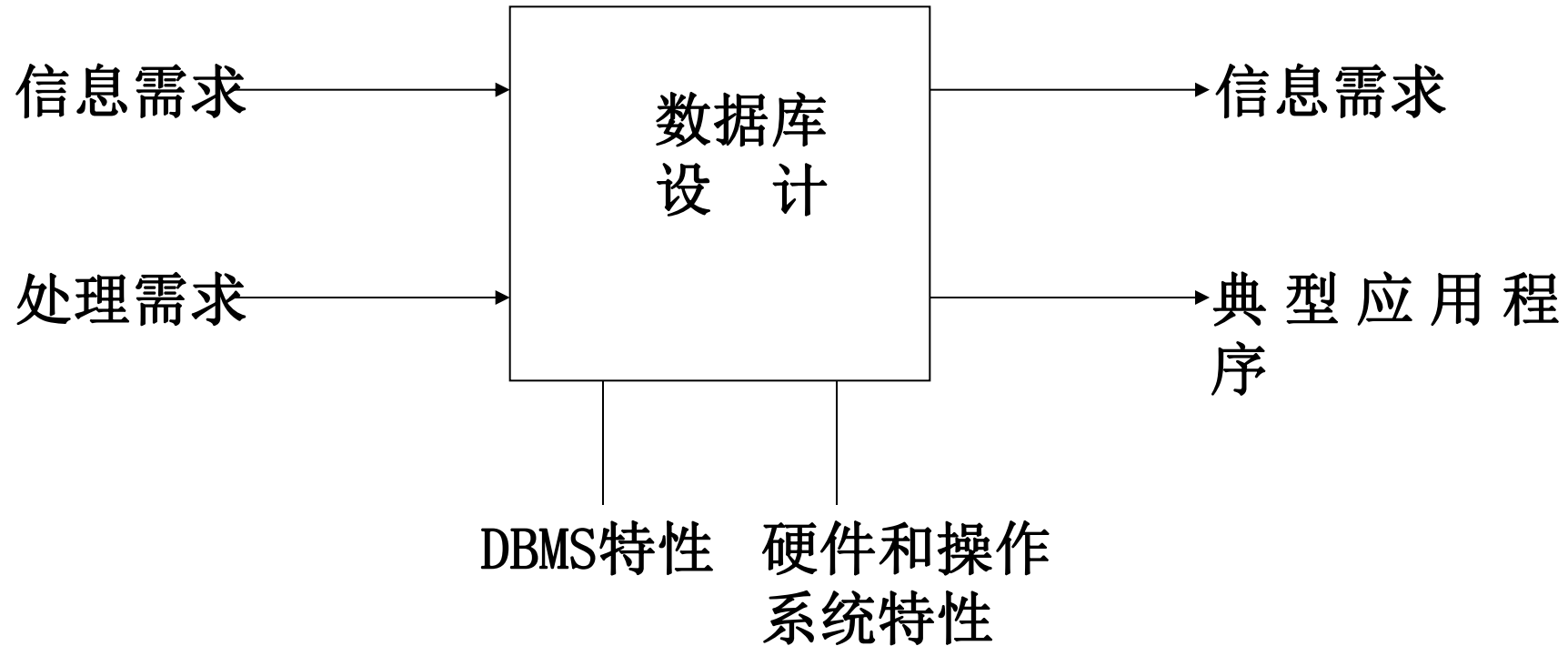


第十四讲 关系数据库设计理论

14.1 数据库设计概述

- 14.1.1 数据库设计的任务、内容和特点
- 数据库设计的任务
 - 指根据用户需求**研制数据库结构的过程**，具体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**。
 - 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**。

14.1 数据库设计的任务

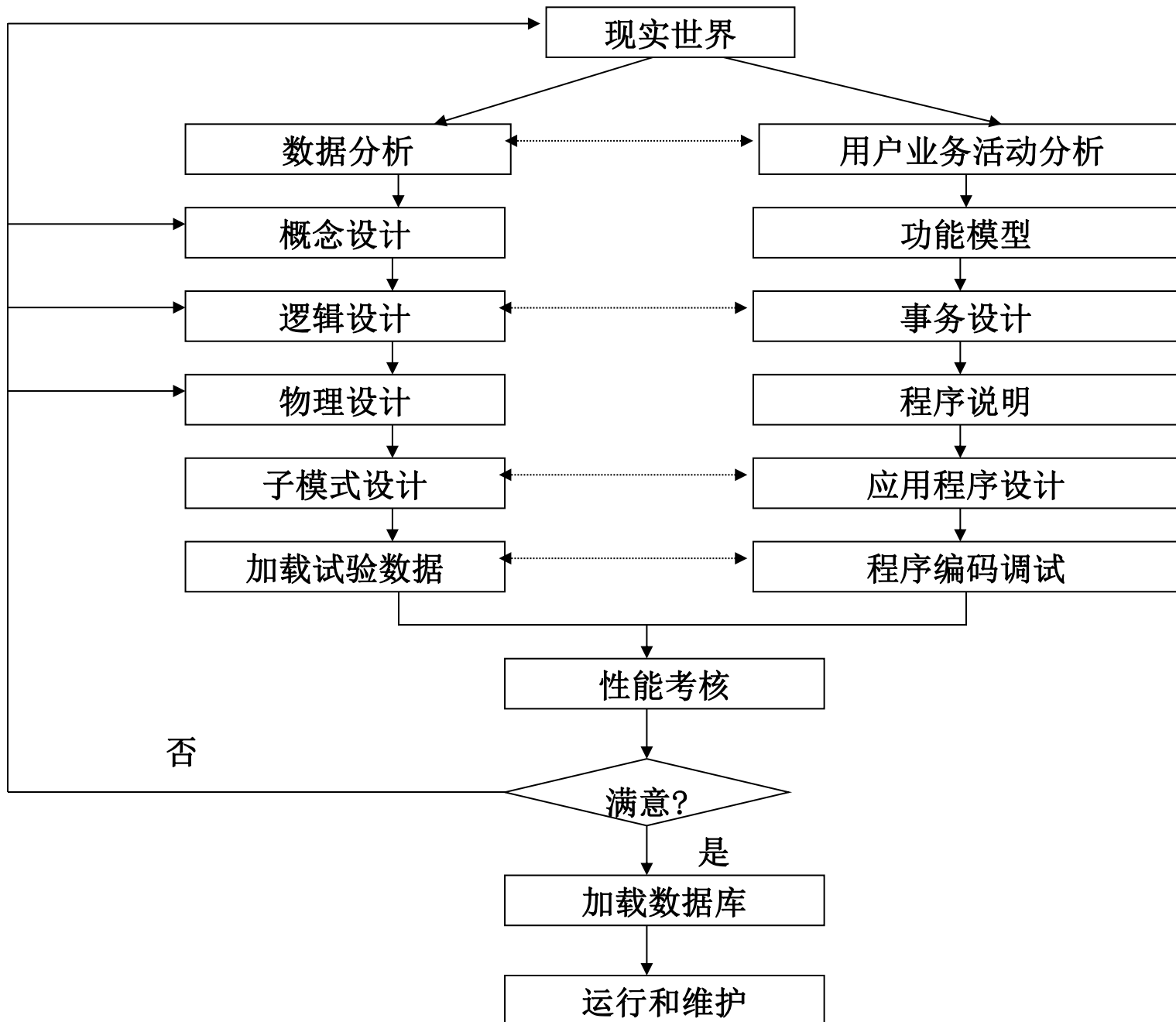


数据库设计的内容

- 1、数据库的结构设计
 - 指根据给定的应用环境，进行数据库的**模式或子模式**的设计。
 - 数据库模式是各应用程序共享的结构，是静态的、稳定的，一经形成通常是不容易改变的，结构设计又称**静态模型设计**
 - 包括数据库的**概念设计、逻辑设计和物理设计**
- 2、数据库的行为设计
 - 指确定数据库用户的行为和动作，即**用户对数据库的操作**，通过应用程序来实现，即**应用程序的设计**。
 - 用户的行为总是使数据库的内容发生变化，是动态的，行为设计又称为**动态模型设计**

数据库设计的特点

- 现代数据库的设计的特点是强调**结构设计与行为设计相结合**，是一种“**反复探寻，逐步求精**”的过程。
- 首先从数据模型开始设计，以数据模型为核心进行展开，数据库设计和应用系统设计相结合，建立一个完整、独立、共享、冗余小、安全有效的数据库系统。



14.1.2 数据库设计方法简述

- 1. 基于**E-R**模型的数据库设计方法
 - 1976年**P.P.S.chen**提出的数据库设计方法
 - 基本思想
 - 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
 - 再将企业模式转换成基于某一特定的**DBMS**的概念模式

2. 基于3NF的数据库设计方法

- 由**S·Atre**提出的结构化设计方法
- 基本思想
 - 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
 - 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。
 - (1) **设计企业模式**，利用规范化得到的3NF关系模式画出企业模式
 - (2) **设计数据库的概念模式**，把企业模式转换成DBMS所能接受的概念模式，并根据概念模式导出各个应用的外模式
 - (3) **设计数据库的物理模式**（存储模式）
 - (4) **对物理模式进行评价**
 - (5) **实现数据库**

3. 基于视图的数据库设计方法

- 先从分析各个应用的数据着手，其基本思想：为每个应用建立自己的视图→再把这些视图汇总起来合并成整个数据库的概念模式。
- 合并过程中要解决以下问题：
 - (1) 消除命名冲突
 - (2) 消除冗余的实体和联系
 - (3) 进行模式重构，在消除了命名冲突和冗余后，需要对整个汇总模式进行调整，使其满足全部完整性约束条件。

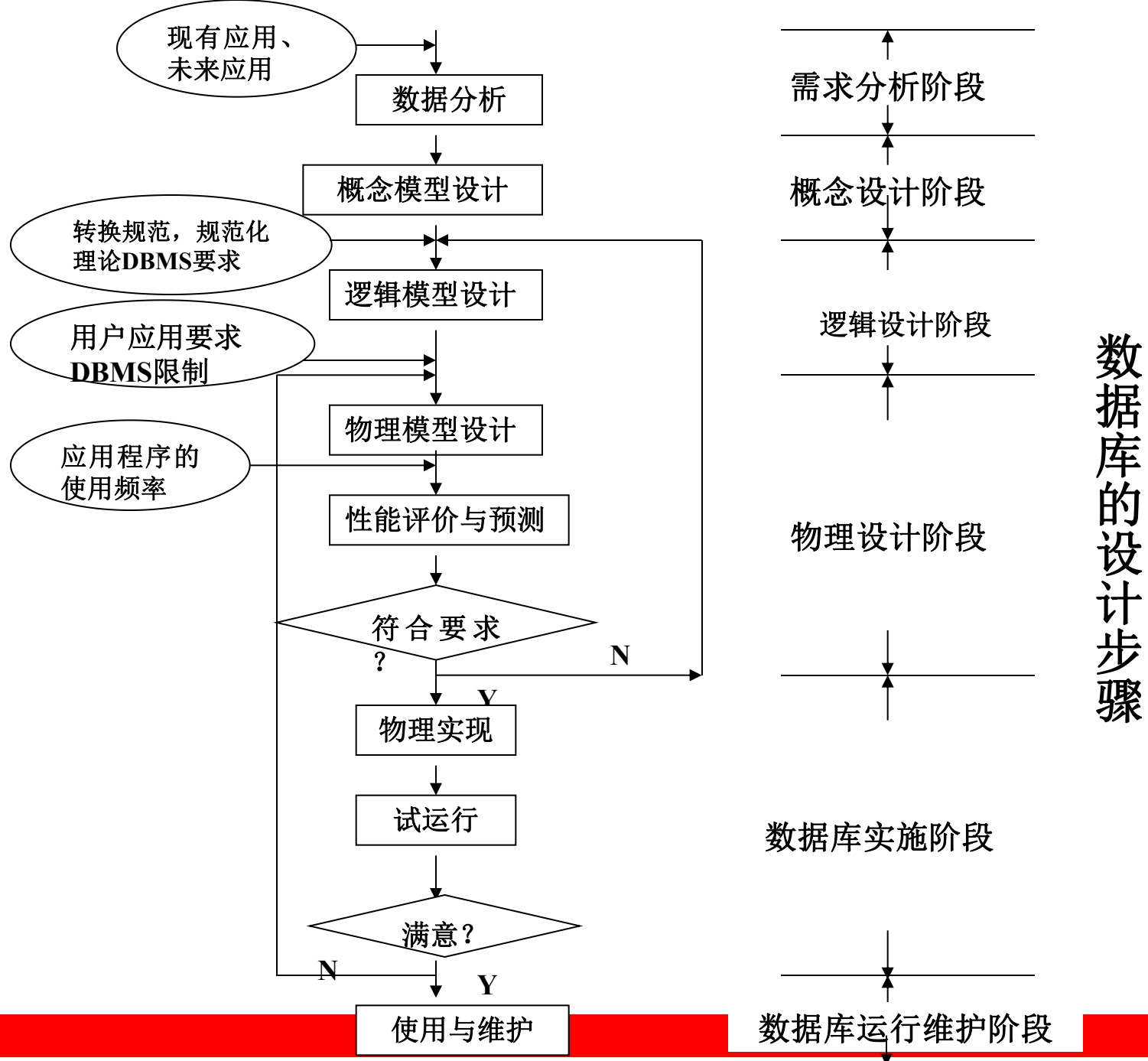
14.1.3 数据库设计的步骤

- 数据库的设计过程可以使用软件工程中的生存周期的概念来说明，称为“**数据库设计的生存期**”，它是指从数据库研制到不再使用它的整个时期。
- 按规范设计法可将数据库设计分为六个阶段
 - (1) 系统需求分析阶段
 - (2) 概念结构设计阶段
 - (3) 逻辑结构设计阶段
 - (4) 物理设计阶段
 - (5) 数据库实施阶段
 - (6) 数据库运行与维护阶段

面向用户的应用要求
面向具体的问题
面向数据库管理系统

分析和设计阶段

面向具体的实现方法 → 实现和运行阶段



数据库设计的六个阶段

1. 系统需求分析阶段

- 需求分析是整个数据库设计过程的基础，要收集数据库所有用户的信息内容和处理要求，并加以规格化和分析
- 是最费时、最复杂的一步，但也是最重要的一步
- 需求分析做得不好，可能会导致整个数据库设计返工重做
- 在分析用户需求时，要确保用户目标的一致性

2. 概念结构设计阶段

- 概念设计是把用户的信息要求统一到一个整体逻辑结构中
- 概念结构能够表达用户的要求，是一个独立于任何DBMS软件和硬件的概念模型。

3. 逻辑结构设计阶段

- 逻辑设计是将上一步所得到的概念模型转换为某个DBMS所支持的数据模型，并对其进行优化。

数据库设计的六个阶段

4. 物理设计阶段

- 物理设计是为逻辑数据模型建立一个完整的能实现的数据库结构，包括**存储结构**和**存取**方法。

5. 数据库实施阶段

- 根据物理设计的结果把原始数据装入数据库，建立一个具体的数据库并编写和调试相应的应用程序。
- 应用程序的开发目标是开发一个可依赖的有效的数据库存取程序，来满足用户的处理要求。

6. 数据库运行与维护阶段

- 收集和记录实际系统运行的数据，数据库运行的记录用来提高用户要求的有效信息，用来评价数据库系统的性能，进一步调整和修改数据库。
- 在运行中，必须保持数据库的完整性，有效地处理数据库故障和进行数据库恢复
- 在运行和维护阶段，可能要对数据库结构进行修改或扩充。

14.2 系统需求分析

- 需求分析是数据库设计的起点，为以后的具体设计作准备。
- 需求分析的结果是否准确的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，必须高度重视系统的需求分析。
- **14.2.1 需求分析的任务**
 - 对现实世界要处理的对象（组织、部门、企业）等进行详细的调查
 - 通过对原系统的了解，收集支持新系统的基础数据并对其进行处理，在此基础上确定新系统的功能

1. 调查分析用户的活动

- 通过对新系统运行目标的研究，对现行系统所存在的主要问题
的分析以及制约因素的分析，明确用户总的需求目标，确定这
个目标的功能域和数据域。具体做法是：
 - (1) 调查组织机构情况：包括该组织的部门组成情况，各部门的职责和任
务等。
 - (2) 调查各部门的业务活动情况：包括各部门输入和输出的数据与格式、
所需的表格与卡片、加工处理这些数据的步骤、输入输出的部门等。

2. 收集和分析需求数据，确定系统边界

- 在熟悉业务活动的基础上，协助用户明确对新系统的各种需求：
 - (1) **信息需求**指目标范围内涉及的所有实体、实体的属性以及实体间的联系等数据对象，也就是用户需要从数据库中获得信息的内容与性质。由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。
 - (2) **处理需求**指用户为了得到需求的信息而对数据进行加工处理的要求，包括对某种处理功能的响应时间，处理的方式（批处理或联机处理）等。
 - (3) **安全性和完整性的需求**。在定义信息需求和处理需求的同时必须相应确定安全性和完整性约束。

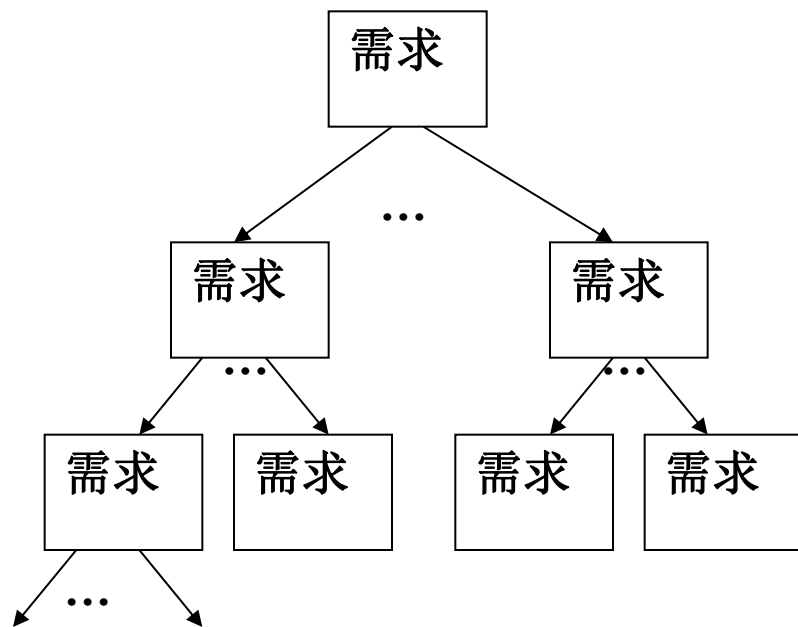
3. 编写需求分析说明书

- 系统分析阶段的最后是编写系统分析报告，通常称为**需求规范说明书**。
- 需求规范说明书是对需求分析阶段的一个总结。编写系统分析报告是一个不断反复、逐步深入和逐步完善的过程
- 系统分析报告应包括如下内容：
 - (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。
- 完成系统的分析报告→技术专家评审→审查通过后由项目方和开发方领导签字认可

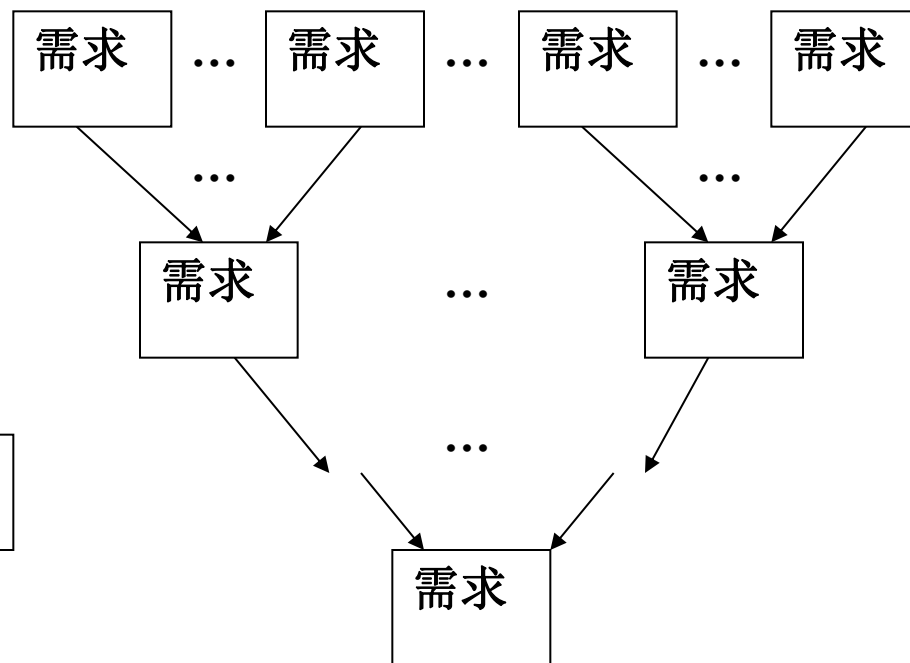
14.2.2 需求分析的方法

- 用户参加数据库设计是数据应用系统设计的特点，是数据库设计理论不可分割的一部分。
- 用于需求分析的方法有多种，主要方法有自顶向下和自底向上两种
- 自顶向下的分析方法（**Structured Analysis**，简称**SA**方法）是最简单实用的方法。**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，用**数据流图**（**Data Flow Diagram, DFD**）和**数据字典**（**Data Dictionary, DD**）描述系统。

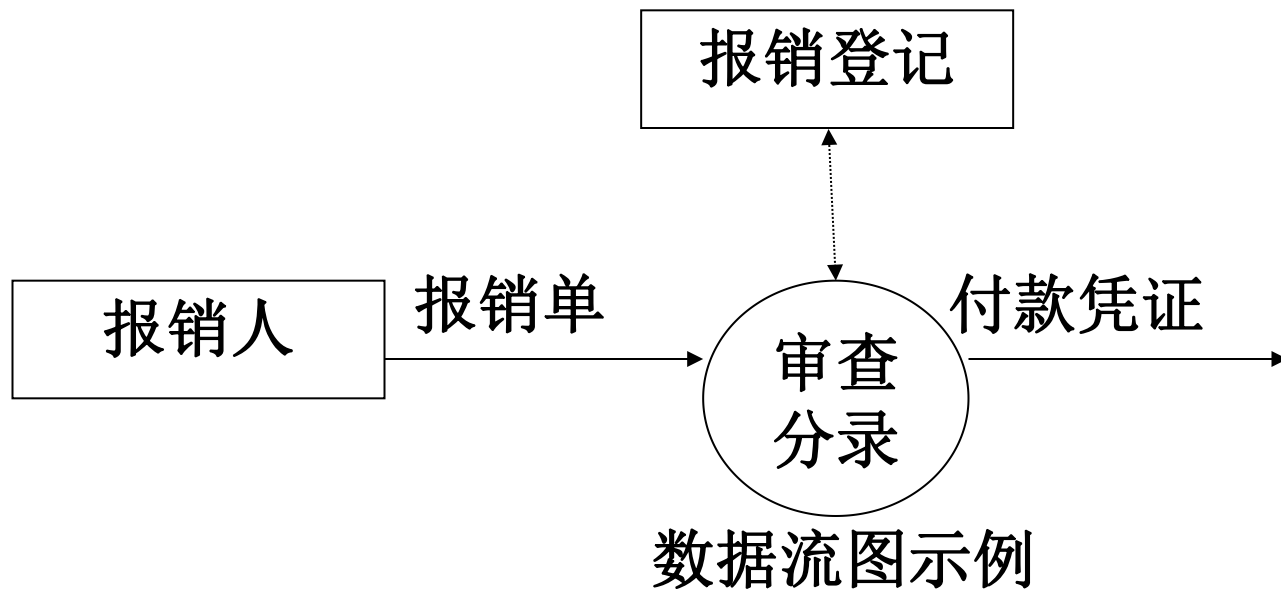
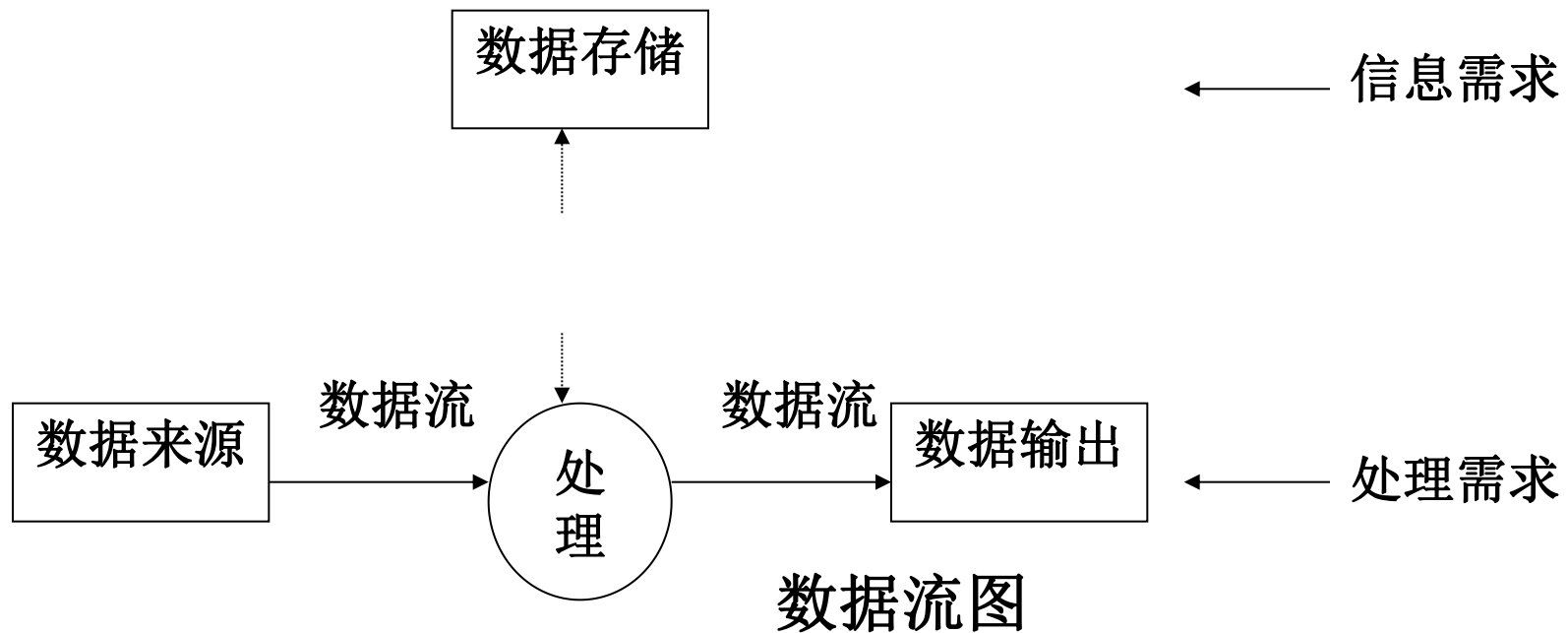
需求分析的方法



(a) 自顶向下的需求分析



(b) 自底向上的需求分析



2. 数据字典

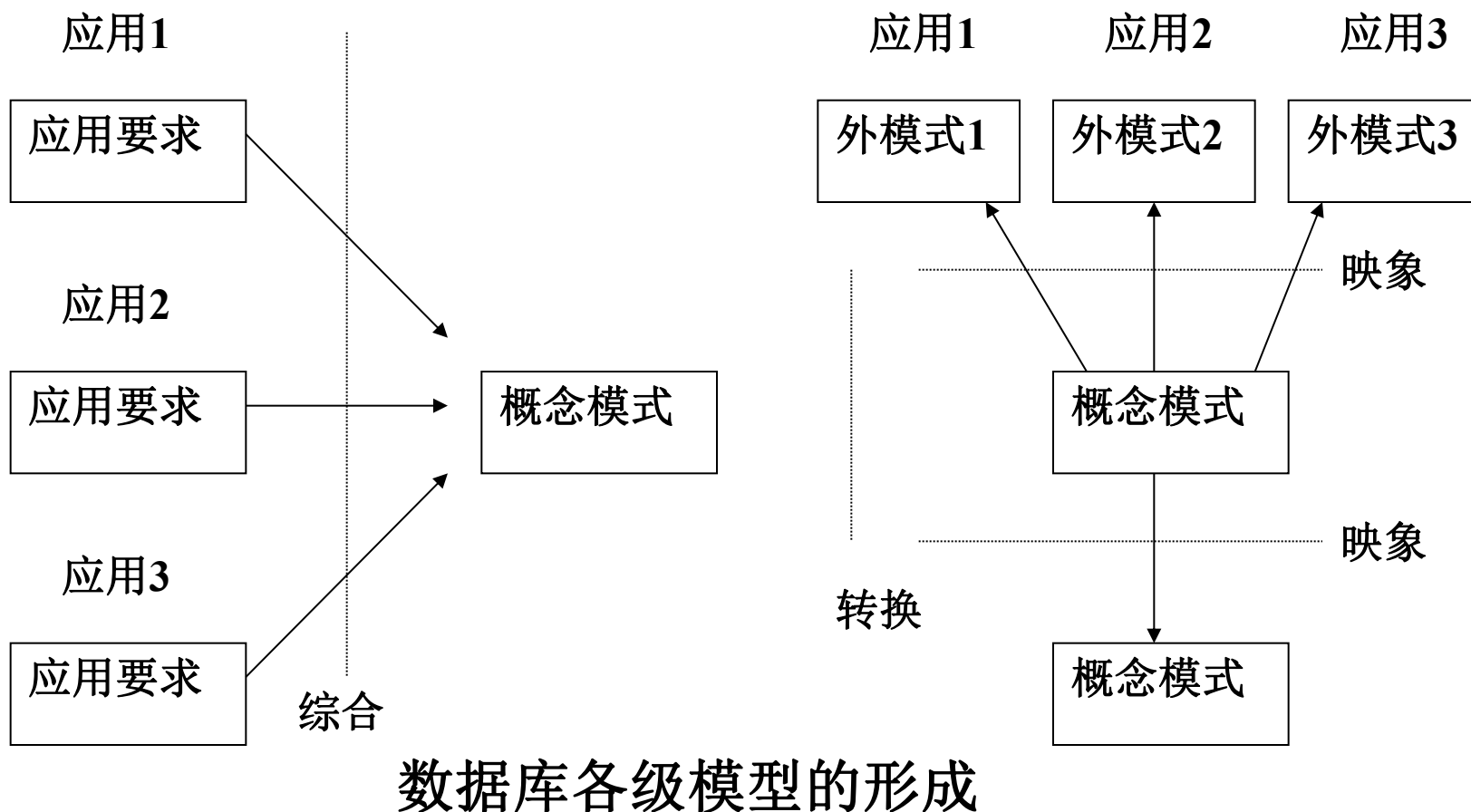
- 数据字典是对系统中数据的详细描述，是各类数据结构和属性的清单。它与数据流图互为注释。
- 数据字典贯穿于数据库需求分析直到数据库运行的全过程，在不同的阶段其内容和用途各有区别。
- 在需求分析阶段，它通常包含以下五部分内容。
 - (1) 数据项
 - (2) 数据结构
 - (3) 数据流
 - (4) 数据存储
 - (5) 处理过程

14.3 概念结构设计

14.3.1 概念结构设计的必要性

- 概念设计：将需求分析得到的用户需求抽象为信息结构，即概念模型。
- **P.P.S.chen**设计了基于**E-R**模型的数据库设计方法，即在需求分析和逻辑设计之间增加了一个概念设计阶段。
- 概念设计好处
 - (1) 各阶段的任务相对单一化，设计复杂程度低，组织管理容易
 - (2) 概念模型不受特定的DBMS的限制，也独立于存储安排和效率方面的考虑，因而比逻辑模型更为稳定。
 - (3) 概念模型不含具体的DBMS所附加的技术细节，更容易为用户所理解，因而更有可能准确反映用户的信息需求。

数据库各级模型的形成



14.3.2 概念模型的特点

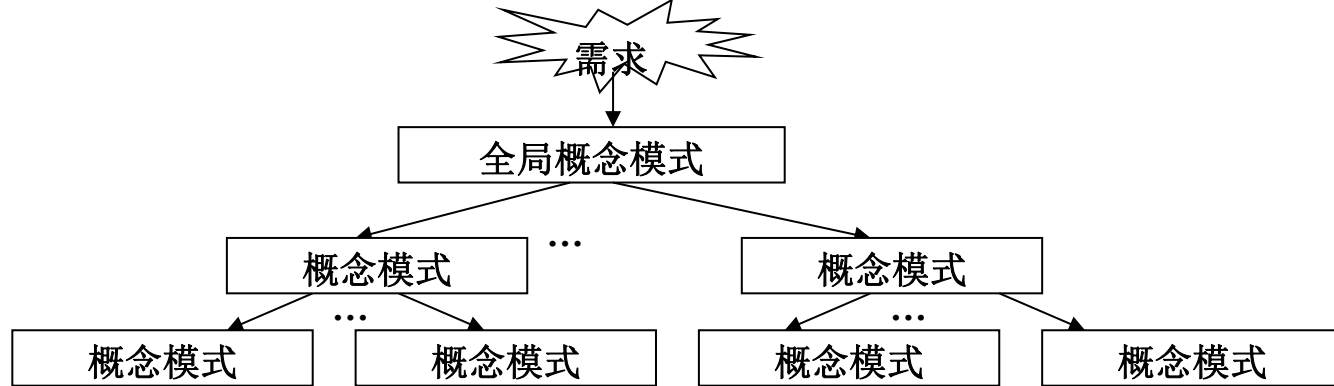
- 概念模型作为概念设计的表达工具，为数据库提供一个说明性结构，是设计数据库逻辑结构即逻辑模型的基础
- 概念模型必须具备以下特点：
 - (1) 语义表达能力丰富。
 - (2) 易于交流和理解。
 - (3) 易于修改和扩充。
 - (4) 易于向各种数据模型转换。
- 人们提出了许多概念模型，其中最著名、最实用的一种是**E-R**模型，它将现实世界的信息结构统一用属性、实体以及它们之间的联系来描述

14.3.3 概念结构设计的方法与步骤

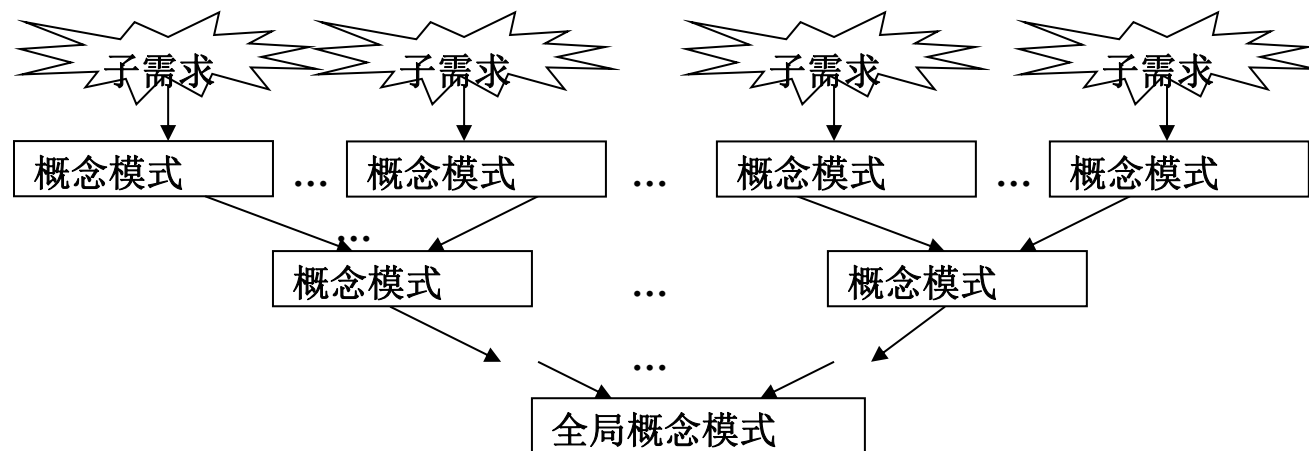
1. 概念结构设计的方法

- 设计概念结构的E-R模型可采用四种方法
 - (1) **自顶向下**。先定义全局概念结构E-R模型的框架，再逐步细化。
 - (2) **自底向上**。先定义各局部应用的概念结构E-R模型，然后将它们集成，得到全局概念结构E-R模型。
 - (3) **逐步扩张**。先定义最重要的核心概念E-R模型，然后向外扩充，以滚雪球的方式逐步生成其他概念结构E-R模型。
 - (4) **混合策略**。该方法采用自顶向下和自底向上相结合的方法，先自顶向下定义全局框架，再以它为骨架集成自底向上方法中设计的各个局部概念结构。
- 最常用的方法是自底向上。即自顶向下地进行需求分析，再自底向上地设计概念结构。

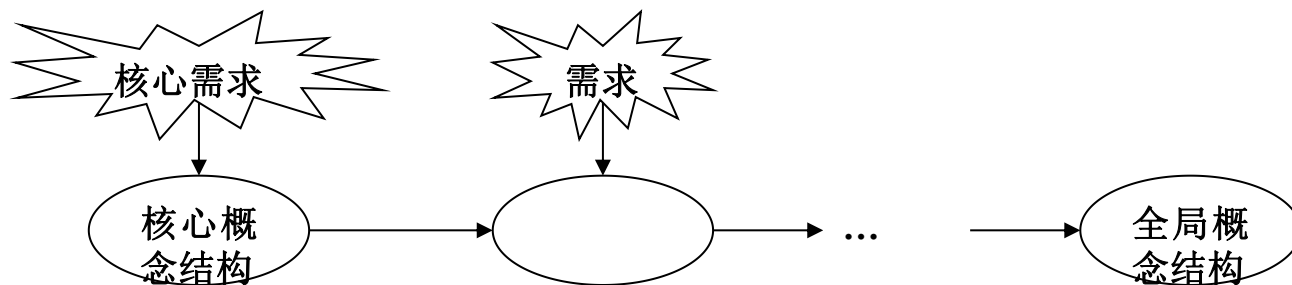
概念结构设计的方法



(a) 自顶向下的设计方法



(b) 自底向上的设计方法



(c) 逐步扩张的设计方法

概念结构设计的步骤

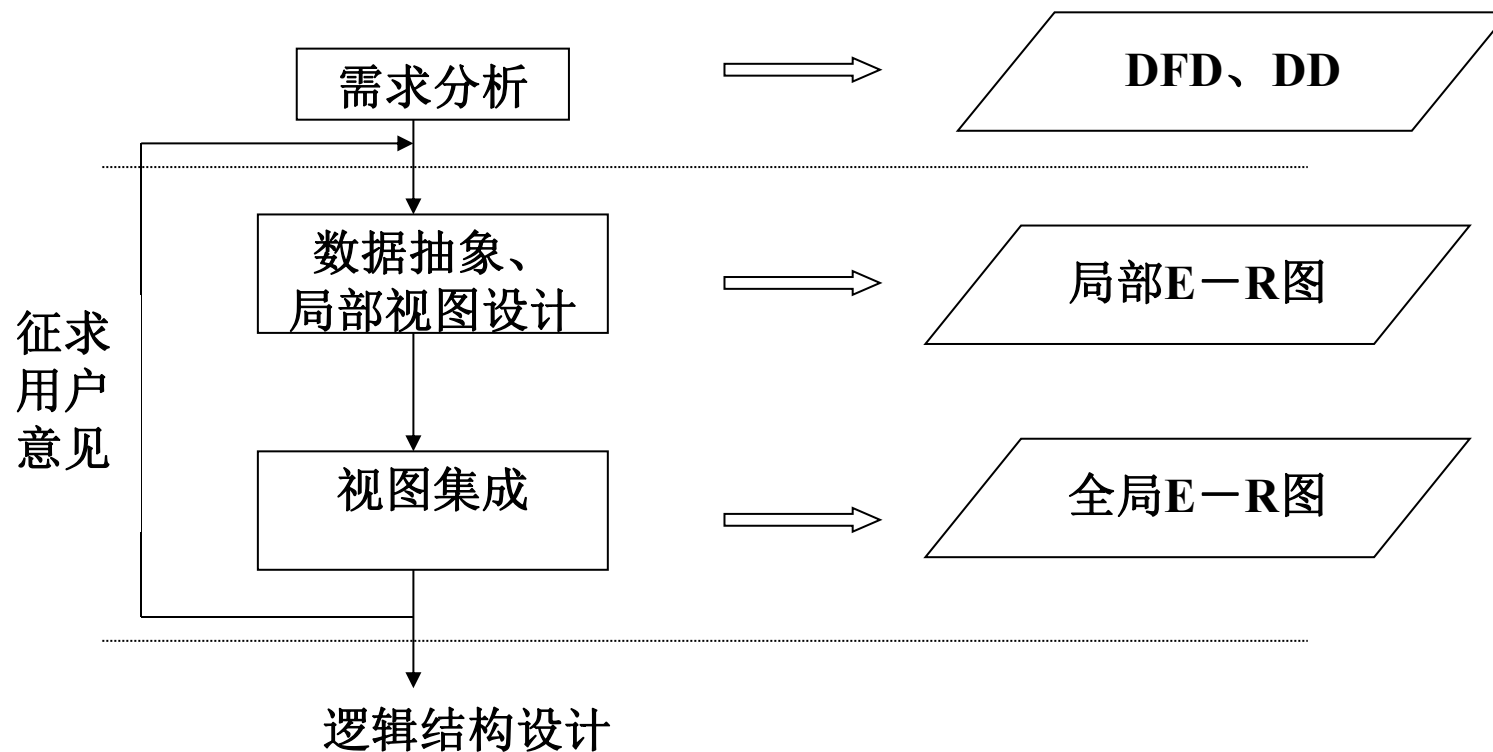
2. 概念结构设计的步骤

- 自底向上的设计方法可分为两步：
 - (1) 进行数据抽象，设计局部E-R模型，即设计用户视图
 - (2) 集成各局部E-R模型，形成全局E-R模型，即视图的集成

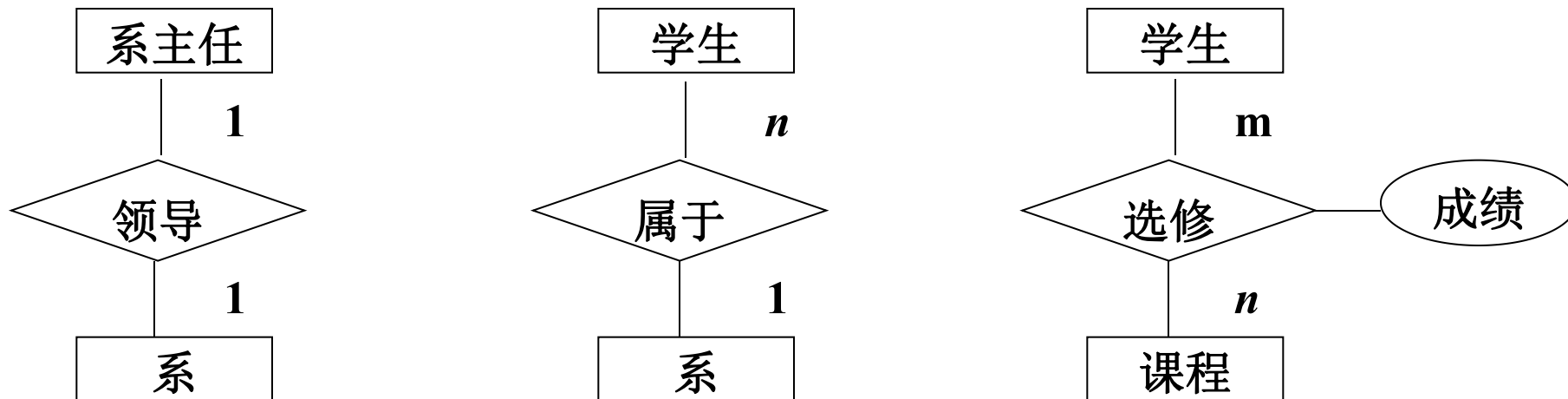
3. 数据抽象与局部E-R模型设计

- 概念结构是对现实世界的一种抽象。
- 所谓抽象是对实际的人、物、事和概念进行人为处理，抽取人们关心的共同特性，忽略非本质的细节，并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
- 概念结构设计首先要根据需求分析得到的结果（数据流图、数据字典等）对现实世界进行抽象，设计各个局部E-R模型

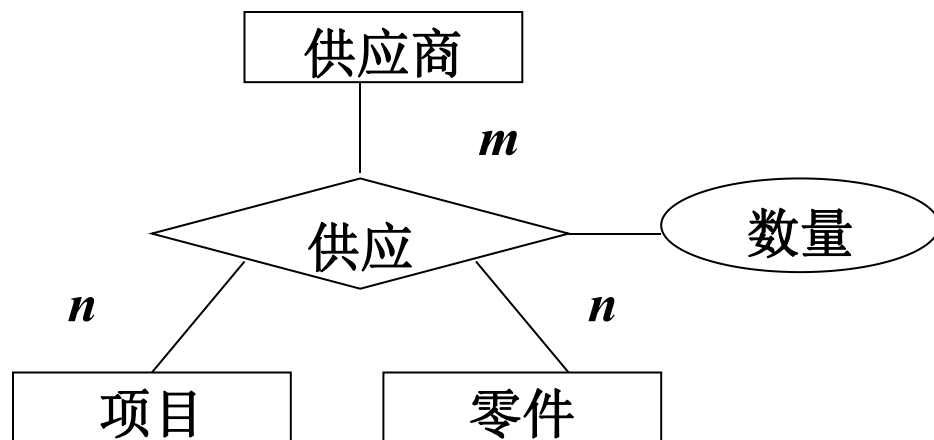
自底向上方法



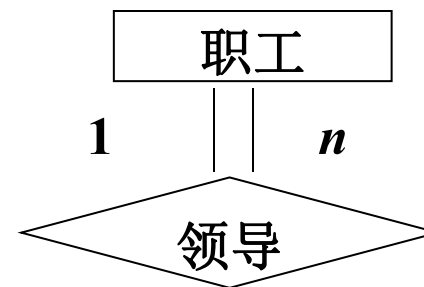
自底向上方法的设计步骤



(a)两个实体之间的联系



(b)多个实体之间的联系



(c)实体集内部的联系

实体及其联系图——局部E-R模型

(2) 数据抽象

- 在系统需求分析阶段，最后得到了多层数据流图、数据字典和系统分析报告。
- 建立局部E-R模型，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，作为设计分E-R图的出发点，让这组图中每一部分对应一个局部应用。
- 在前面选好的某一层的数据流图中，每个局部应用都对应了一组数据流图，局部应用所涉及的数据存储在数据字典中。
- 设计局部**E-R**模型的关键就是正确划分实体和属性。实体和属性之间在形式上并无可以明显区分的界限，通常是按照现实世界中事物的自然划分来定义实体和属性，将现实世界中的事物进行数据抽象，得到实体和属性。
- 一般有两种数据抽象：分类和聚集。

两种数据抽象

①分类 (Classification)

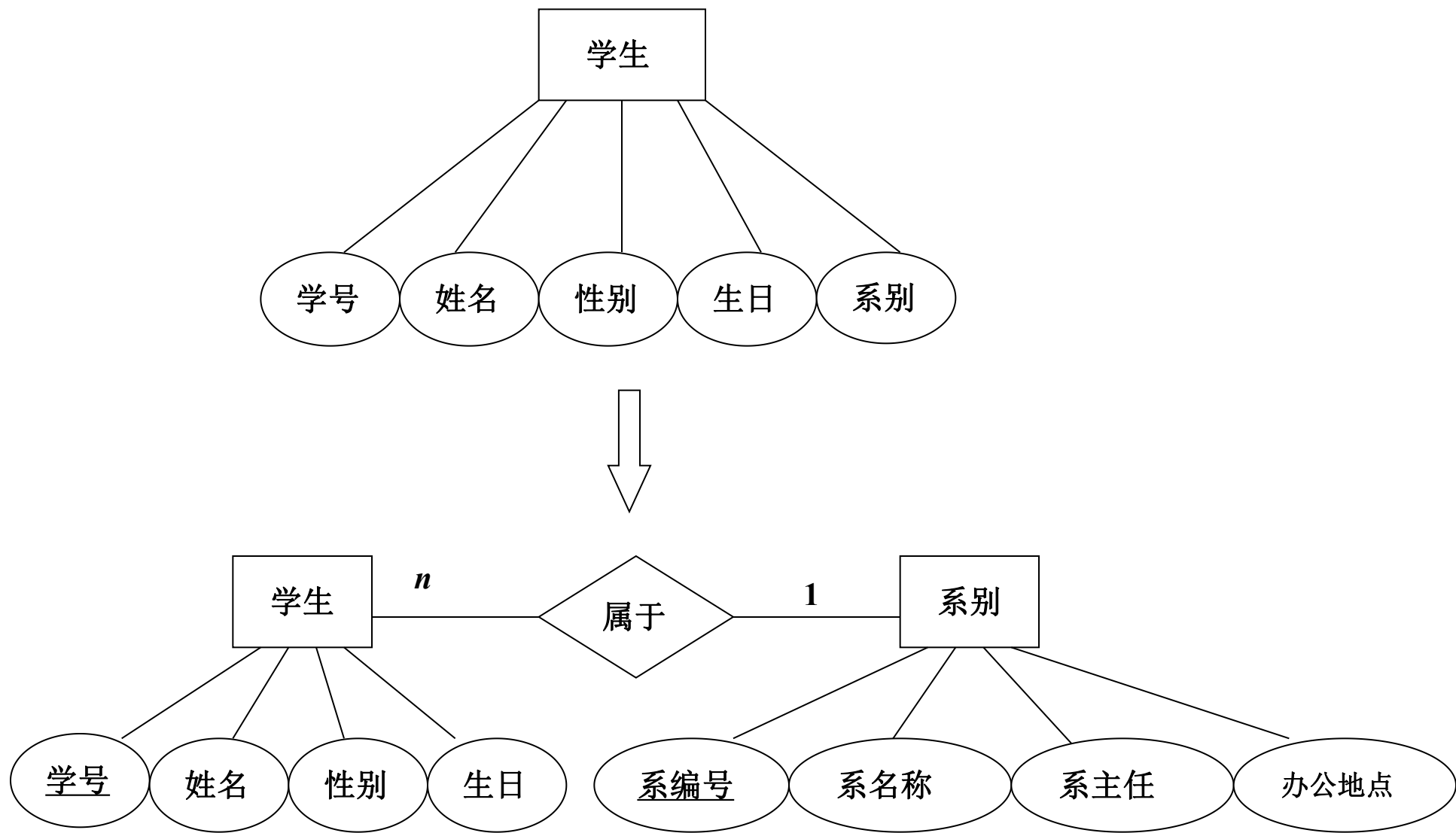
- 分类定义某一类概念作为现实世界中一组对象的类型，将一组具有某些共同特性和行为的对象抽象为一个实体。对象和实体之间是“is member of”的关系。
- 例如，在教学管理中，“赵亦”是一名学生，表示“赵亦”是学生中的一员，具有学生们共同的特性和行为

②聚集 (Aggregation)

- 聚集定义某一类型的组成成份，将对象类型的组成成份抽象为实体的属性。组成成份与对象类型之间是“is part of”的关系。
- 例如，学号、姓名、性别、生日、系别等可以抽象为学生实体的属性，其中学号是标识学生实体的主键。

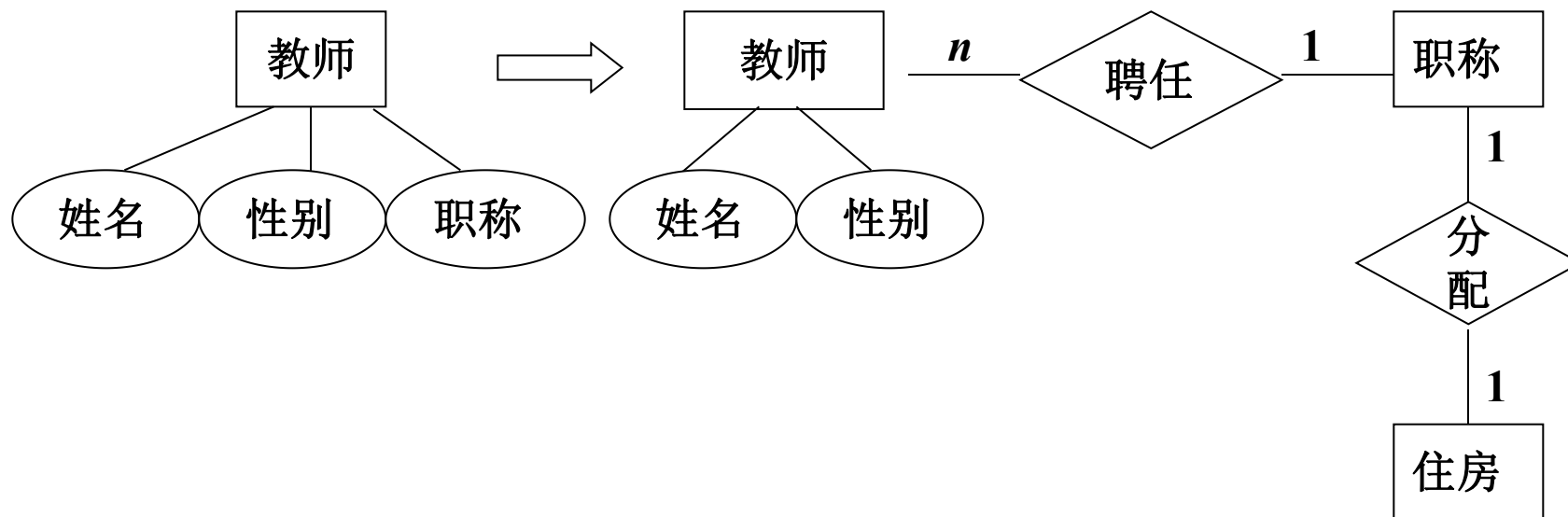
(2) 局部E-R模型设计

- 数据抽象后得到了实体和属性，实际上实体和属性是相对而言的，往往要根据实际情况进行必要的调整。在调整中要遵循两条原则：
 - ①实体具有描述信息，而属性没有。即属性必须是不可分的数据项，不能再由另一些属性组成。
 - ②属性不能与其他实体具有联系，联系只能发生在实体之间。
 - 例如：学生是一个实体，学号、姓名、性别、生日、系别等是学生实体的属性。则根据原则①可以作为学生实体的属性。但如果考虑一个系的系主任、学生人数、教师人数、办公地点等，则系别应看作一个实体。
 - “职称”为教师实体的属性，但在涉及住房分配时，由于分房与职称有关，即职称与住房实体之间有联系，则根据原则②，职称应作为一个实体。



系别作为一个属性或实体

职称作为一个属性或实体

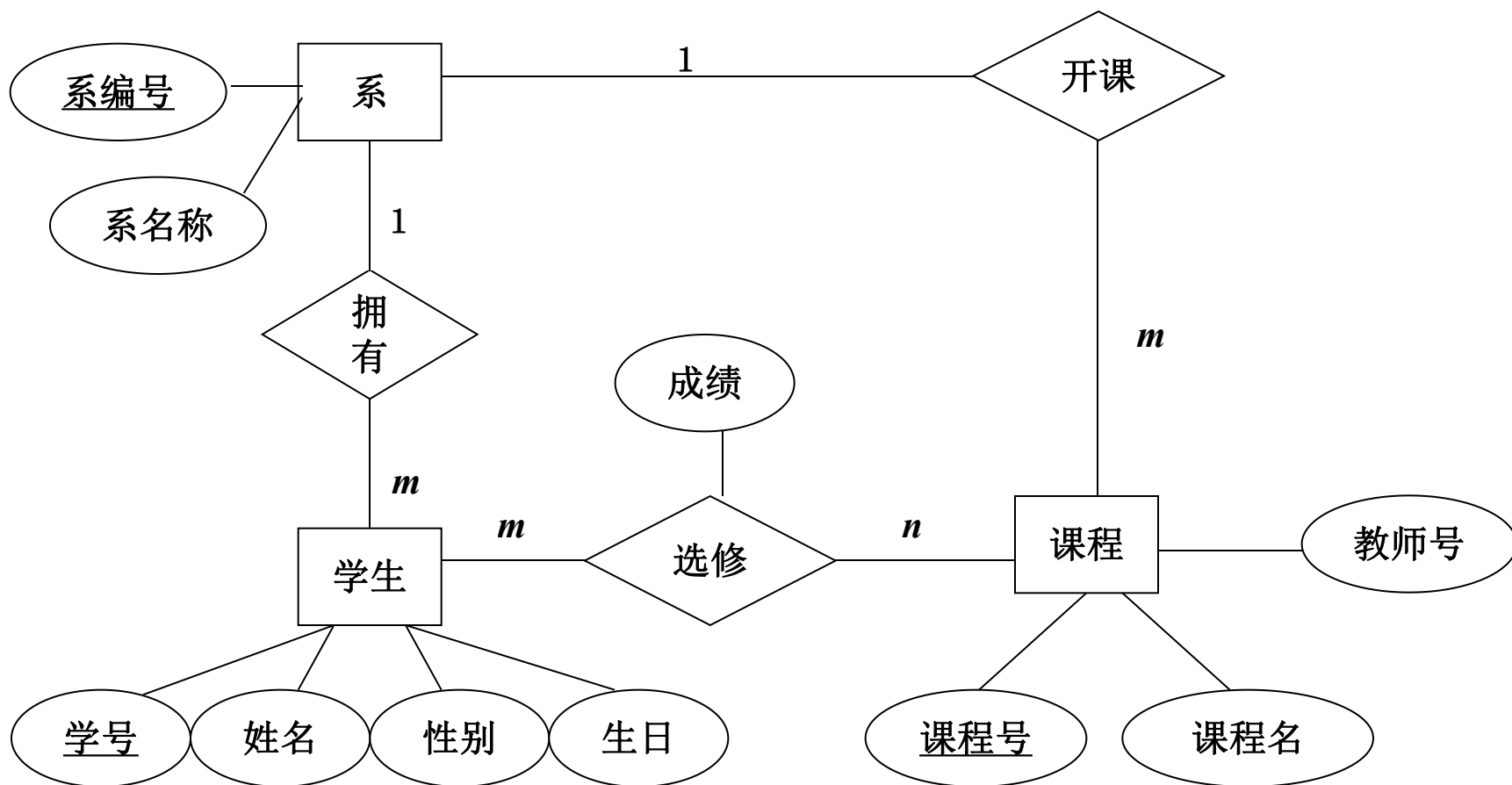


职称作为一个属性或实体

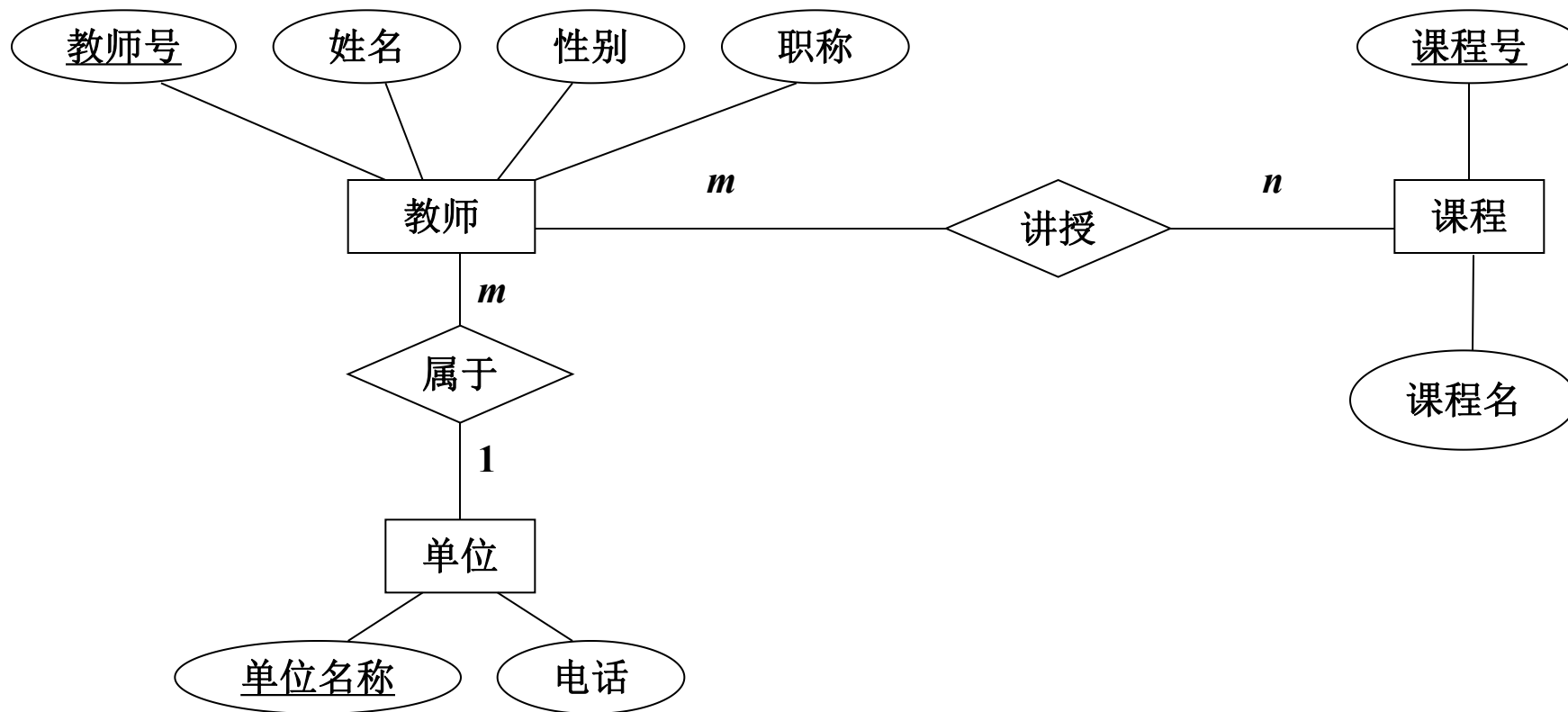
局部E-R模型设计举例

- 在简单的教务管理系统中，有如下语义约束。
 - ①一个学生可选修多门课程，一门课程可为多个学生选修，因此学生和课程是多对多的联系；
 - ②一个教师可讲授多门课程，一门课程可为多个教师讲授，因此教师和课程也是多对多的联系；
 - ③一个系可有多个教师，一个教师只能属于一个系，因此系和教师是一对多的联系，同样系和学生也是一对多的联系
- 根据上述约定，可以得到学生选课局部 E — R 图和教师任课局部 E — R 图。形成局部**E-R**模型后，应该返回去征求用户意见，以求改进和完善，使之如实地反映现实世界。
- **E-R**图的优点就是易于被用户理解，便于交流。

学生选课局部 E-R 图

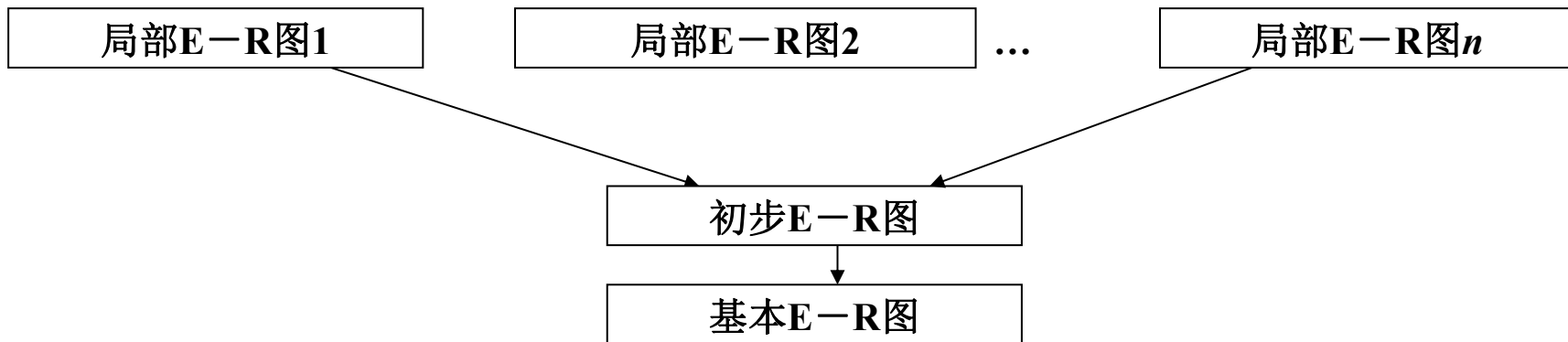


教师任课局部 E-R 图

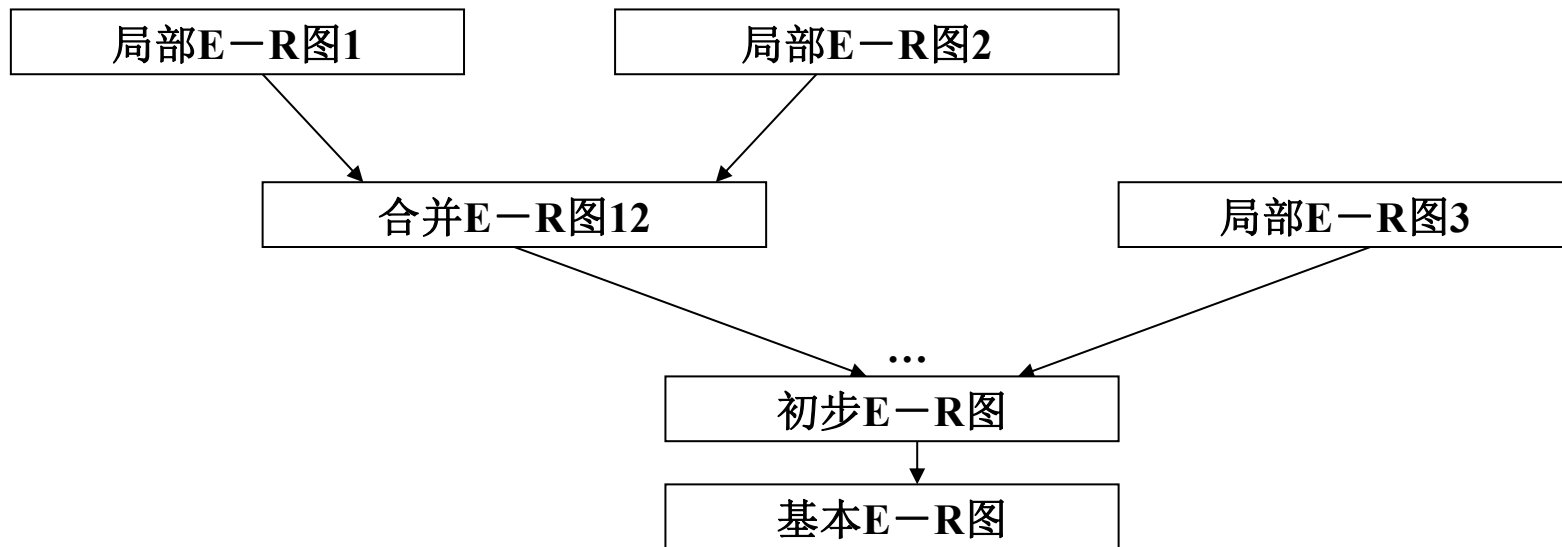


4. 全局E-R模型设计

- 局部E-R模型设计完成之后，下一步就是集成各局部E-R模型，形成全局E-R模型，即视图的集成。
- 视图集成的方法有两种：
 - ①**多元集成法**，一次性将多个局部E-R图合并为一个全局E-R图，如果局部视图比较简单，可以采用多元集成法
 - ②**二元集成法**，首先集成两个重要的局部视图，以后用累加的方法逐步将一个新的视图集成进来，采用二元集成法，即每次只综合两个视图，可降低难度
- 在实际应用中，可以根据系统复杂性选择这两种方案。无论使用哪一种方法，视图集成均分成两个步骤
 - ①**合并**，消除各局部E-R图之间的冲突，生成初步E-R图
 - ②**优化**，消除不必要的冗余，生成基本E-R图



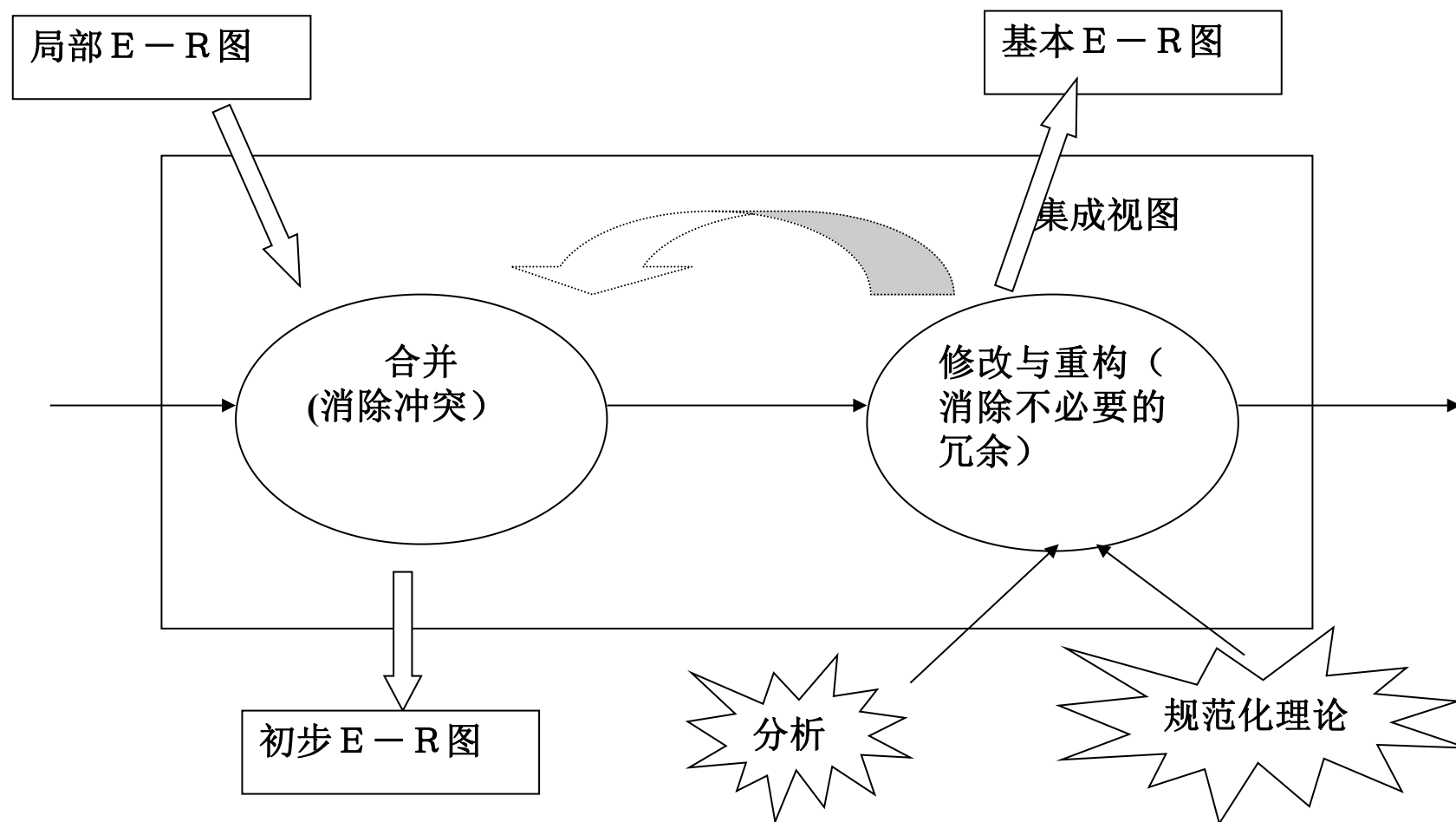
(a) 多元集成法



(b) 二元集成法

局部视图合并成全局视图

视图集成



(1) 合并局部E-R图，生成初步E-R图

- 全局概念结构支持所有的局部E-R模型，且必须合理地表示一个完整、一致的数据库概念结构
- 由于各个局部应用不同，通常由不同的设计人员进行局部**E-R**图设计，因此，各局部**E-R**图不可避免地会有许多不一致的地方，我们称之为**冲突**。
- 合并局部**E-R**图时必须消除各个局部**E-R**图中的不一致，使合并后的全局概念结构不仅支持所有的局部**E-R**模型，而且必须是一个能为全系统中所有用户共同理解和接受的完整的概念模型。
- **合并局部E-R图的关键就是合理消除各局部E-R图中的冲突。**

①属性冲突

- E-R图中冲突有三种：属性冲突、命名冲突和结构冲突
- ①属性冲突
- 属性冲突又分为属性值域冲突和属性的取值单位冲突
 - a. 属性值域冲突，即属性值的类型、取值范围或取值集合不同。比如学号，有些部门将其定义为数值型，而有些部门将其定义为字符型。又如生日，有的可能用出生年月表示，有的则用整数表示
 - b. 属性的取值单位冲突。比如零件的重量，有的以公斤为单位，有的以斤为单位，有的则以克为单位
- 属性冲突属于用户业务上的约定，必须与用户协商后解决

②命名冲突

- 命名不一致可能发生在实体名、属性名或联系名之间，其中属性的命名冲突更为常见。
- 一般表现为同名异义或异名同义（实体、属性、联系名）
 - a. **同名异义**，即同一名字的对象在不同的部门中具有不同的意义。
例：“单位”在某些部门表示为人员所在的部门，而在某些部门可能表示物品的重量、长度等属性
 - b. **异名同义**，即同一意义的对象在不同的部门中具有不同的名称。
例：对于“房间”这个名称，在教务管理部门中对应着为教室，而在后勤管理部门对应为学生宿舍。
- 命名冲突的解决方法同属性冲突，需要与各部门协商、讨论后加以解决。

③结构冲突

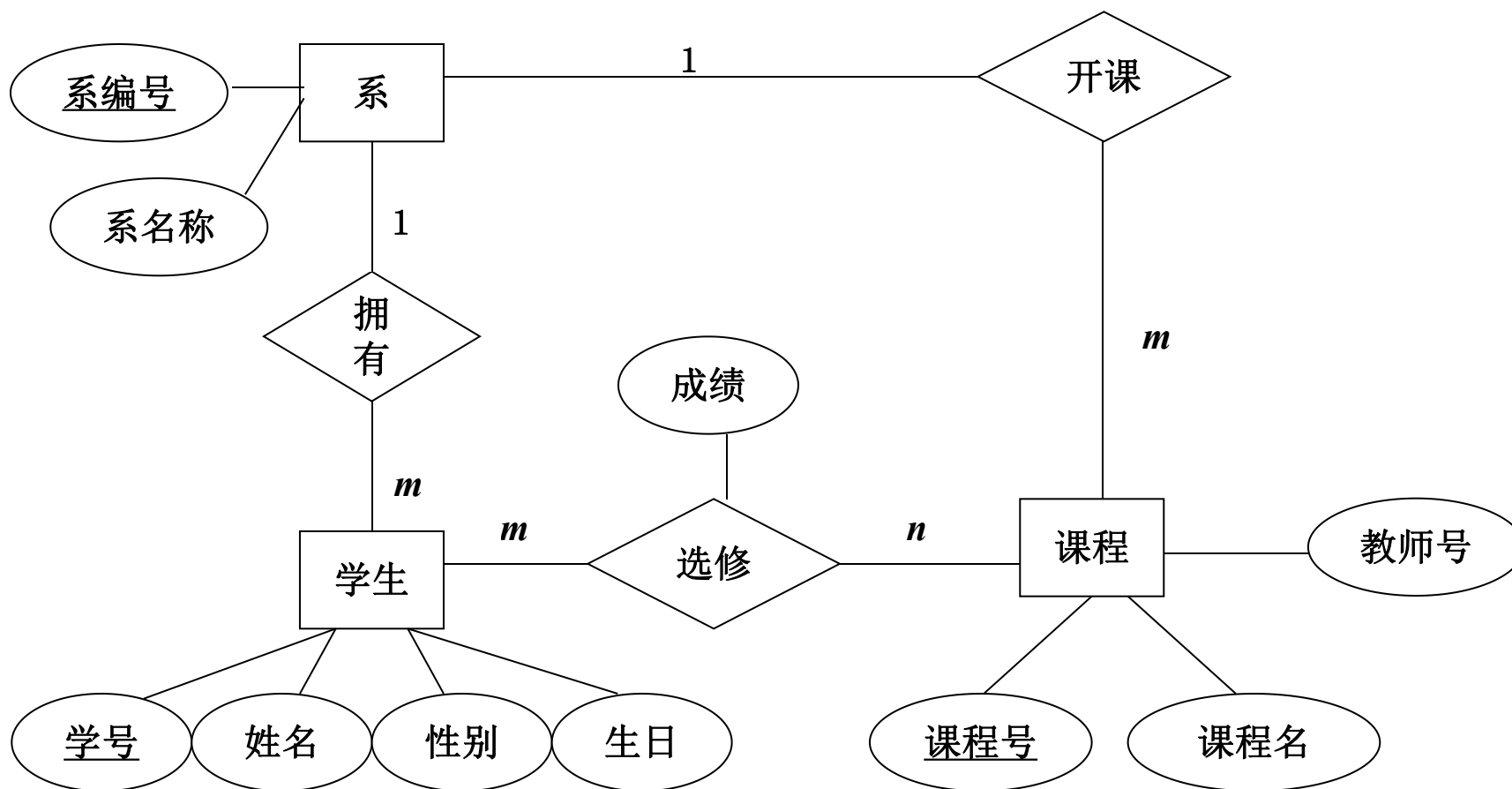
a. 同一对象在不同应用中有不同的抽象，可能为实体，也可能为属性。例如，教师的职称在某一局部应用中被当作实体，而在另一局部应用中被当作属性。

解决办法：使同一对象在不同应用中具有相同的抽象，或把实体转换为属性，或把属性转换为实体。

b. 同一实体在不同应用中属性组成不同，可能是属性个数或属性次序不同。解决办法：合并后实体的属性组成为各局部E-R图中的同名实体属性的并集，然后再适当调整属性的次序。

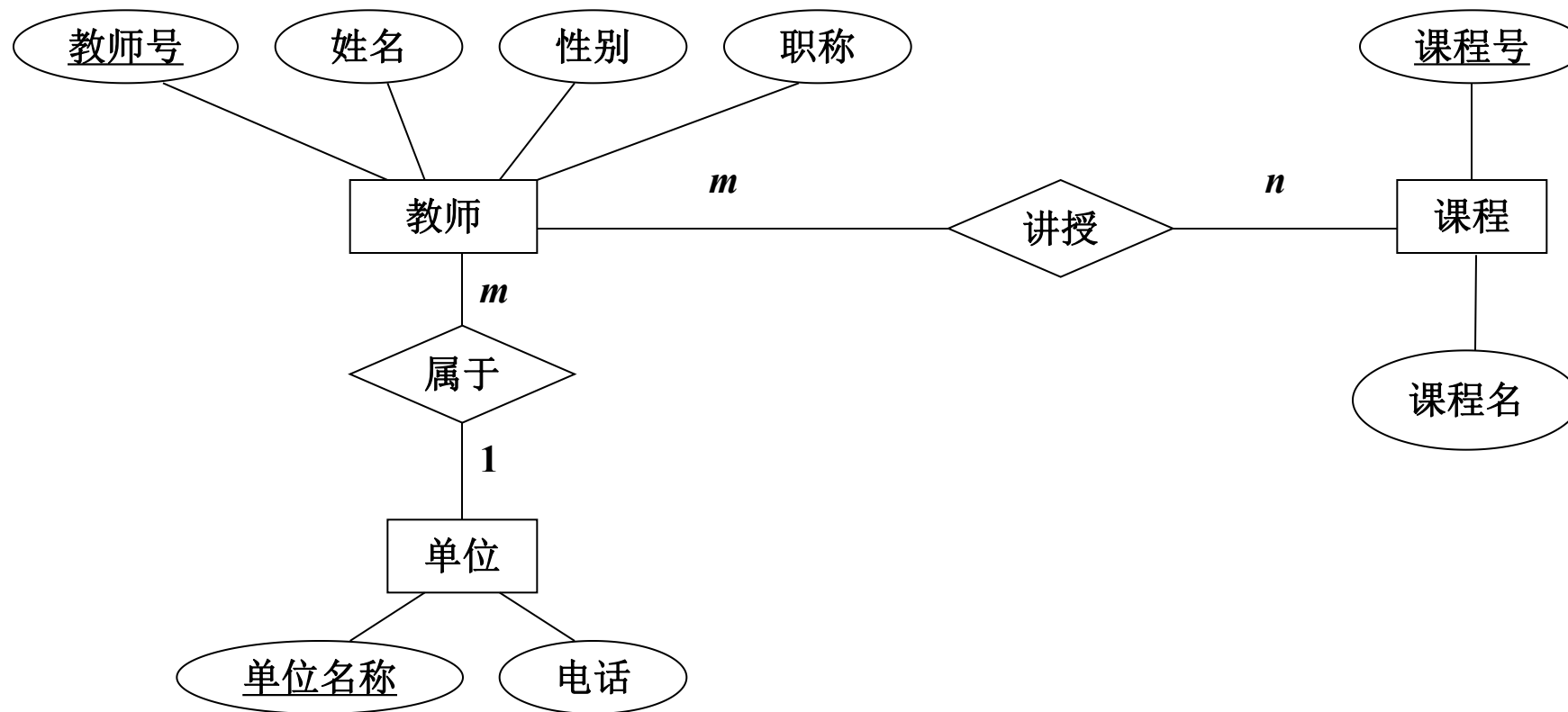
c. 同一联系在不同应用中呈现不同的类型。比如E1与E2在某一应用中可能是一对一联系，而在另一应用中可能是一对多或多对多联系，也可能是在E1、E2、E3三者之间有联系。解决办法：根据应用的语义对实体联系的类型进行综合或调整。

消除各局部E-R图之间的冲突生成初步E-R图



学生选课局部 E - R 图

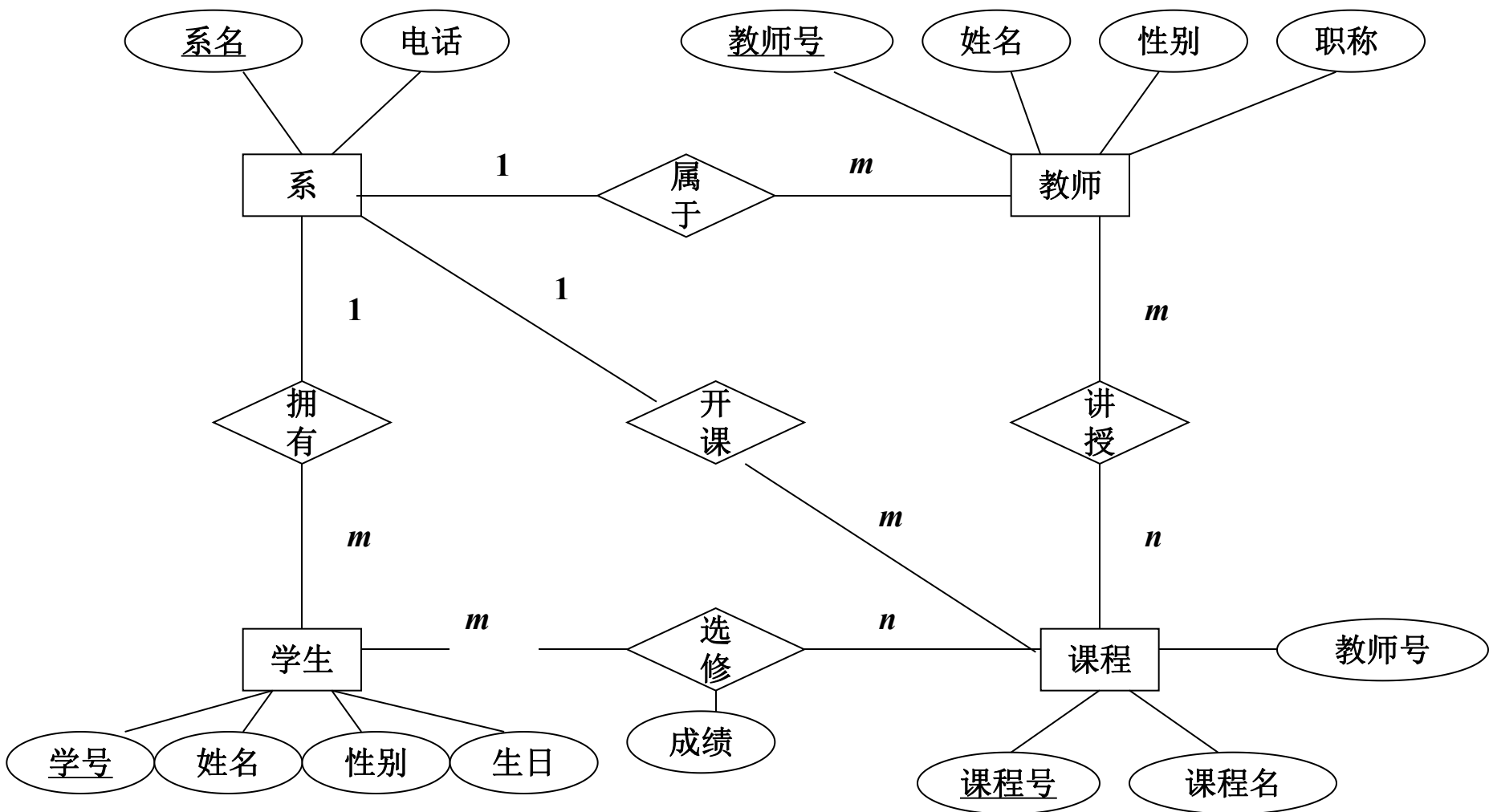
消除各局部E-R图之间的冲突生成初步E-R图



教师任课局部 E — R 图

消除各局部E-R图之间的冲突

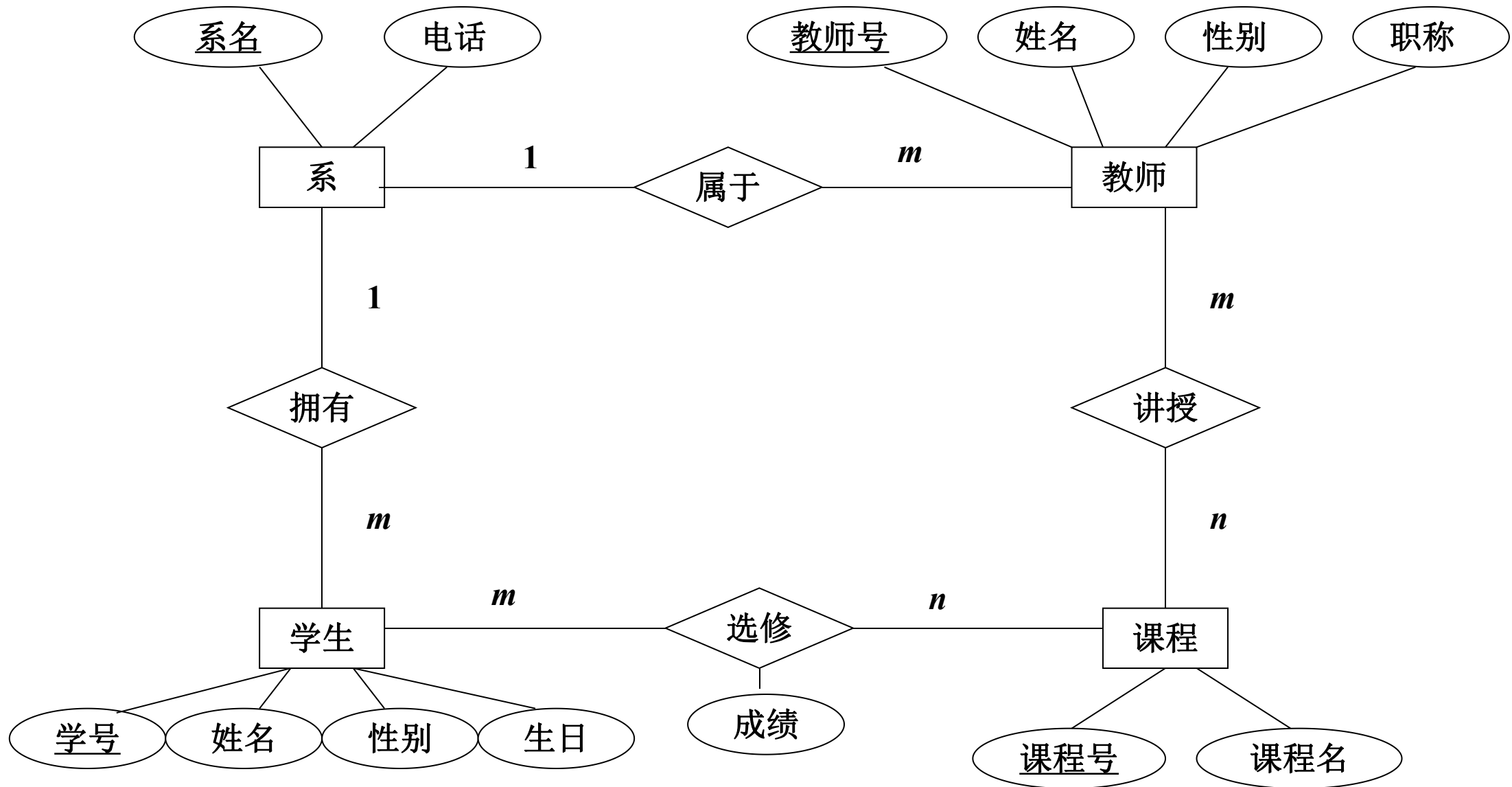
- 以教务管理系统中的两个局部E-R图为例, 消除各局部E-R图之间的冲突, 进行局部E-R模型的合并, 生成初步E-R图。
 - 首先, 这两个局部E-R图中存在着命名冲突, 学生选课局部 E - R 图中的实体“系”与教师任课局部 E - R 图中的实体“单位”, 都是指“系”, 即所谓的异名同义, 合并后统一改为“系”, 这样属性“名称”和“单位”即可统一为“系名”。
 - 其次, 还存在着结构冲突, 实体“系”和实体“课程”在两个不同应用中的属性组成不同, 合并后这两个实体的属性组成为原来局部E-R图中的同名实体属性的并集。解决上述冲突后, 合并两个局部E-R图, 生成初步的全局E-R图。



教务管理系统的初步E—R图

(2) 消除不必要的冗余，生成基本E-R图

- 所谓冗余，指冗余的数据和实体之间冗余的联系。冗余的数据是指可由基本的数据导出的数据，冗余的联系是由其他的联系导出的联系
- 冗余的存在容易破坏数据库的完整性，给数据库的维护增加困难，应该消除。把消除了冗余的初步E-R图称为基本E-R图。
- 通常采用分析的方法消除冗余。数据字典是分析冗余数据的依据，还可以通过数据流图分析出冗余的联系。
- 如图所示的初步E-R图中，“课程”实体中的属性“教师号”可由“讲授”这个教师与课程之间的联系导出



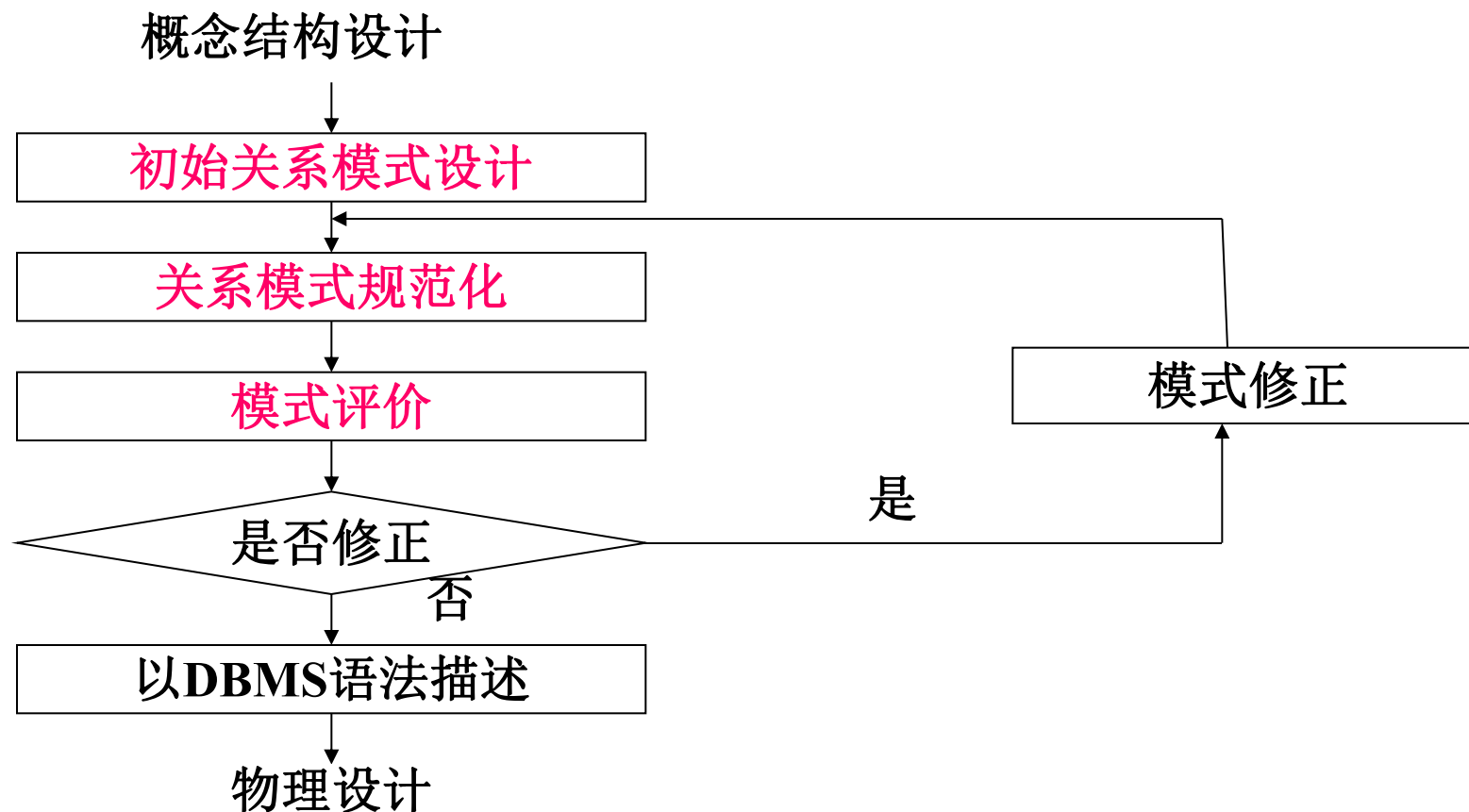
教务管理系统的基本E-R图

14.4 逻辑结构设计

14.4.1 逻辑结构设计的任务和步骤

- 概念结构设计阶段得到的E-R模型是用户的模型，它独立于任何一种数据模型，独立于任何一个具体的DBMS。
- 为了建立用户所要求的数据库，需要把上述概念模型转换为某个具体的DBMS所支持的数据模型。
- 数据库逻辑设计的任务是将概念结构转换成特定DBMS所支持的数据模型的过程。实现设计阶段，需要考虑到具体的DBMS的性能、具体的数据模型特点。
- 从E-R图所表示的概念模型可以转换成任何一种具体的DBMS所支持的数据模型，如网状模型、层次模型和关系模型。下面介绍E-R图如何向关系模型进行转换。
- 一般的逻辑设计分为以下三步
 - (1) 初始关系模式设计；
 - (2) 关系模式规范化；
 - (3) 模式的评价与改进。

关系数据库的逻辑设计

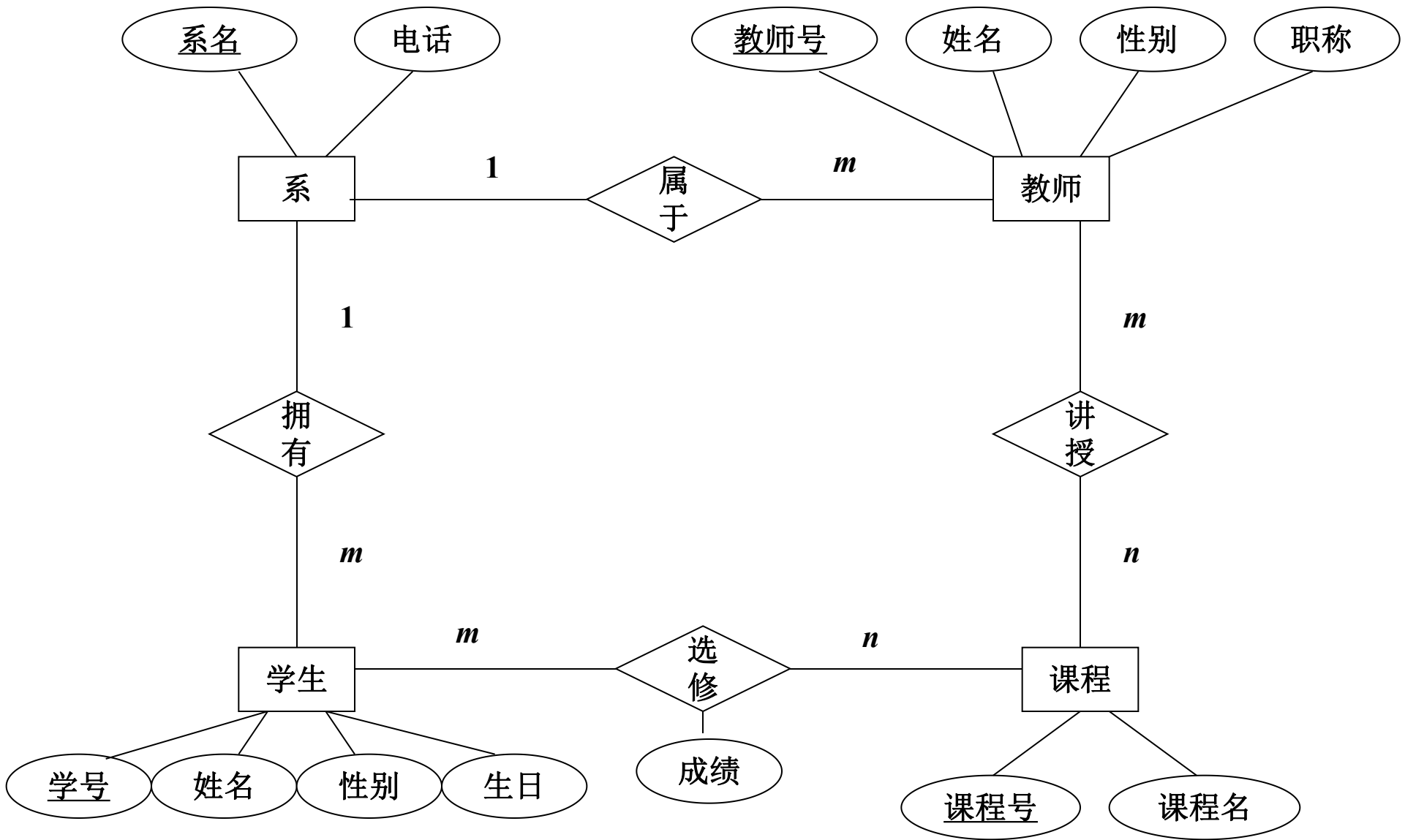


关系数据库的逻辑设计

14.4.2 初始关系模式设计

1. 转换原则

- 概念设计中得到的E-R图是由实体、属性和联系组成的，而关系数据库逻辑设计的结果是一组关系模式的集合。
- 将E-R图转换为关系模型实际就是将实体、属性和联系转换成关系模式
- E-R图转换为关系模型转换原则：
 - (1) 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，实体的键就是关系的键。
 - (2) 一个联系转换为一个关系模式，与该联系相连的各实体的键以及联系的属性均转换为该关系的属性。该关系的键有三种情况：
 - ①如果联系为1:1，则每个实体的键都是关系的候选键；
 - ②如果联系为1: n ，则 n 端实体的键是关系的键；
 - ③如果联系为 $m:n$ ，则各实体键的组合是关系的键。



教务管理系统的基本E-R图

2. 具体做法

(1) 把每一个实体转换为一个关系

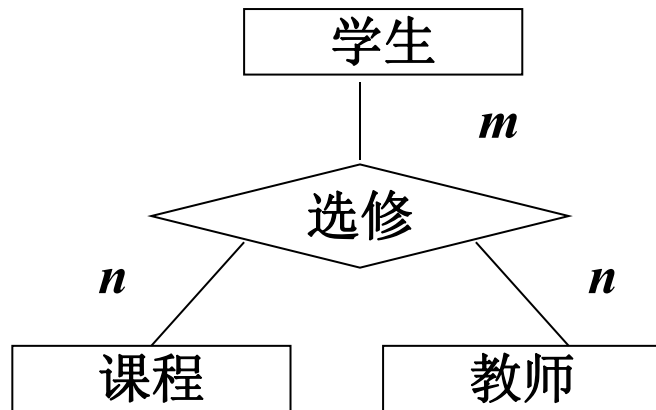
- 首先分析各实体的属性，从中确定其主键，然后分别用关系模式表示。例如，以教务管理系统的基本E-R图为例，四个实体分别转换成四个关系模式：
 - 学生（学号，姓名，性别，生日）
 - 课程（课程号，课程名）
 - 教师（教师号，姓名，性别，职称）
 - 系（系名，电话）
- 其中，有下划线者表示是主键。

(2) 把每一个联系转换为关系模式

- 由联系转换得到的关系模式的属性集中，包含两个发生联系的实体中的主键以及联系本身的属性，其关系键的确定与联系的类型有关。
- 例如，以教务管理系统的基本E—R图为例，四个联系也分别转换成四个关系模式：
 - 属于（教师号，系名）
 - 讲授（教师号，课程号）
 - 选修（学号，课程号，成绩）
 - 拥有（系名，学号）

(3) 特殊情况的处理

- 三个或三个以上实体间的一个多元联系在转换为一个关系模式时，与该多元联系相连的各实体的主键及联系本身的属性均转换成为关系的属性，转换后所得到的关系的主键为各实体键的组合。
- 例如，表示学生、课程、教师的选课关系



多个实体之间的联系

- 选修（学号，课程编号，教师编号）

14.4.3 关系模式规范化

- 应用规范化理论对上述产生的关系的逻辑模式进行初步优化，以减少乃至消除关系模式中存在的各种异常，改善完整性、一致性和存储效率。
- 规范化过程两个步骤
 - 1. 确定范式级别
 - 考查关系模式的函数依赖关系，确定范式等级，逐一分析各关系模式
 - 考查是否存在部分函数依赖，传递函数依赖等，确定它们分别属于第几范式
 - 2. 实施规范化处理
 - 确定范式级别后，利用规范化理论，逐一考察各个关系模式
 - 根据应用要求，判断它们是否满足规范要求

规范化理论在数据库设计中的应用

- (1) 在需求分析阶段，用数据依赖概念分析和表示各个数据项之间的联系。
- (2) 在概念结构设计阶段，以规范化理论为指导，确定关系键，消除初步E-R图中冗余的联系。
- (3) 在逻辑结构设计阶段，从E-R图向数据模型转换过程中，用模式合并与分解方法达到规范化级别。

14.4.4 模式评价与改进

1. 模式评价

- 关系模式的规范化不是目的而是手段，数据库设计的目的是最终满足应用需求。为了提高数据库应用系统的性能，还应该对规范化后产生的关系模式进行评价、改进，经过反复多次的尝试和比较，最后得到优化的关系模式。
- 模式评价的目的是检查所设计的数据库模式是否满足用户的功能要求、效率，确定加以改进的部分。模式评价包括功能评价和性能评价。

2 性能评价

- 对于目前得到的数据库模式，由于缺乏物理设计所提供的数量测量标准和相应的评价手段，所以性能评价是比较困难的，只能对实际性能进行估计，包括逻辑记录的存取数、传送量以及物理设计算法的模型等。

2. 模式改进

- 根据模式评价的结果，对已生成的模式进行改进
 - 如果因为需求分析、概念设计的疏漏导致某些应用不能得到支持，则应该增加新的关系模式或属性。
 - 如果因为性能考虑而要求改进，可采用合并或分解的方法

(1) 合并

- 如果有若干个关系模式具有相同的主键，并且对这些关系模式的处理主要是查询操作，而且经常是多关系的查询，那么可对这些关系模式按照组合使用频率进行合并，减少联接操作而提高查询效率。

(2) 分解

- 为了提高数据操作的效率和存储空间的利用率，最常用和最重要的模式优化方法就是分解，根据应用的不同要求，可以对关系模式进行垂直分解和水平分解。

14.5 数据库物理设计

- 数据库最终要存储在物理设备上
- **数据库物理设计** 是对于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
- 物理设计的任务是为了有效地实现逻辑模式，确定所采取的存储策略
- 物理设计阶段是以逻辑设计的结果作为输入，结合具体DBMS的特点与存储设备特性进行设计，选定数据库在物理设备上的存储结构和存取方法
- 数据库的物理设计可分为两步：
 - (1) **确定物理结构**，在关系数据库中主要指存取方法和存储结构；
 - (2) **评价物理结构**，评价的重点是时间和空间效率。

14.5.1 确定物理结构

- 设计人员必须深入了解给定的DBMS的功能，DBMS提供的环境和工具、硬件环境,特别是存储设备的特征。
- 了解应用环境的具体要求，如各种应用的数据量、处理频率和响应时间等
- 确定物理结构内容
 - 1. 存储记录结构的设计
 - 2. 访问方法的设计
 - 3. 数据存放位置的设计
 - 4. 系统配置的设计

1. 存储记录结构的设计

- 在物理结构中，数据的基本存取单位是存储记录
- 逻辑记录结构确定后，就可以设计存储记录结构，一个存储记录可以和一个或多个逻辑记录相对应
- 存储记录结构包括记录的组成、数据项的类型和长度，以及逻辑记录到存储记录的映射。
- 文件是某一类型的所有存储记录的集合。文件的存储记录可以是定长的，也可以是变长的。
- 文件组织或文件结构是组成文件的存储记录的表示法，文件结构表示文件格式、逻辑次序、物理次序、访问路径、物理设备分配
- 物理数据库就是指数据库中实际存储记录的格式、逻辑次序和物理次序、访问路径、物理设备的分配。
- 决定存储结构的主要因素包括存取时间、存储空间和维护代价三个方面。设计时应根据实际情况进行综合权衡。一般**DBMS**提供一定的灵活性可供选择，包括聚簇和索引

(1) 聚簇 (Cluster)

- 聚簇就是为了提高查询速度，把在一个（或一组）属性上具有相同值的元组集中地存放在一个物理块中。如果存放不下，可以存放在相邻的物理块中。其中，这个（或这组）属性称为聚簇码。
- 聚簇有两个作用：
 - ①节省存储空间：使用聚簇以后，聚簇码相同的元组集中在一起了，因而聚簇值不必在每个元组中重复存储，只要在一组中存储一次即可
 - ②聚簇功能可以大大提高按聚簇码进行查询的效率。例如，假设要查询学生关系中计算机系的学生名单，设计算机系有300名学生。在极端情况下，这些学生的记录会分布在300个不同的物理块中，这时查询计算机系的学生需要做300次的I/O操作，影响系统查询的性能。如果按照系别建立聚簇，使同一个系的学生记录集中存放，则每做一次I/O操作，就可以获得多个满足查询条件和记录，从而显著地减少了访问磁盘的次数。

(2) 索引

- 存储记录是属性值的集合，主关系键惟一确定一个记录，在主关系键建立惟一索引，可以提高查询速度，避免关系键重复值的录入，确保数据的完整性
- 用户在数据库中访问的最小单位是属性。对某些检索频繁的非主属性建立索引文件。
- 索引文件对存储记录重新进行内部链接，从逻辑上改变了记录的存储位置，从而改变了访问数据的入口点。关系中数据越多索引的优越性也就越明显。
- 建立多个索引文件可以缩短存取时间，但是增加了索引文件所占用的存储空间以及维护的开销。应该根据实际需要综合考虑。

2. 访问方法的设计

- 访问方法是为存储在物理设备（通常指辅存）上的数据提供存储和检索能力的方法。
- 一个访问方法包括存储结构和检索机构两个部分。
 - 存储结构限定了可能访问的路径和存储记录
 - 检索机构定义了每个应用的访问路径，但不涉及存储结构的设计和分配。
- 存储记录是属性的集合，属性是数据项类型，可用作主键或辅助键。主键唯一地确定了一个记录。辅助键是用作记录索引的属性，可能并不惟一确定某一个记录。
- 访问路径的设计分成主访问路径与辅访问路径的设计。
 - 主访问路径与初始记录的装入有关，通常是用主键来检索的
 - 辅访问路径是通过辅助键的索引对存储记录重新进行内部链接，从而改变访问数据的入口点。用辅助索引可缩短访问时间，但增加了辅存空间和索引维护的开销

3. 数据存放位置的设计

- 为了提高系统性能，应该根据应用情况将数据的易变部分、稳定部分、经常存取部分和存取频率较低部分分开存放。
- 例如：
 - 可以将表和索引分别存放在不同的磁盘上，查询时，因两个磁盘驱动器并行工作，可以提高物理读写的速度
 - 在多用户环境下，可能将日志文件和数据库对象（表、索引等）放在不同的磁盘上，以加快存取速度。
 - 数据库的数据备份、日志文件备份等，只在数据库发生故障进行恢复时才使用，而且数据量很大，可以存放在磁带上，以改进整个系统的性能。

4. 系统配置的设计

- DBMS产品一般都提供了一些系统配置变量、存储分配参数，供设计人员和DBA对数据库进行物理优化。系统为这些变量设定了初始值，但是这些值不一定适合每一种应用环境，在物理设计阶段，要根据实际情况重新对这些变量赋值，以满足新的要求。
- 系统配置变量和参数很多，例如，同时使用数据库的用户数、同时打开的数据库对象数、内存分配参数、缓冲区分配参数（使用的缓冲区长度、个数）、存储分配参数、数据库的大小、时间片的大小、锁的数目等，这些参数值影响存取时间和存储空间的分配，在物理设计时要根据应用环境确定这些参数值，以使系统的性能达到最优。

14.5.2 评价物理结构

- 在确定了数据库的物理结构之后，要进行评价，重点是时间和空间的效率
- 如果评价结果满足设计要求，则可进行数据库实施
- 实际上，往往需要经过反复测试才能优化物理设计

14.6 数据库实施

- 数据库实施是指根据逻辑设计和物理设计的结果，在计算机上建立起实际的数据库结构、装入数据、进行测试和试运行的过程。
- 数据库实施主要工作
 - 建立实际数据库结构
 - 装入数据
 - 应用程序编码与调试
 - 数据库试运行（功能测试、性能测试）
 - 整理文档

14.7 数据库运行和维护

- 数据库系统投入正式运行，进入运行和维护阶段，标志着数据库应用开发工作的基本结束，但并不意味着设计过程已经结束。
- 由于应用环境不断发生变化，用户的需求和处理方法不断发展，数据库在运行过程中的存储结构也会不断变化，从而必须修改和扩充相应的应用程序。
- 数据库运行和维护阶段的主要任务：
 - (1) 维护数据库的安全性与完整性
 - (2) 监测并改善数据库性能
 - (3) 重新组织和构造数据库