第七章 数据库系统设计

第七章 数据库系统设计

7.1 数据库系统设计概述

数据库系统设计指的是对于一个给定的**应用环境**，构造一个**最优的**数据库模式，并据此建立一个既能**反映现实**世界信息和信息联系、满足用户对**数据要求和加工要求**，又能被某个**DBMS所接受**的数据**库**及其**应用**系统。

数据库系统生存期

数据库应用系统从开始**规划**、**分析**、**设计**、**实现**、**投入**运行后的**维护**到最后**被新的系统取代**而停止使用的整个期间称为**数据库系统的生存期**。

数据库系统生存期的七个阶段

(1)规划阶段

(2)需求分析阶段

(3)概念设计阶段

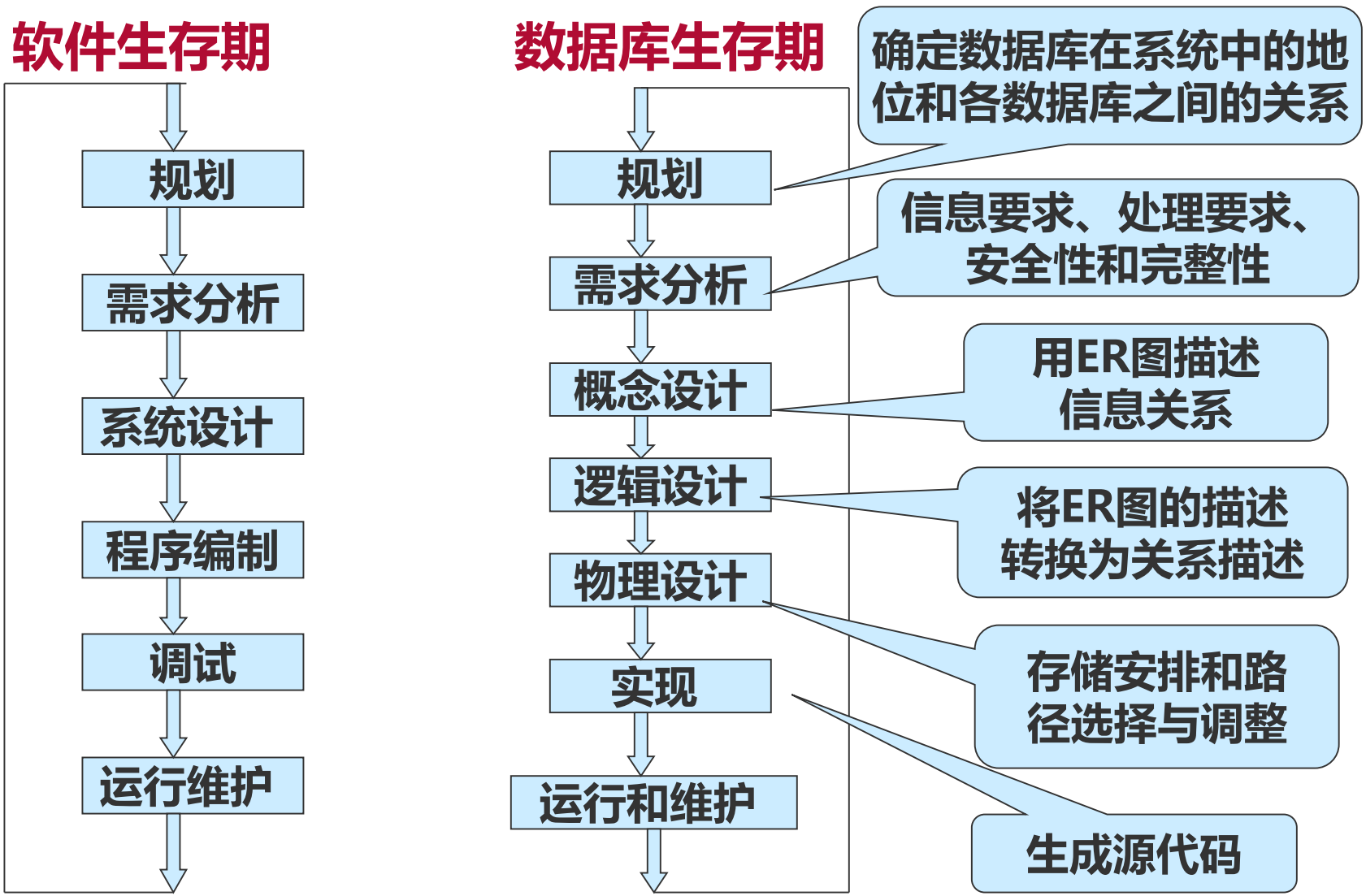
(4)逻辑设计阶段

(5)物理设计阶段

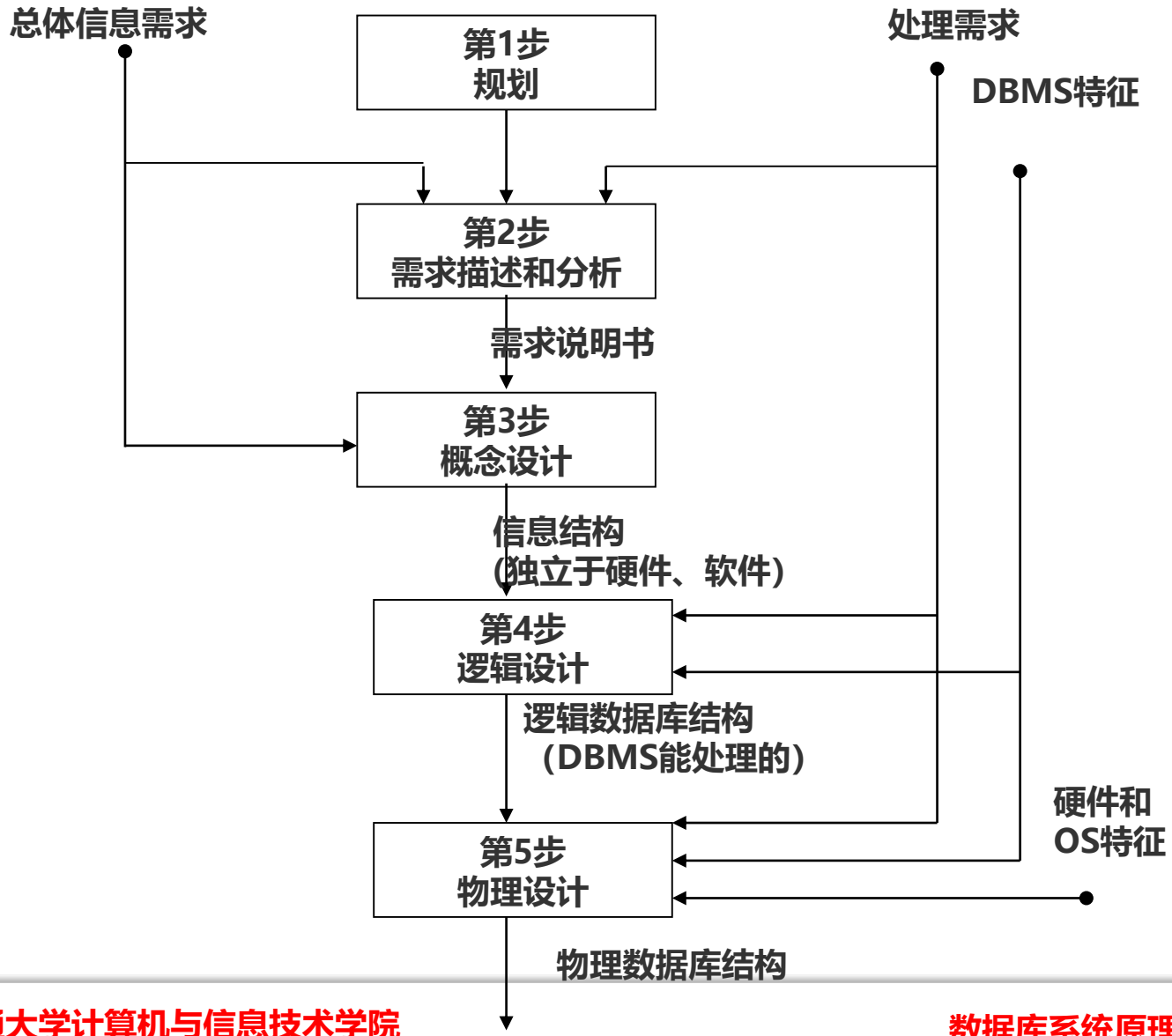
(6)数据库实施阶段

(7)运行维护阶段

数据库系统生存期的七个阶段



数据库系统生存期的七个阶段



数据库设计的步骤

数据库系统生存期的七个阶段

(1) 规划阶段

进行建立数据库的**必要性及可行性分析**，确定数据库系统在组织中和信息系统中的地位，以及各个数据库之间的联系。

输出：

可行性分析报告

数据库系统生存期的七个阶段

- 步骤：
  - 系统调查：

    对企业组织作全面的调查，画出组织层次图，以了解企业的组织结构
  - 可行性分析

    从技术、经济、效益、法律等方面对建立数据库的可行性进行分析；写出可行性分析报告；组织专家进行讨论其可行性
  - 确定数据库系统的总目标和制定项目开发计划

数据库系统生存期的七个阶段

(2)需求分析阶段

通过调查研究，了解用户的要求，其中包括:

(a)信息要求

(b)处理要求

(c)安全性和完整性要求

输出：

数据流图和数据字典

数据库系统生存期的七个阶段

(3)概念设计阶段

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，整个数据库设计的关键

输出：

数据库系统生存期的七个阶段

(4)逻辑设计阶段

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

输出：

关系模式

数据库系统生存期的七个阶段

(5)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：

(a)确定DB的存储安排；

(b)存取路径的选择和调整；

(c)确定系统的配置，如硬件和OS、DBMS等；

输出：物理设计说明书

数据库系统生存期的七个阶段

(6) 数据库实施阶段

运用DBMS提供的数据库语言（如SQL）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
- 进行试运行

输出：

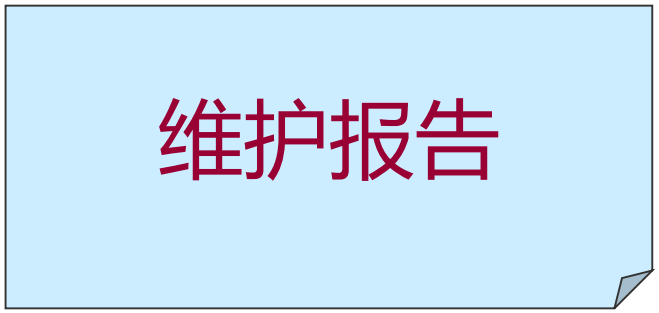
程序代码

数据库系统生存期的七个阶段

(7) 运行维护阶段

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

输出：



数据库设计方法

- 新奥尔良（New Orleans）方法
  - 将数据库设计分为若干阶段和步骤
- 基于E-R模型的数据库设计方法
  - 概念设计阶段广泛采用
- 3NF（第三范式）的设计方法
  - 逻辑阶段可采用的有效方法
- ODL（Object Definition Language）方法
  - 面向对象的数据库设计方法

参与设计的人员

(1) 系统分析人员和数据库设计人员：是数据库设计的核心人员，自始至终参与数据库设计，他们的水平决定数据库质量。

(2)用户和数据库管理员：主要参与需求分析和数据库运行维护。

(3) 程序员：参与系统实施，负责编制程序和准备软硬件环境。

7.2 需求分析

需求分析的任务：

- 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）
- 明确用户的各种需求
- 确定**新系统的功能**
- 充分考虑今后可能的扩充和改变

7.2 需求分析

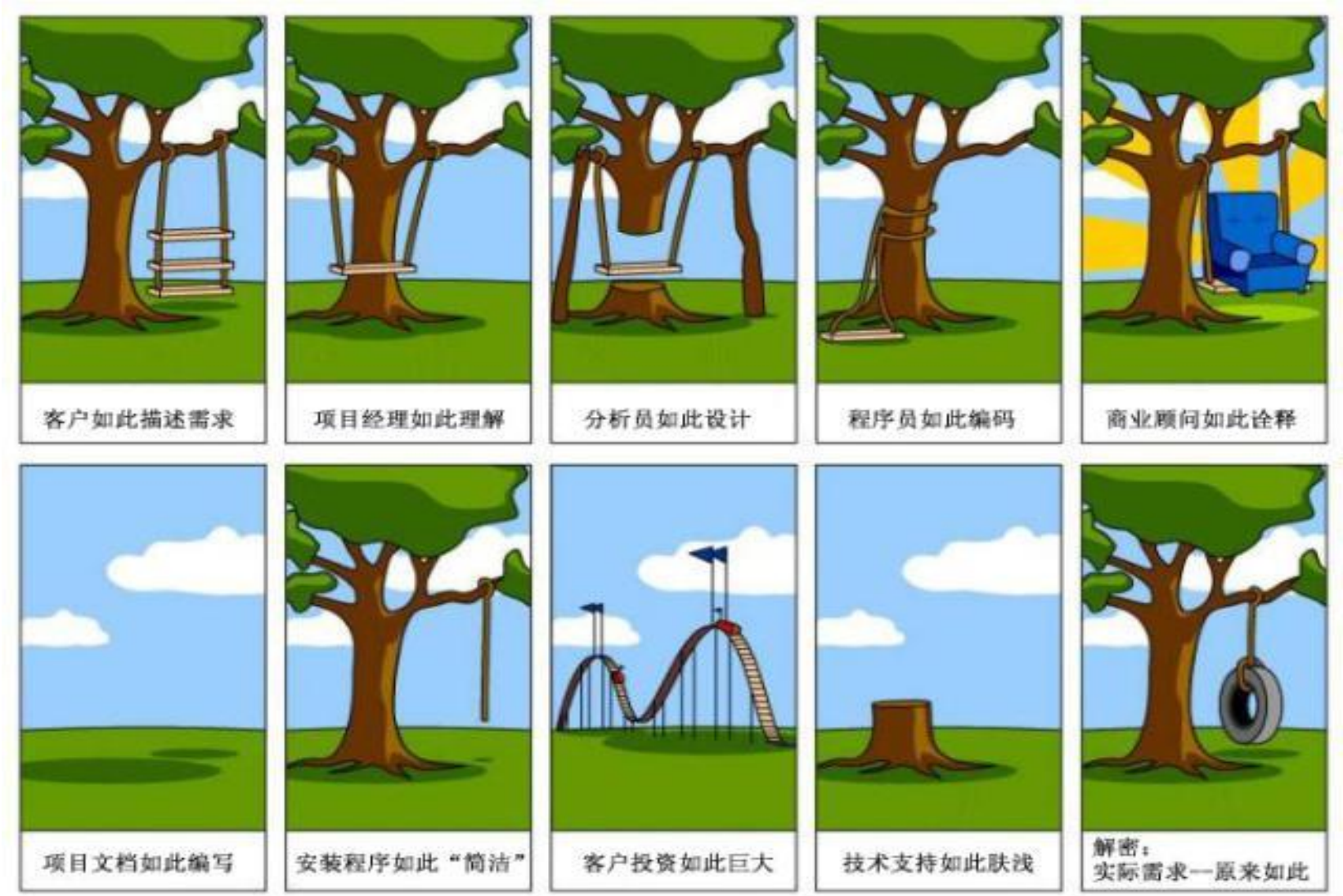
需求分析的目标

- 分析用户活动，产生**业务流程图**
- 确定系统范围，产生**系统范围图**
- 分析用户活动涉及的数据，产生**数据流图**
- 分析系统数据，产生**数据字典**

需求分析的方法

- 调查需求
- 达成共识
- 分析表达需求

需求分析的方法


客户如此描述需求
项目经理如此理解
分析员如此设计
程序员如此编码
商业顾问如此诠释
项目文档如此编写
安装程序如此“简洁”
客户投资如此巨大
技术支持如此肤浅
解密：
实际需求一原来如此

调查用户需求的具体步骤

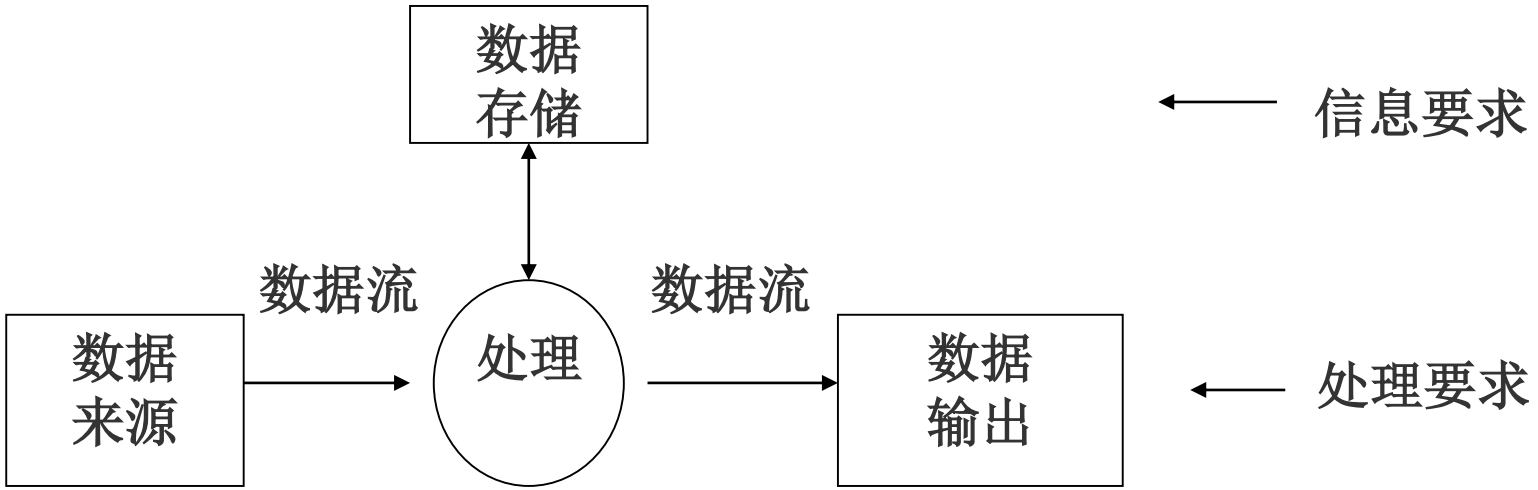
- 调查组织机构情况
- 调查各部门的业务活动情况。
- 在熟悉业务活动的基础上，协助用户明确对新系统的各种要求。
- 确定新系统的边界

进一步分析和表达用户需求

- **结构化分析方法**（Structured Analysis，简称SA方法）
  - 从最上层的系统组织机构入手
  - 自顶向下、逐层分解分析系统

进一步分析和表达用户需求

1．首先把任何一个系统都抽象为：



进一步分析和表达用户需求

2．分解处理功能和数据

(1)分解处理功能

- 将处理功能的具体内容分解为若干子功能

(2)分解数据

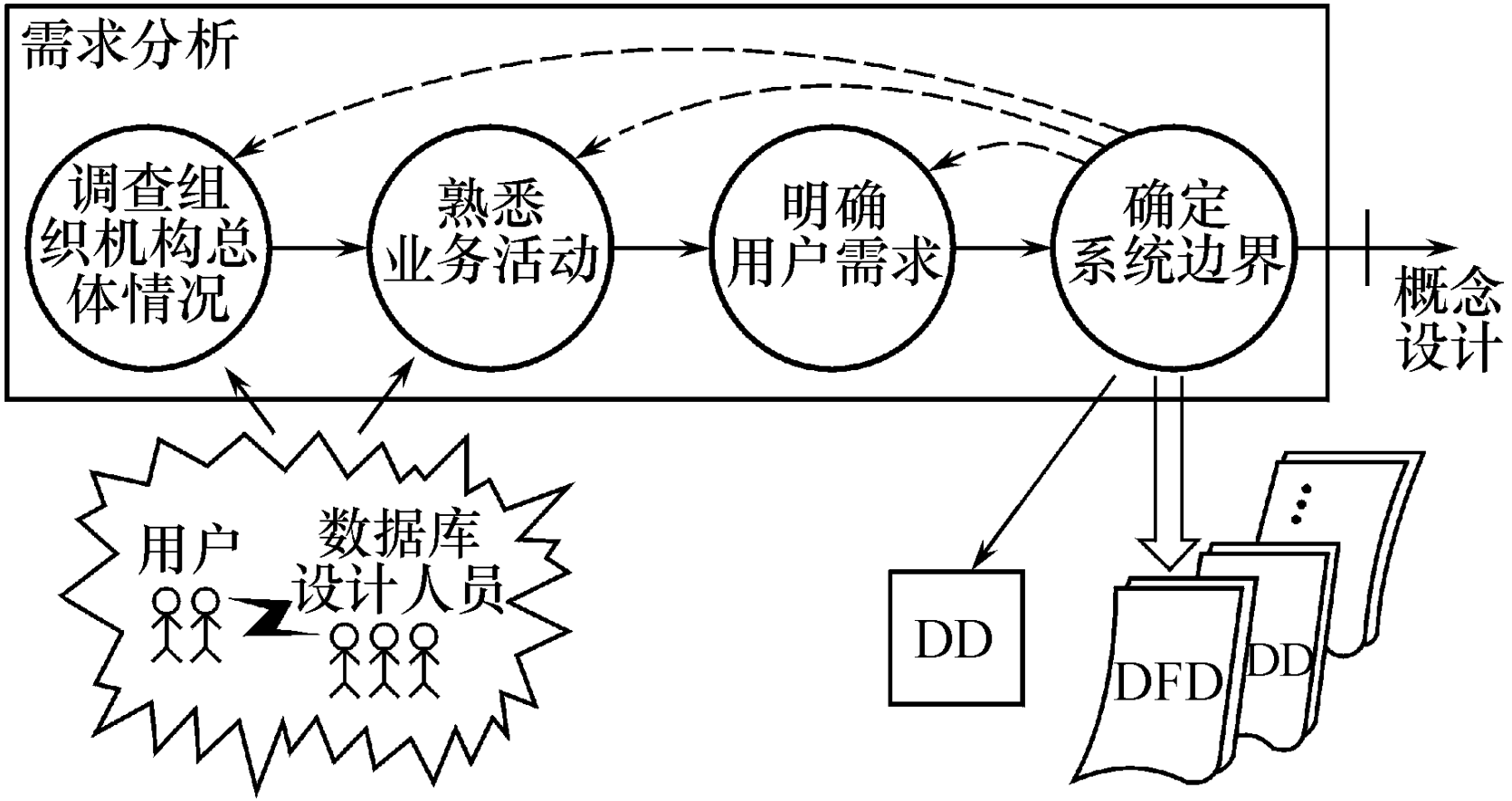
- 处理功能逐步分解同时，逐级分解所用数据，形成若干层次的数据流图

(3)表达方法

- 处理逻辑：用判定表或判定树来描述
- 数据：用数据字典来描述

3．将分析结果再次提交给用户，征得用户的认可

需求分析过程


需求分析
调查组织机构总体情况
熟悉业务活动
明确用户需求
确定系统边界
概念设计
用户
数据库设计人员
DD
DFD
DD

数据字典

- **数据字典**是系统中各类数据描述的集合，是进行详细的数据收集和数据分析所获得的主要成果。数据字典中的内容在数据库设计过程中还要不断修改、充实和完善。
- 一般来说数据字典中应包括对以下几部分数据的描述：
  - 数据项、数据结构、数据流、数据存储、处理过程

数据字典

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，数据项之间的联系 }

数据字典

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}

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，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}

5 处理过程

- 具体处理逻辑一般用判定表或判定树来描述
- 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} }

数据字典举例

例：学生学籍管理子系统的数据字典。

数据项，以“学号”为例：

数据项：　学号

含义说明：唯一标识每个学生

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：前两位标别该学生所在年级，后六位按顺序编号

数据结构举例

数据结构，以“学生”为例

“学生”是该系统中的一个核心数据结构：

数据结构：　学生

含义说明：　是学籍管理子系统的主体数据结构，定义了一个学生的有关信息

组成：学号，姓名，性别，年龄，所在系，年级

数据流

数据流，“体检结果”可如下描述：

数据流：　　体检结果

说明：学生参加体格检查的最终结果

数据流来源：体检

数据流去向：批准

组成：　　……

平均流量：　……

(单位时间里的传输次数)

高峰期流量：……

(高峰时期的数据流量)

数据字典

数据存储，“学生登记表”可如下描述：

数据存储：　学生登记表

说明：　记录学生的基本情况

流入数据流：……

流出数据流：……

组成：　……

数据量：　每年3000张

存取方式：　随机存取

数据字典

处理过程“分配宿舍”可如下描述：

处理过程：分配宿舍

说明：　　为所有新生分配学生宿舍

输入：　　学生，宿舍

输出：　　宿舍安排

处理：　　在新生报到后，为所有新生分配学生宿舍。要求同一间宿舍只能安排同一性别的学生，同一个学生只能安排在一个宿舍中。每个学生的居住面积不小于3平方米。安排新生宿舍其处理时间应不超过15分钟。

数据字典

- 数据字典是关于数据库中数据的描述，是元数据，而不是数据本身
- 数据字典在需求分析阶段建立，在数据库设计过程中不断修改、充实、完善

需求分析小结

- 设计人员应充分考虑到可能的扩充和改变，使设计易于更改，系统易于扩充
- 必须强调用户的参与

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

概念结构设计

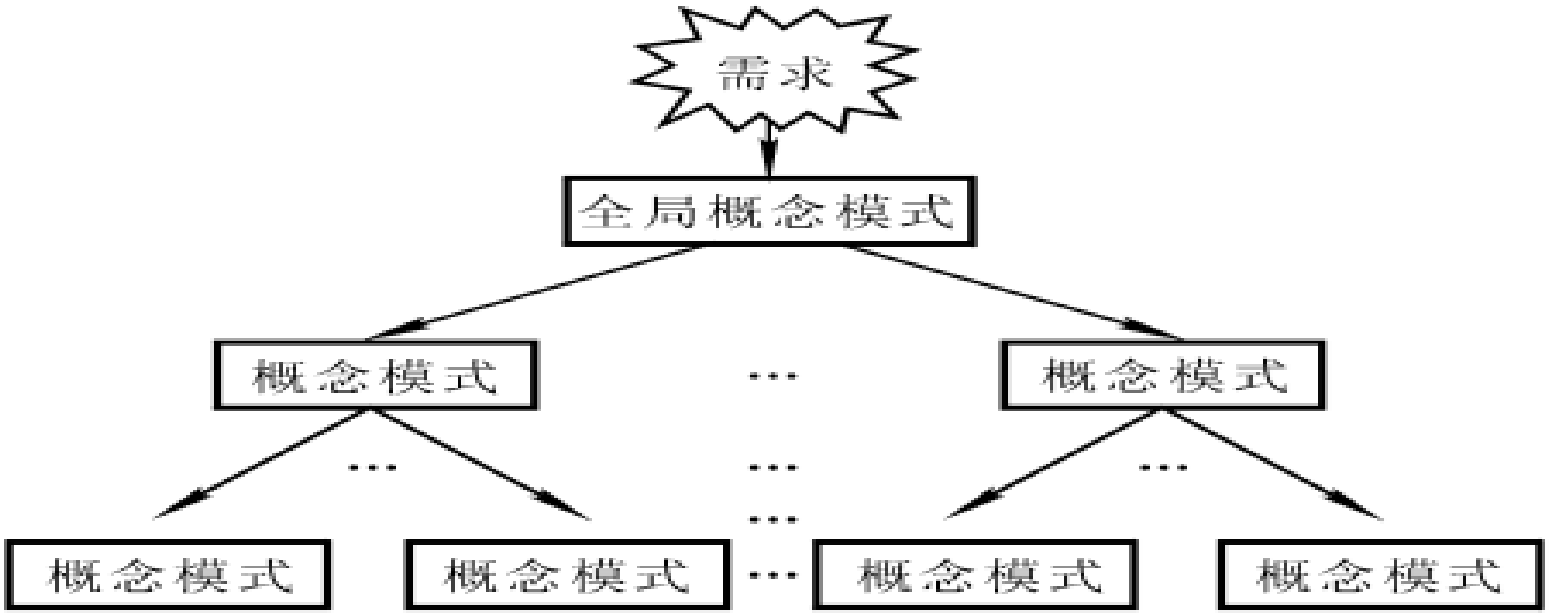
根据需求分析阶段形成的新系统需求分析说明书，把用户的信息需求抽象为信息结构即概念模型的过程就是概念结构设计。用E-R图来描述现实世界的概念模型。

概念结构设计的方法与步骤

设计概念结构通常有自顶向下、自底向上、逐步扩张和混合策略等4类方法。

1.自顶向下

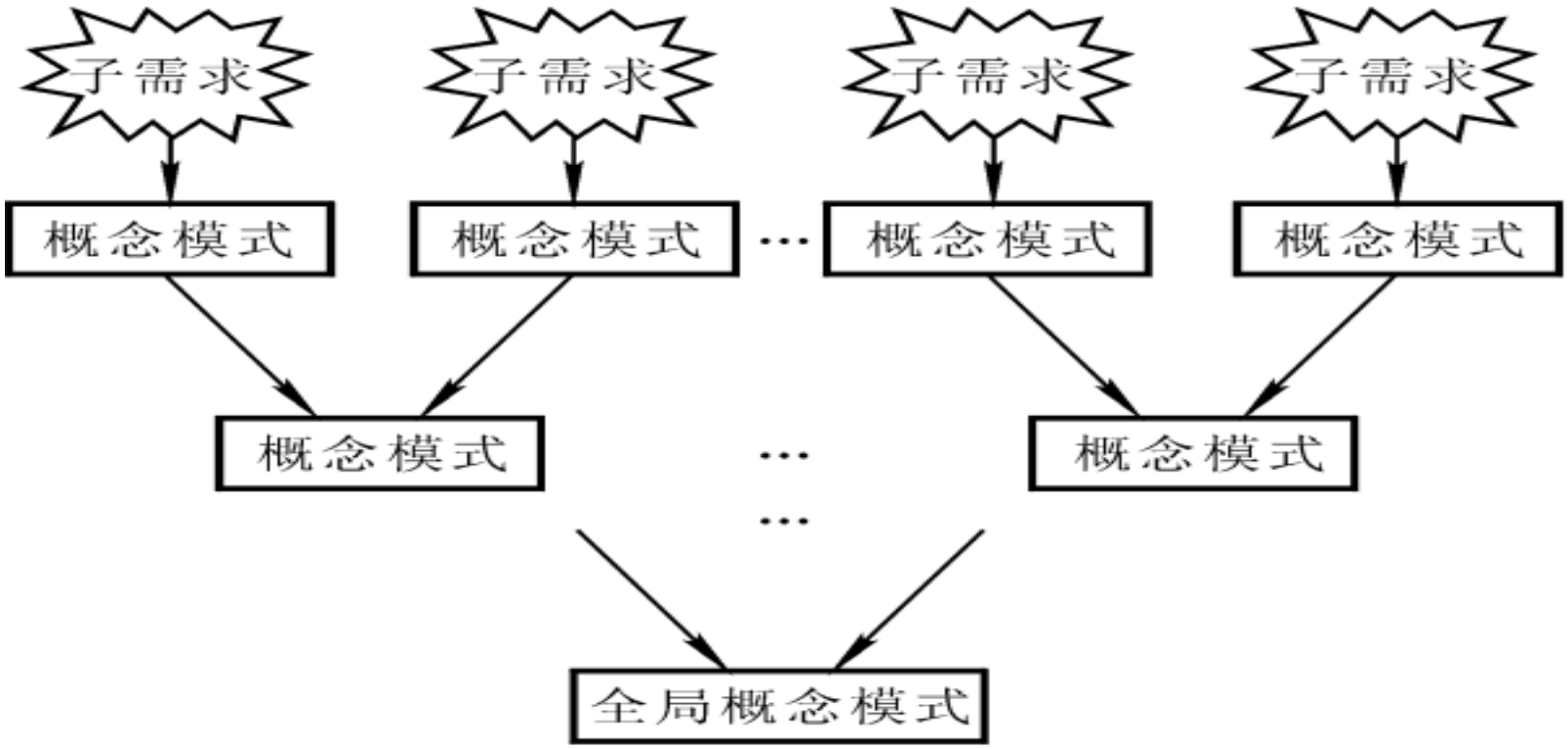
- 首先定义全局概念结构的框架，然后逐步细化



概念结构设计的方法与步骤

2.自底向上

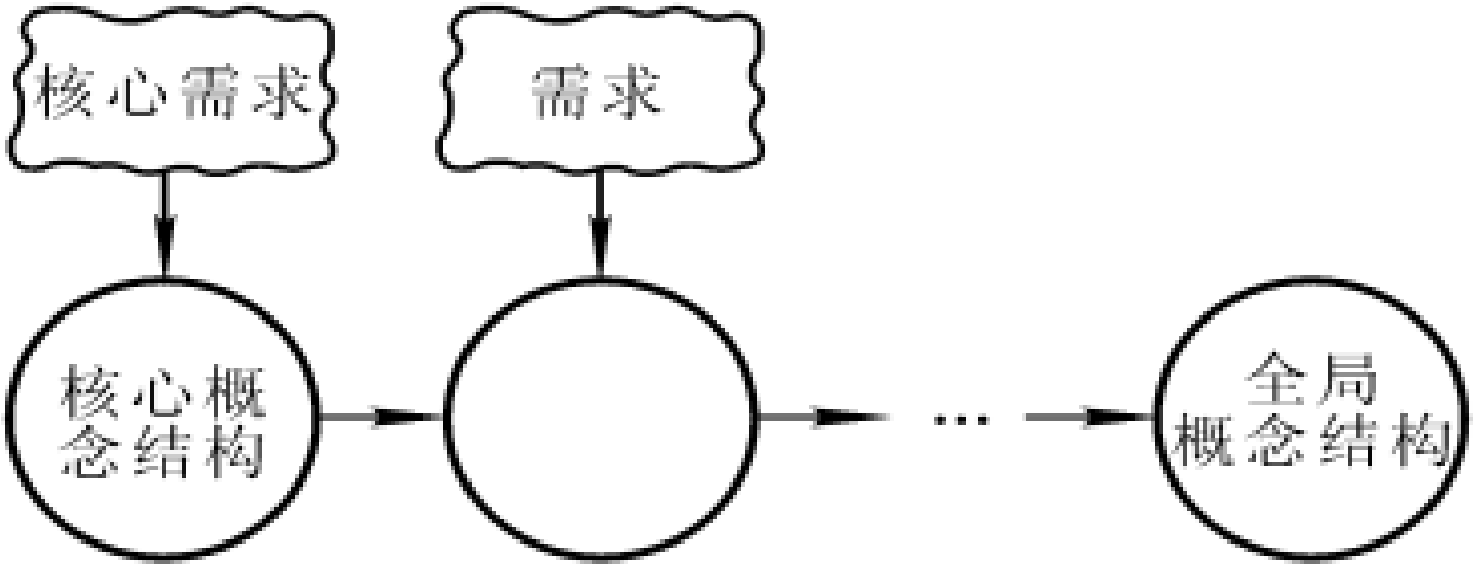
- 首先定义各局部应用的概念结构，然后将它们集成起来，得到全局概念结构



概念结构设计的方法与步骤

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



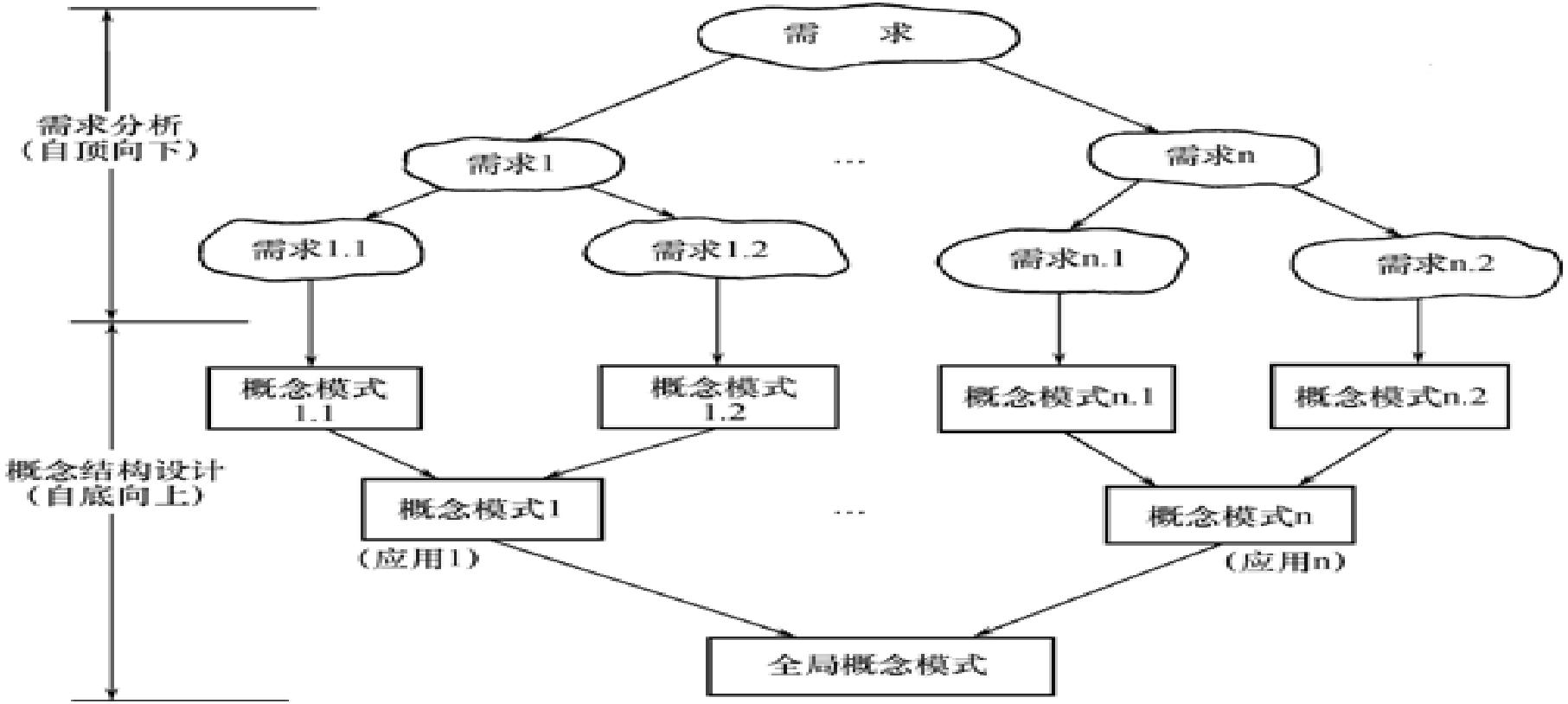
概念结构设计的方法与步骤

4.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

- 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

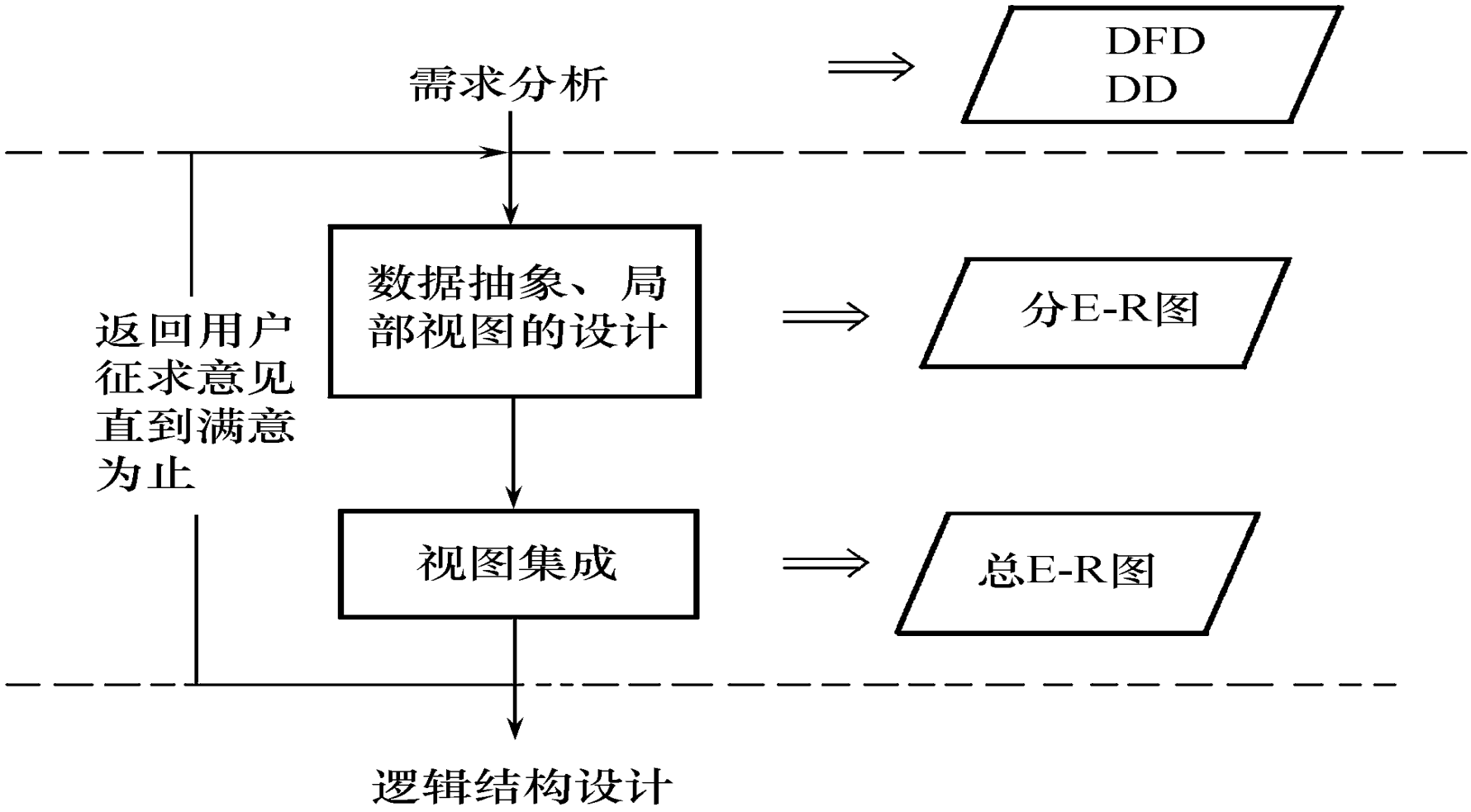


概念结构的设计步骤

第一步：抽象数据并设计**局部视图**

第二步：集成局部视图，得到**全局的概念结构**

概念结构的设计步骤


需求分析
DFD
DD
数据抽象、局部视图的设计
分E-R图
返回用户征求意见直到满意为止
视图集成
总E-R图
逻辑结构设计

常用的概念模型 — 实体联系模型

Entity relationship model,简称ER模型,是由美籍华人陈平山于1976 年提出的。ER图提供了表示实体,属性,联系的方法

(1) ER模型中的三要素

A. 实体 (Entity)：表示客观事物。
B. 属性 (Attributes)：
表示客观事物的特征（属性）
C. 联系 (Relations)：
客观事物之间的联系

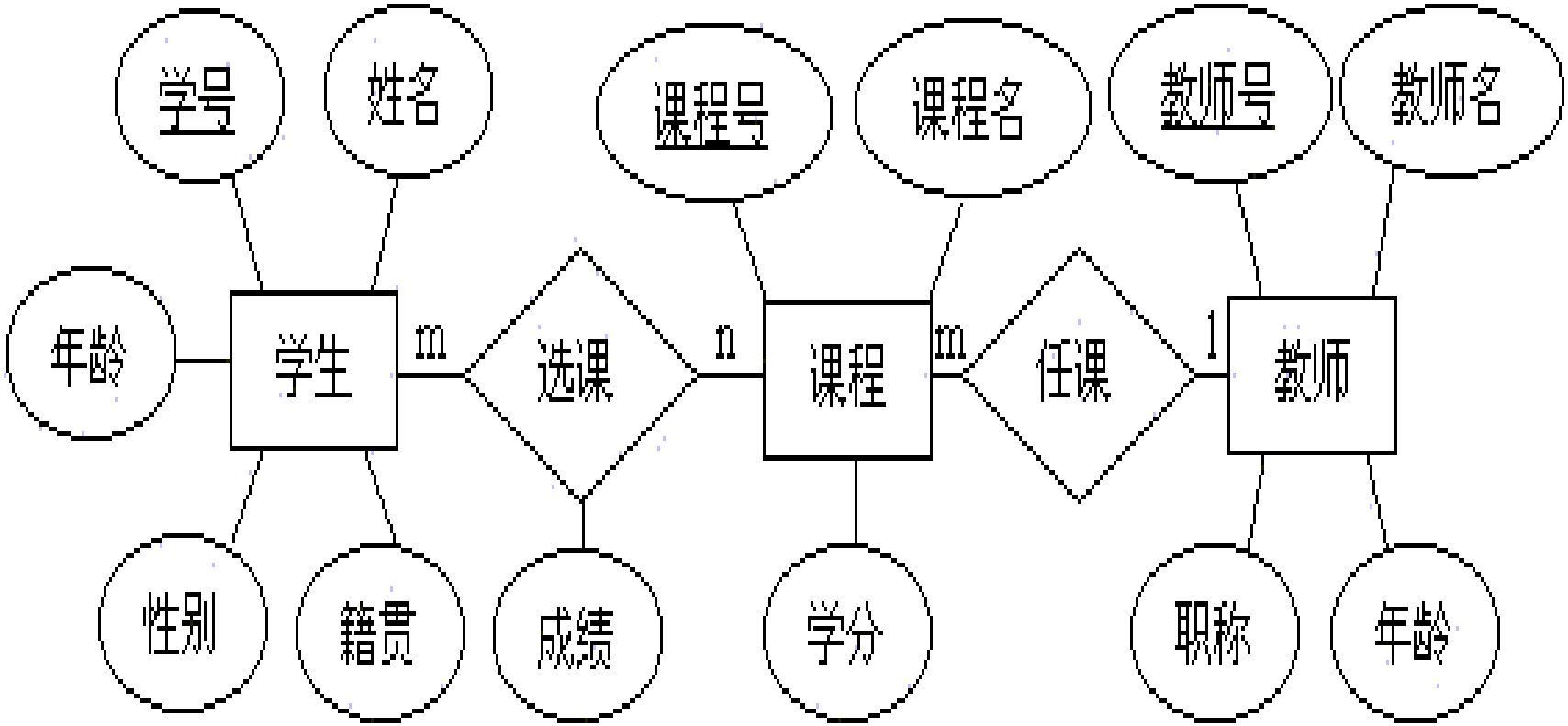
(2) 刻划工具 -实体联系图(ER图)

ER图表示方法：

用矩形表示实体、用椭圆表示属性、用菱形表示实体间的联系，属性和实体间、实体和联系间用线段连接(同时在线段边上标上联系的类型)

例如：假设一个学生可选多门课程，而一门课程又有多个学生选修，一个教师可讲多门课程，一门课程至多只有一个教师讲授。

刻划工具 -实体联系图(ER图)



如何建立实体-联系模型

(a) 确定实体以及实体的属性
(b) 确定实体之间的联系以及联系的属性
(c) 将实体的属性都加上就完成了完整的ER图

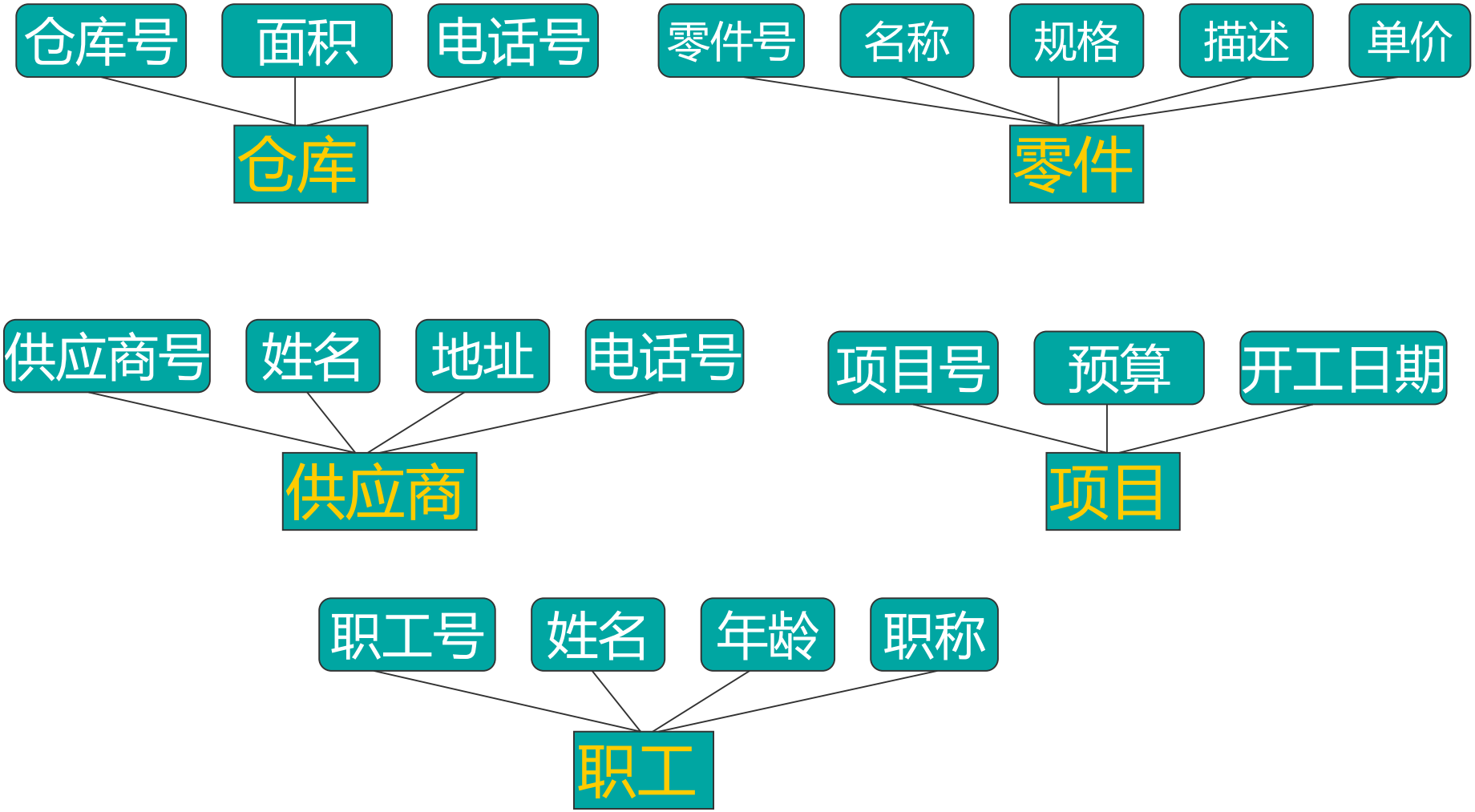
示例

用ER图表示某个工厂物资管理的概念模型

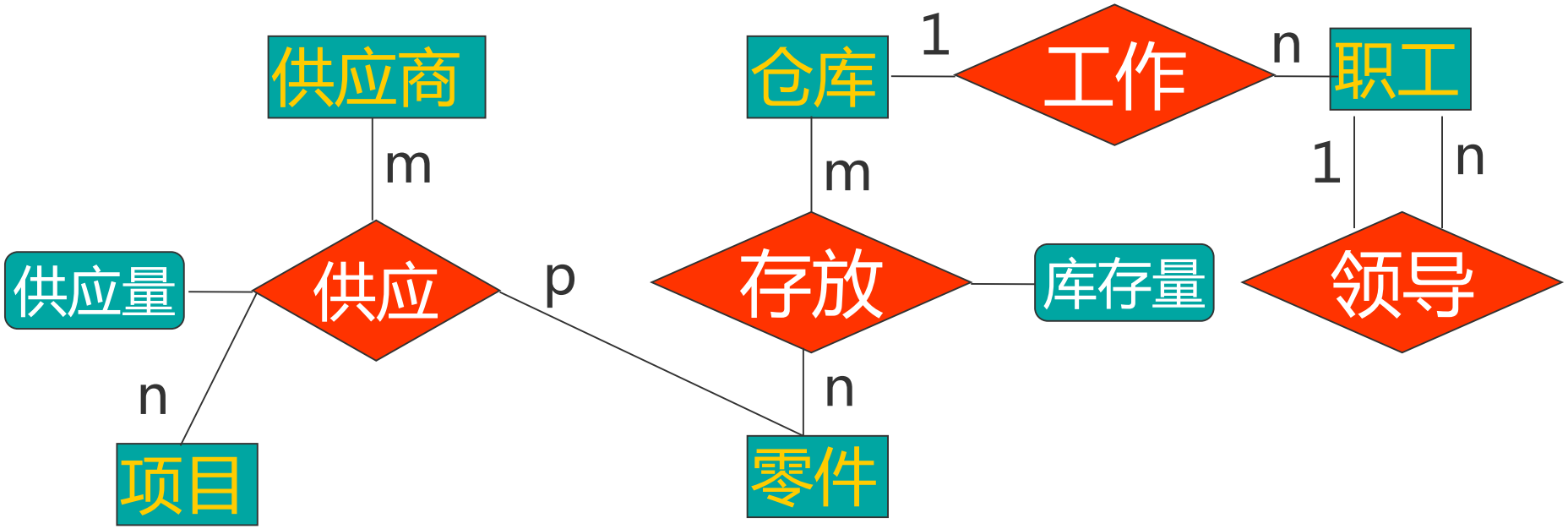
1. 仓库（仓库号，面积，电话号码）
2. 零件（零件号，名称，规格，单价，描述）
3. 供应商（供应商号，姓名，地址，电话号码）
4. 项目（项目号，预算，开工日期）
5. 职工（职工号，姓名，年龄，职称）

- 一个仓库可以存放多种零件，一种零件可以存放在多个仓库
- 一个仓库有多个职工当仓库保管员，一个职工只能在一个仓库
- 职工之间具有领导被领导关系，即仓库主任领导若干保管员
- 供应商，项目，零件之间具有多对多关系

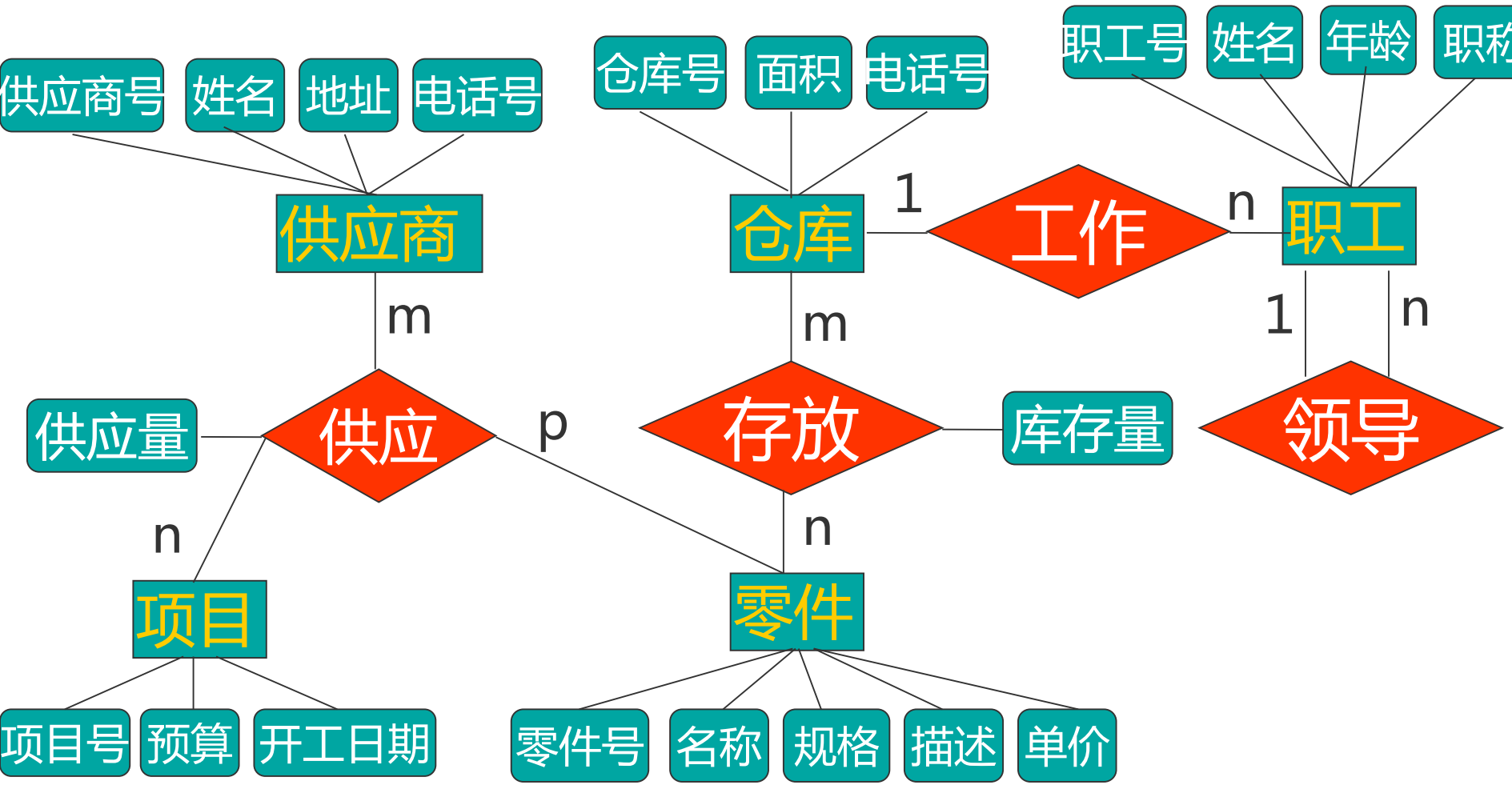
建立过程(a) - 确定实体及属性



建立过程(b) - 确定联系及属性



建立过程(c) - 综合


供应商号
姓名
地址
电话号
供应商
m
供应量
供应
p
n
项目
项目号
预算
开工日期
仓库号
面积
电话号
仓库
1
工作
n
职工
职工号
姓名
年龄
职称
m
存放
库存量
n
零件
零件号
名称
规格
描述
单价
1
n
领导

局部视图设计

概念结构设计的第一步

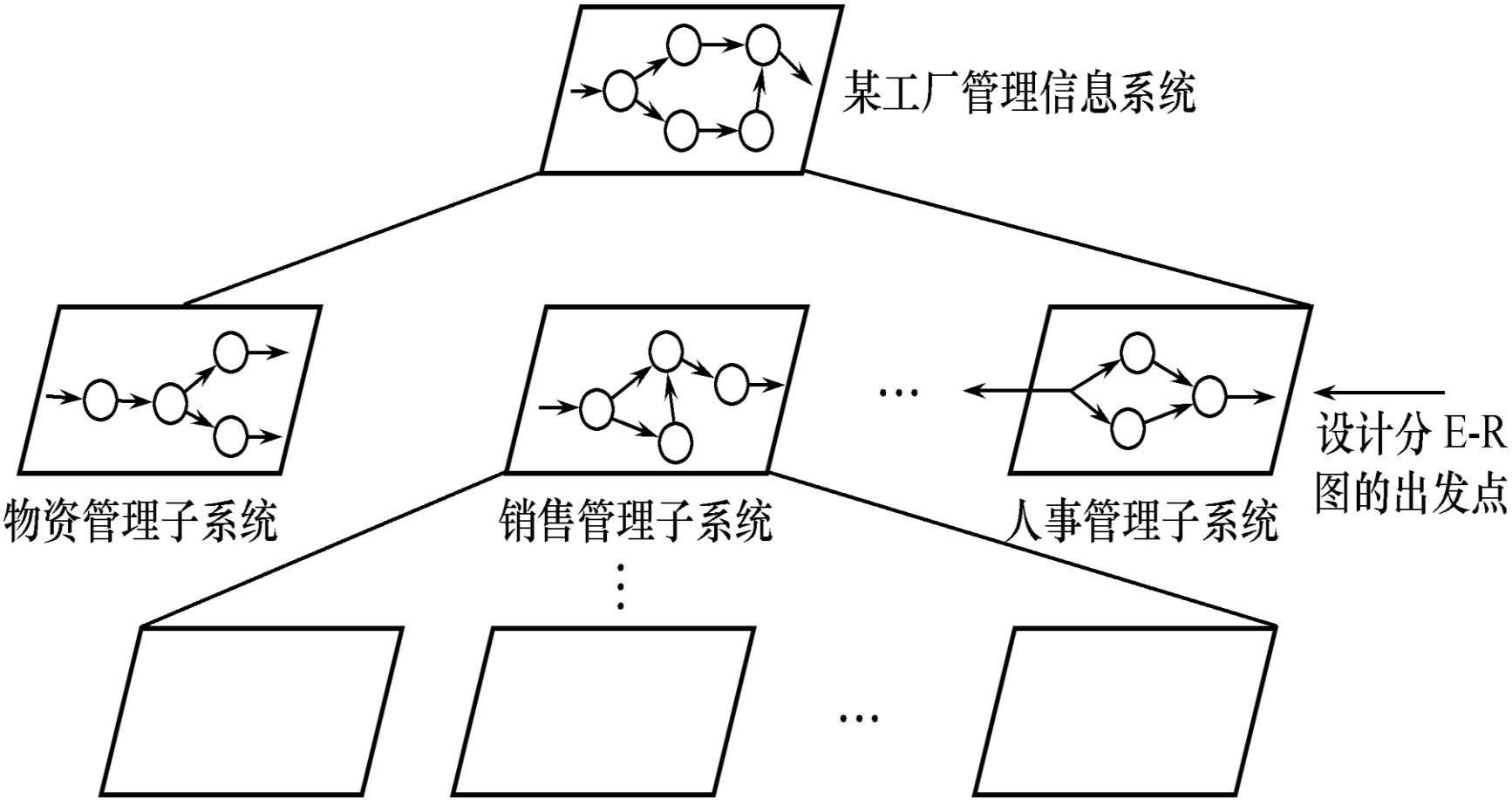
就是对需求分析阶段收集到的数据按照ER模型的要求进行**分类、组织，形成实体、实体的属性，标识实体的码**，确定实体之间的联系类型（1∶1，1∶*n*，*m*∶*n*），设计分E-R图。 具体做法是：

1．选择局部应用

根据应用系统的具体情况，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。

往往以**中层数据流图**作为设计分E-R图的依据

分层数据流图示意



局部视图设计

2．逐一设计分E-R图

选择好局部应用之后，就要对每个局部应用逐一设计分E-R图。

在现实世界中具体的应用环境常常对实体和属性已经作了大体的自然的划分。在系统分析阶段得到的“数据存储”、数据字典中的“数据结构”和“数据流”都是若干属性有意义的聚合，就体现了这种划分。

可以先从这些内容出发定义E-R图。

实体属性设计准则

两条准则：

（1）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

（2）属性不能与其他实体具有联系。**联系只发生在实体之间**

[实例] 销售管理子系统分E-R图的设计

销售管理子系统的主要功能：

- 处理顾客和销售员送来的订单
- 工厂是根据订货安排生产的
- 交出货物同时开出发票
- 收到顾客付款后，根据发票存根和信贷情况进行应收款处理

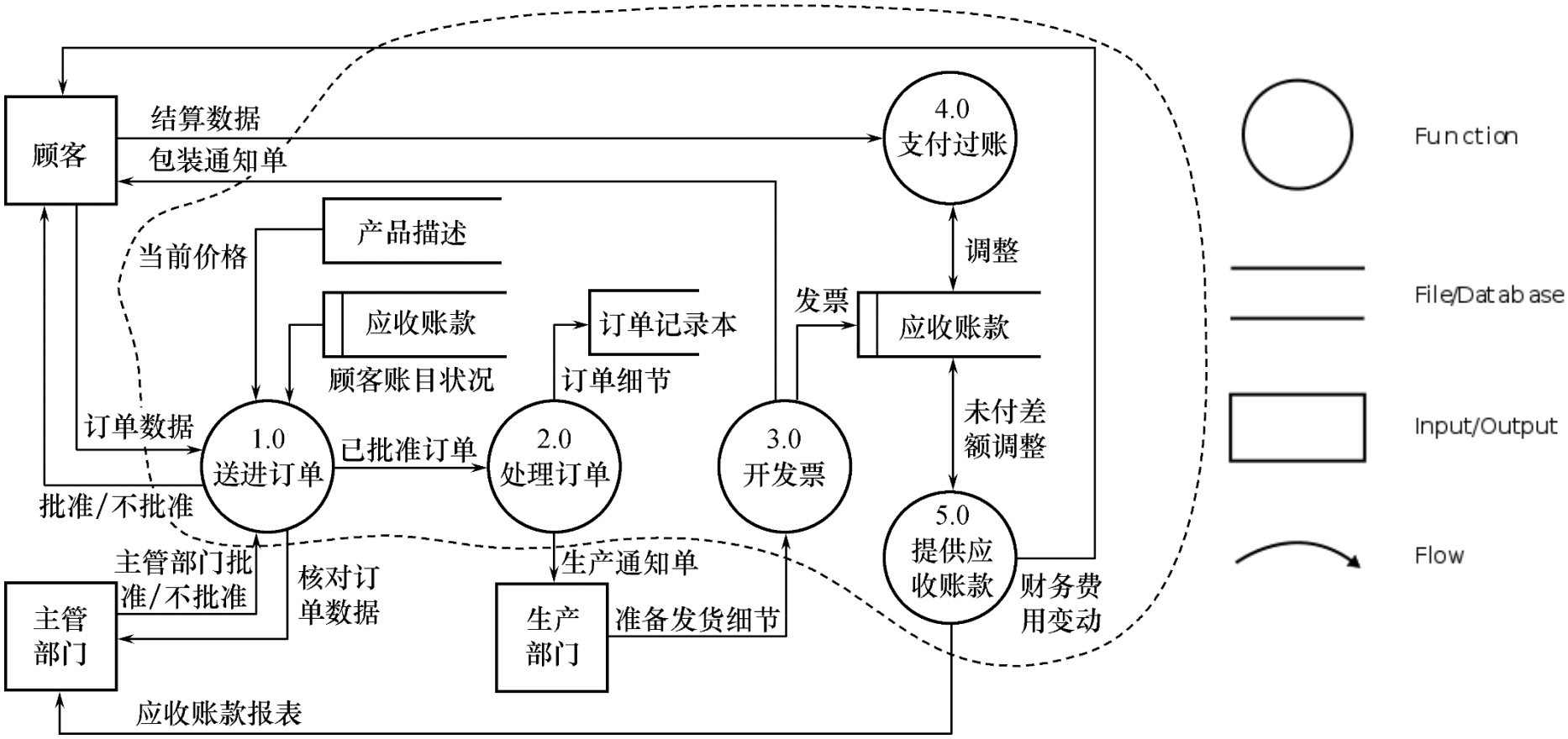
Data flow diagram

- A **data flow diagram** (**DFD**) is a **graphical representation** of the "flow" of data through **an information system**, modelling its ***process* aspects**. A DFD is often used as a **preliminary step** to create an **overview of the system** without going into great detail, which can later be elaborated.

- A DFD shows what kind of information will be **input to** and **output from** the system, how the data will **advance through** the system, and where the data will be stored. It **does not show** information about process **timing** or whether processes will operate in **sequence** or in **parallel**, unlike a traditional **structured flowchart** which focuses on **control flow**, or a UML **activity workflow diagram**, which presents both control and data flows as a unified model.

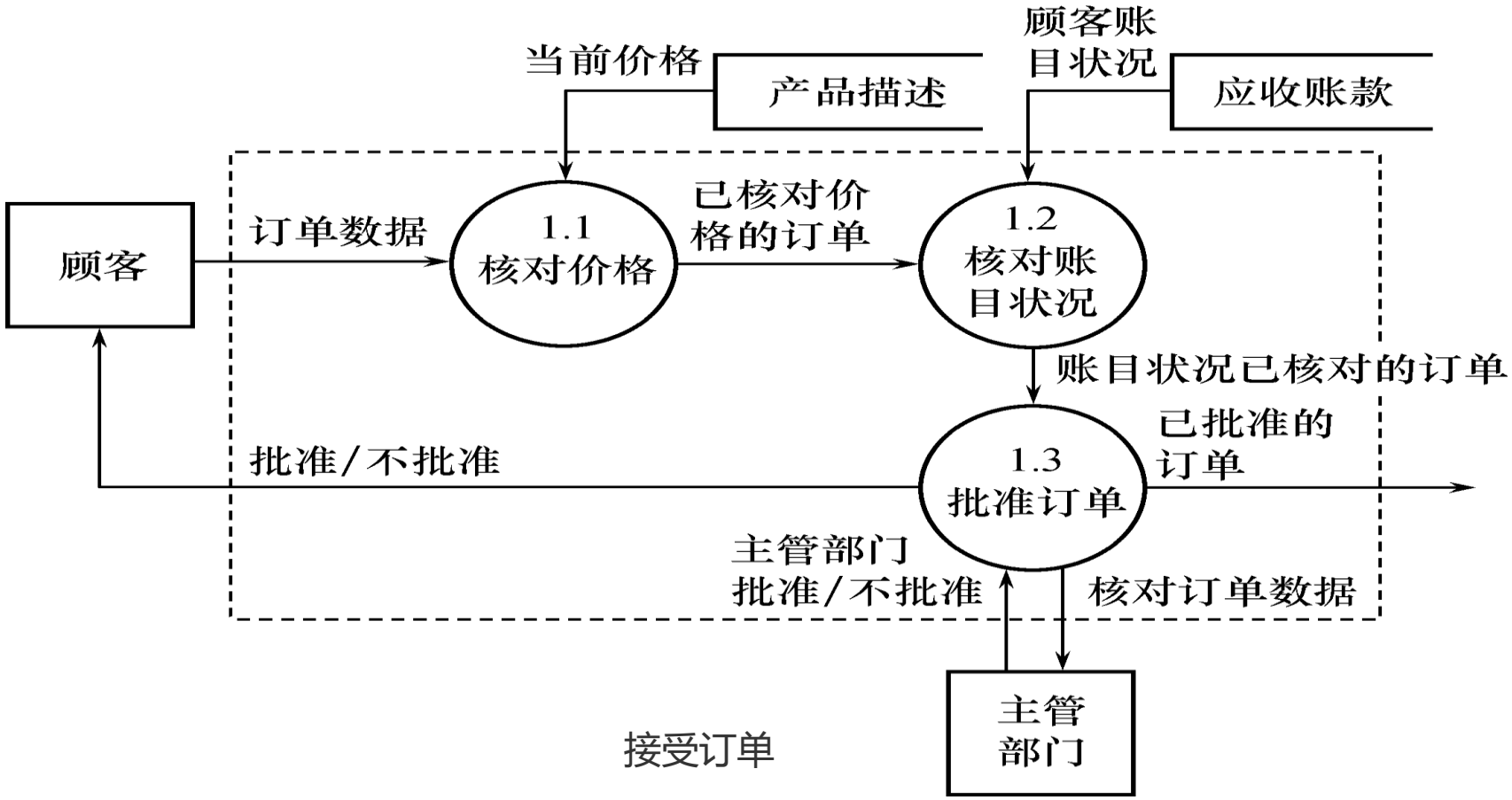
第一层数据流图

下图是第一层数据流图，虚线部分划出了系统边界

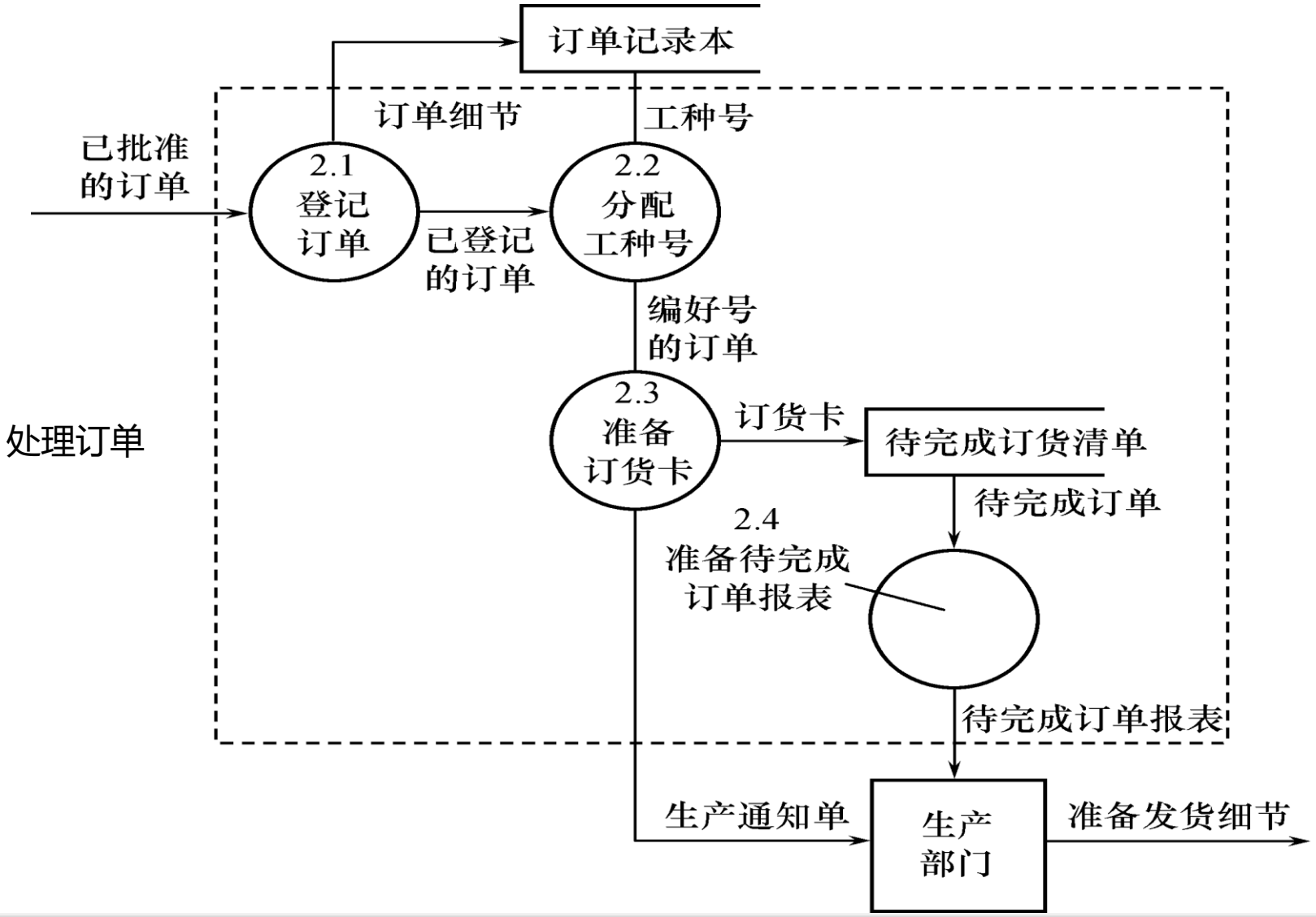


第二层数据流图—订单生成与批准

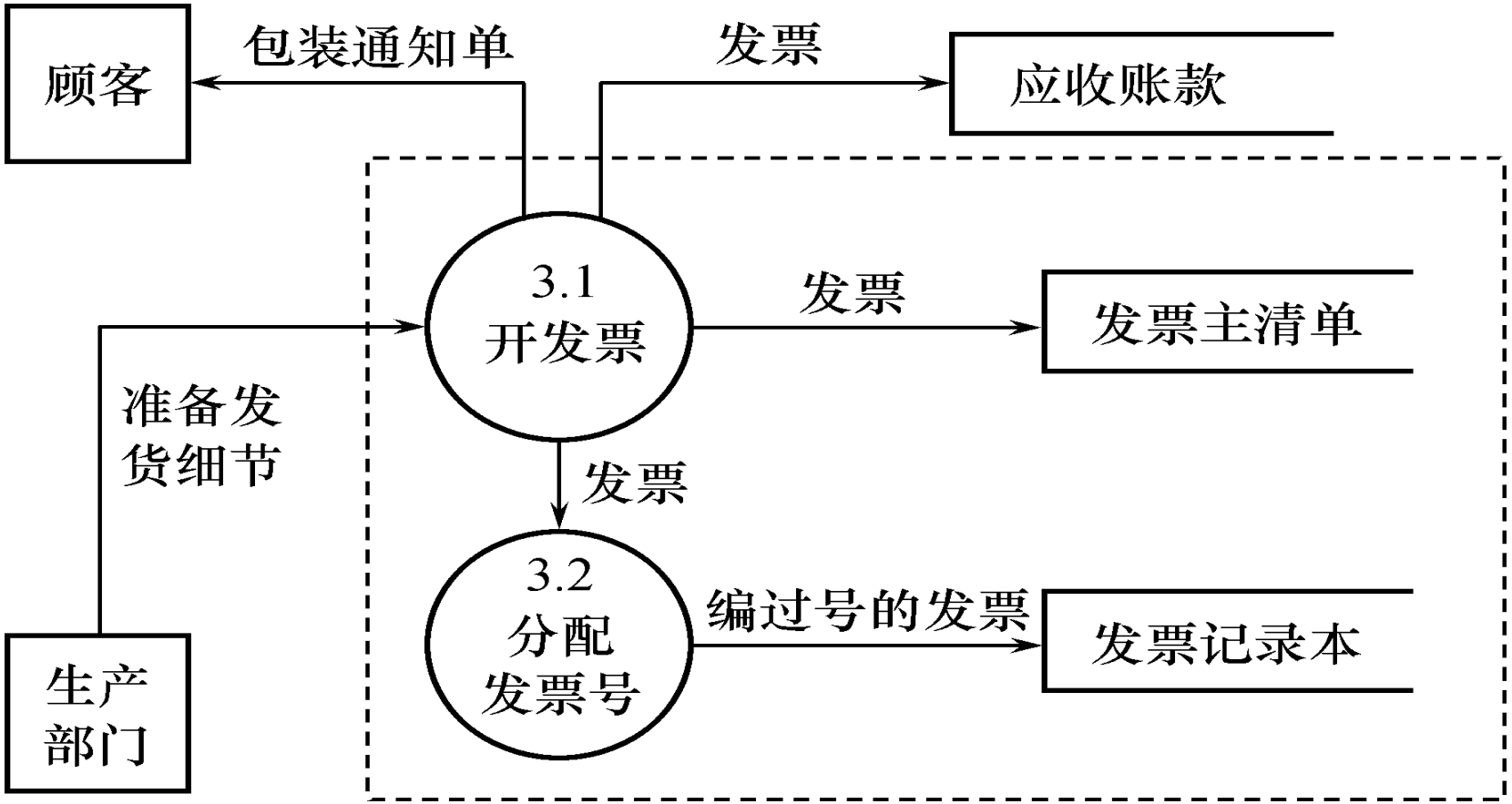
上图中把系统功能又分为4个子系统，下面四个图是第二层数据流图



第二层数据流图—处理订单

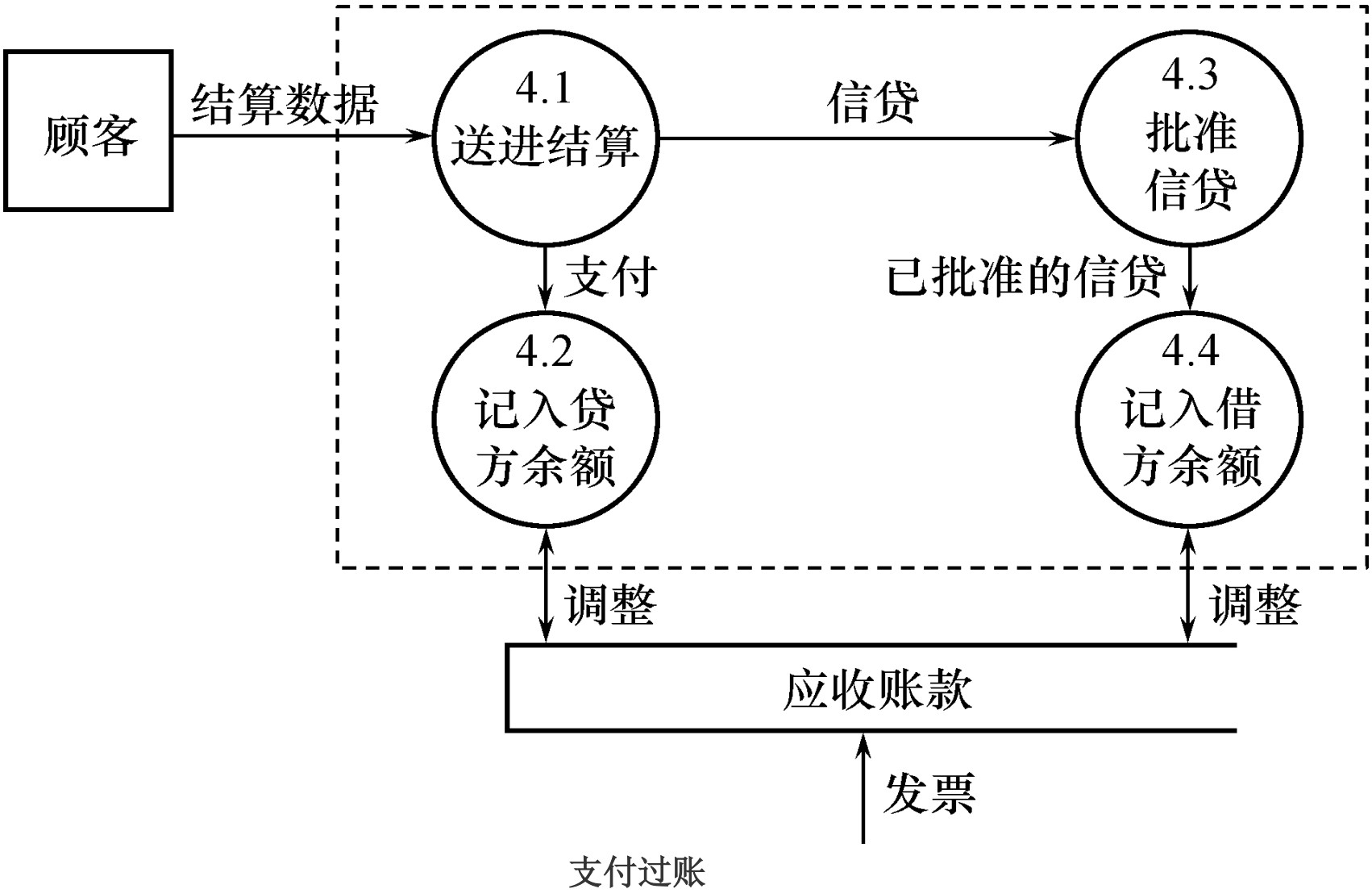


第二层数据流图—发票处理



开发票

第二层数据流图—财务手续

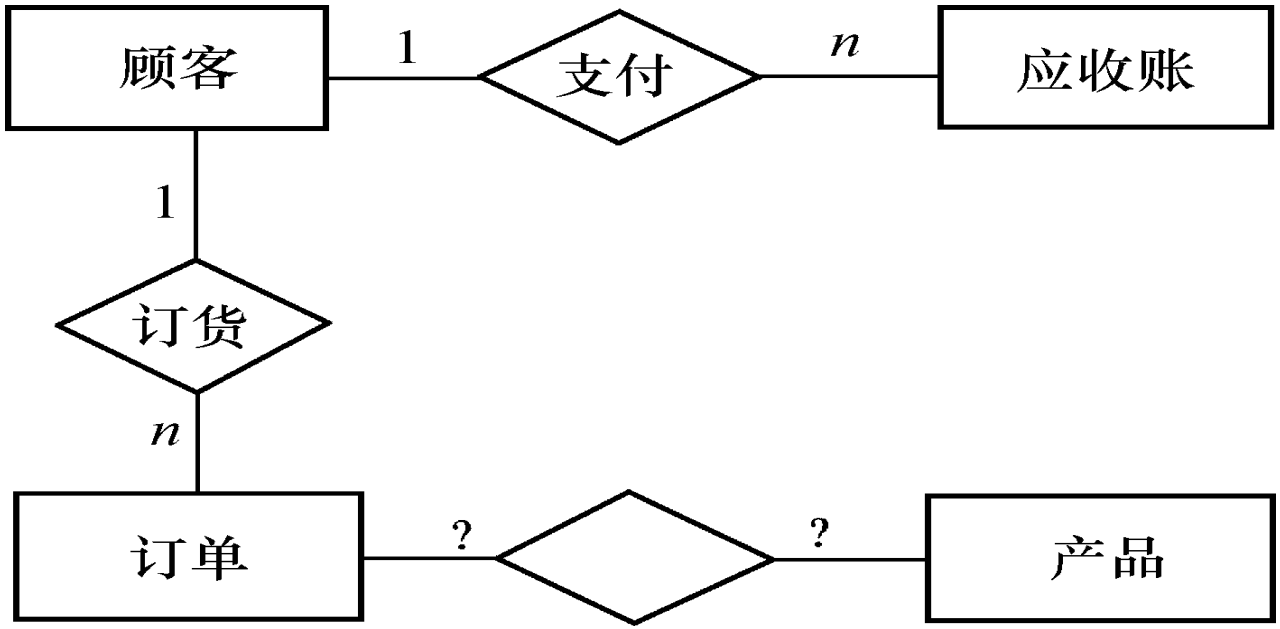


设计总结

分析数据流图和数据字典，知道整个系统功能围绕了“订单”和“应收账款”的处理。

数据结构中订单、顾客、顾客应收账目用得最多，它们是许多子功能和数据流共享的数据，因此先设计该分E-R图的草图

局部视图设计



设计细化

参照第二层数据流图和数据字典，遵循两个准则，进行如下调整：

（1）每张订单由订单号、若干头信息和订单细节组成。订单细节又有订货的零件号、数量等来描述。按照准则，订单细节就不能作订单的属性处理而应该上升为实体。

（2）一张订单可以订若干产品，所以订单与订单细节两个实体之间是1：n的联系。

（3）原订单和产品的联系实际上是订单细节和产品的联系。每条订货细节对应一个产品描述，订单处理时从中获得当前单价、产品重量等信息。

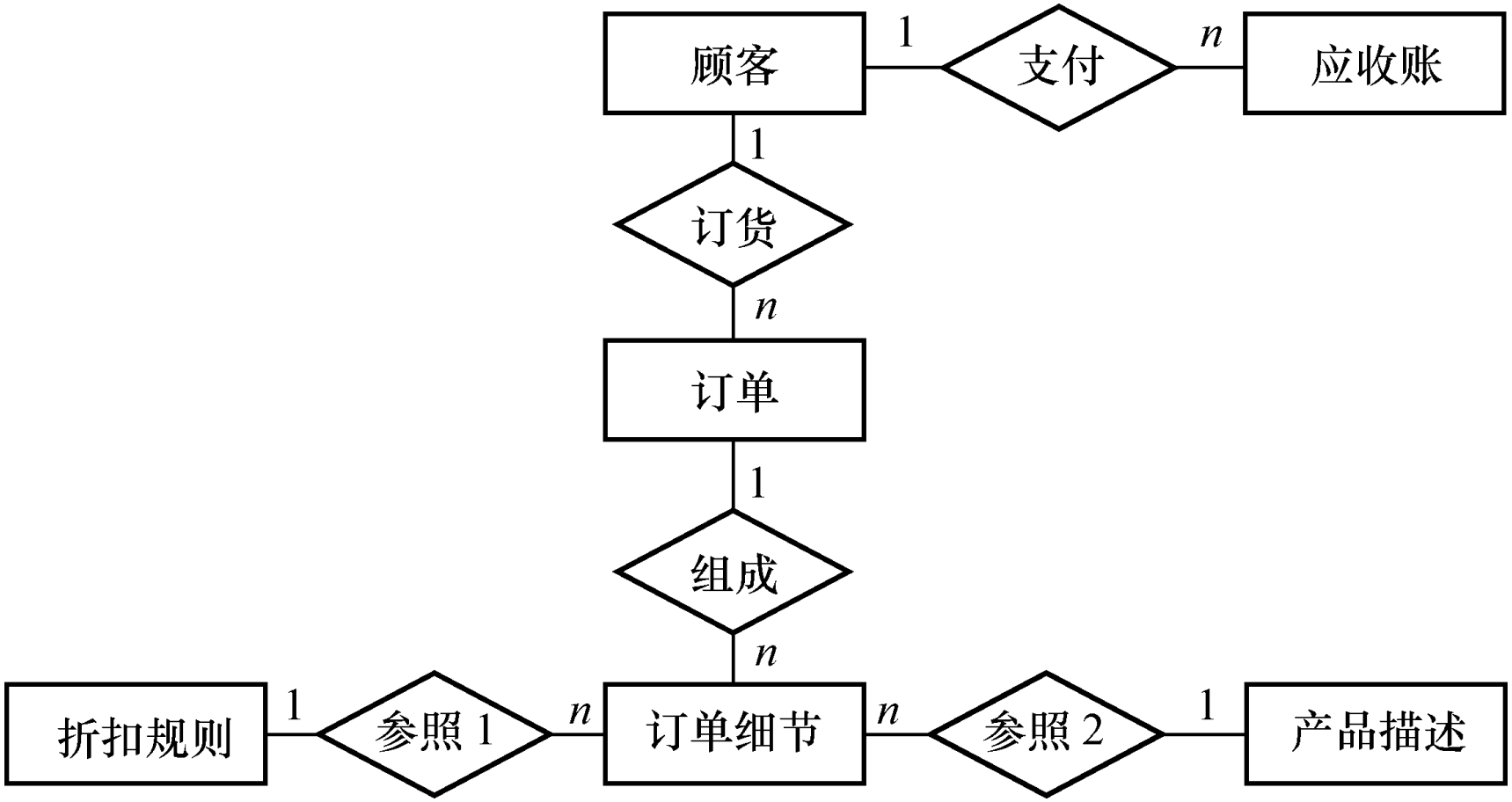
设计细化

（4）“发票清单”是一个数据存储，无须作为实体加入分E-R图。这里的数据存储对应手工凭证，发票上的信息在开具发票同时已及时存入应收账款中了。

（5）工厂对大宗订货给予优惠。每种产品都规定了不同订货数量的折扣，应增加一个“折扣规则”实体存放这些信息，而不应把它们放在产品描述实体中。

局部ER图结果

得到分E-R图如下图所示



销售管理子系统的分E-R图

实体属性设计

- 对每个实体定义的属性如下：
  - 顾客：{顾客号，顾客名，地址，电话，信贷状况，账目余额}
  - 订单：{订单号，顾客号，订货项数，订货日期，交货日期，工种号，生产地点}
  - 订单细则：{订单号，细则号，零件号，订货数，金额}
  - 应收账款：{订单号，顾客号,发票号，应收金额，支付日期，支付金额，当前余额，货款限额}
  - 产品描述：{产品号，产品名，单价，重量}
  - 折扣规则：{产品号，订货量，折扣}

视图的集成

一般说来，视图集成可以有两种方式：

（1）多个分E-R图一次集成。

（2）逐步集成，用累加的方式一次集成两个分E-R图。

第1种方法比较复杂，做起来难度较大。

第2种方法每次只集成两个分E-R图，可以降低复杂度。

视图的集成步骤

无论采用哪种方式，每次集成局部E-R图时都需要分两步走

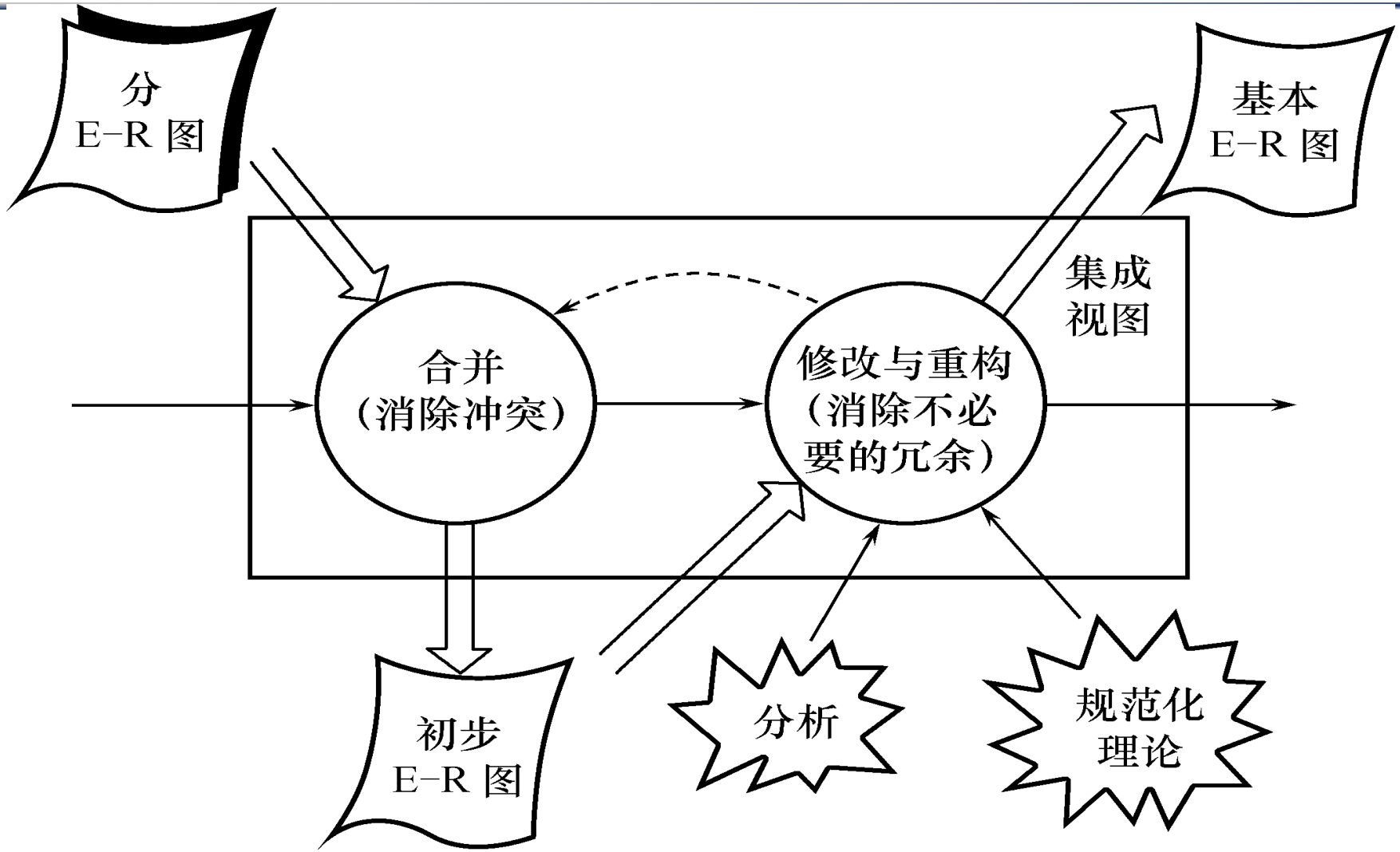
（1）**合并**。

解决各分E-R图之间的冲突，将各分E-R图合并起来生成初步E-R图。

（2）**修改和重构**。

消除不必要的冗余，生成基本E-R图。

视图的集成



验证整体概念结构

- 视图集成后形成一个整体的数据库概念结构，对该整体概念结构还必须进行**进一步验证**，确保它能够满足下列条件：
  - 整体概念结构内部必须具有**一致性**，不存在互相矛盾的表达
  - 整体概念结构能准确地**反映原来的每个视图结构**，包括属性、实体及实体间的联系
  - 整体概念结构能**满足需求分析阶段所确定的所有要求**

验证整体概念结构（续）

- 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。

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

7.4 逻辑结构设计

逻辑结构设计的任务就是把概念结构设计阶段设计好的基本E-R图转换为与选用的DBMS产品所支持的数据模型相符合的逻辑结构。

设计逻辑结构时一般要分两步进行：

（1）将概念结构**转换**为关系模型；

（2）对关系数据模型进行**优化**。

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

如何将实体和实体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。

关系模型的逻辑结构是一组关系模式的集合。

E-R图向关系模型的转换

转换时遵循下列原则：

1） **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，实体的码即为关系的码。

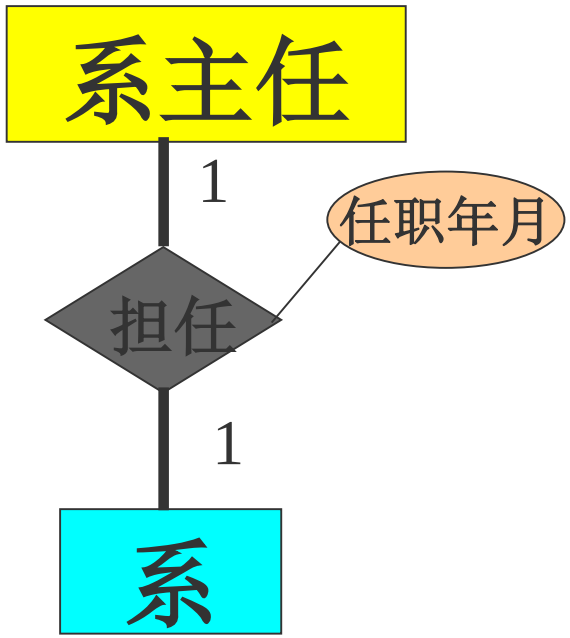
如：学生（学号，姓名，年龄）

2） 一个联系转换为一个关系模式，有三种情况：

1：1， 1：N， M：N

E-R图向关系模型的转换

(1)若联系为1:1



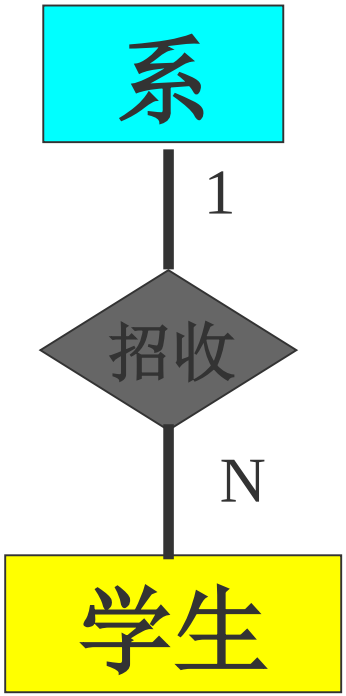
这个模式有什么问题?

系(<u>系编号</u>, 系名, 电话, 地址, **系主任名**, 任职年月)

系主任（<u>系主任名</u>， 年龄， 性别， 职称）

E-R图向关系模型的转换

(2)若联系为1:N

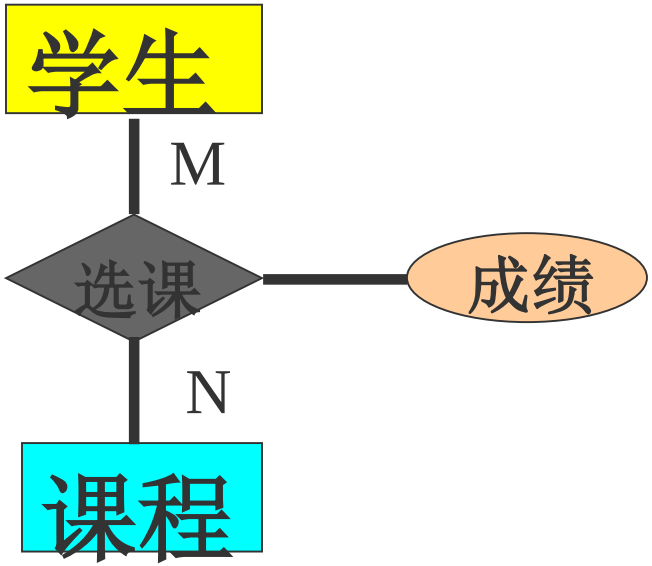


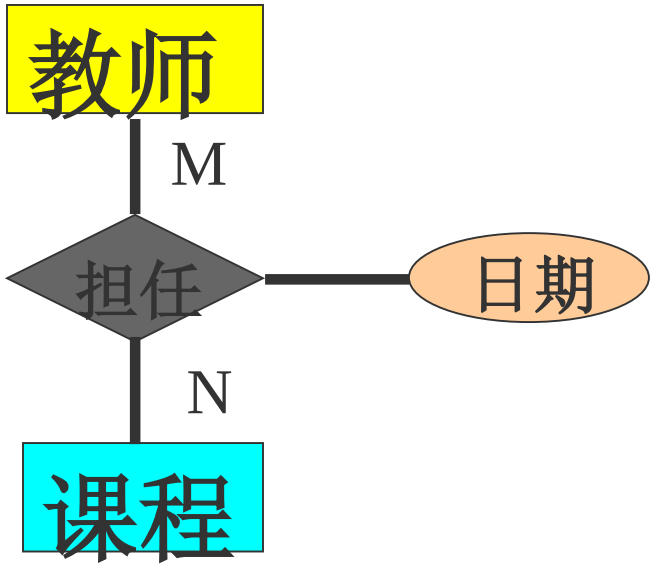
系(<u>系编号</u>， 系名,电话,地址,系主任)

学生(<u>学号</u>, **系号**, 姓名, 年龄, 性别)

E-R图向关系模型的转换

(3)若联系为M:N





C(C#, CN, CREDIT)

S(S#, SN, AGE, SEX)

SC(S#,C#,G)

T(T#, TN, AGE)

C(C#, CN, CREDIT)

TC(T#, C#, DATE)

ER模型到关系模型的转换实例

仓库（仓库号，仓库名，地址）

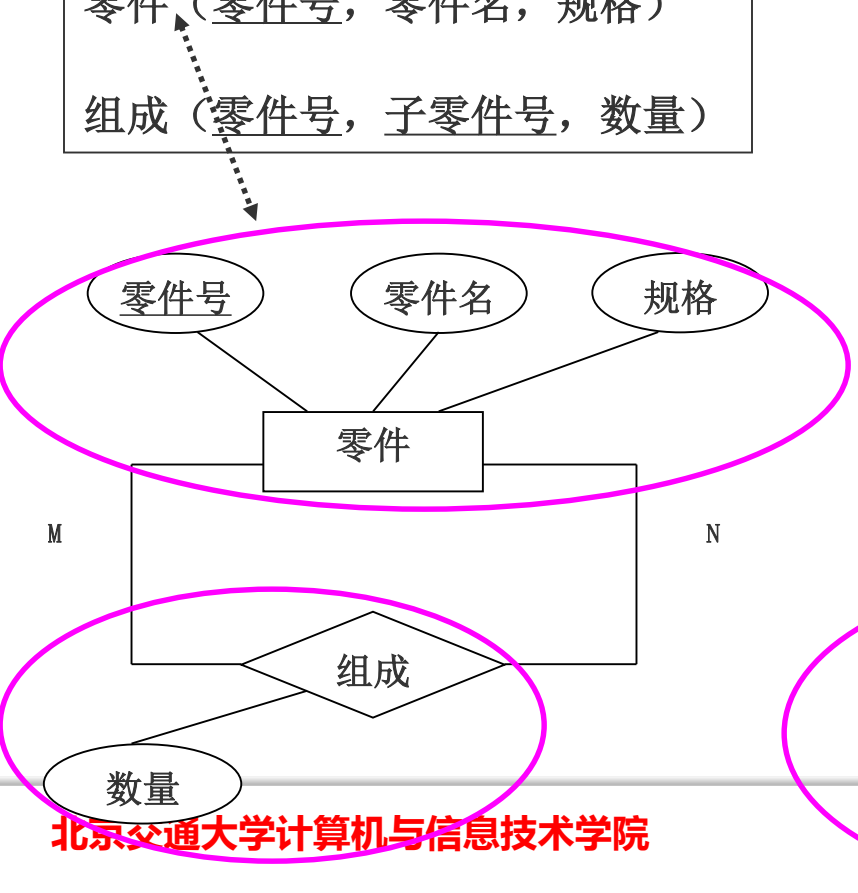
商店（商店号，商品名）

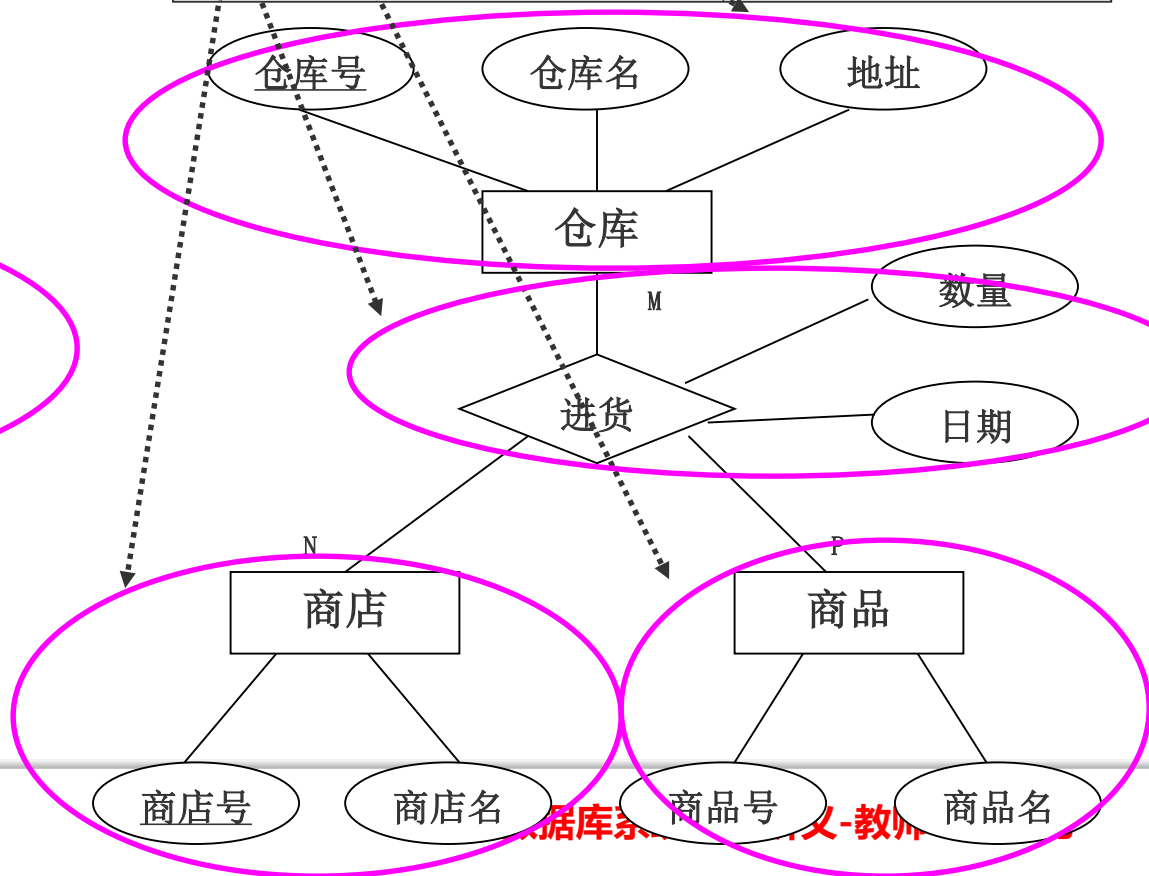
商品（商品号，商品名）

进货（商店号，商品号，仓库号，日期，数量）

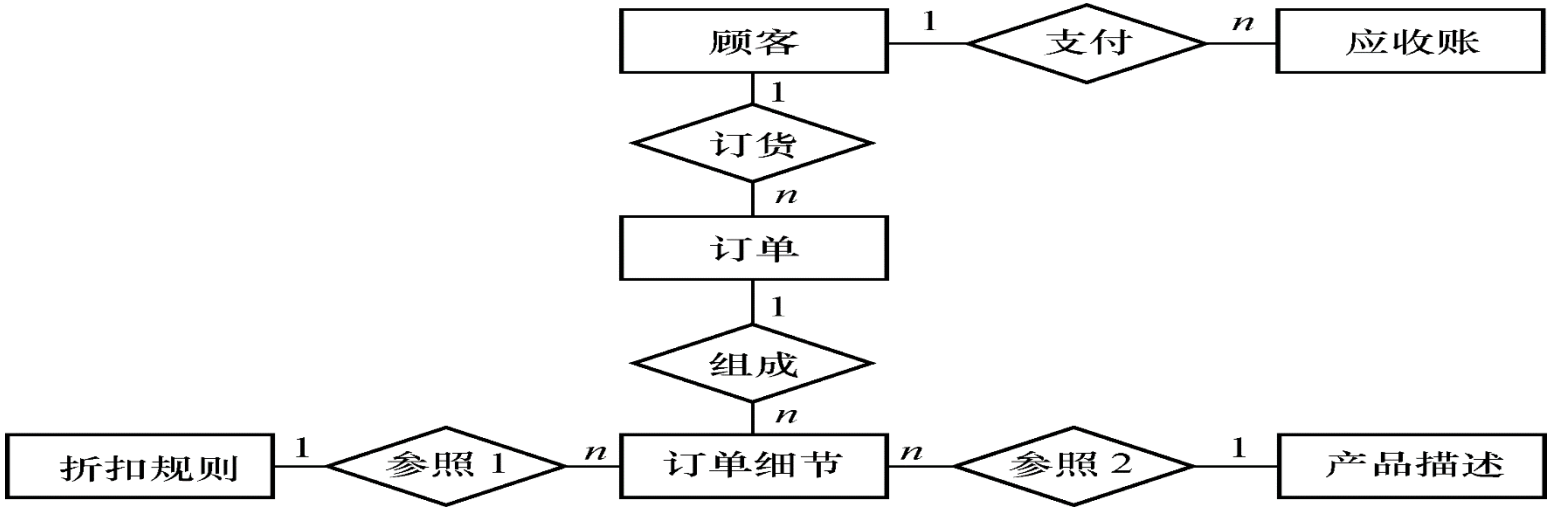
零件（零件号，零件名，规格）

组成（零件号，子零件号，数量）





ER模型到关系模型的转换实例



顾客（顾客号，顾客名，地址，电话，信贷状况，账目余额）
订单（订单号，顾客号，订货项数，订货日期，交货日期，工种号，生产地点）
订单细节：（订单号，细则号，产品号，订货数，金额）
应收账款：（订单号，顾客号，发票号，应收金额，支付日期，支付金额，当前余额，货款限额）
产品描述：（产品号，产品名，单价，重量）
折扣规则：（产品号，订货量，折扣）

数据模型的优化

数据库逻辑设计的结果**不是唯一的**。

为了提高数据库应用系统的性能，还应该根据应用需要适当地修改、调整关系模式，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。

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

2. **消除冗余的联系**

   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. **确定所属范式**

   - 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

优化数据模型的方法

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**。

注意：并不是规范化程度越高的关系就越优，一般说来，第三范式就足够了

数据模型调整案例

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

显然有： 学号→(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，仍然可保留该冗余数据，对关系模式不再做进一步分解

优化数据模型的方法

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**，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

优化数据模型的方法

- **水平分解**
  - ➢什么是水平分解
    把(基本)关系的元组分为若干子集合，定义每个子集合为一个子关系，以提高系统的效率
  - ➢水平分解的适用范围
    满足“80/20原则”的应用
    并发事务经常存取不相交的数据
- **垂直分解**
  - ➢什么是垂直分解
    把关系模式*R*的属性分解为若干子集合，形成若干子关系模式
  - ➢垂直分解的适用范围
    取决于分解后*R*上的所有事务的总效率是否得到了提高

设计用户子模式

定义**用户外模式**时应该注重的问题：

(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View，以满足系统对安全性的要求。

(3) 简化用户对系统的使用

设计用户子模式案例

[例] 关系模式产品（产品号，产品名，规格，单价，生产车间，生产负责人，产品成本，产品合格率，质量等级），可以在产品关系上建立两个视图：

为一般顾客建立视图：

产品1（产品号，产品名，规格，单价）

为产品销售部门建立视图：

产品2（产品号，产品名，规格，单价，车间，生产负责人）

- 顾客视图中只包含允许顾客查询的属性
- 销售部门视图中只包含允许销售部门查询的属性
- 生产领导部门则可以查询全部产品数据
- 可以防止用户非法访问不允许他们查询的数据，保证系统的安全性

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
- 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

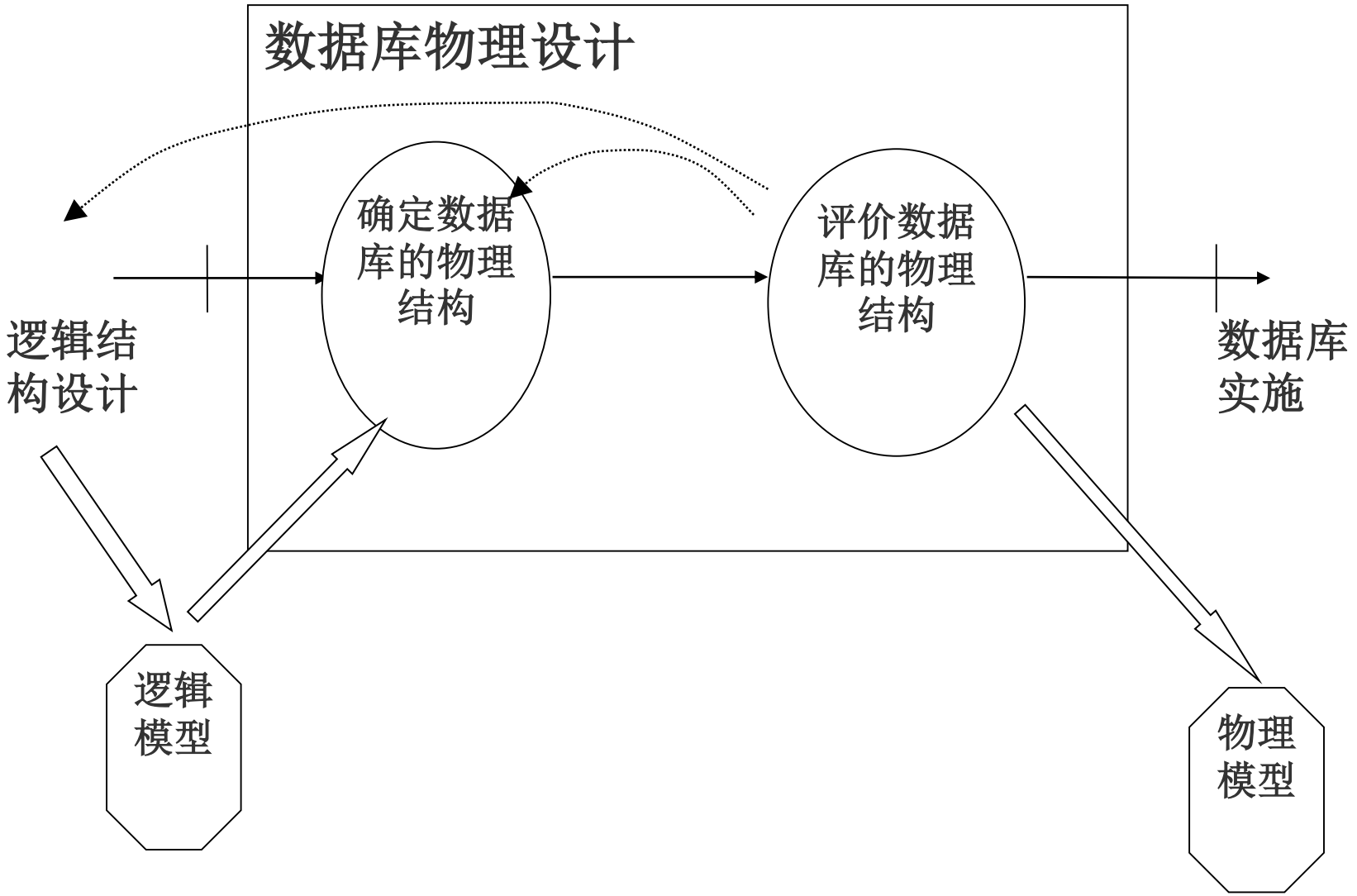
7.5 数据库的物理设计

- 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**，它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 数据库物理设计的步骤

- 确定数据库的物理结构，在关系数据库中主要指存取方法和存储结构
- 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

数据库物理设计的步骤



7.5 数据库的物理设计

7.5.1 数据库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 数据库物理设计的内容和方法

- 设计物理数据库结构的准备工作
  - 对要运行的事务进行详细分析，获得选择物理数据库设计所需参数
  - 充分了解所用RDBMS的内部特征，特别是系统提供的存取方法和存储结构

7.5.1 数据库物理设计的内容和方法

- 选择物理数据库设计所需参数

1.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

2.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

3.每个事务在各关系上运行的**频率和性能要求**

7.5.1 数据库物理设计的内容和方法

- 关系数据库物理设计的内容

1、为关系模式选择存取方法(建立存取路径)

2、设计关系、索引等数据库文件的物理存储结构

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
- 物理设计的任务之一就是要确定选择哪些**存取方法**，即建立**哪些存取路径**
- DBMS常用存取方法
  - 索引方法
    - 目前主要是B+树索引方法
    - 经典存取方法，使用最普遍
  - 聚簇（Cluster）方法
  - HASH方法

一、索引存取方法的选择

- 根据应用要求确定
  - 对哪些属性列建立索引
  - 对哪些属性列建立组合索引
  - 对哪些索引要设计为唯一索引

索引存取方法的选择

- 选择索引存取方法的一般规则
  - 如果一个(或一组)属性**经常在查询条件中出现**，则考虑在这个(或这组)属性上建立索引(或组合索引)
  - 如果一个属性经常作为**最大值和最小值**等聚集函数的参数，则考虑在这个属性上建立索引
  - 如果一个(或一组)属性经常在连接操作的**连接条件中出现**，则考虑在这个(或这组)属性上建立索引
- 关系上定义的索引数过多会带来较多的额外开销
  - **维护索引**的开销
  - **查找索引**的开销

二、聚簇存取方法的选择

- 聚簇
  - 为了提高某个属性（或属性组）的查询速度，把这个或这些属性（称为聚簇码）上**具有相同值的元组集中存放在连续的物理块称为聚簇**

聚簇存取方法的选择

■ 聚簇的用途

1. 大大提高按聚簇码进行查询的效率

例：假设学生关系按所在系建有索引，现在要查询信息系的所有学生名单。

- 信息系的500名学生分布在500个不同的物理块上时，至少要执行500次I/O操作
- 如果将同一系的学生元组集中存放，则每读一个物理块可得到多个满足查询条件的元组，从而显著地减少了访问磁盘的次数

2. 节省存储空间

聚簇以后，聚簇码相同的元组集中在一起了，因而聚簇码值

不必在每个元组中重复存储，只要在一组中存一次就行了

聚簇存取方法的选择（续）

■聚簇的局限性

1. 聚簇只能提高某些特定应用的性能
2. 建立与维护聚簇的开销相当大
   - 对已有关系建立聚簇，将导致关系中元组移动其物理存储位置，并使此关系上原有的索引无效，必须重建
   - 当一个元组的聚簇码改变时，该元组的存储位置也要做相应移动

聚簇存取方法的选择（续）

■聚簇的适用范围

1. 既适用于单个关系独立聚簇，也适用于**多个关系组合聚簇**

例：假设用户经常要按系别查询学生成绩单，这一查询涉及学生关系和选修关系的连接操作，即需要按学号连接这两个关系，为提高连接操作的效率，可以把具有相同学号值的学生元组和选修元组在物理上聚簇在一起。这就相当于把多个关系按“预连接”的形式存放，从而大大提高连接操作的效率。

聚簇存取方法的选择（续）

2. 如果通过聚簇码进行访问或连接是该关系的主要应用模式，而与聚簇码无关的其他访问很少或者是次要的时，可以使用聚簇。

- 尤其当SQL语句中包含有与聚簇码有关的ORDER BY，GROUP BY，UNION，DISTINCT等子句或短语时，使用聚簇特别有利，可以省去对结果集的排序操作

聚簇存取方法的选择（续）

设计候选聚簇

- 对经常在一起进行连接操作的关系可以建立聚簇
- 如果一个关系的一组属性经常出现在相等比较条件中，则该单个关系可建立聚簇
- 如果一个关系的一个(或一组)属性上的值重复率很高，则此单个关系可建立聚簇。即对应每个聚簇码值的平均元组数不太少。太少了，聚簇的效果不明显

聚簇存取方法的选择（续）

- **优化聚簇设计**
  - 从聚簇中删除经常进行全表扫描的关系；
  - 从聚簇中删除更新操作远多于连接操作的关系；
  - 一个关系可以在某一个聚簇中，但不能同时加入多个聚簇
    - 从多个聚簇方案(包括不建立聚簇)中选择一个较优的，即在这个聚簇上运行各种事务的总代价最小

三、HASH存取方法的选择

选择HASH存取方法的规则

当一个关系满足下列两个条件时，可以选择HASH存取方法

（1）该关系的属性主要出现在等值连接条件中或主要出现在相等比较选择条件中

（2）该关系的大小可预知，而且不变；

或

该关系的大小动态改变，但所选用的DBMS提供了动态HASH存取方法

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

- 确定数据存放位置和存储结构的因素
  - 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--tradeoff

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**部分分开存放
  - **存取频率较高**部分与**存取频率较低**部分，分开存放

确定数据的存放位置（续）

例：

- 数据库数据备份、日志文件备份等由于只在故障恢复时才使用，而且数据量很大，可以考虑存放在磁带上
- 如果计算机有多个磁盘或磁盘阵列，可以考虑将表和索引分别放在不同的磁盘上，在查询时，由于磁盘驱动器并行工作，可以提高物理I/O读写的效率
- 可以将比较大的表分别放在两个或多个磁盘上，以加快存取速度，这在多用户环境下特别有效
- 可以将日志文件与数据库对象（表、索引等）放在不同的磁盘以改进系统的性能

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
  - 内存分配参数
  - 使用的缓冲区长度、个数
  - 存储分配参数
  - .....

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

评价方法（完全依赖于所选用的DBMS）

- 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

完成数据库的物理设计之后的实施工作

(1) 编写脚本

(2) 建库

(3) 组织数据入库

(4) 应用程序编写与调试

(5) 试运行

(6) 正式运行和维护

数据的载入和应用程序的调试

- 数据库实施阶段包括两项重要的工作
  - 数据的载入
  - 应用程序的调试

数据库的试运行

- 输入一小部分数据后就可以开始对数据库系统进行联合调试，这又称为数据库的试运行。
- 这一阶段要实际运行数据库应用程序，执行对数据库的各种操作，测试应用程序的功能是否满足设计要求。如果不满足，对应用程序部分则要修改和调整，直到达到设计要求为止。

数据库的运行和维护

在数据库运行阶段，对数据库经常性的维护工作主要由DBA完成，包括：

1．数据库的转储和恢复

数据库的转储和恢复是系统正式运行后最重要的维护工作之一。

2．数据库的安全性、完整性控制

数据库的运行和维护

3.数据库性能的监督、分析和改造

在数据库运行过程中，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是DBA的又一重要任务。

4.数据库的重组织与重构造

数据库运行一段时间后，由于记录不断增、删、改，会使数据库的物理存储情况变坏，降低了数据的存取效率，数据库性能下降，这时DBA就要对数据库进行重组织，或部分重组织（只对频繁增、删的表进行重组织）。RDBMS一般都提供数据重组织用的实用程序。

7.7 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
  - 实施和维护

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

本章作业

1.完成书中的两个习题

(第四版p.38 12，13,第五版P.241 7,8)

其中，各实体的属性如下（联系的属性根据需要添加）：

系：系编号，系名

班级：班级编号，班级名

教研室：教研室编号，教研室

学生：学号，姓名，学历

课程：课程编号，课程名

教员：职工号，姓名，职称

产品：产品号，产品名

零件：零件号，零件名

原材料：原材料号，原材料名，类别

仓库：仓库号，仓库名

2.将上题的E-R图转换成关系模型，指明每个关系模式的主键和外键，在函数依赖的范畴分析关系模式满足第几范式，并将不满足BCNF的关系模式分解成BCNF。

下课了。。。

攀登

休息一会儿。。。