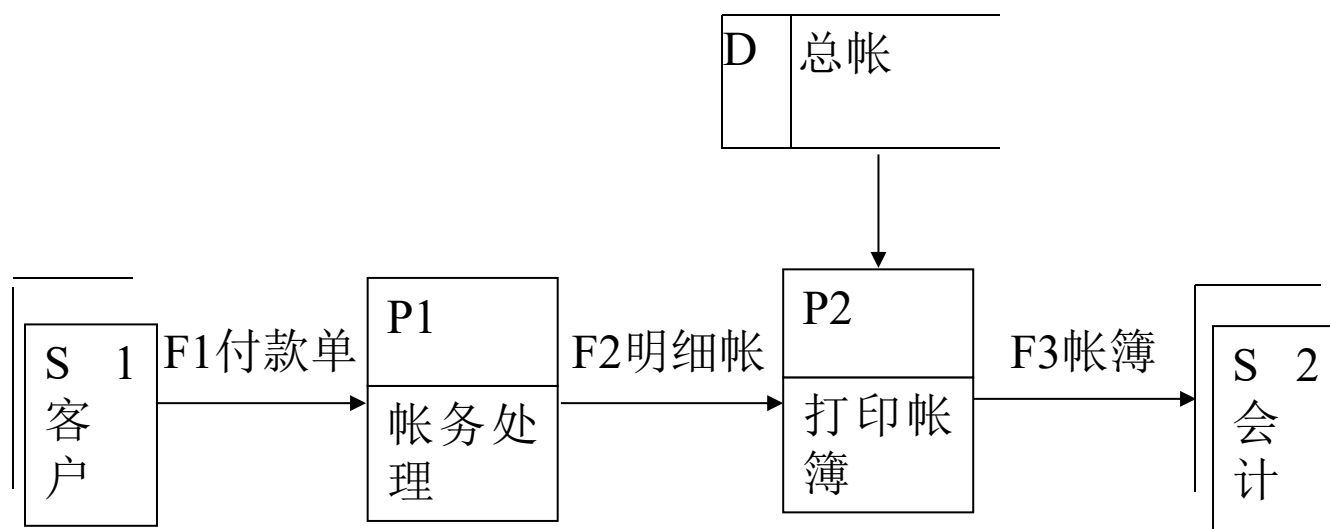
数据加工 (P)

数据存储 (D)

数据流 (F)

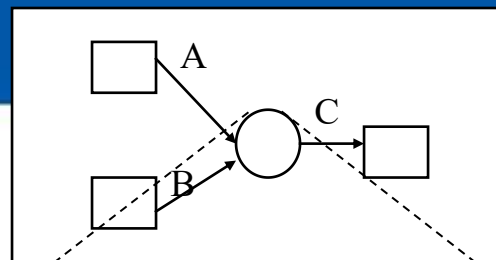


下图是一个简单的DFD。它表示数据流“付款单”从外部项“客户”（源点）流出，经加工“帐务处理”转换成数据流“明细帐”，再经加工“打印帐簿”转换成数据流“帐簿”，最后流向外部项“会计”（终点），加工“打印帐簿”在进行转换时，从数据存储“总帐”中读取数据。

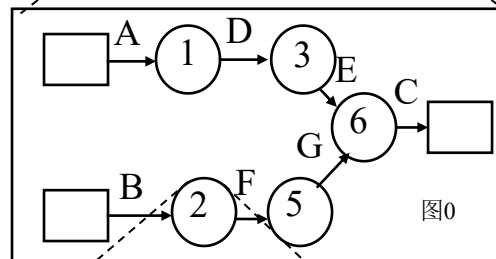


绘制数据流图的过程

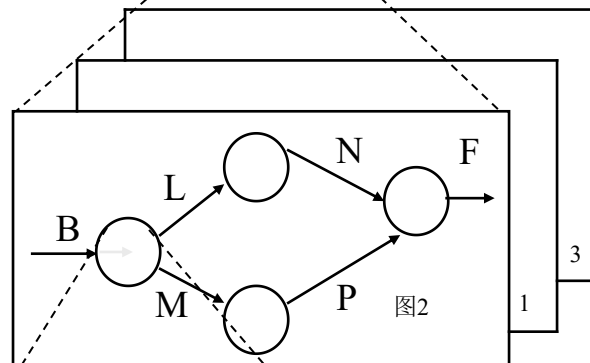
a 关联图



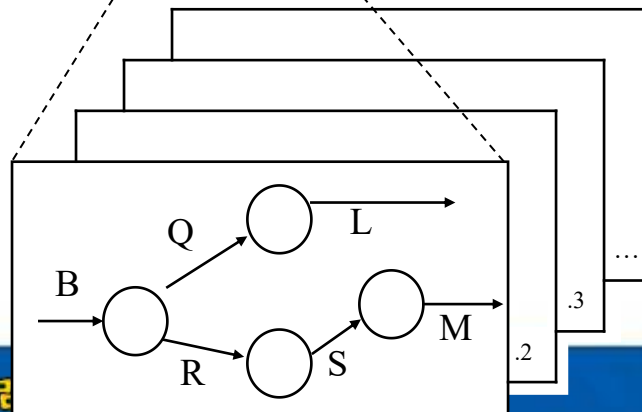
b 顶层图



c 二层图



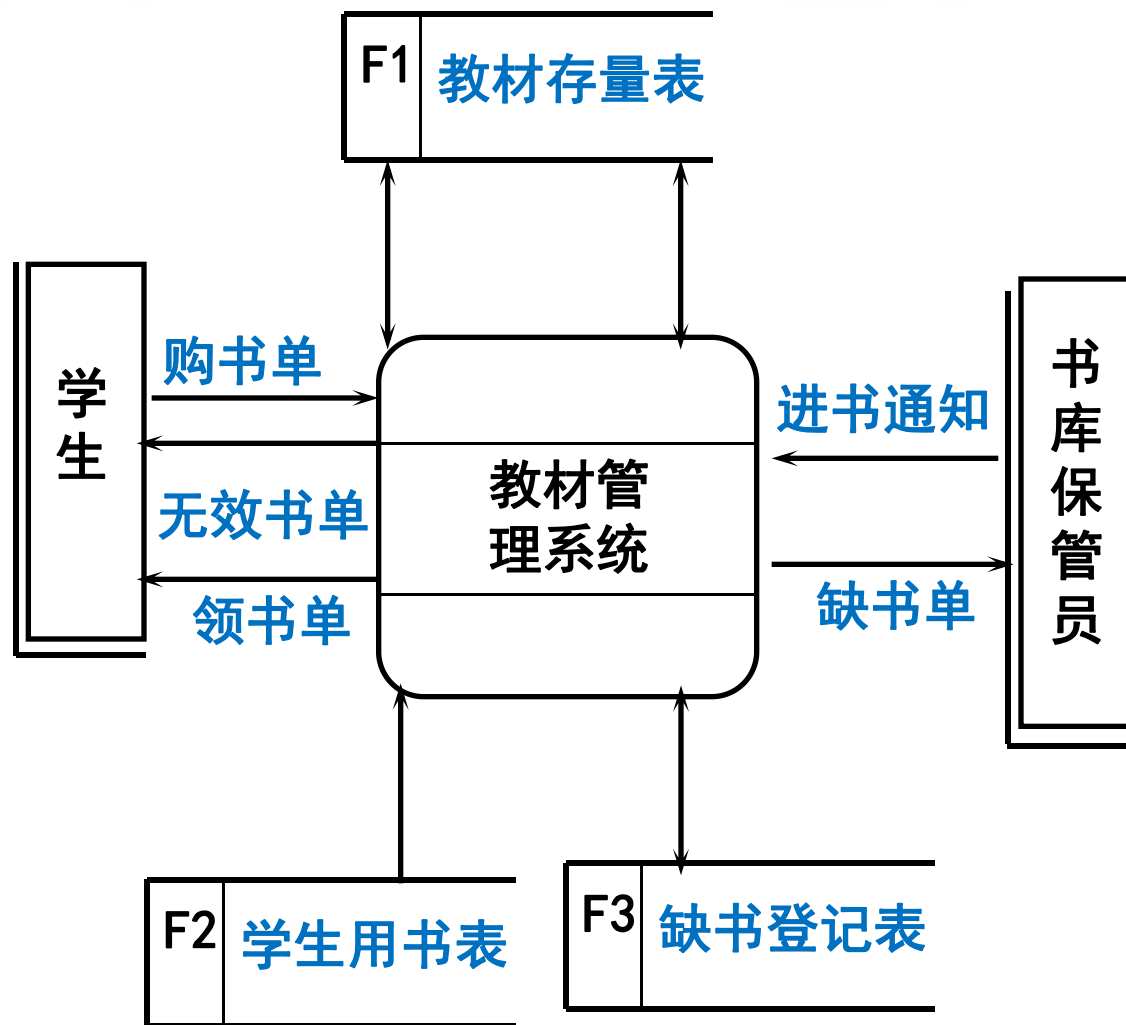
d 三层图



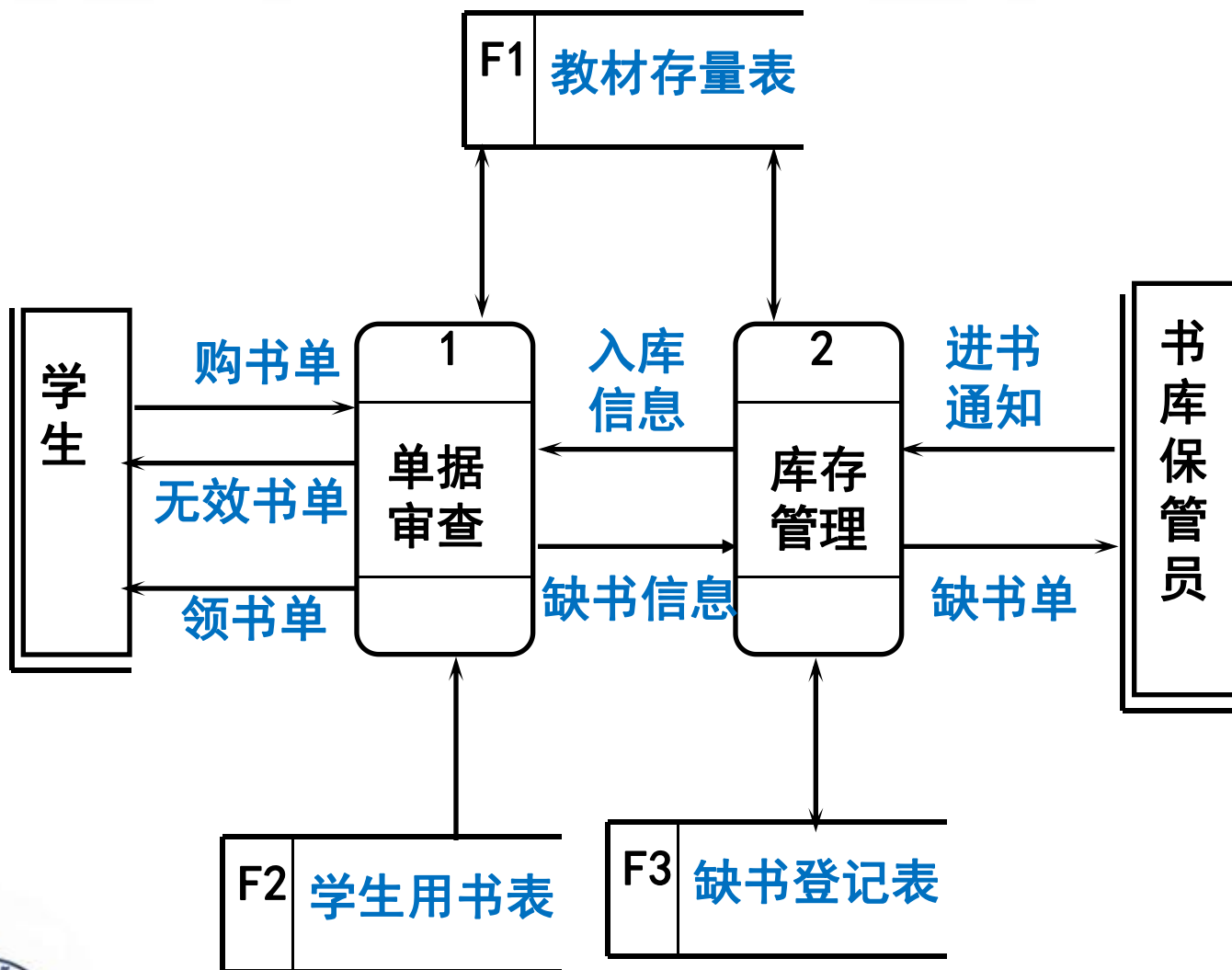
绘制数据流图过程示意图



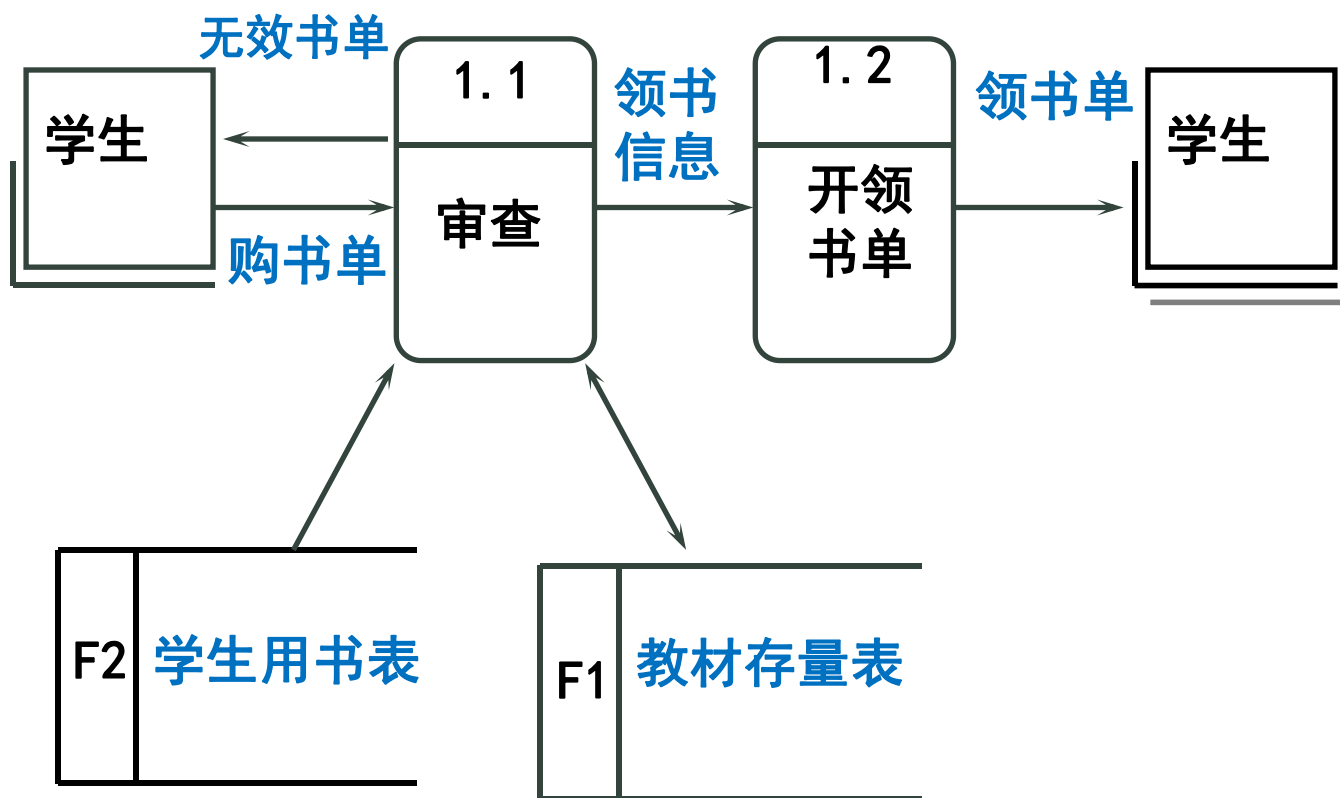
顶层数据流图



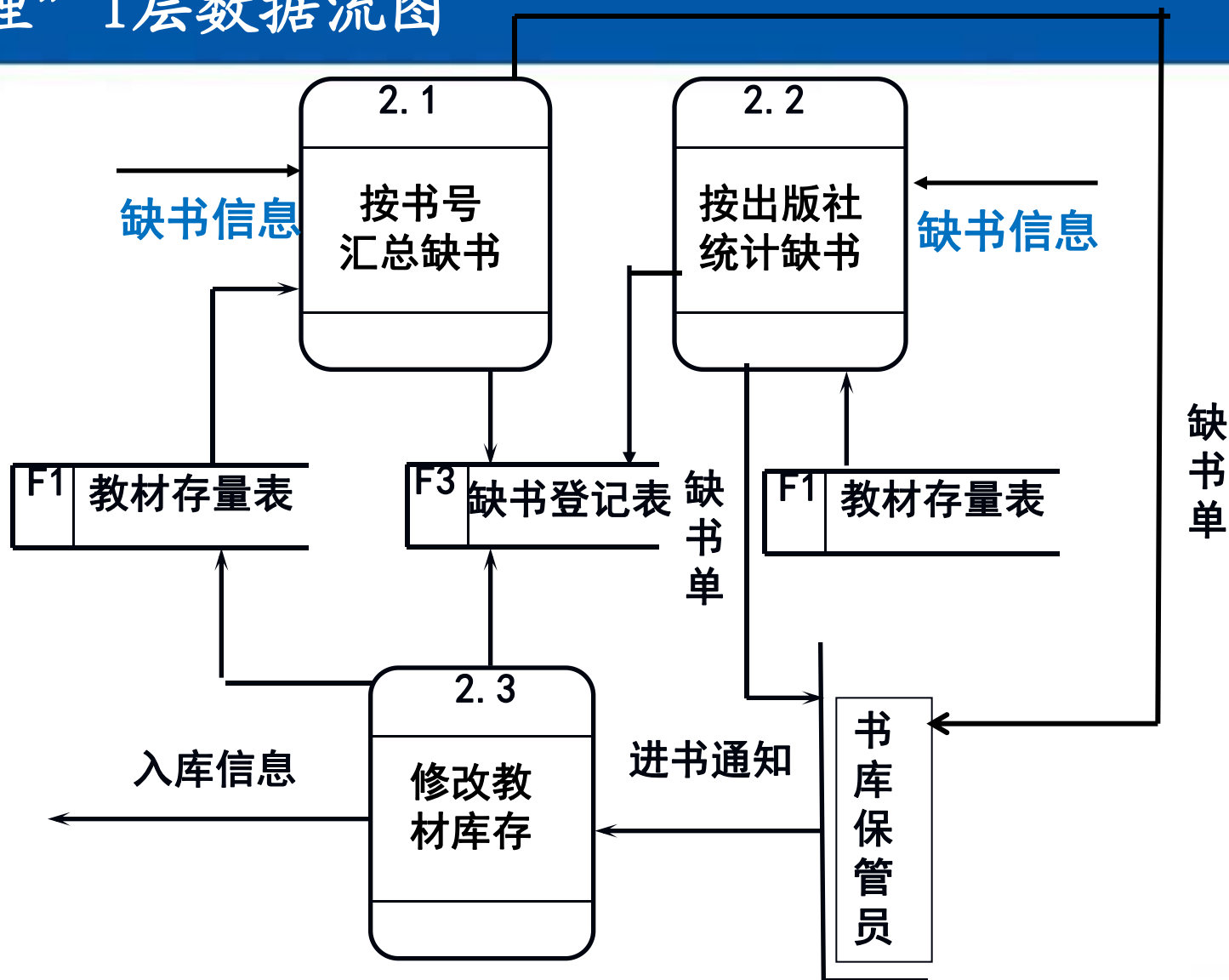
0层数据流图



“单据审查” 1层数据流图



“库存管理” 1层数据流图



2. 数据字典（DD）

- 数据字典是关于数据的数据库，它是对数据流程图上各个元素作出详细的定义和说明。

数据字典	数据项
	数据结构
	数据流
	数据存储
	处理过程



(1). 数据项

➤ 数据项是不可再分的数据单位。

➤ 数据项的描述=

{数据项名, 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, 数据项之间的联系}

- 数据项名: 学号
- 别名: 学生编号
- 取值: 8{数字}8
- 注释: * 例如: 20110101 *



(2). 数据结构

➤ 数据结构反应了数据之间的组合关系。

➤ 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}}

➤ 数据流名: 领书单

➤ 别名: 购书发票

➤ 组成: (学号) + 姓名 + 书号 + 单价 + 数量 + 总价



(3). 数据流

➤ 数据流是数据结构在系统内传输的途径。

➤ 数据流的描述=

{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}

➤ 数据流名: 领书信息

➤ 数据流来源: 审查

➤ 数据流去向: 开领书单

➤ 组成: (学号) + 姓名 + 书号 + 单价 + 数量 + 总价

➤ 数据量: 100次/天

➤ 高峰值: 开学期间400次/天



(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。它可以是手工文档或手工凭单，也可以是计算机文档。

➤数据存储描述=

{数据存储名，说明，编号，输入的数据流，输出的数据流，组成{数据结构}，数据量，存取频度，存取方式}

➤文件名：学生用书表

➤输入的数据流：学生用书申请信息

➤输出的数据流：学生用书信息

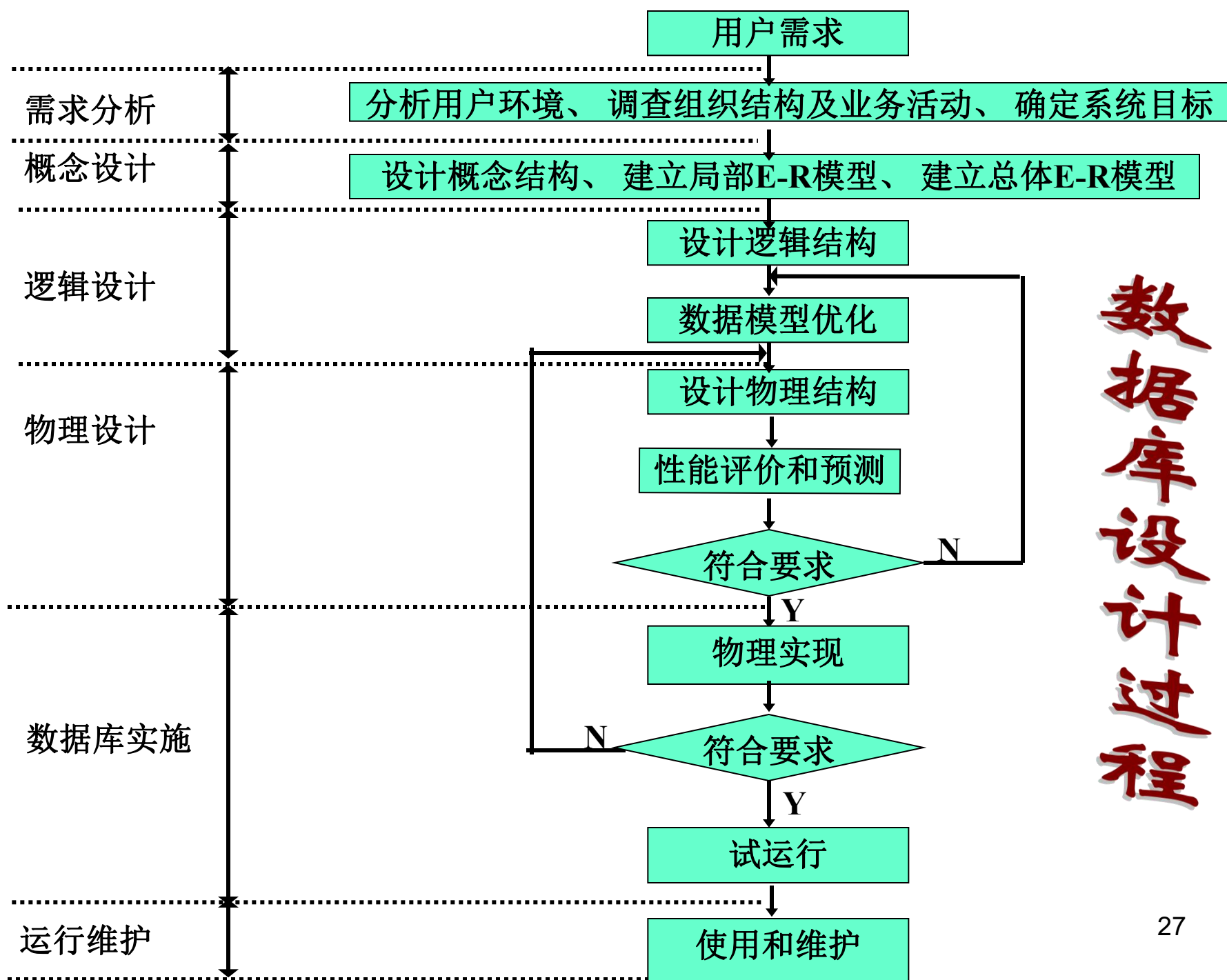
➤组成：{系编号+专业和班编号+年级+{书号}}

➤存取要求：关键字是专业和班编号

(5). 处理过程

- 加工名： 开领书单
- 编号： 2.0
- 加工激活条件： 得到图书采购员采购操作命令
- 加工逻辑：2.0采购= {2.1 按书号汇总缺书
+ 2.2 按出版社统计缺书
+ 2.3 修改教材库存和待购量}
- 执行频率： 随时



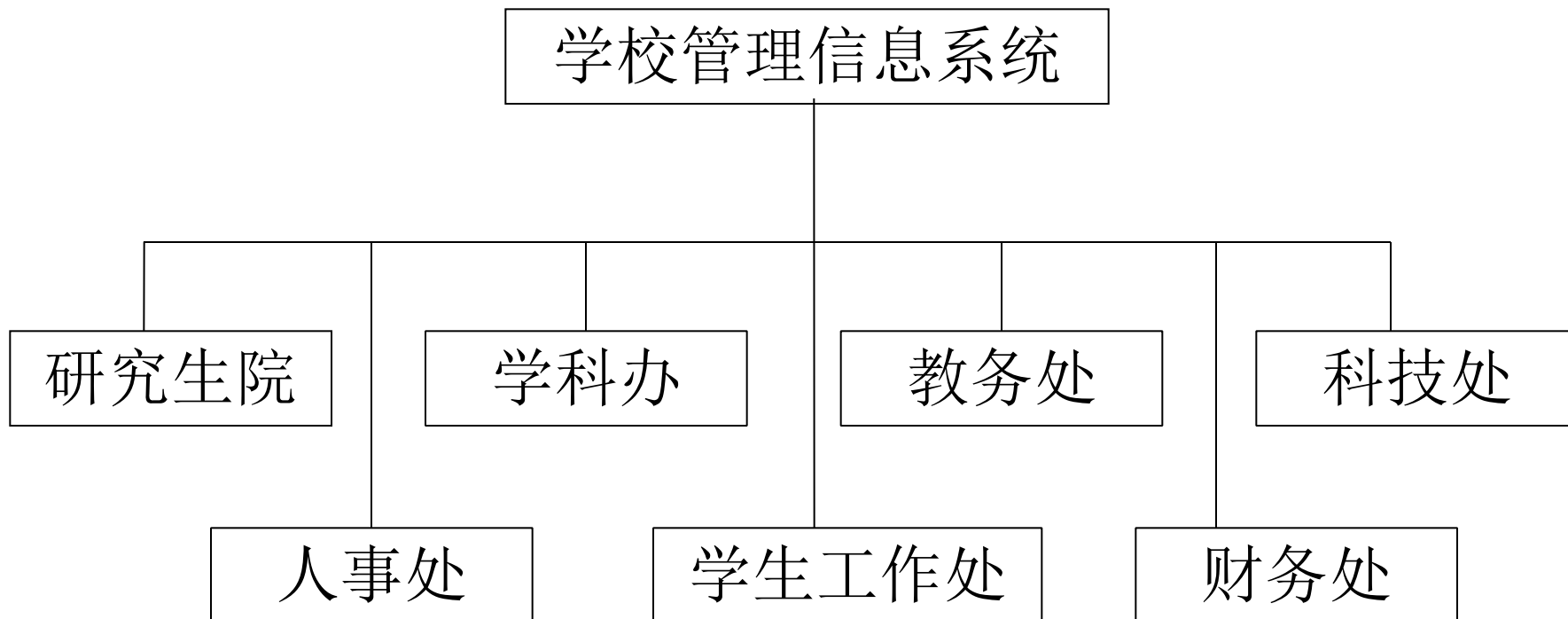


数据库设计过程

学校管理信息系统

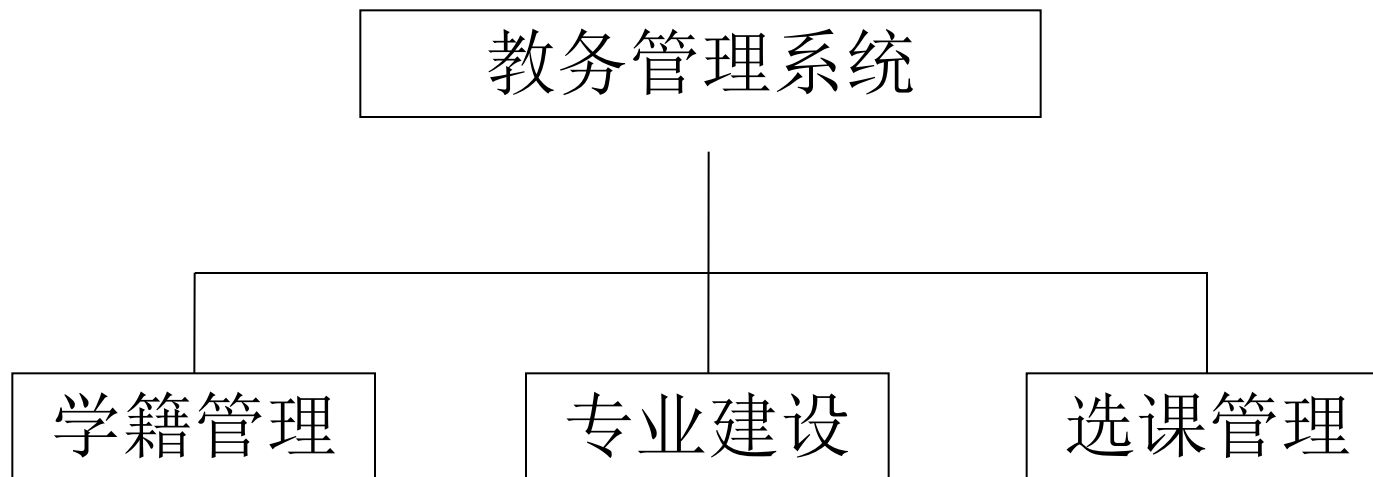


学校管理信息系统



学校组织结构图

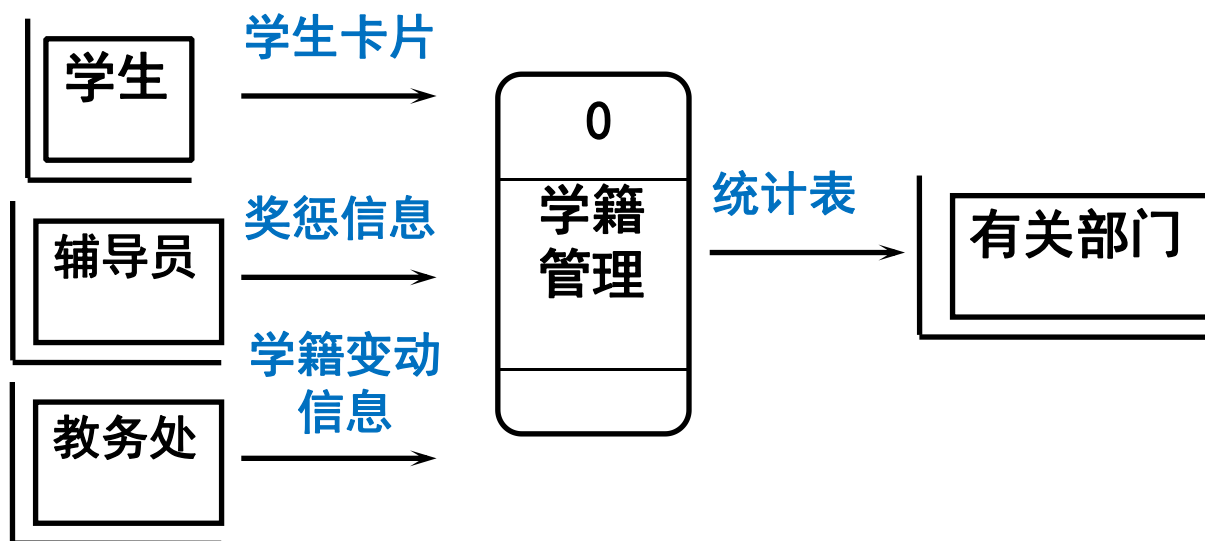




教务管理系统功能结构图



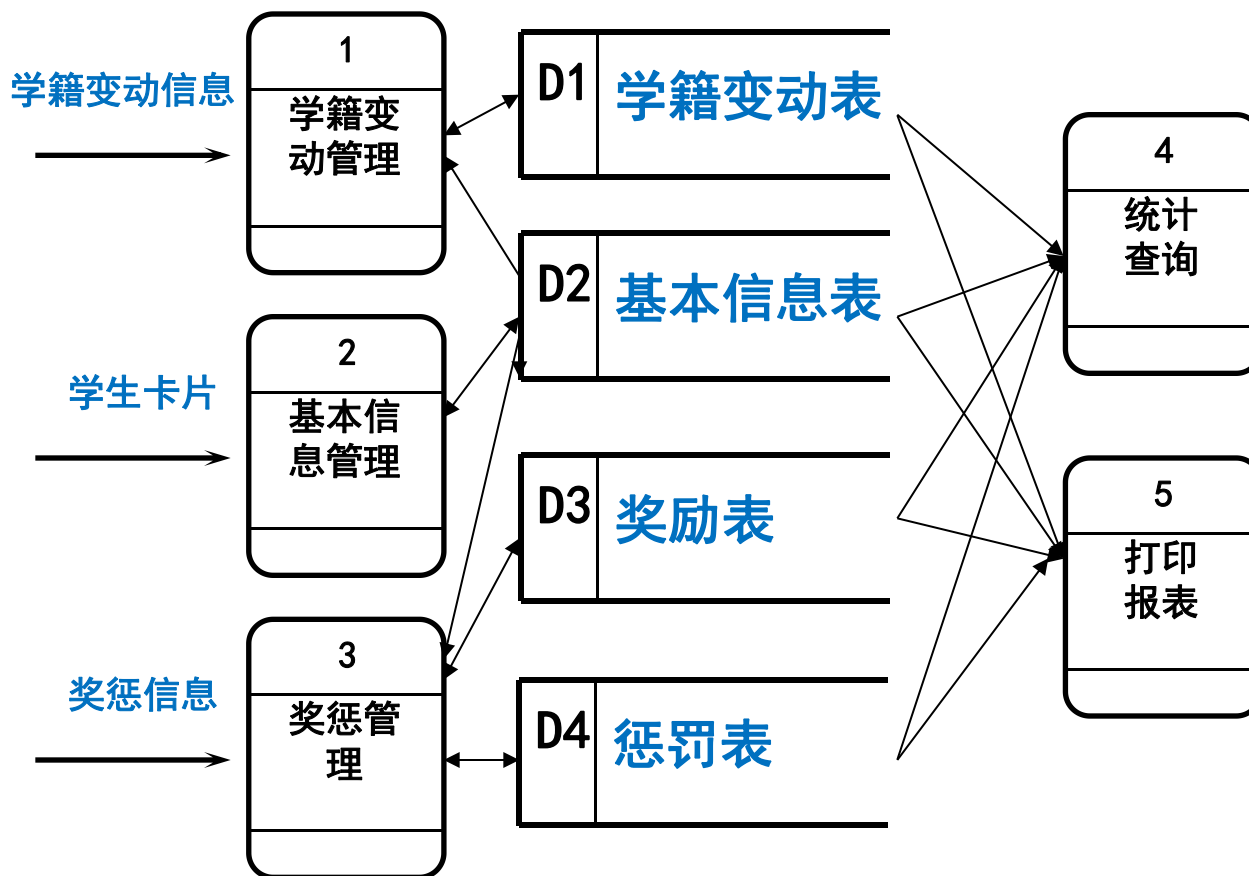
顶层数据流图



学籍管理系统顶层数据流图



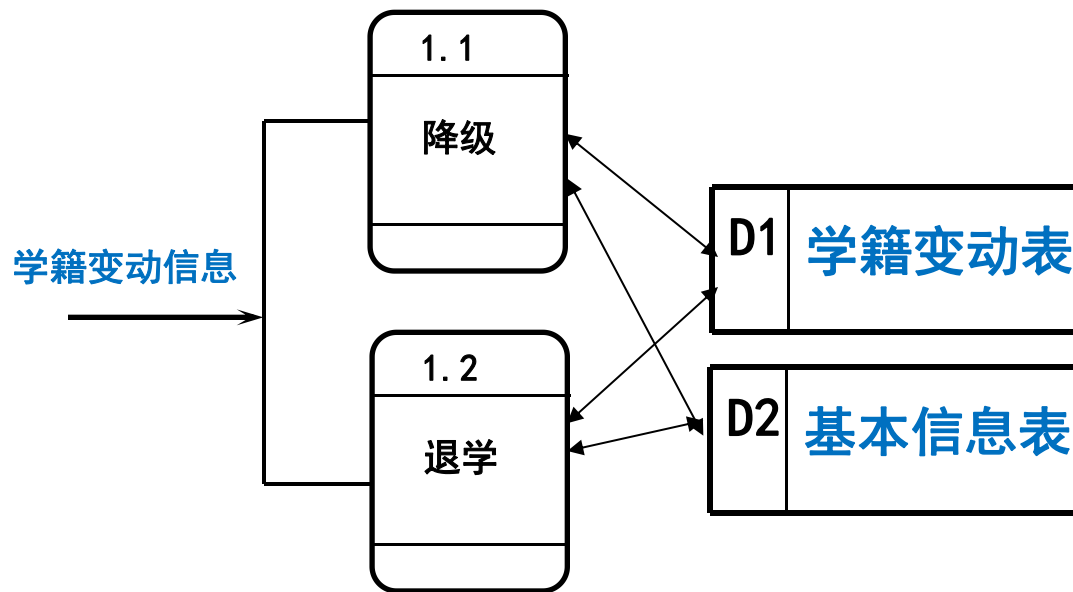
0层数据流图



学籍管理系统一级细化数据流图



1层数据流图



学籍管理系统二级细化数据流图



数据项	学号
含义说明	唯一标识某个学生
别名	学生编号
类型	字符型
长度	8
取值范围	00000000至 99999999
取值含义	前两位表示该学生所在院系， 3、4位表示所在年级，5、6位 表示班级、后2位按顺序标号

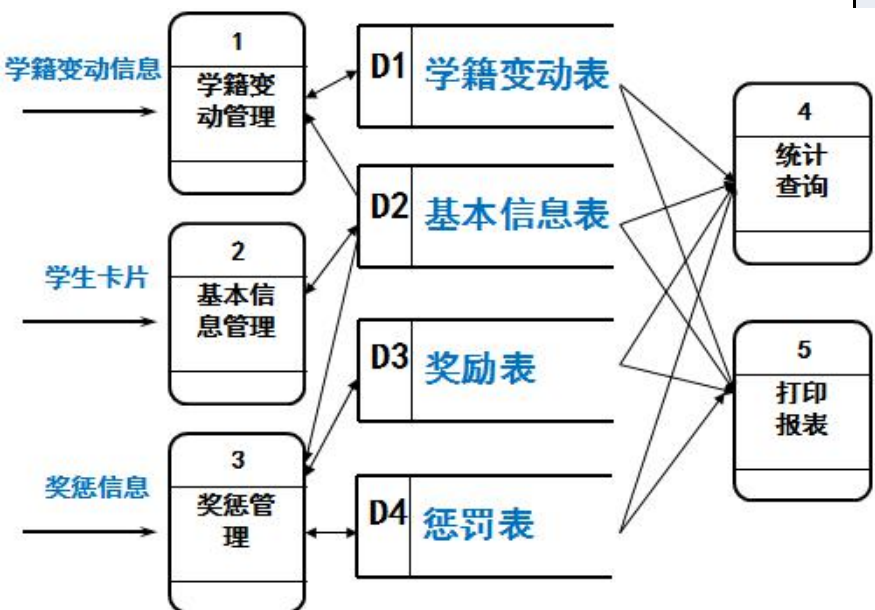


数据结构	学生
含义说明	是学籍管理模块的主体数据结构，定义了一个学生的有关信息
组成	学号，姓名，性别，年龄，院系，年级，专业



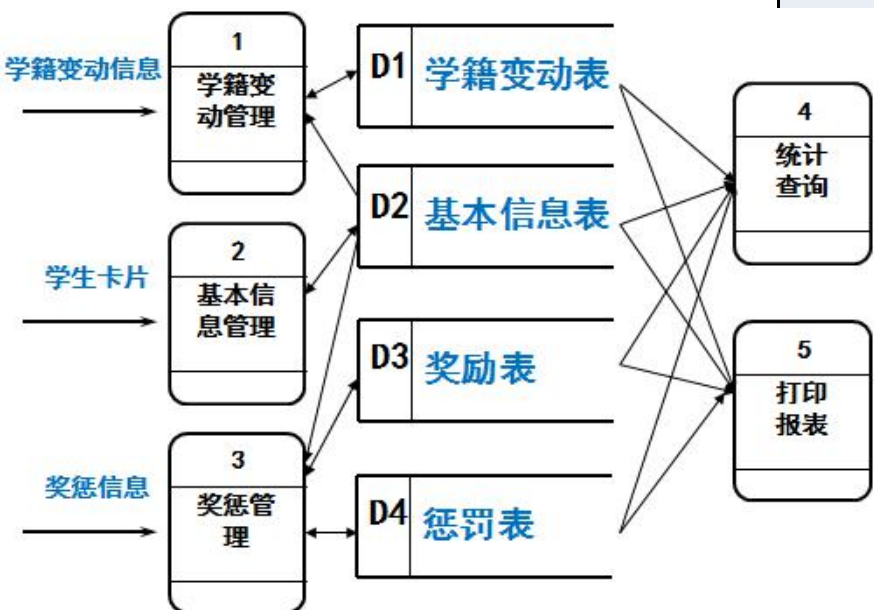
数据字典— 数据流

数据流	学籍变动信息
含义说明	学籍变化的相关信息
数据流来源	教务人员
数据流去向	学籍变动管理
组成	{{学生}, 变动内容}



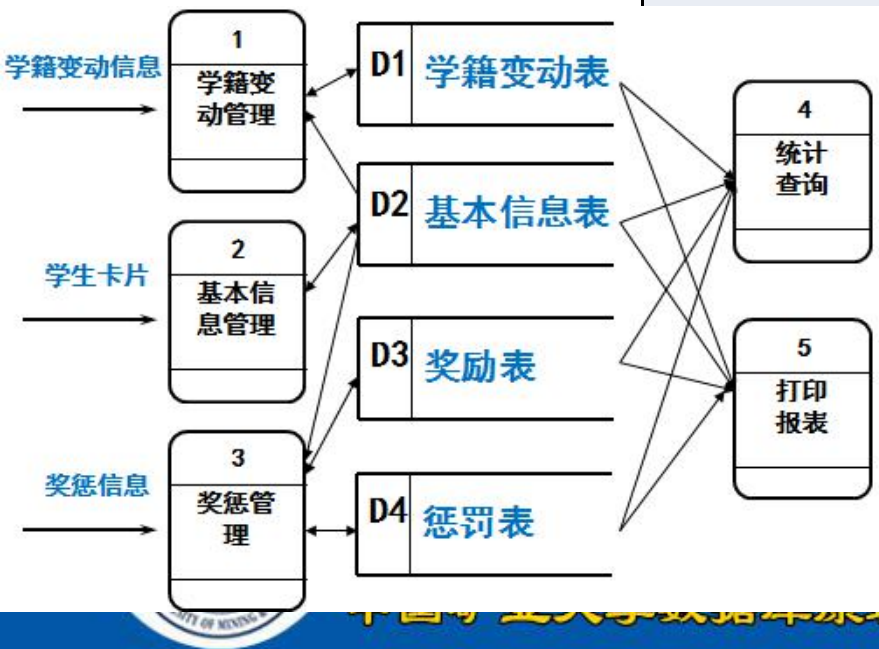
数据字典— 数据存储

数据存储	基本信息表
含义说明	学生学籍基本信息
流入的数据流	新的学籍信息
流出的数据流	查询的学籍信息
组成	学生学号，学生姓名，性别，入学年份、学院、宿舍
存取方式	顺序存取
存取量	每年4000个新记录

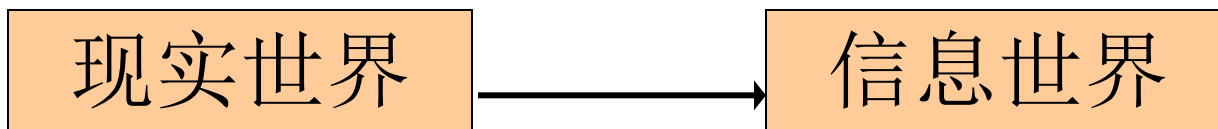


数据字典— 处理过程

处理过程	基本信息管理
含义说明	实现新生在校学籍信息的增删改
输入数据流	学生卡片
输出数据流	学生在校基本信息
处理	新生报到后，根据录取专业，为所有新生录入宿舍，院系，专业，班级等在校信息



5.3 概念结构设计



E-R图



5.3.1、概念结构设计的定义

- 不同DBMS基于不同的数据模型，而现实应用环境复杂多变，将现实世界中的事物直接转换为机器中的对象，非常不方便。
- 概念模型是现实世界到机器世界的一个中间层次，使用接近计算机存储的方式表示数据，同时又不涉及具体的DBMS。

概念模型的描述工具通常是E_R模型，该模型不依赖于具体的硬件环境和DBMS。

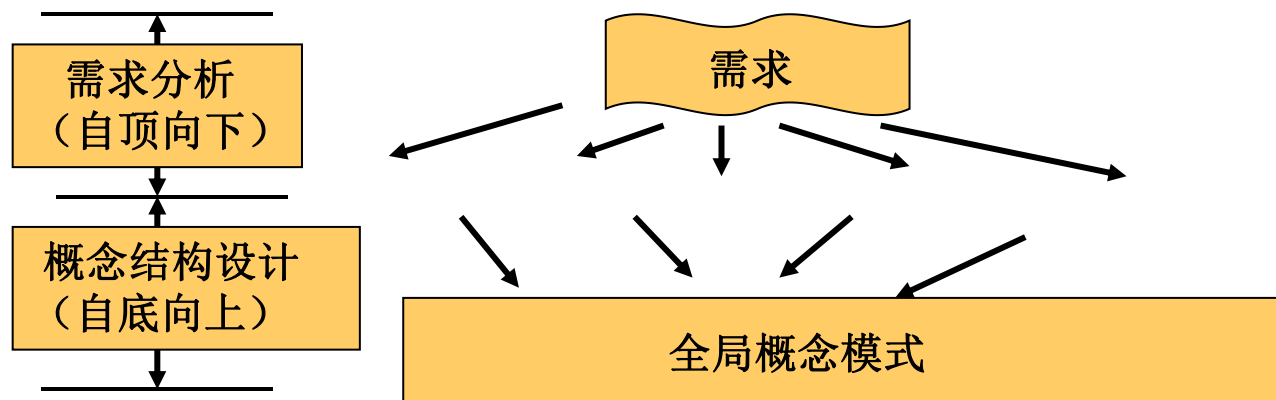


5.3.2、概念结构设计方法

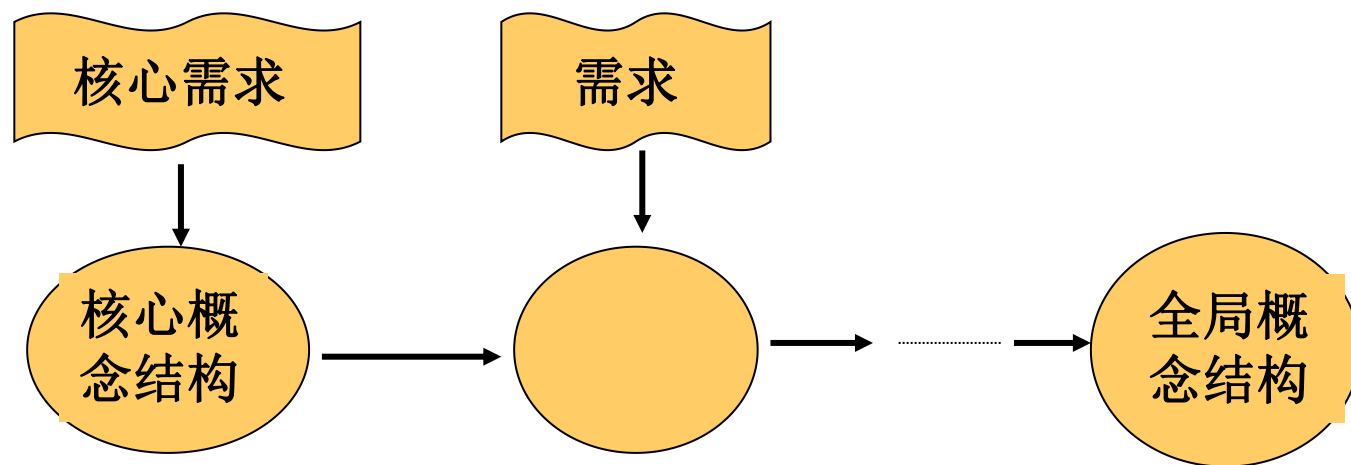
1、自顶向下（集中模式设计）：先定义全局，然后再逐步细化。

2、自底向上（视图集成法）：先定义局部，然后再集成起来。

3、混合策略：自底向上和自顶向下相结合。



4、逐步扩张（由内向外法）：先定义核心，然后再逐步向外扩充。



5.3.3、局部（分）E-R图设计

- 选择局部应用。一般而言，中层数据流图能较好反应系统各局部应用的子系统组成，因此通常以中层数据流图作为设计分E-R图的依据。
- 从数据字典中抽取数据。依据数据流图，标定局部应用中的实体、实体属性、标识实体的码，确定联系及类型。



(1) 标定局部应用中的实体

- 实体 is member of

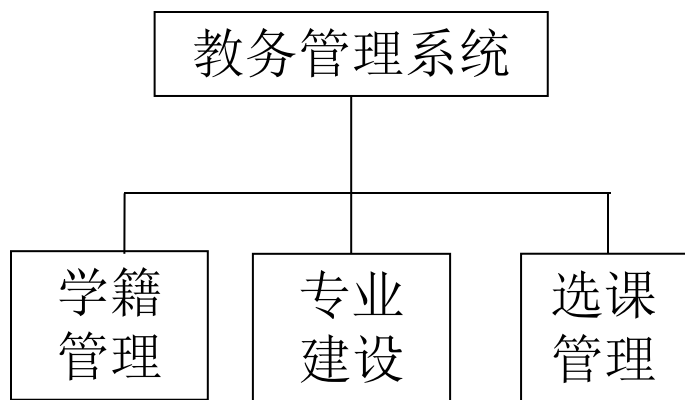
(2) 标定实体的属性、标识实体的码

- 实体 is part of

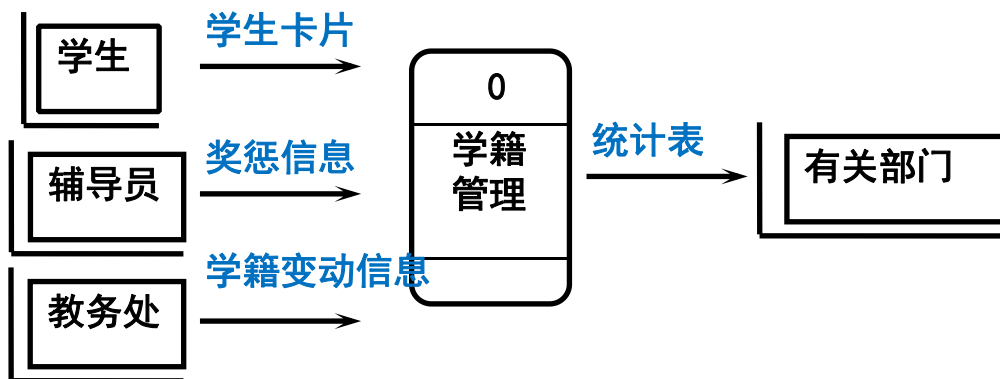
(3) 确定实体之间的联系



学校管理信息系统



教务管理系统功能结构图



学籍管理顶层数据流图



局部视图设计

➤在“学籍管理模块”中，涉及到的实体有哪些？

学生，班级，学院，专业，宿舍，班主任



请分析各个实体的属性有哪些？



请分析实体之间的联系及类型。

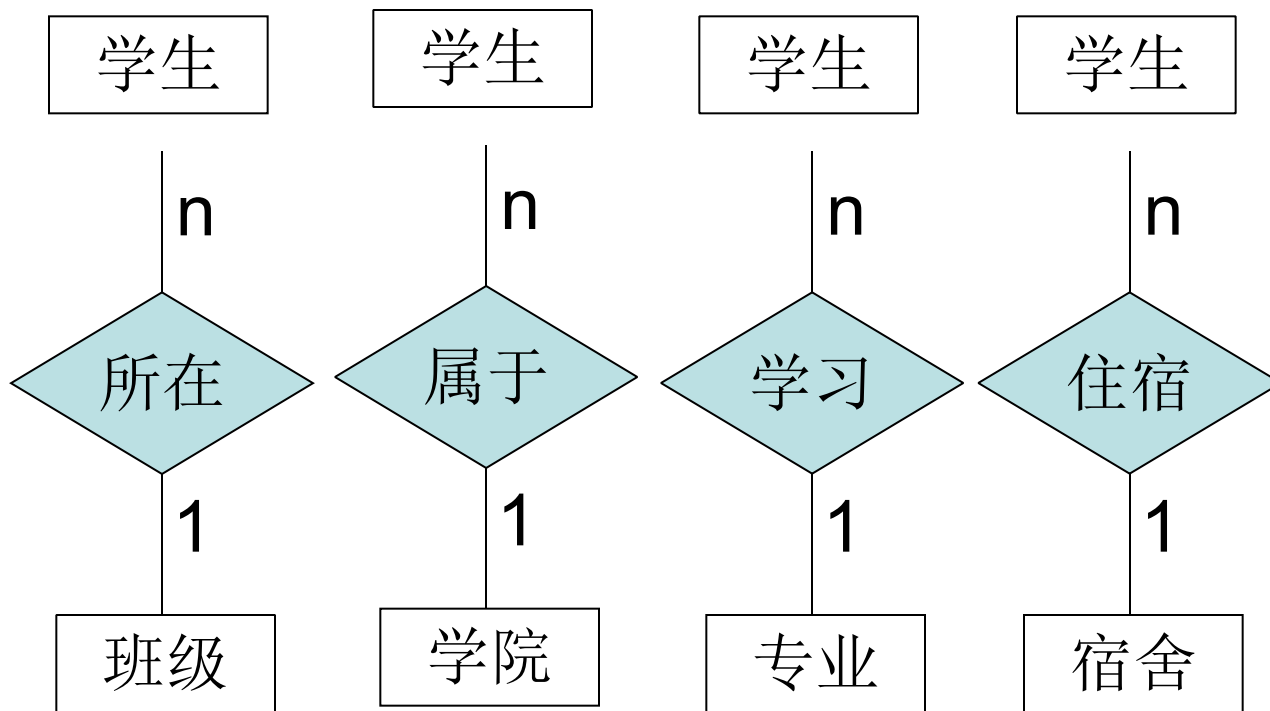


➤ 局部应用中的实体及属性

- 学生：{学号，姓名，性别，年龄，出生日期，院系，专业，班级，宿舍}
- 班级：{班级编号，班级备注信息}
- 宿舍：{宿舍编号，地址，电话}
- 学院：{学院编号，学院名称，院长，办公电话}
- 专业：{专业编号，专业名称}
- 班主任：{职工号，姓名，性别}
- 教师：{职工号，姓名，性别，职称}

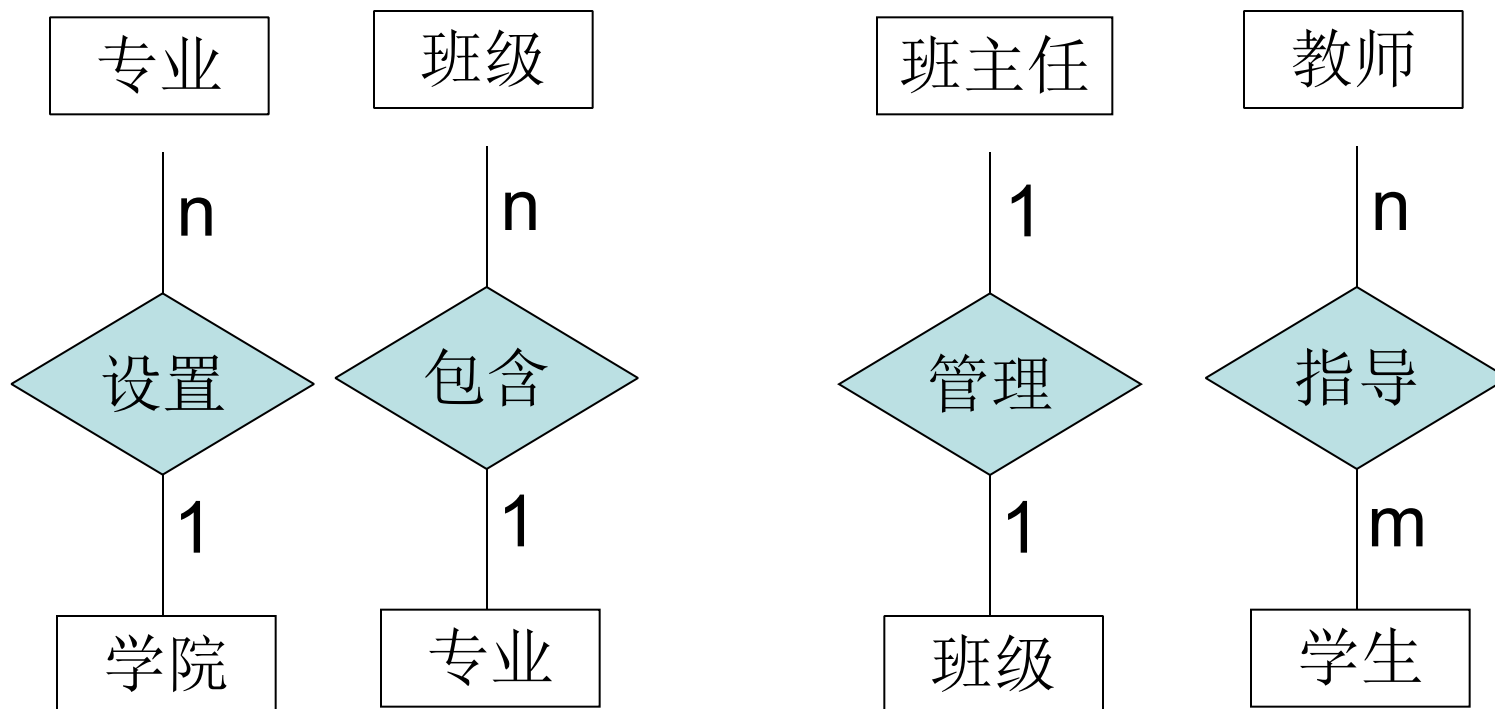


➤ 实体之间的联系及类型



学籍管理子系统分E_R图





学籍管理子系统分E_R图



5.3.4 集成全局视图

视图集成要解决的问题

(1) 确定模式之间的对应和冲突

- 属性冲突 (属性域冲突 取值单位冲突)
- 命名冲突 (同名异议 异名同义)
- 结构冲突

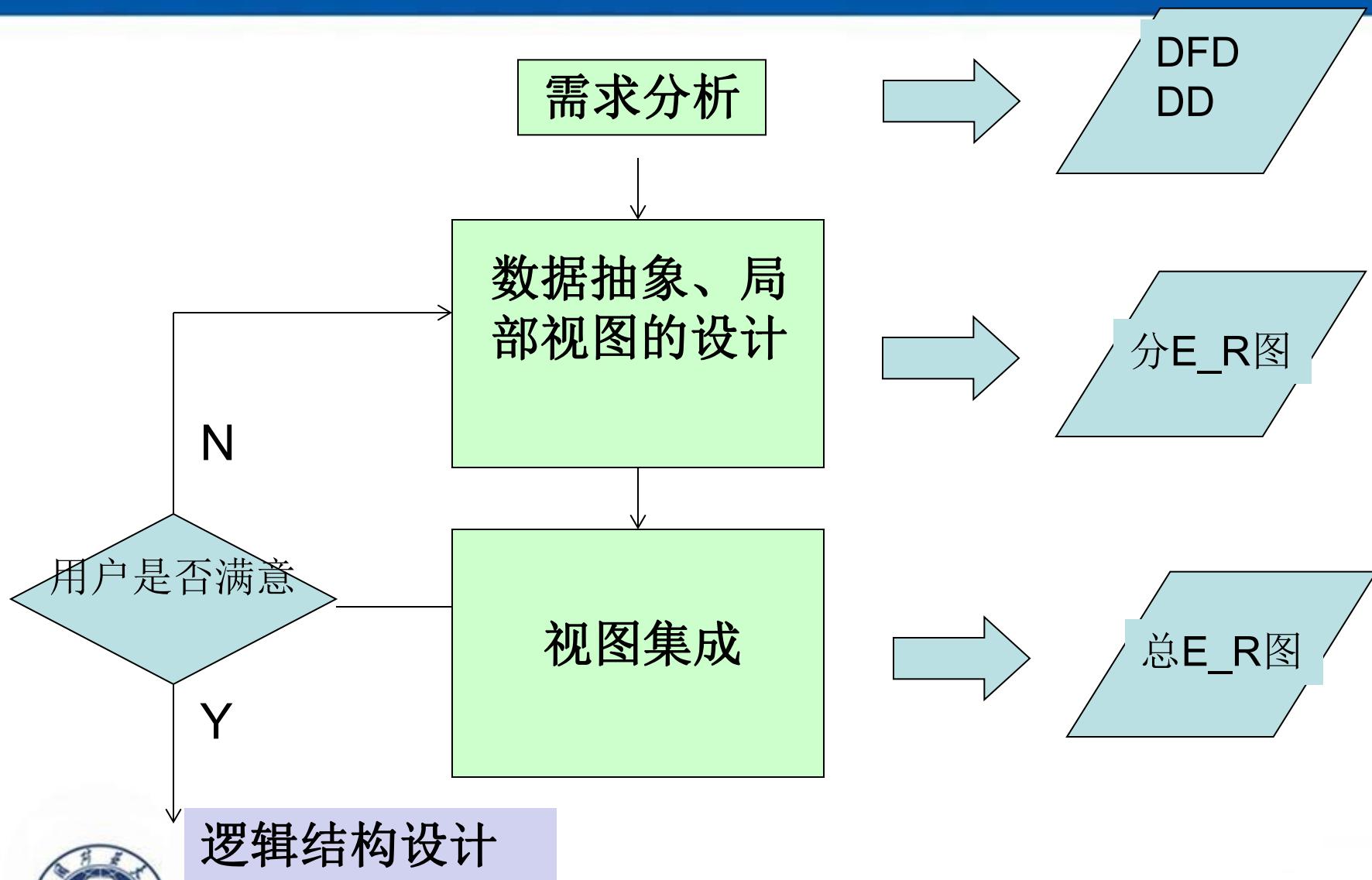
(2) 修改视图使得相互一致

(3) 合并视图

(4) 重构(可选)



5.3.4 集成全局视图



学籍管理子系统

- 学生：{学号，**姓名**，性别，年龄，出生日期，院系，专业，**班级**，宿舍}
 - **班级**：{**班级编号**，班级信息备注}
 - 宿舍：{宿舍编号，地址，电话}
 - 学院：{学院编号，学院名称，院长，办公电话}
 - 专业：{专业编号，专业名称}
 - 班主任：{职工号，**姓名**，性别}
-
- 命名冲突
- 结构冲突



应用举例——消除冲突

教师：{职工号，姓名，性别，职称}

班主任：{职工号，姓名，性别}

结构冲突

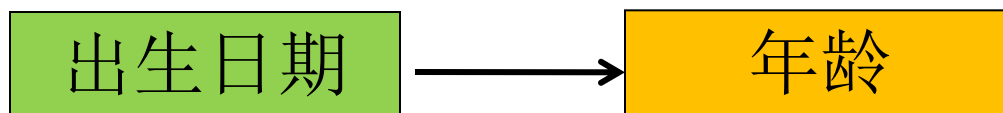
统一

教师：{职工号，教师姓名，性别，职称}

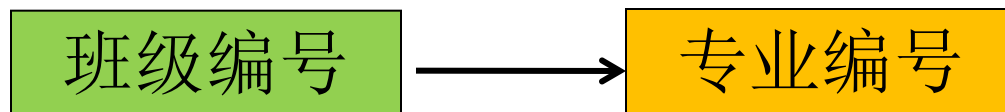


消除冗余属性和联系

- 学生：{学号，姓名，性别，出生日期，学院编号，专业编号，班级编号，宿舍编号}

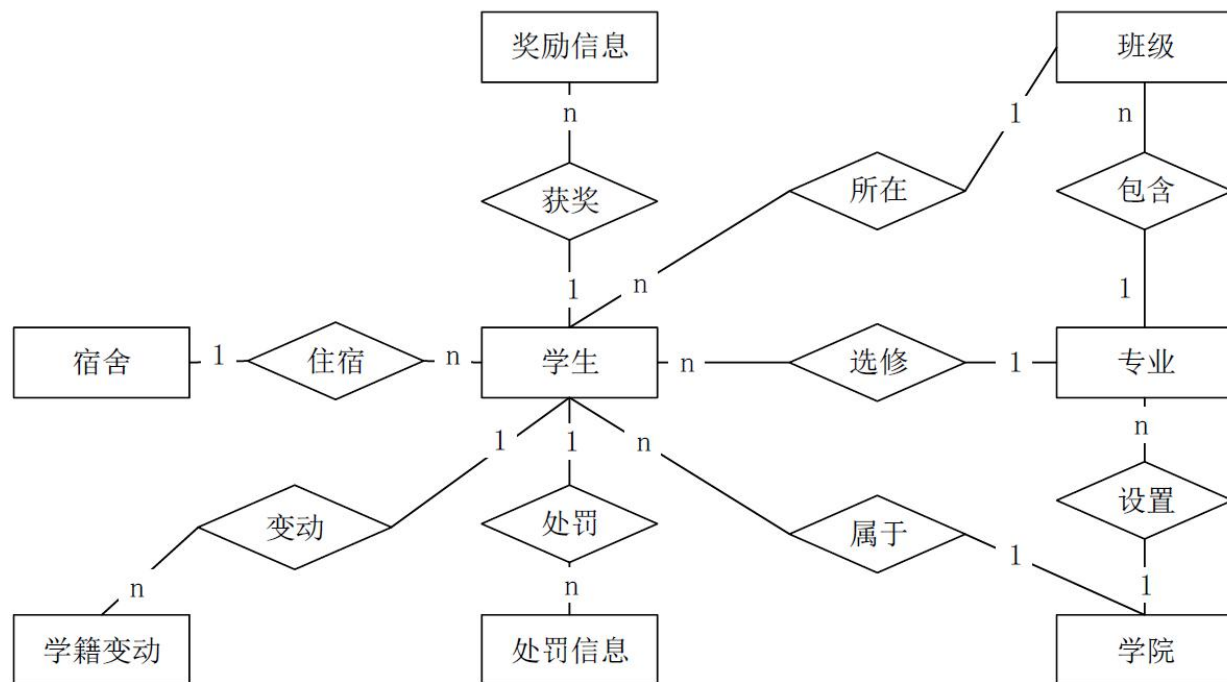


- 学生：{学号，姓名，性别，出生日期，学院编号，班级编号，宿舍编号}



- 学生：{学号，姓名，性别，出生日期，学院编号，班级编号，宿舍编号}





学籍管理子系统总E-R图



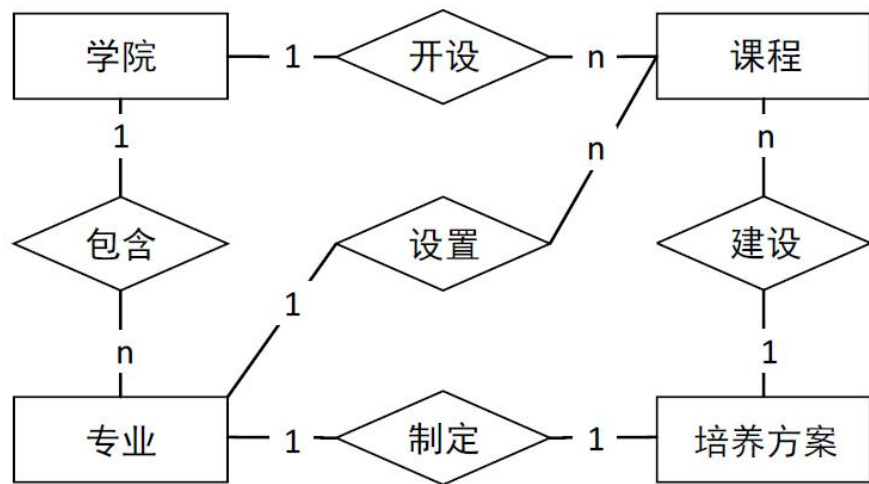


图 5-15 专业建设模块 E-R 图

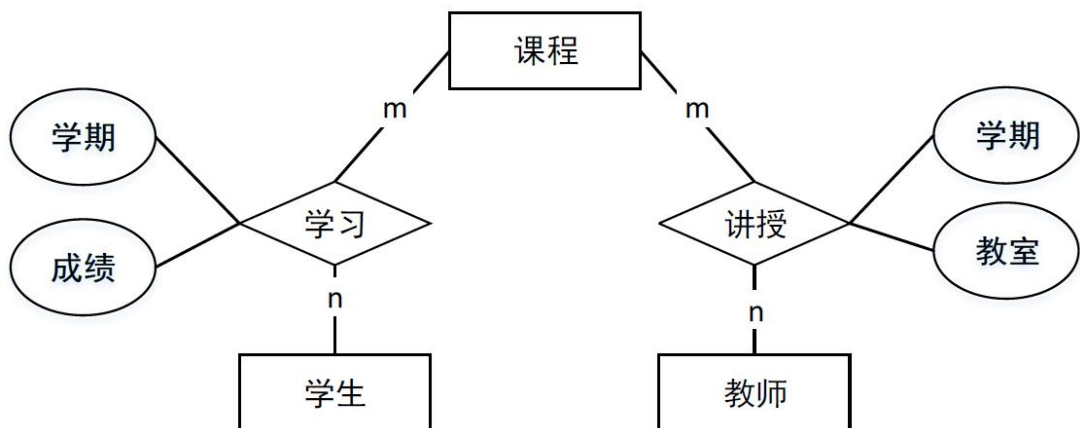


图 5-16 选课管理模块 E-R 图



5.4 逻辑结构设计

数据库逻辑结构设计的任务是将概念结构转换成特定DBMS所支持的数据模型，需要考虑具体的DBMS的性能、具体的数据模型特点。

E_R图转换为关系模式，实质就是将**实体**和**联系**转换为**关系模式**。

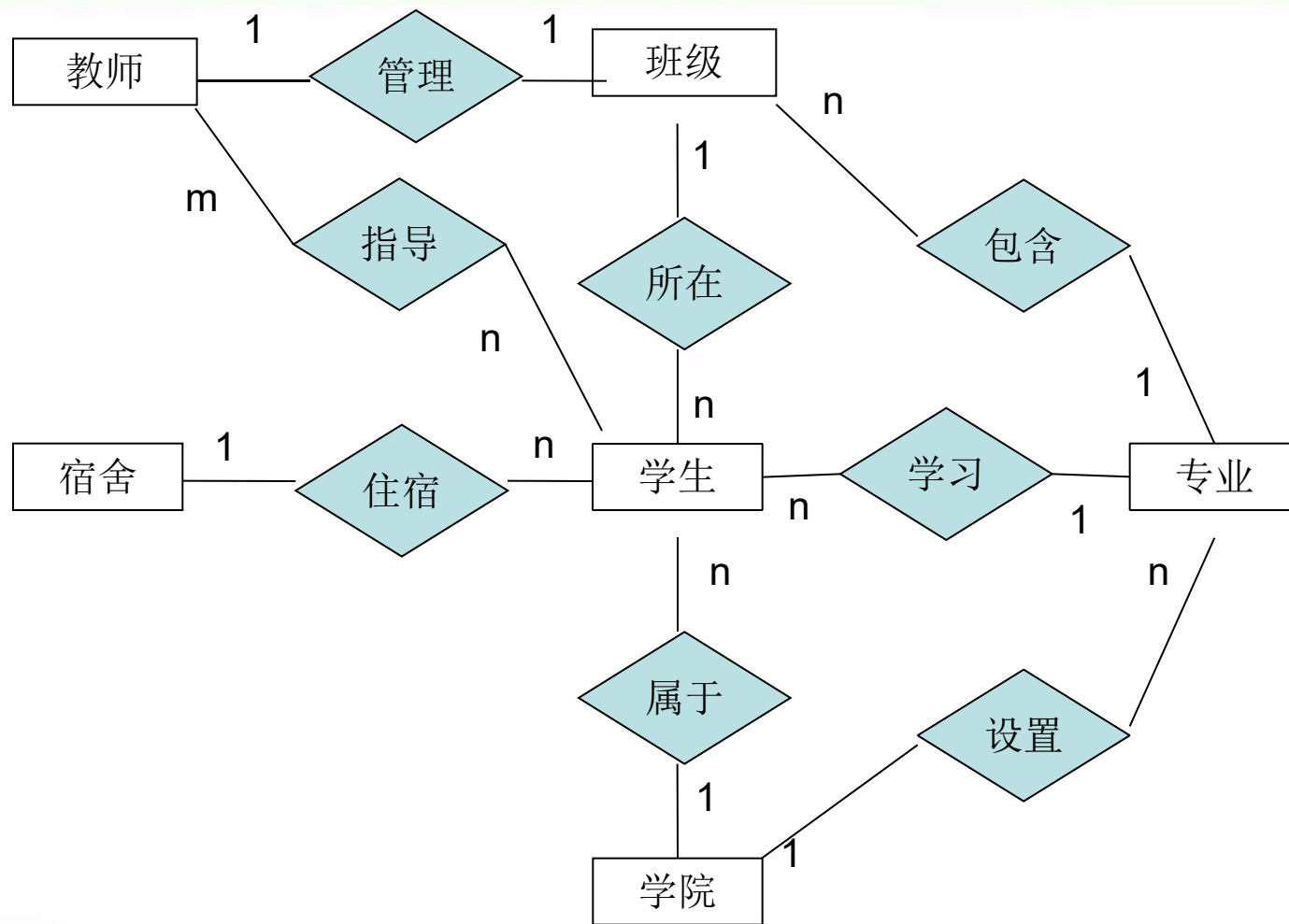


5.4.1 逻辑结构设计的任务和步骤

- 将概念模型转化为等价的关系模式
- 按需要对关系模式进行规范化
- 对规范化后的模式进行评价
- 根据局部应用的需要，设计用户外模式



学籍管理子系统总E-R图



5.4.2 E-R图向关系模型转换的原则

➤ **实体**：一个实体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**。

学生



学生（学号，姓名，出生日期，学院编号，班级编号，宿舍编号）

班级

学院

专业

宿舍

教师



5.4.2 E-R图向关系模型转换的原则

- 类型为1:1 联系的转换规则
- 类型为1:n联系的转换规则
- 类型为n:m联系的转换规则



5.4.2 E-R图向关系模型转换的原则

➤ 联系类型为1:1

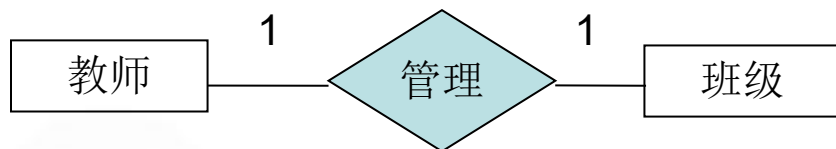
- 可以将联系转换为一个新的关系。

与该联系相连的各实体的码以及联系本身的属性构成新关系的属性，每个实体的码均是该关系的候选码。

管理（教师号，班级号）

管理（教师号，班级号）

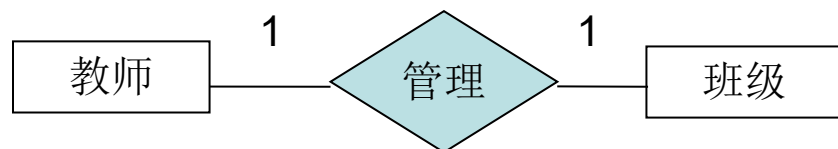
管理（教师号，班级号）



5.4.2 E-R图向关系模型转换的原则

➤ 联系类型为1:1

- 可以与任意一端对应的关系模式合并。在该关系模式中加入另一关系的码和联系的属性，该关系的码不变。



教师： { 职工号，教师姓名，性别，职称 ， 班级号 }

班级： { 班级编号，班级信息备注] ， 职工号 }

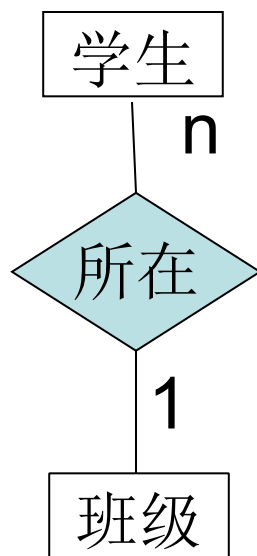


5.4.2 E-R图向关系模型转换的原则

➤ 联系类型为1:n

- 可以将联系转换为一个新的关系。

将联系转换为一个新的关系模式：与该联系相连的各实体的码以及联系本身的属性构成新关系的属性，该关系的码是n端关系模式的码。



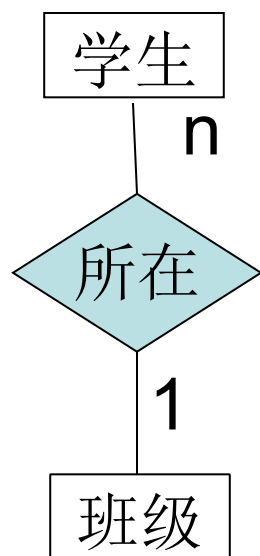
所在（学号，班级号）



5.4.2 E-R图向关系模型转换的原则

➤ 联系类型为1:n

- 可以与n端对应的关系模式合并。在n端关系模式中加入1端关系模式的码和联系的属性，关系的码仍为n端关系的码。



学生（学号，姓名，性别，出生日期，*班级号*）



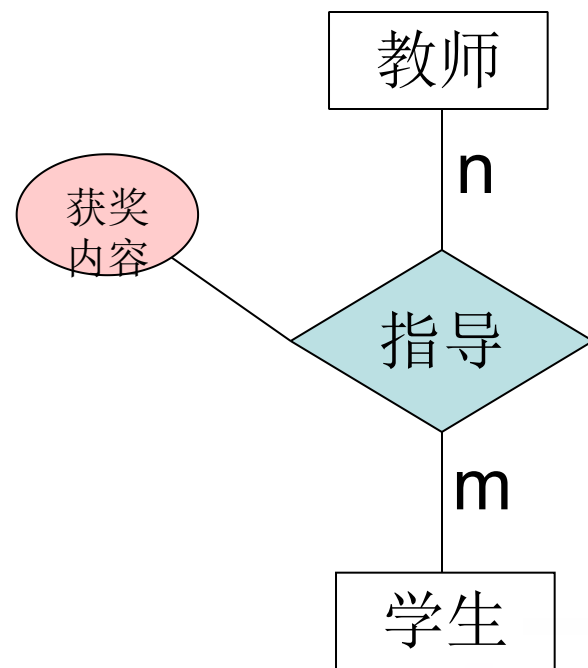
5.4.2 E-R图向关系模型转换的原则

➤ 联系类型为n:m

- 只能将联系转换为一个新的关系。

与该联系相连的各实体的码以及联系本身的属性构成新关系的属性，该关系的码是两端关系模式码的组合。

指导（学号，教师号，获奖内容）



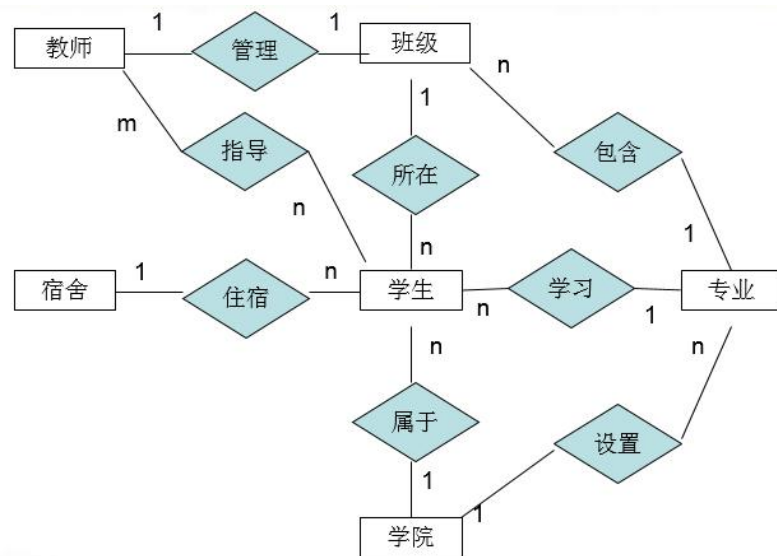
学籍管理子系统的关系模式

- 学生：{学号，姓名，性别，出生日期，院系编号，班级编号，宿舍编号}
- 班级：{班级编号，班级信息备注}
- 宿舍：{宿舍编号，地址，电话}
- 学院：{学院编号，学院名称，院长，办公电话}
- 专业：{专业编号，专业名称}
- 教师：{职工号，教师姓名，性别，职称}



采用生成一个新关系的策略

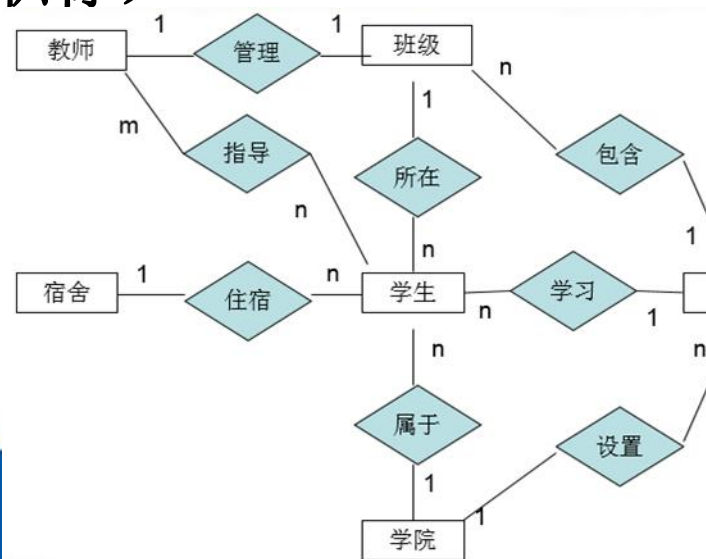
- 管理：（教师编号，班级编号）
- 所在：（学号，班级编号）
- 住宿：（学号，宿舍编号）
- 属于：（学号，学院编号）
- 学习：（学号，专业编号）
- 包含：（班级编号，专业编号）
- 设置：（专业编号，学院编号）
- 指导：（教师编号，学号，获奖内容）



- 学生：（学号，姓名，性别，出生日期，学院编号，班级编号，宿舍编号）
- 班级：（班级编号，班级信息备注）
- 宿舍：（宿舍编号，地址，电话）
- 学院：（学院编号，学院名称，院长，办公电话）
- 专业：（专业编号，专业名称）
- 教师：（教师编号，教师姓名，性别，职称）
- 所在：（学号，班级编号）
- 住宿：（学号，宿舍编号）
- 属于：（学号，学院编号）
- 管理：（教师编号，班级编号）
- 包含：（班级编号，专业编号）
- 设置：（专业编号，学院编号）
- 指导：（教师编号，学号，获奖内容）

采用合并的策略

- 学生：（学号，姓名，性别，出生日期，*学院编号*，*班级编号*，*宿舍编号*）
- 班级：（班级编号，班级信息备注，*教师编号*，*专业编号*）
- 宿舍：（宿舍编号，地址，电话）
- 学院：（学院编号，学院名称，院长，办公电话）
- 专业：（专业编号，专业名称，*学院编号*）
- 教师：（教师编号，教师姓名，性别，职称）
- 指导：（教师编号，学号，获奖内容）



5.4.3 逻辑结构的优化

- 确定数据依赖
- 对于各关系模式间的数据依赖进行极小化处理，消除冗余的联系。
- 规范化



- 学生：（学号，姓名，性别，出生日期，学院编号，班级编号，宿舍编号）
- 班级：（班级编号，班级信息备注，教师编号，专业编号）
- 宿舍：（宿舍编号，地址，电话）
- 学院：（学院编号，学院名称，院长，办公电话）
- 专业：（专业编号，专业名称，学院编号）
- 教师：（教师编号，教师姓名，性别，职称）
- 指导：（教师编号，学号，获奖内容）



5.4.4 设计用户外模式

- 使用更符合用户习惯的别名。
- 对于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。
 - view1(学号, 姓名, 教师编号, 获奖内容)
 - view2(学号, 姓名, 教师编号, 教师姓名, 职称, 获奖内容)
- 简化用户对系统的使用。



5.8 数据库设计实例

➤ 医院管理信息系统



➤ 实例 医院管理信息系统

1 需求分析

医院有若干科室，**科室**包括科编号、名称、人数、地点、负责人。医院每一个科室有若干名**医生**，医生包括医生编号、姓名、职务、学历、职称、简历，每个医生的**简历**包括开始时间、终止时间、单位、担任职务、证明人。一个医生要负责几个**病房**病人的医疗工作，每个病房又可以有多个医生为病人治疗，但一个病人只能由一个医生负责。对于**病人**，医院关心病人编号、姓名、性别、年龄、住院时间、出院时间、病因等信息，对于病房关心病房号、床位数、床位号、床位是否为空等信息。



2 概念结构设计

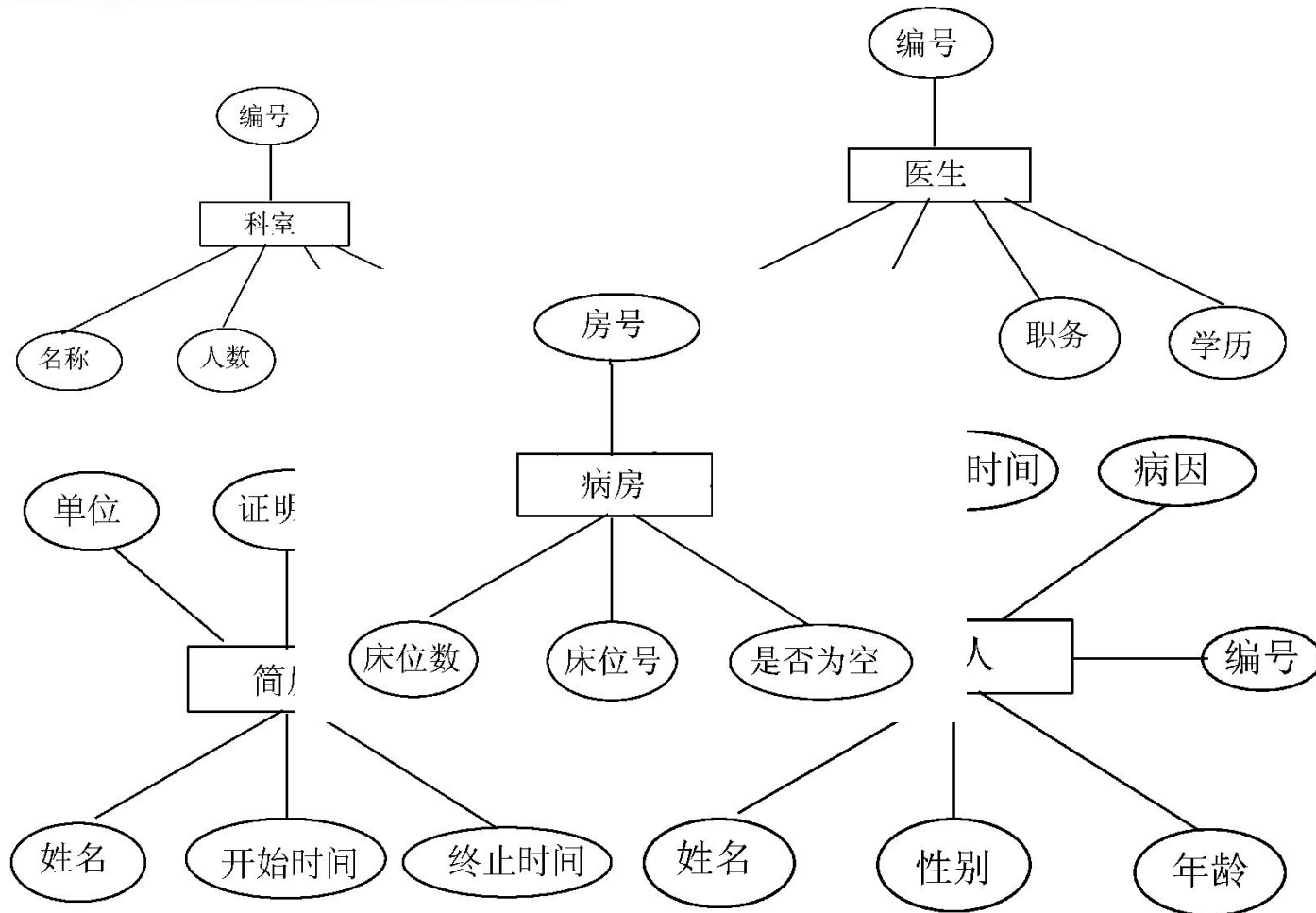
科室

医生

简历

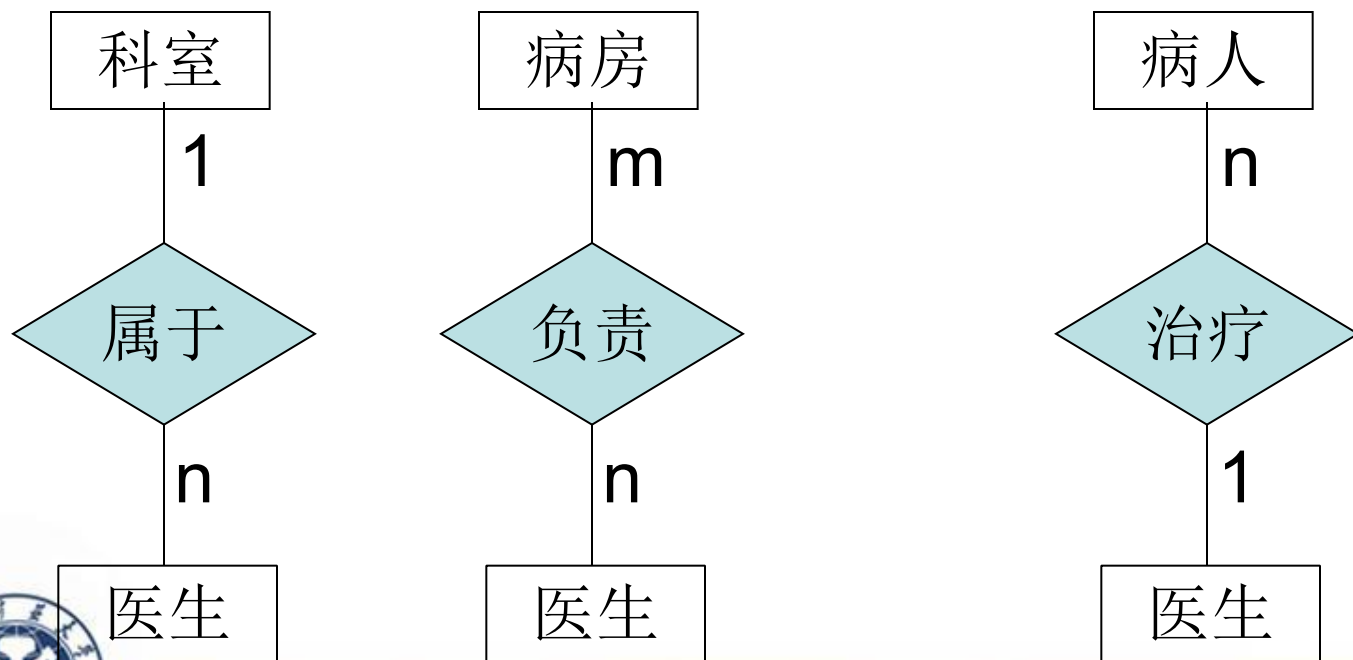
病人

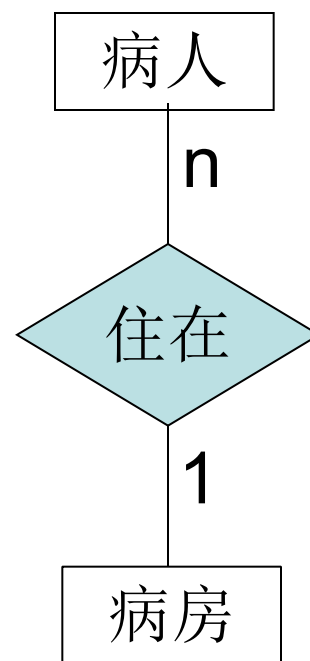
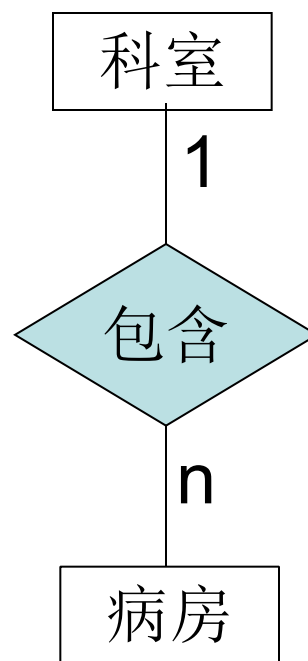
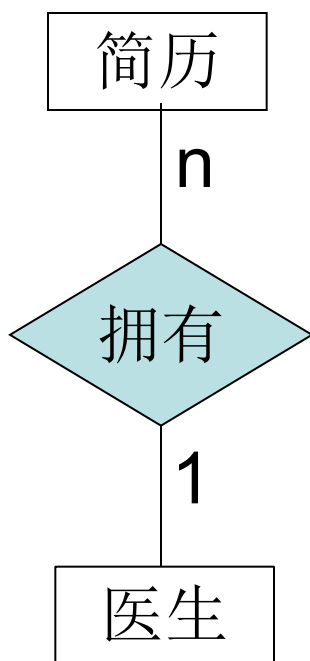
病房

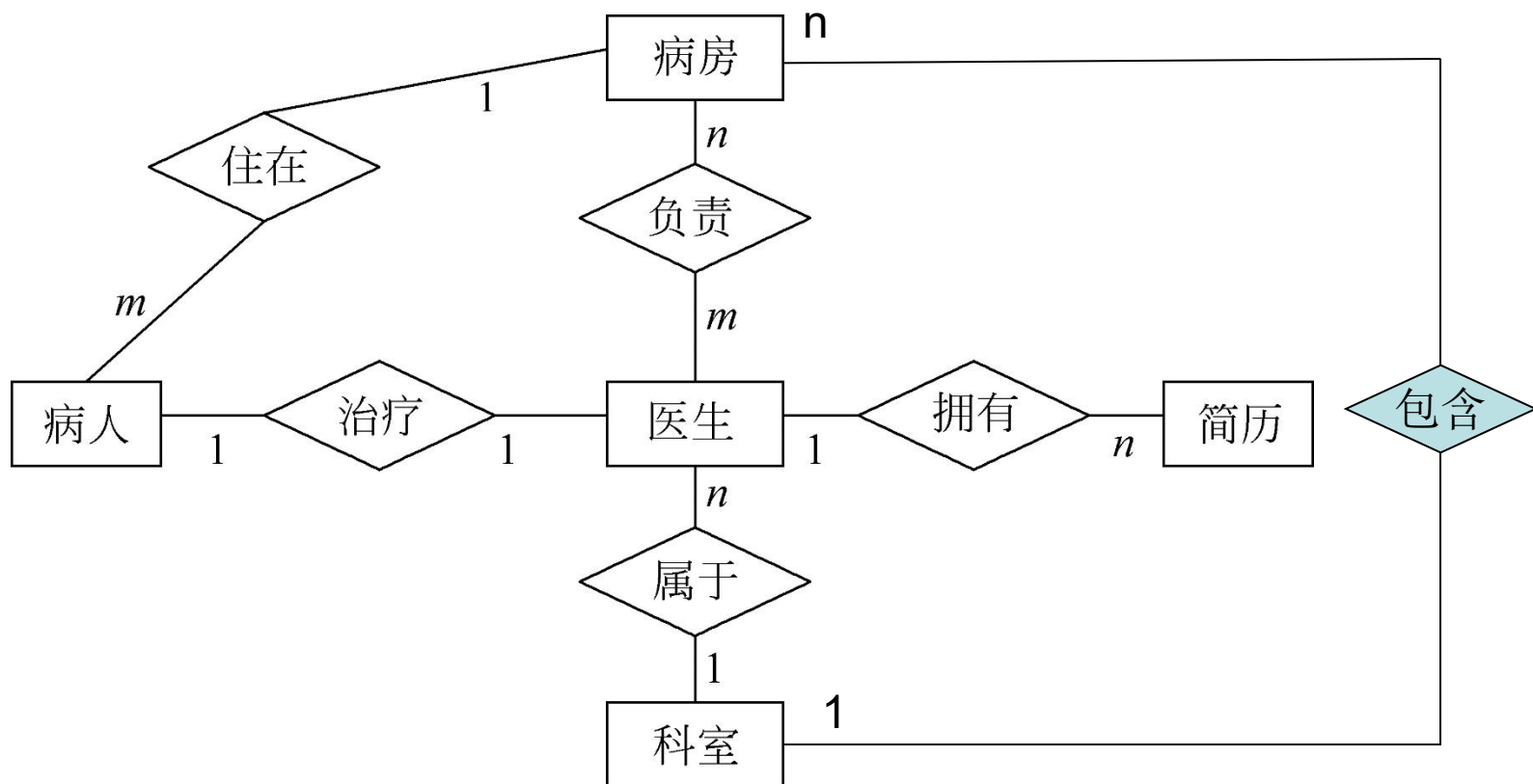


2 概念结构设计

- 医院每一个科室有若干名医生
- 一个医生要负责几个病房病人的医疗工作，每个病房又可以有多个医生为病人治疗
- 一个病人只能由一个医生负责







3逻辑结构设计

医生（医生编号，姓名，职称，职务，学历）

科室（科室编号，名称，人数，地点，负责人）

简历（简历编号，医生姓名，单位，担任职务，开始时间，终止时间，证明人）

病人（病人编号，姓名，性别，年龄，住院时间，出院时间，病因）

病房（病房号，床位号，病床数，床位是否为空）



采用生成一个新关系的策略

- 治疗（病人编号，医生编号）
- 拥有（简历编号，医生编号）
- 属于（医生编号，科室编号）
- 包含（病房号，科室编号）
- 住在（病人编号，病房编号）
- 负责（病房号，医生编号）



1:1 和1:n关系采用合并的策略

医生 (医生编号, 姓名, 职称, 职务, 学历, 科室
编号)

科室 (科室编号, 名称, 人数, 地点, 负责人)

简历 (简历编号, 医生姓名, 单位, 担任职务, 开始时间, 终止时间, 证明人, 医生编号)

病人 (病人编号, 姓名, 性别, 年龄, 住院时间, 出院时间, 病因, 医生编号, 病房编号)

病房 (病房号, 床位号, 病床数, 床位是否为空, 科室编号)



病房（病房号，床位号，病床数，床位是否为空，*科室编号*）



病房（病房号，床位号，床位是否为空，*科室编号*）

病房信息（病房号，病床数）

