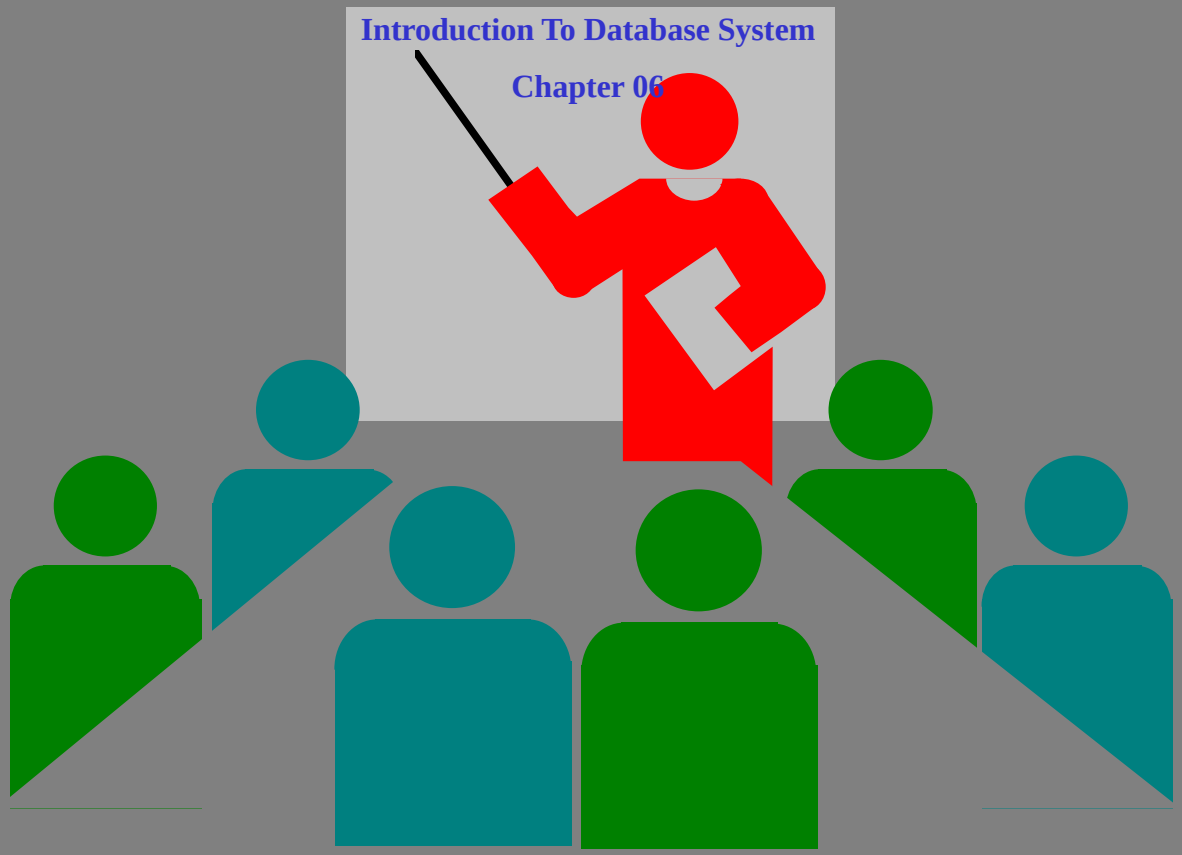
第六章　数据库设计



6.1　数据库设计概述

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要求（信息要求和处理要求）。

6.1.1 数据库和信息系统

信息系统：它是提供信息、辅助人们对环境进行控制和进行决策的系统。

数据库：是信息系统的核心和基础。它把信息系统中大量的数据按照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。

数据库和信息系统（续）

从事数据库设计的专业人员来讲，应该具备多方面的技术和知识，主要包括：

- 数据库的基本知识和数据库设计技术；
- 计算机科学的基础知识和程序设计的方法和技巧；
- 软件工程的原理和方法；
- 应用领域的知识；

6.1.2 数据库设计的特点

数据库设计的特点之一：

数据库建设是硬件、软件和干件（技术与管理的界面）的结合。

数据库设计的特点之二：

在数据库的整个设计过程中要把结构（数据）设计和行为（处理）设计密切结合起来。

数据库设计的特点（续）

早期的数据库设计致力于数据模型和建模方法研究，着重结构特性的设计而忽视了对行为的设计。

传统数据库设计的特点

图 6.1　结构和行为分离的设计

6.1.3 数据库设计方法简述

手工试凑法：

与设计人员的经验和水平有直接关系，数据库设计成为一种技艺而不是工程技术，缺乏科学理论和工程方法的支持，工程的质量难以保证，增加了系统维护的代价。

规范设计法：

运用软件工程的思想和方法，提出了各种设计准则和规范。其基本思想是过程迭代和逐步求精。

专门的数据库设计工具：

可以自动地或辅助设计人员完成数据库设计过程中的很多任务。

6.1.4　数据库设计的基本步骤

，将数据库设计分为以下六个阶段：

- 需求分析；
- 概念结构设计；
- 逻辑结构设计；
- 物理结构设计；
- 数据库实施；
- 数据库运行和维护；

参与数据库设计的主要人员

数据库设计的人员，包括系统分析人员、数据库设计人员、程序员、用户和数据库管理员。

系统分析和数据库设计人员是数据库设计的核心人员，他们将自始至终参与数据库设计.

用户和数据库管理员 主要参与需求分析和数据库的运行维护.

程序员则在系统实施阶段参与进来，分别负责编写程序和准备软硬件环境等。

如果所设计的数据库比较复杂，还应该考虑是否需要使用数据库设计工具和CASE工具以提高数据库设计质量，并减少设计工作量。

数据库设计中各阶段简要介绍

一、需求分析阶段

了解与分析用户需求， 是整个设计过程的基础。是最困难、最耗时的一步。需求分析做得充分与准确，决定了在其上构建数据库大厦的速度与质量。需求分析做得不好，可能会导致整个数据库设计返工重做。

二、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，形成概念模型。

数据库设计中各阶段简要介绍--续

三、逻辑结构设计阶段

是将概念结构转换为某个**DBMS**所支持的数据模型，并对其进行优化。

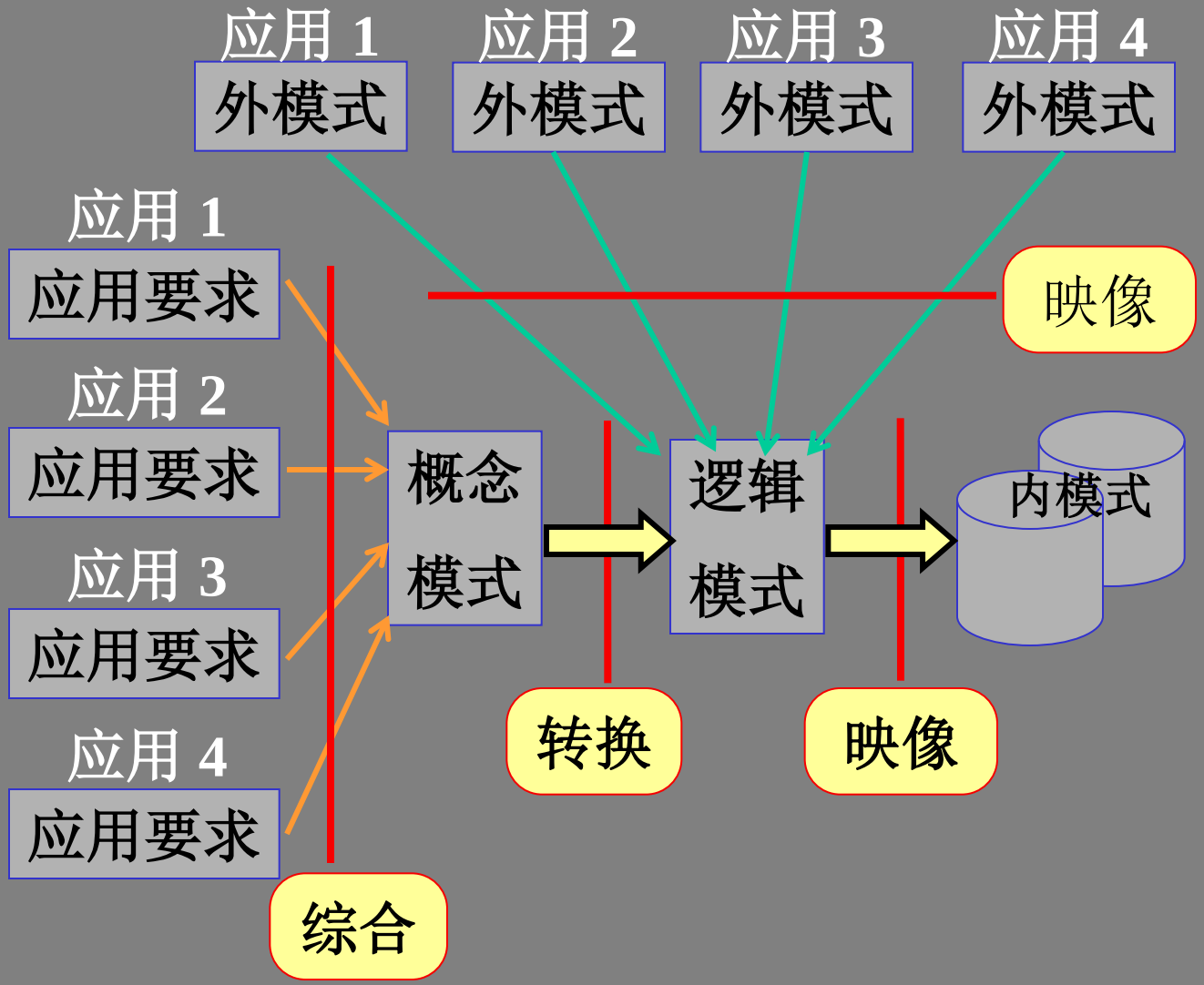
四、数据库物理设计阶段

是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）。

- 五、数据库实施阶段
- 设计人员运用DBMS提供的数据语言及其宿主语言，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。

- 六、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。

数据库的各级模式


应用 1
外模式
应用 2
外模式
应用 3
外模式
应用 4
外模式
应用 1
应用要求
应用 2
应用要求
应用 3
应用要求
应用 4
应用要求
映像
概念
模式
逻辑
模式
内模式
转换
映像
综合

6.2　需求分析

简单地说，需求分析就是分析用户的要求。它是数据库设计的起点，需求分析的结果是否准确地反映了用户的实际要求，将直接影响到数据库设计过程中后面的各个阶段，并影响设计结果是否合理和使用。

用户在数据库设计过程中的作用

确定用户的最终需求是很困难的，这主要是因为：

- 用户缺少计算机知识
- 数据库设计人员缺少用户所在领域的专业知识

- 基于以上两点，在数据库设计的过程中，要求设计人员必须不断深入地与用户交流，才能逐步确定用户的实际需求。在整个设计和开发过程中，必须自始至终地强调用户的参与。

6.2.2 需求分析的方法

调查用户需求的具体步骤是：

1) 1) 调查组织机构的情况

2) 2) 调查各部门的业务活动情况

3) 3) 在熟悉了业务活动的基础上，协助用户明确对新系统的

4) 各种 要求。

4) 确定新系统的边界 由计算机完成的功能就是新系统应该实现的功能。

需求分析中常用的调查方法

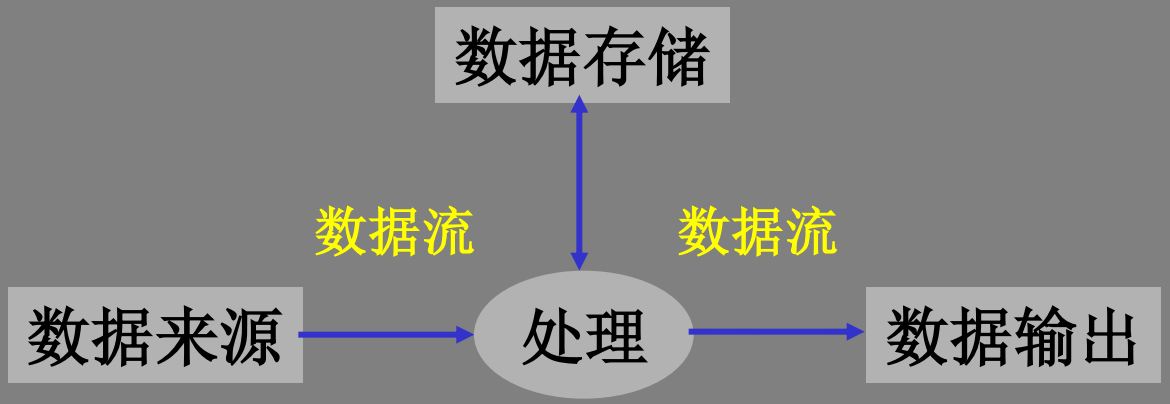
在需求分析的调查过程中，常用的调查方法有：

- 跟班作业。可准确理解用户需求，但费时间。
- 开调查会。通过座谈了解业务活动及用户需求。
- 请专人介绍。请该领域的专家或技术专员介绍。
- 询问。对某些调查中的问题，找专人询问。
- 设计调查表请用户填写。调查表须设计合理。
- 查阅纪录。查阅与原系统有关的数据记录。

总之，做需求调查时，往往需要同时采用上面的多种方法，但是无论使用何种调查方法，都必须有用户的积极参与和配合。

分析和表达用户的需求的方法

调查了解了用户的需求后，数据库设计人员还需要进一步分析和表达用户的需求。在众多的分析方法中结构化分析方法（SA方法）是一种简单实用的方法。SA方法从最上层的系统组织结构入手，采用自顶向下、逐层分解的方式分析系统。它把任何一个系统都抽象为下图的形式：



系统高级抽象图

数据流图表达了数据和处理的关系

系统高级抽象图与数据流图

上图中给出了最高层次抽象的系统概貌，要反映更详细的内容，可以将处理功能分解为若干子功能，每个子功能还可以继续分解，直到把系统工作过程描述清楚为止。在处理功能逐步分解的同时，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。在SA方法中，处理过程的处理逻辑常常借助判定表或判定树来描述。系统中的数据则借助数据字典来描述。

对用户需求进行分析和表达后，必须提交给用户，征得用户的认可。

6.2.3 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。

数据字典通常包括数据项、数据结构、数据流、数据存储和处理过程五个部分。其中，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。

数据项

数据项是不可再分的数据单位。对数据项的描述通常包括以下内容：

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}；

其中“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，是设计数据校验功能的依据。

数据结构 与 数据流

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。

数据结构描述={数据结构名，含义说明，

组成:{数据项或者数据结构} };

数据流是数据结构在系统内传输的路径。

数据流描述={数据流名，说明，流来源，流去向，

组成:{数据结构}，平均流量，高峰期流量 };

其中“平均流量”是指在单位时间里的数据传输次数。“高峰期流量”则是指在高峰时期的数据流量。“数据流来源”是说明该数据流来自哪个过程，“数据流去向”是说明该数据流将流到哪个过程中去。

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。可以是手工文件或者是计算机文件等。

数据存储表述={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};

其中“存取频度”指每小时或每天或者每周存取几次、每次存取多少数据等信息。“存取方式”包括指明是批处理还是联机处理，是检索还是更新，是顺序检索还是随机检索等。

处理过程

处理过程的具体处理逻辑一般用判定树或判定表来描述。数据字典只需描述处理过程的说明性信息。

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}}；

其中“简要说明”中主要说明该处理过程的功能（指该处理过程用来做什么，而不是怎么做）及处理要求（包括频率要求，如单位时间里处理多少事务、多少数据量、响应时间要求等）。这些处理是数据库物理设计阶段的输入和性能评价的标准。

需求分析阶段的注意事项

这一阶段收集到的基础数据（用数据字典来表达）和一组数据流程图是后续概要设计的基础。

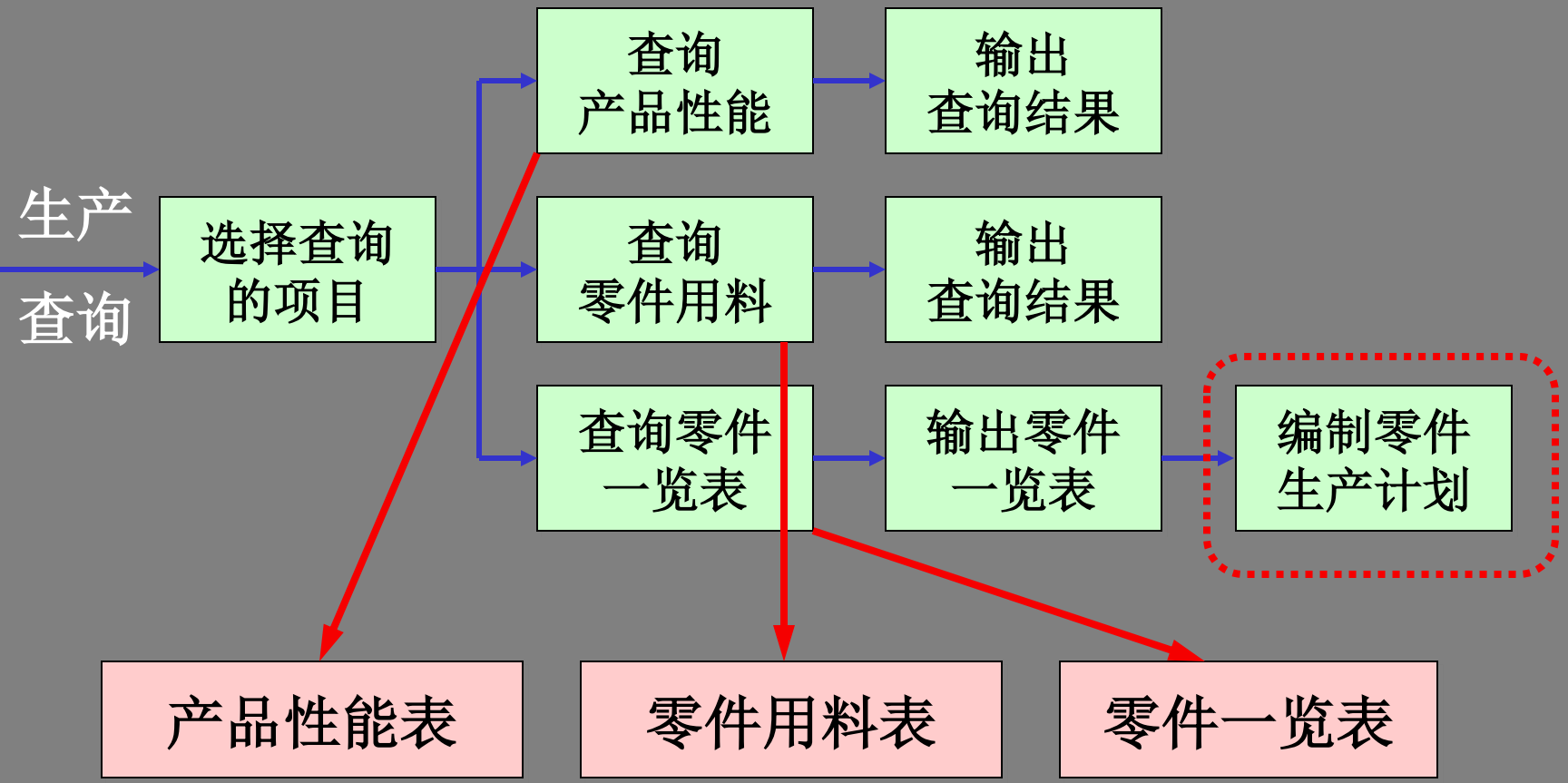
在需求分析阶段应注意下面两点：

一、系统易于扩充

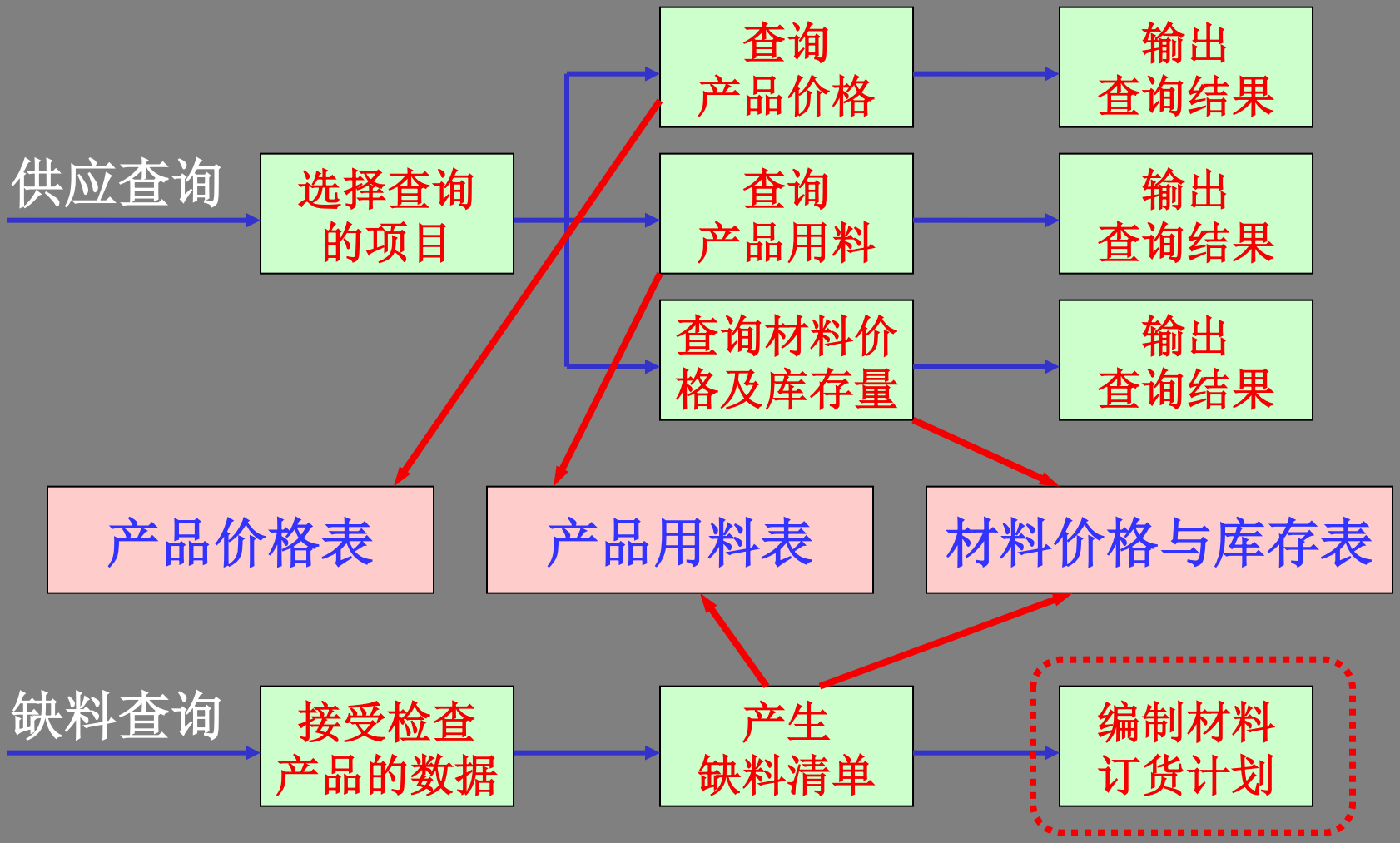
二、用户的参与

用户需求分析--实例

下面，我们简单地以某个工厂的物资管理系统为例，说明，如何进行需求分析。



用户需求分析--实例



数据流图：描述了事务处理与所需数据之间的关系；

形象描述了系统功能。

6.3.1 概念结构

概念结构设计是整个数据库设计的关键，它将需求分析得到的用户需求抽象为信息结构（即：概念结构）。

在需求分析阶段所得到的应用需求应该首先抽象为信息世界的结构，才能更好地更准确地用某一DBMS实现这些需求。

概念结构的主要特点

概念结构的主要特点有：

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。

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，通常用**E-R图**来描述概念模型。

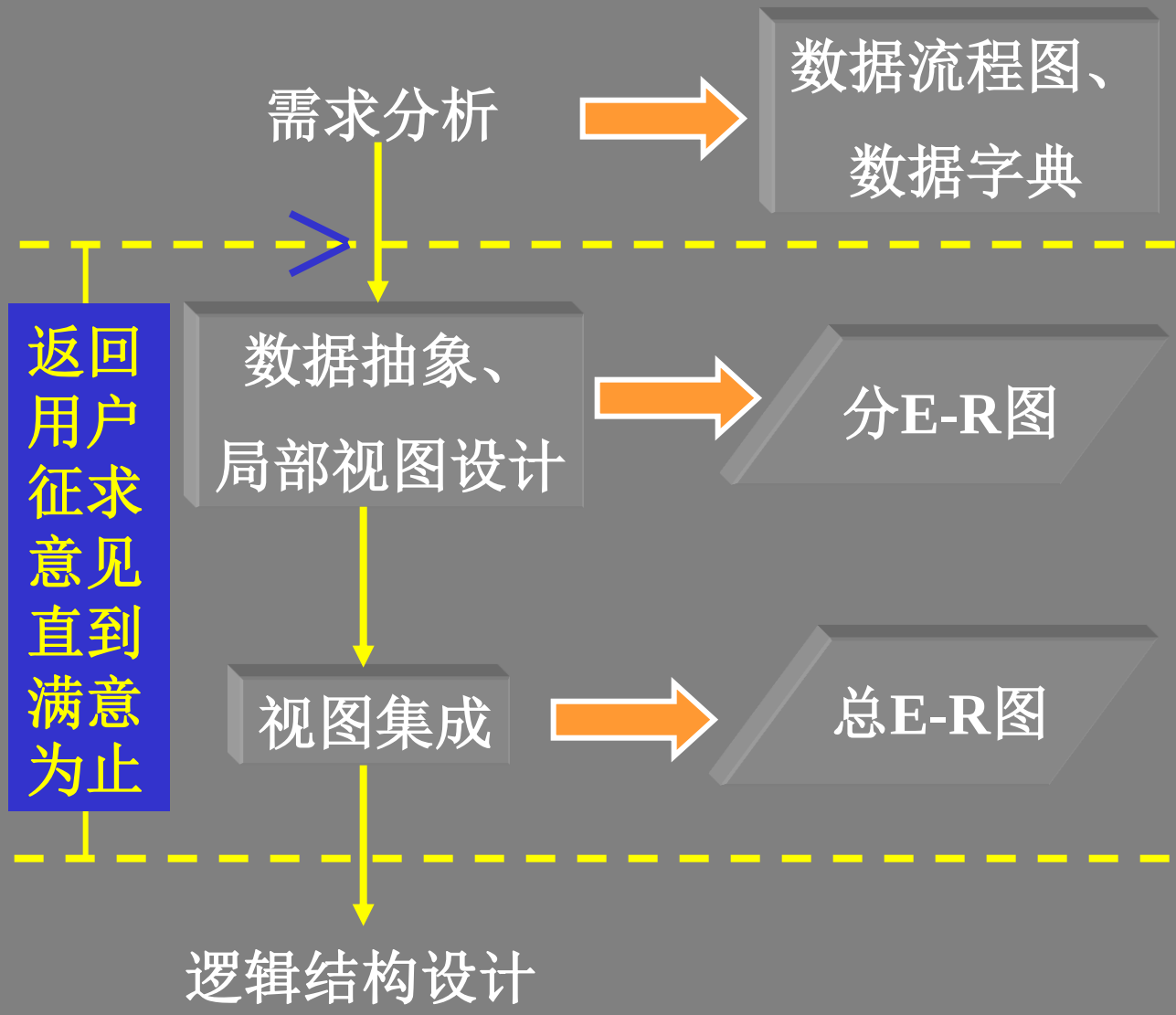
6.3.2 概念结构设计的方法与步骤

设计概念结构通常有四类方法，分别为：

- 自顶向下：即首先定义全局概念结构的框架，然后再细化。
- 自底向上：即首先定义各局部应用的概念结构，然后将他们集成起来，得到全局概念结构。
- 逐步扩张：首先定义最重要的核心概念结构，然后向外扩充，类似于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的方法相结合，用自顶向下的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。

常用的是自底向上的设计方法。即：自顶向下进行需求分析，然后再自底向上地设计概念结构。

概要设计的步骤


需求分析
数据流程图、
数据字典
数据抽象、
局部视图设计
分E-R图
视图集成
总E-R图
返回用户征求意见直到满意为止
逻辑结构设计

6.3.3 数据抽象与局部视图设计

概念结构是对现实世界的一种抽象。抽象就是对实际的人、物、事和概念进行人为处理，抽取所关心的共同特性，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。

常用的数据抽象方法有以下三种：

分类（Classification）

聚集（Aggregation）

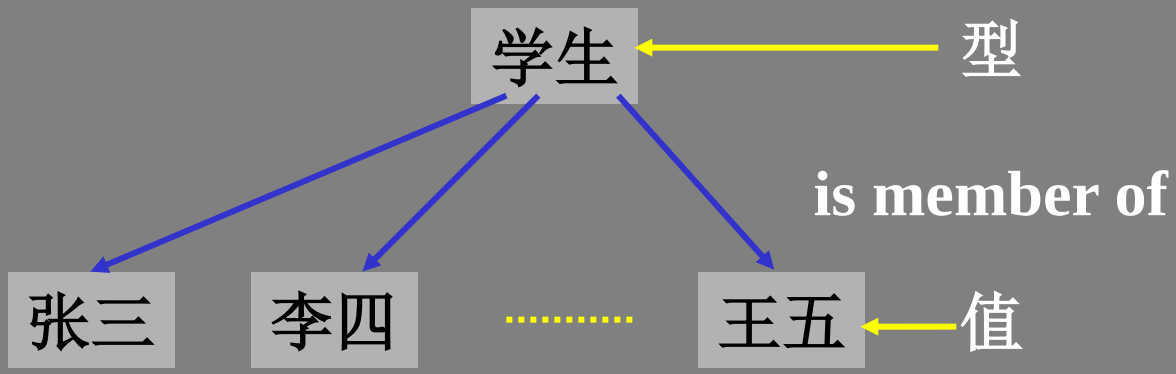
概括（Generalization）

数据抽象--分类

分类：

定义某一类概念作为现实世界中的一组对象的类型。这些对象具有某些共同的特性和行为。在E-R图中，实体型就属于分类抽象。

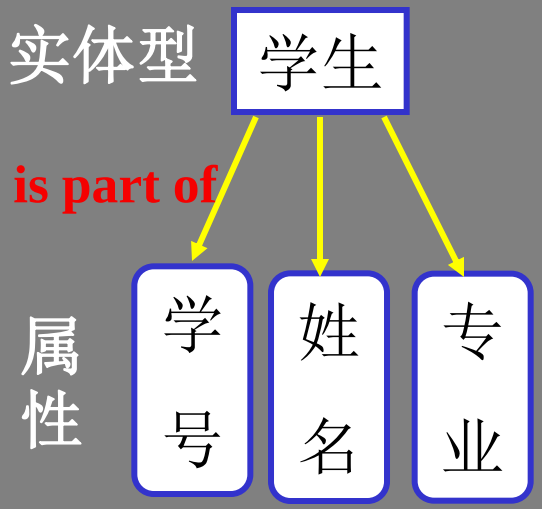
分类抽象举例：

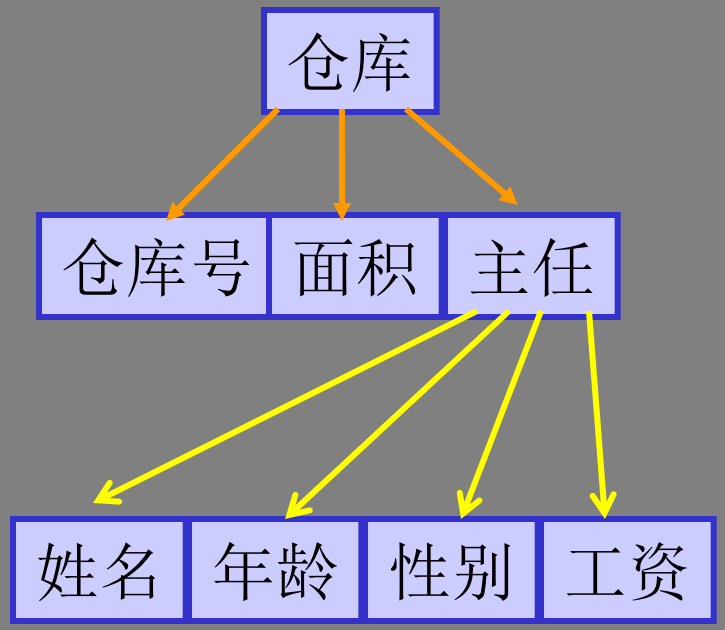


数据抽象--聚集

聚集：

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的“is part of”的语义关系。在E-R图中若干属性的聚集组成了实体型，这就是聚集抽象。举例如下：

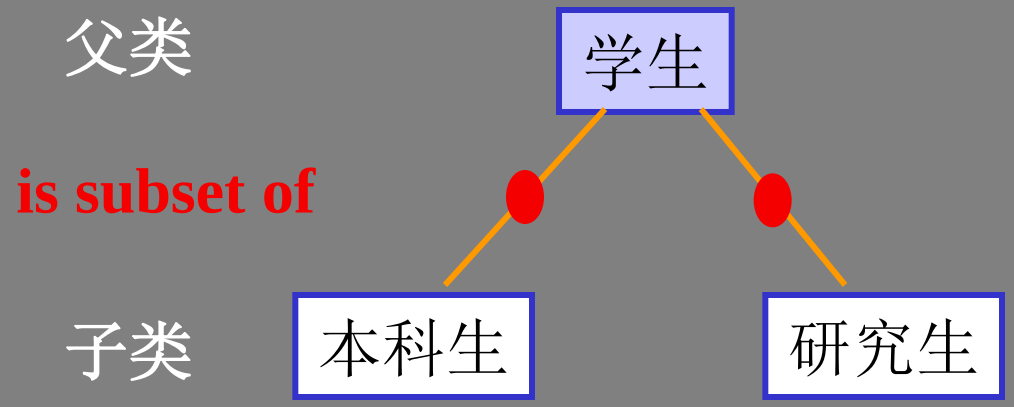




数据抽象--概括

概括：

定义类型之间的一种子集联系。它有一个很重要的特性就是：继承性。子类继承父类上定义的所有抽象，同时子类可以自定义某些特殊属性。举例如下：



数据抽象的具体步骤

概念结构设计的第一步就是利用刚才介绍的抽象机制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，码，确定实体之间的联系类型（1：1，1：m，m：n），设计分E-R图。具体做法如下：

1、选择局部应用

根据某个系统的具体情况，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图的起点，让这组图中的每一个部分对应于一个局部应用。具体的说就是将应用系统进行不同层次的划分，从而将一个大的系统变成几个子系统，将子系统作为分E-R图的出发点。

2、逐一设计分E-R图

局部E-R图的设计

当选择好局部应用之后，就要求对每个局部应用逐一设计分E-R图。

在设计局部E-R图时，将数据字典中与该局部应用相关的数据提取出来，参照数据流图，标定局部应用中的实体、实体的属性集、标识实体的码，确定实体之间的联系及其类型。

在实际应用中，往往实体和属性会有一个较自然的划分。可以先从这些内容出发定义E-R图，然后再进行必要的调整，但是在调整的过程中应遵循下列原则：

为了简化E-R图的处置，现实世界的事物能作为属性对待的，尽量作为属性对待。

区分实体与属性的两条准则

实体与属性之间并没有本质上可以截然划分的界限，那么符合什么条件的事物可以作为属性对待呢？为此，我们给出两条基本准则：

- 作为“属性”，不能再具有需要描述的性质。“属性”必须是不可再分的数据项，不能包含其他属性。
- “属性”不能与其他实体具有联系，即E-R图中所表示的联系是实体之间的联系。

凡是满足上述两条准则的事物，一般都可以作为属性对待。

销售管理子系统的分E-R图的设计

教材中P_{220}页中给出了一个关于销售管理子系统的局部E-R图的设计过程的实例，请同学们认真阅读并能够参考该例设计其他应用系统的局部E-R图。

6.3.4 视图的集成

各个子系统的分E-R图设计好以后，下一步就是要将所有的分E-R图综合成一个系统的总E-R图。这个过程就是视图的集成。通常采用下列两种方式进行视图的集成：

- 多个分E-R图一次集成；
- 逐步集成，用累加的方式一次集成两个分E-R图。

无论采用哪种方式，每次集成局部E-R图时都需要按照下列两个步骤进行：

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。

合并分E-R图，生成初步E-R图

各个局部应用所面向的问题不同，且通常是由不同的设计人员进行局部视图设计的，这可能会导致各个分E-R图之间必定会存在许多不一致的地方，称之为冲突。

各个分E-R图之间的冲突主要有三类：

属性冲突、命名冲突 和 结构冲突。

属性冲突

属性冲突主要包括两个方面：

属性域冲突　和　属性取值单位冲突。

属性域冲突：

包括属性值的类型、取值范围或取值集合不同。如：属性类型可能定义为整数或者字符型；职工的年龄可能用出生日期或者整数来表示等。

属性取值单位冲突：

如：重量可能以公斤或者以克为单位等。

冲突的解决方法：

需要各部门协商讨论，最后确定解决方案。

命名冲突

命名冲突主要包括两个方面：

同名异义　　和　　异名同义。

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。

异名同义：

同一意义的对象在不同的局部应用中具有不同的名字。如：对科研项目，财务科可能叫做项目，而在科研处可能成为课题等。

命名冲突可能发生在实体、联系一级上，也可能发生在属性一级上。后者较为常见。

解决方法：各部门协商解决。

常见的三种结构冲突及其解决方法

一、同一对象在不同应用中具有不同的抽象。如：同一对象可能同时被抽象为实体和属性。

解决方法：使同一对象具有相同的抽象。

二、同一实体在不同的分E-R图中所包含的属性个数和属性排列次序不完全相同。

解决方法：使该实体的属性取各分E-R图中所包含的属性的并集，再适当调整属性的次序。

三、实体间的联系在不同的分E-R图中为不同的类型，即可能在不同的分E-R图中分别为多对多联系和一对多联系等等。

解决方法：根据应用的语义对实体联系的类型进行综合或调整。

消除不必要的冗余，设计基本E-R图

在初步E-R图中，可能存在一些冗余的数据和实体间冗余的联系。所谓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。由于冗余的数据和联系容易破坏数据库的完整性，给数据库的维护增加困难，所以必须消除。

称消除了冗余后的初步E-R图为**基本E-R图**。

消除冗余的主要方法是分析方法，即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的逻辑关系的说明来消除冗余。

注意：并不是所有的冗余数据与冗余联系都必须加以消除的，有时为了提高效率，可以适当地以冗余信息为代价。因此，在实际应用中，应该视用户的整体需求来确定。如果保留了冗余数据，则必须把数据字典中数据关联的说明作为完整性约束条件。

运用规范化理论消除冗余

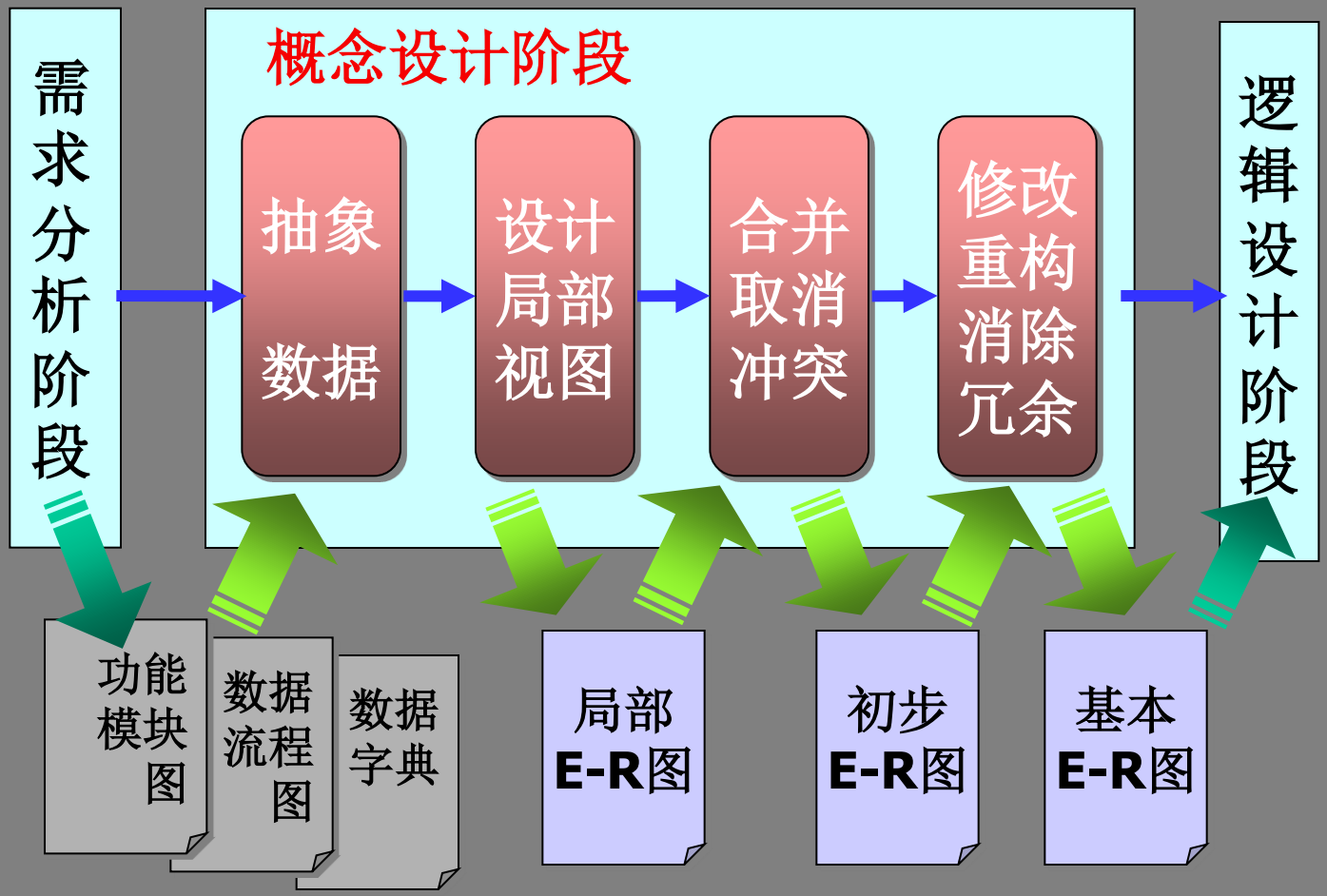
除了刚才介绍的分析方法外，还可以用规范化理论来消除冗余。在规范化理论中，函数依赖的概念提供了消除冗余联系的形式化工具。具体方法如下：

1、确定局部E-R图实体之间的数据依赖。实体之间的一对一、一对多、多对多的联系可以用实体码之间的函数依赖来表示。

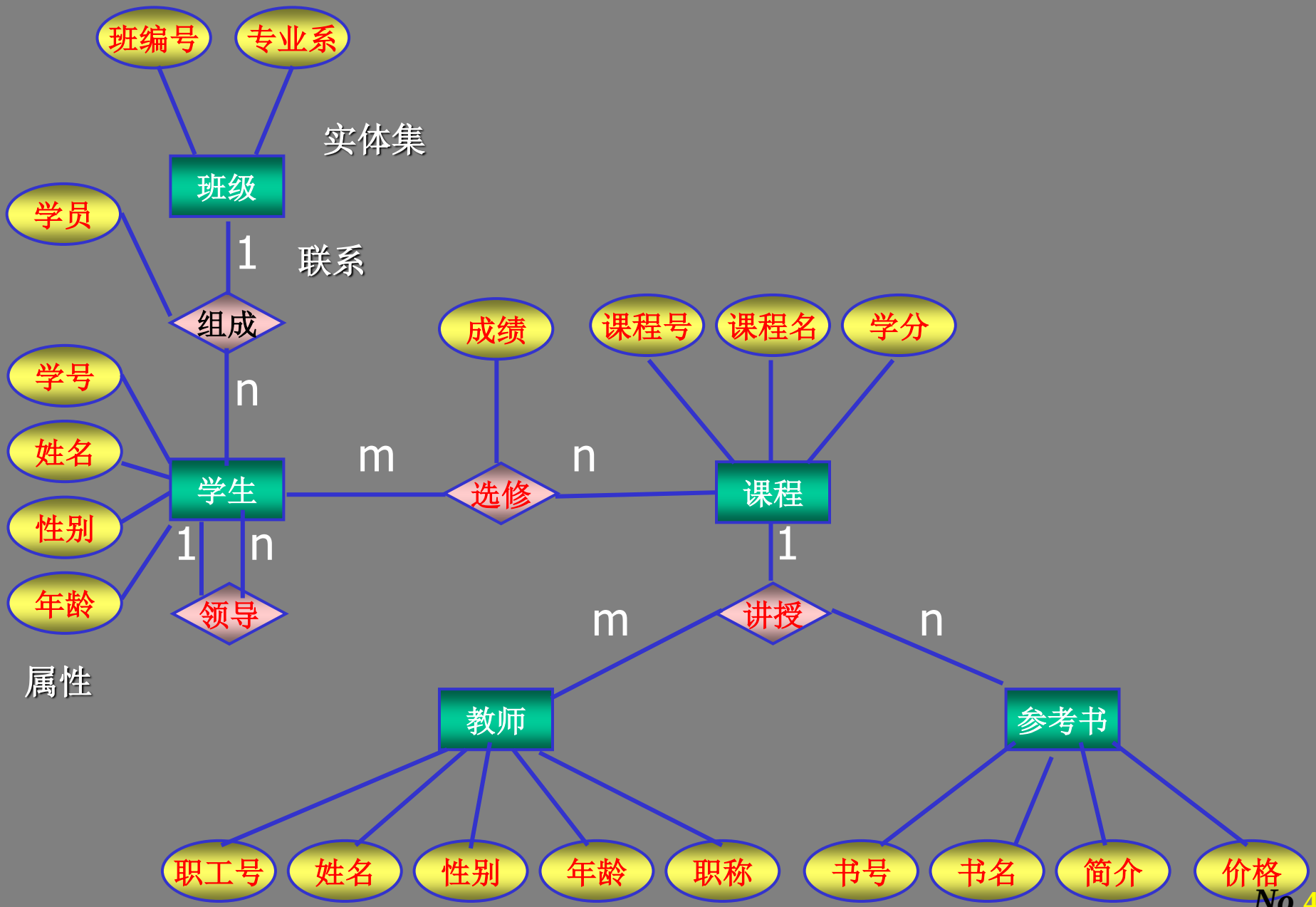
2、求上述函数依赖集F_L的最小覆盖G_L，差集为D = F_L - G_L。逐一考察D中的函数依赖，确定是否是冗余的联系。

请参考教材P_{228}中给出的实例。

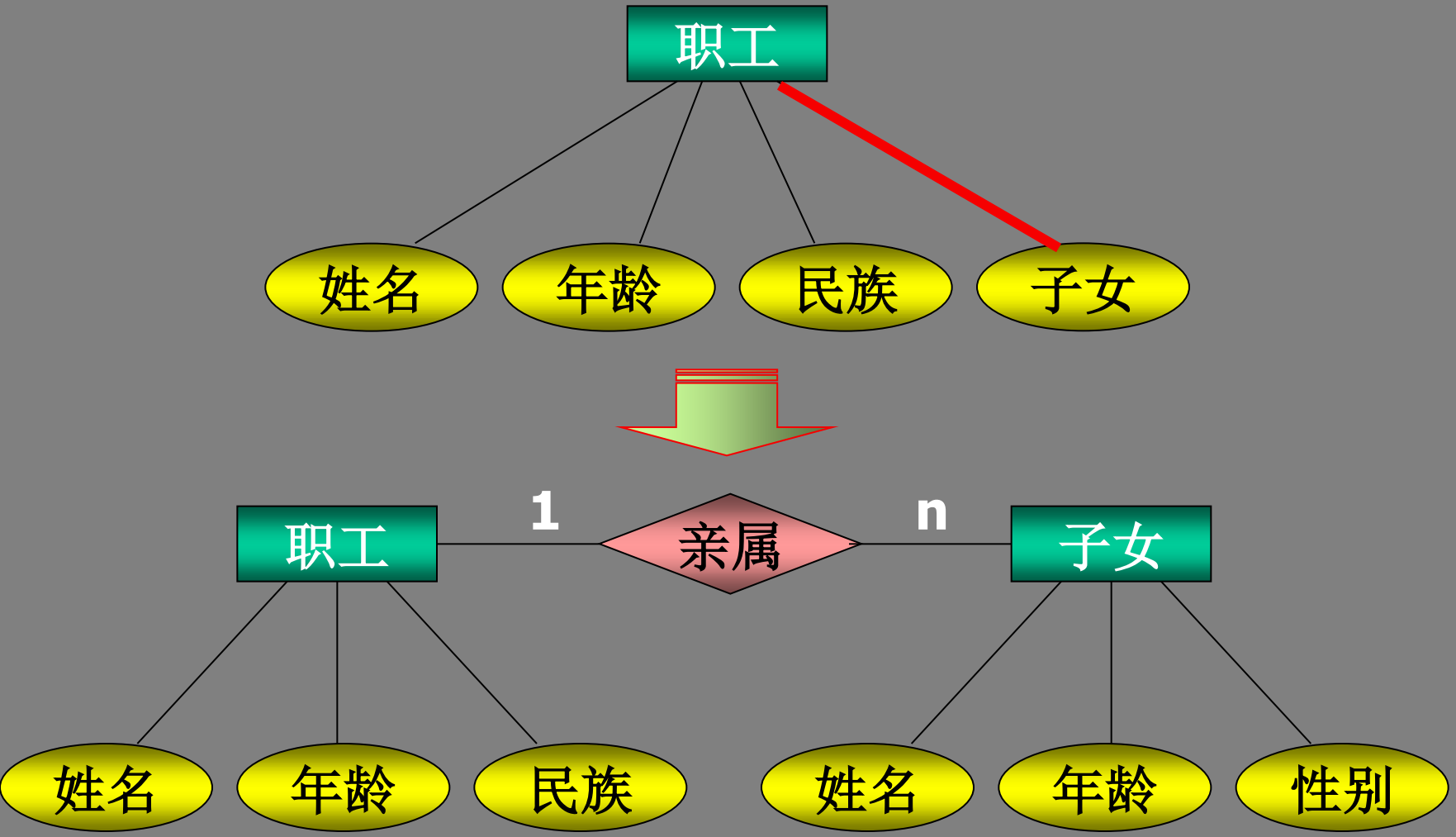
概要设计阶段小结


需求分析阶段
概念设计阶段
抽象数据
设计局部视图
合并取消冲突
修改重构消除冗余
逻辑设计阶段
功能模块图
数据流程图
数据字典
局部E-R图
初步E-R图
基本E-R图

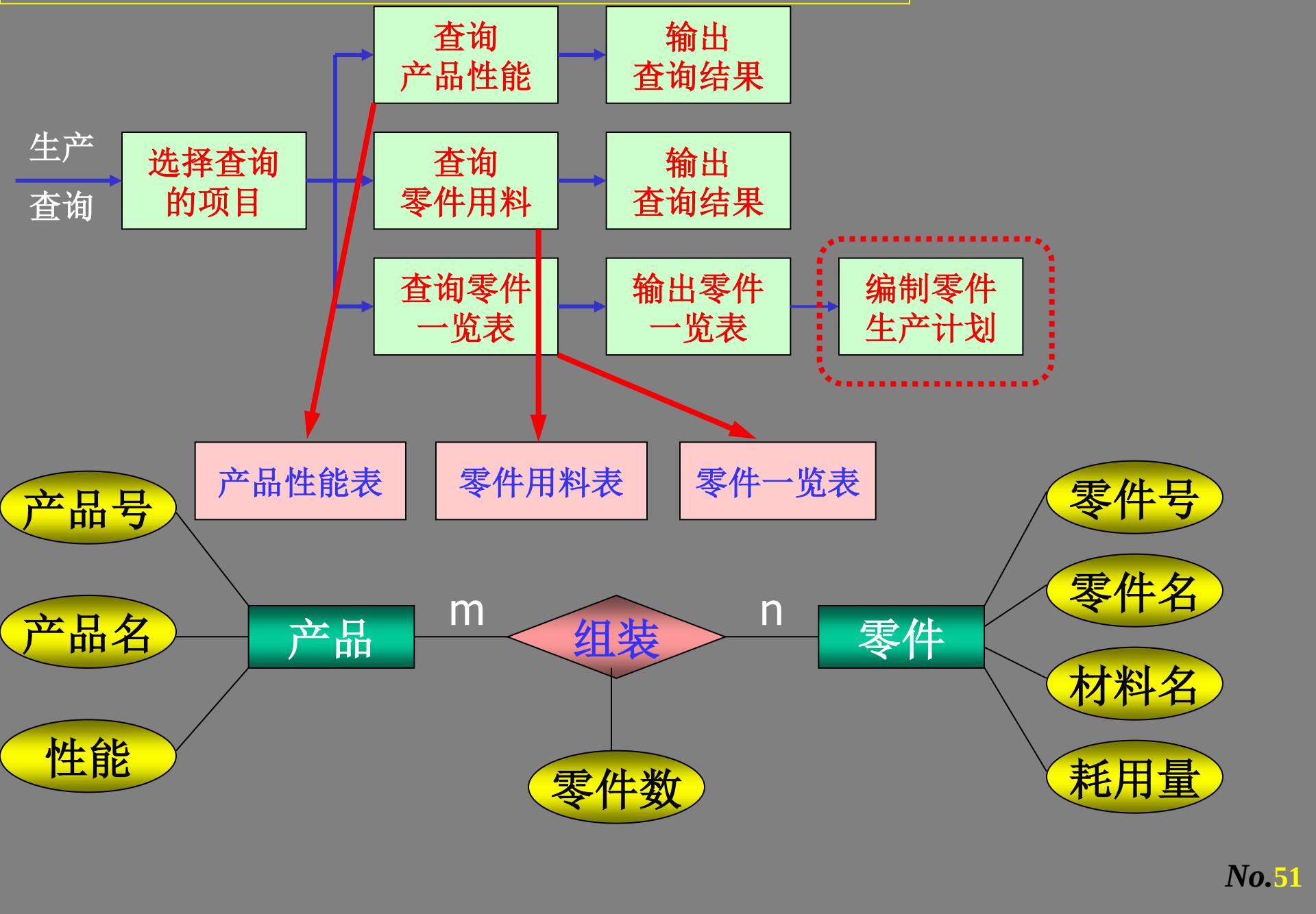
E-R图实例


班编号
专业系
实体集
班级
学员
1
联系
组成
n
学号
姓名
性别
年龄
学生
m
选修
n
成绩
课程号
课程名
学分
课程
1
n
1
领导
m
讲授
n
属性
教师
参考书
职工号
姓名
性别
年龄
职称
书号
书名
简介
价格

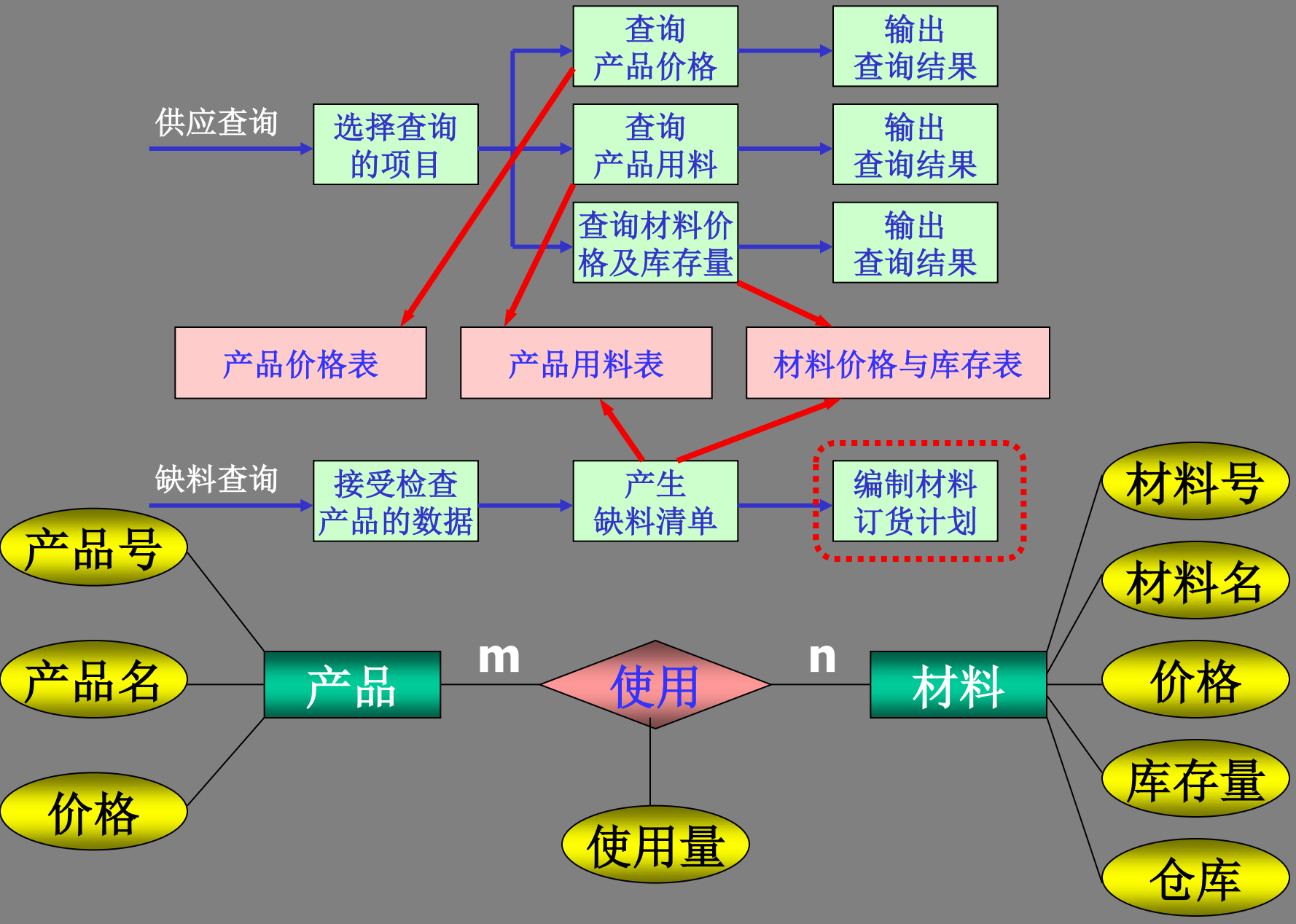
属性不可再分数据项



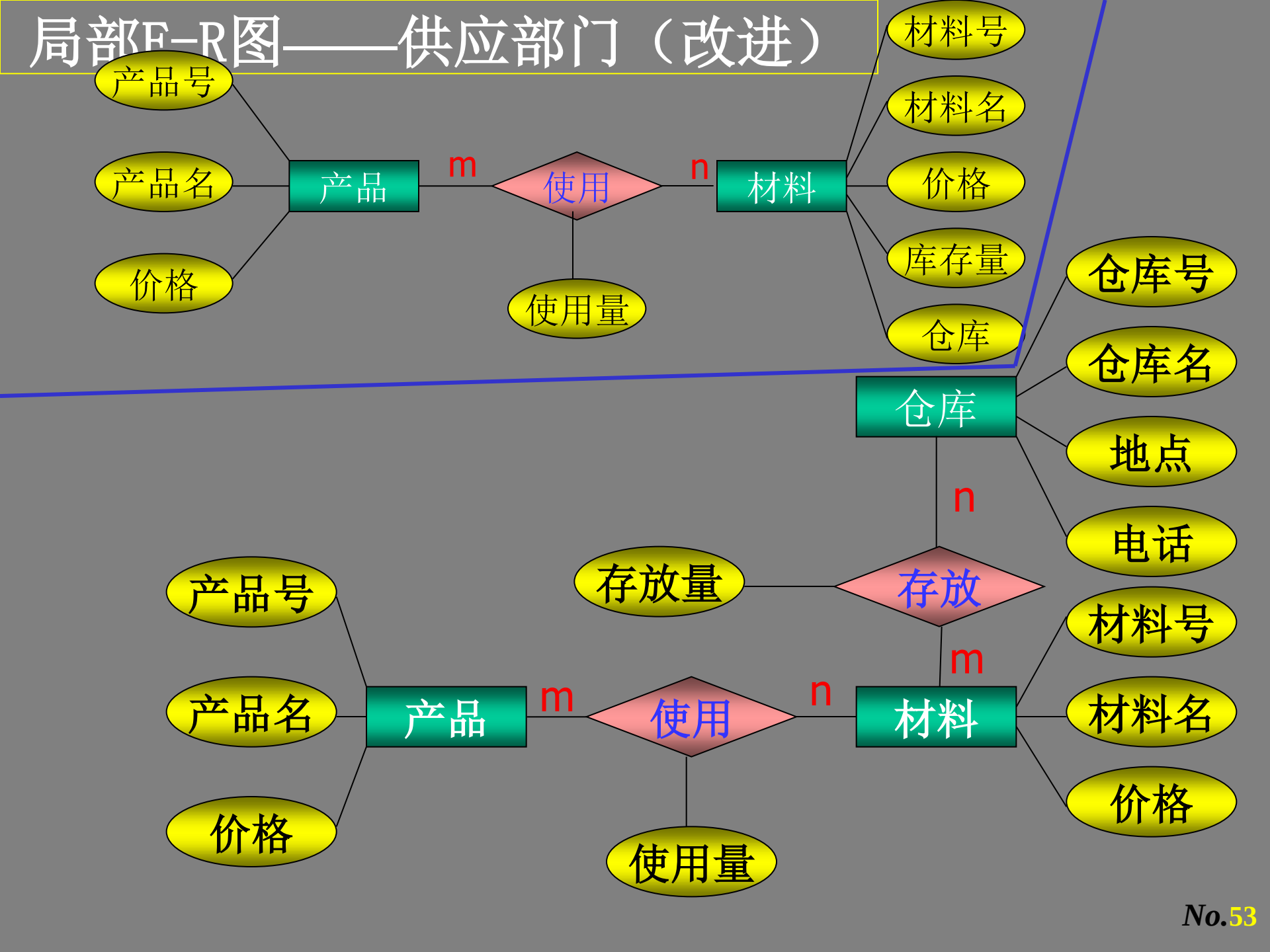
局部E-R图——生产部门

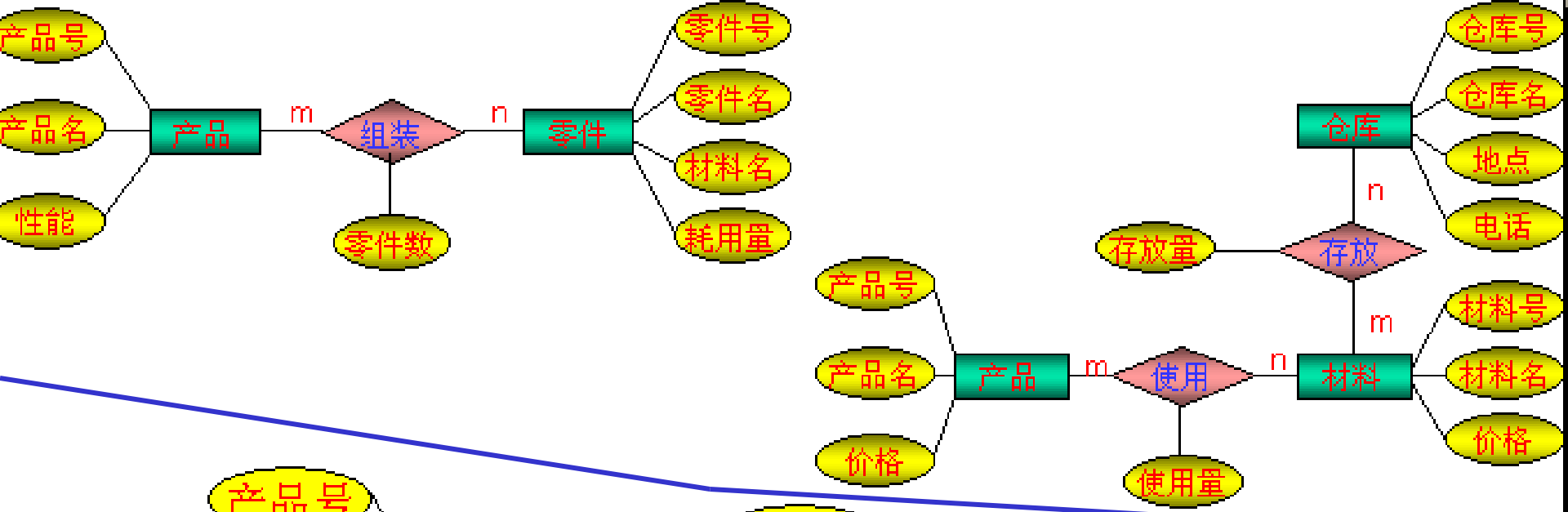

生产
查询
选择查询的项目
查询产品性能
查询零件用料
查询零件一览表
输出查询结果
输出查询结果
输出零件一览表
编制零件生产计划
产品性能表
零件用料表
零件一览表
产品号
产品名
性能
产品
m
组装
n
零件
零件数
零件号
零件名
材料名
耗用量

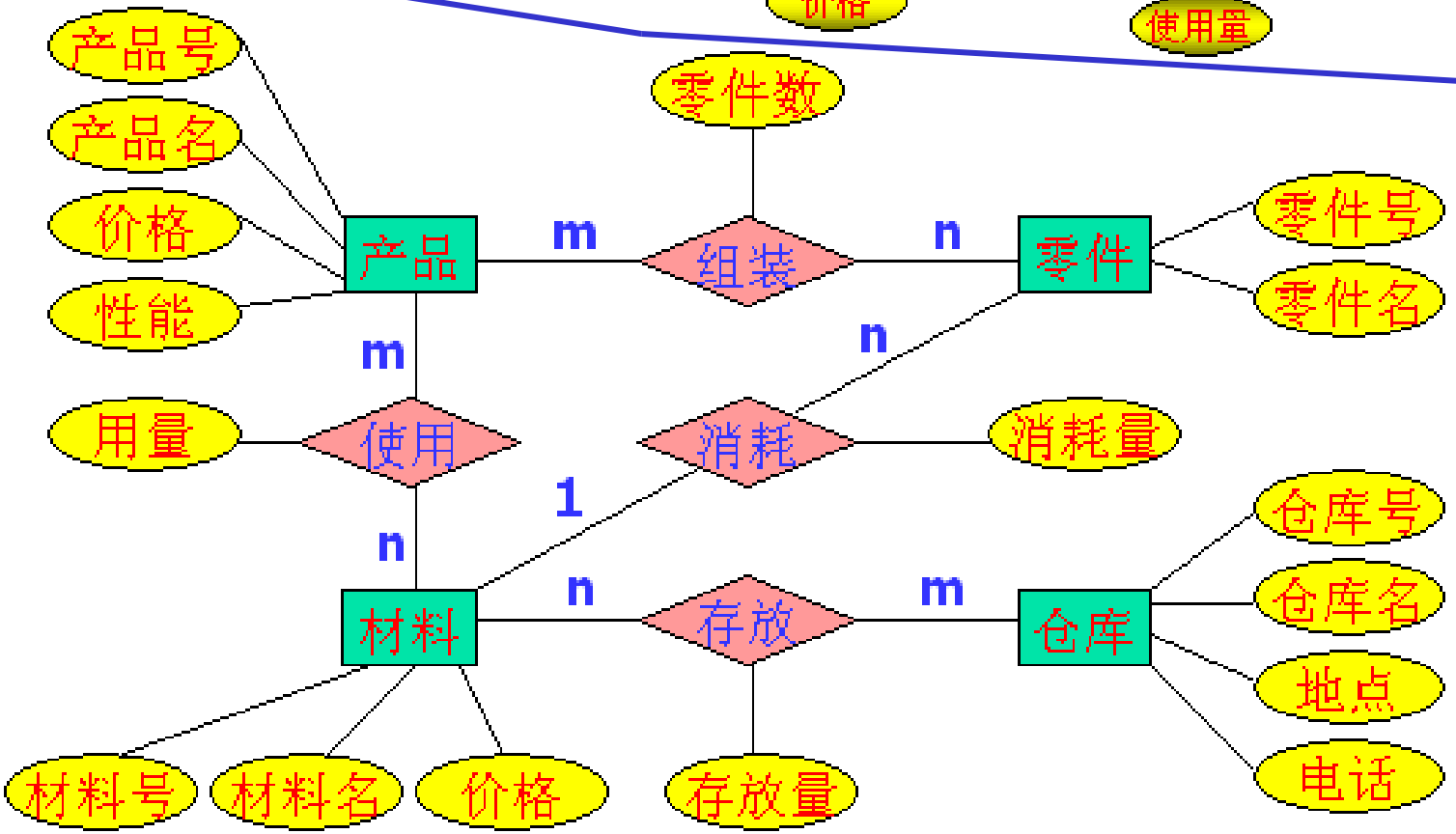
局部E-R图——供应部门


查询
产品价格
输出
查询结果
供应查询
选择查询
的项目
查询
产品用料
输出
查询结果
查询材料价
格及库存量
输出
查询结果
产品价格表
产品用料表
材料价格与库存表
缺料查询
接受检查
产品的数据
产生
缺料清单
编制材料
订货计划
产品号
产品名
价格
产品
m
使用
n
材料
材料号
材料名
价格
库存量
仓库
使用量

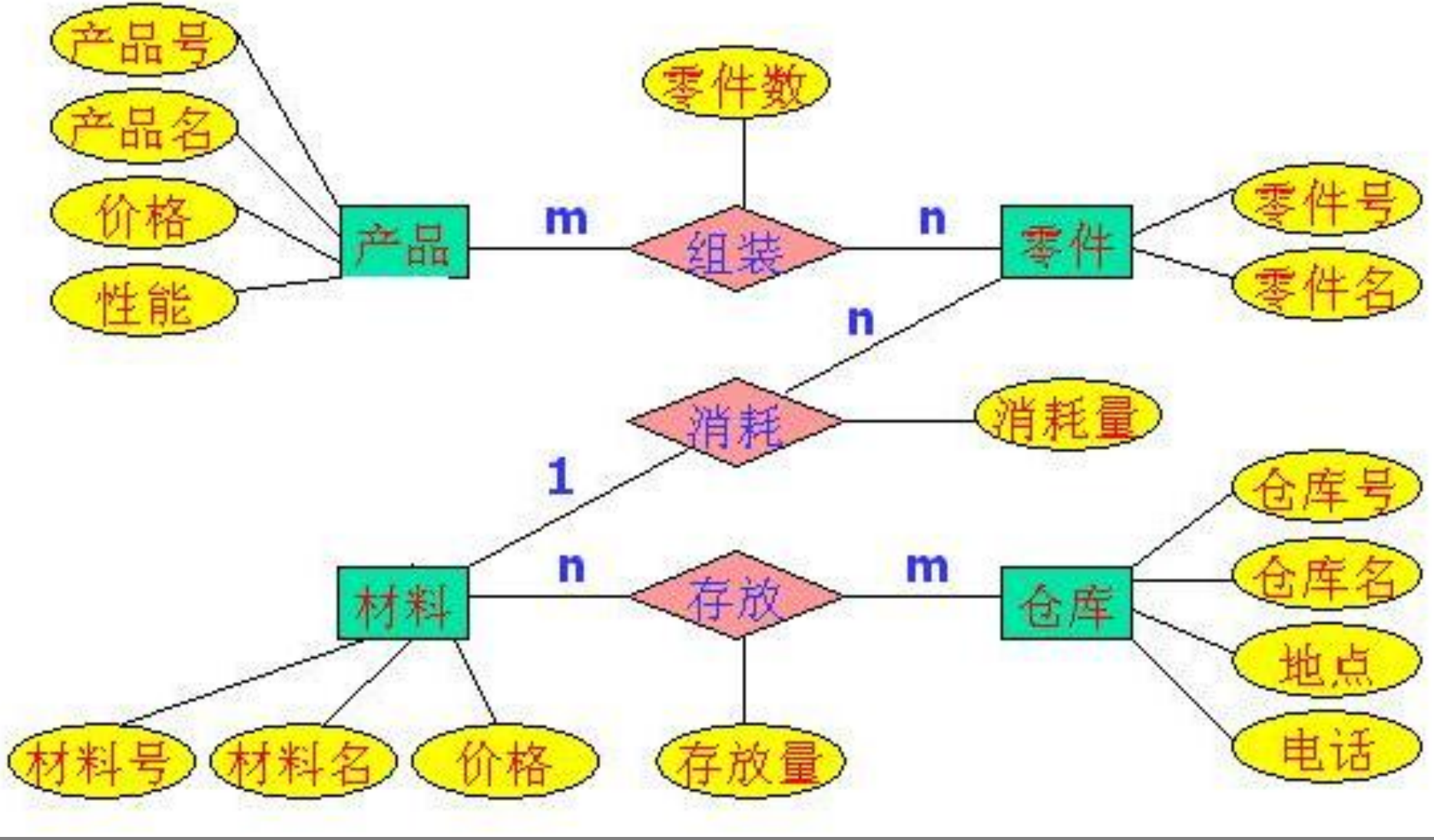
局部E-R图——供应部门（改进）




产品号
产品名
性能
产品
m
组装
n
零件
零件号
零件名
材料名
耗用量
零件数
仓库号
仓库名
地点
电话
仓库
n
存放量
存放
m
产品号
产品名
价格
产品
m
使用
n
材料
材料号
材料名
价格
使用量


产品号
产品名
价格
性能
产品
m
组装
n
零件
零件数
零件号
零件名
m
用量
使用
n
消耗
消耗量
1
n
材料
n
存放
m
仓库
仓库号
仓库名
地点
电话
材料号
材料名
价格
存放量

总体E-R图（改进）



消除冗余的联系——使用

6.4 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DBMS产品所支持的数据模型相符合的逻辑结构。

理论上讲，设计逻辑结构应该选择最适于相应概念结构的数据模型，然后对支持这种数据模型的各种DBMS进行比较，从中选择较合适的DBMS。但是，在实际开发过程中，往往是给定了DBMS，所以设计逻辑结构时一般要分成以下三步来进行。

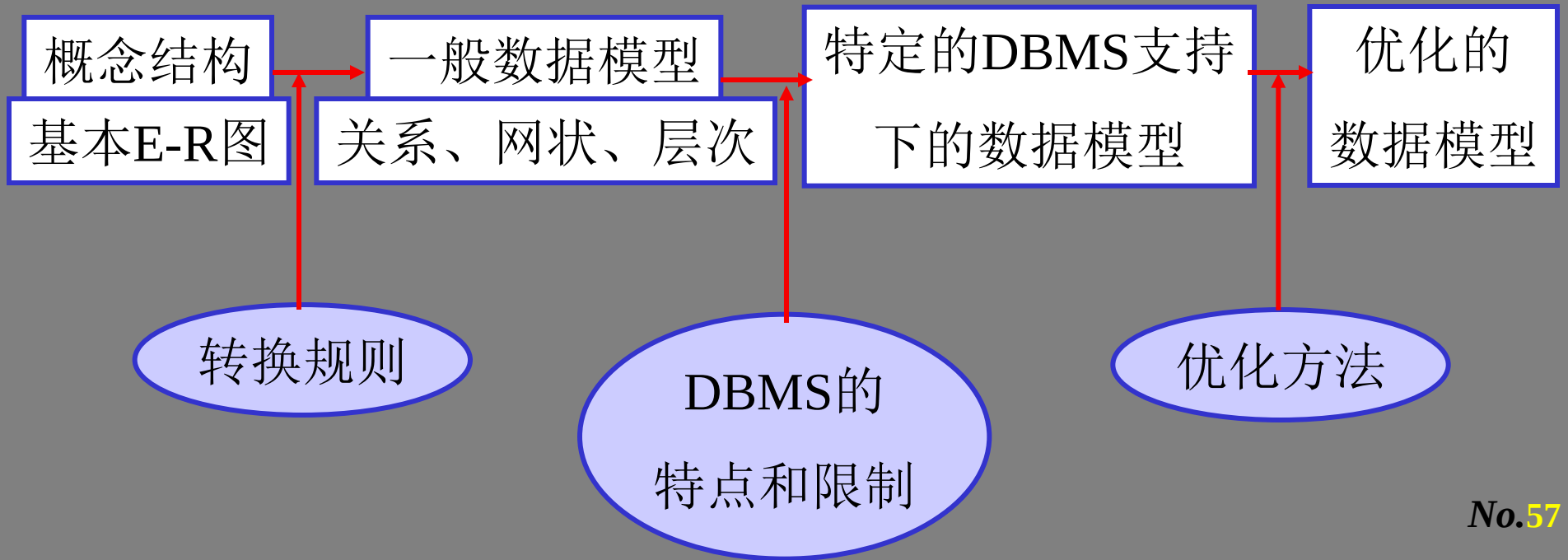
逻辑结构设计的三个步骤

逻辑结构设计一般按照下列三个步骤来进行：

1、将概念结构转换为一般的关系、网状、层次模型；

2、将转换来的关系、网状、层次模型向特定DBMS支持下的数据模型转换；

3、对数据模型进行优化。



6.4.1 E-R图向关系模型的转换

E-R图向关系模型转换就是将实体、实体的属性和实体之间的联系转换为关系模式，一般转换原则如下：

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，实体的码就是关系的码。

2、一个1:1的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。

如果转换为一个独立的关系模式，则与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，每个实体的码均是该关系的候选码，

如果与某一端实体对应的关系模式合并，则需要在该关系模式的属性中加入另一个关系模式的码和联系本身的属性。

转换原则（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。如果转换为一个独立的关系模式，则与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，而关系的码为n端实体的码。

4、一个m:n联系转换为一个关系模式。与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，而关系的码为各实体码的组合。

5、三个或三个以上实体间的一个多元联系可以转换为一个关系模式。与该多元联系相连的各实体的码以及联系本身的属性均转换为关系的属性，而关系的码为各实体码的组合。

6、具有相同码的关系模式可以合并。

形成了一般的数据模型后，下一步就是向特定的RDBMS的模型转换。设计人员必须熟悉所用RDBMS的功能与限制。

6.4.2 数据模型的优化

数据库逻辑设计的结果不是唯一的。需要进行数据模型优化。优化通常以规范化理论为指导，方法如下：

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、按照数据依赖的理论对关系模式逐一进行分析，考察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处理要求，分析这些模式对于这样的应用环境是否合适，确定是否要对某些模式进行合并或分解。

数据模型的优化（续）

NOTE: 连接运算的代价是非常高的，可以说关系模型低效的主要原因就是连接运算引起的。需要权衡响应时间和潜在问题两者的利弊决定。

5、对关系模式进行必要的分解，提高数据操作的效率和存储空间的利用率。常用的两种分解方法是水平分解和垂直分解。

水平分解是把基本关系的元组分为若干个子集合。定义每个子集合为一个子关系，以提高系统的效率。

垂直分解是把关系模式的属性分解为若干个子集合，形成若干个字段关系模式。垂直分解的原则是：经常在一起使用的属性从原关系模式中分解出来形成一个子关系模式。

6.4.3 设计用户子模式

将概念模型转换为全局逻辑模型后，还应该根据局部应用需求，结合具体DBMS的特点，设计用户外模式。设计人员可以利用“视图”来设计更符合局部用户需要的用户外模式。

定义数据库全局模式主要是从系统的时间效率、空间效率、易维护等角度出发的。由于用户外模式与模式是相对独立的，因此

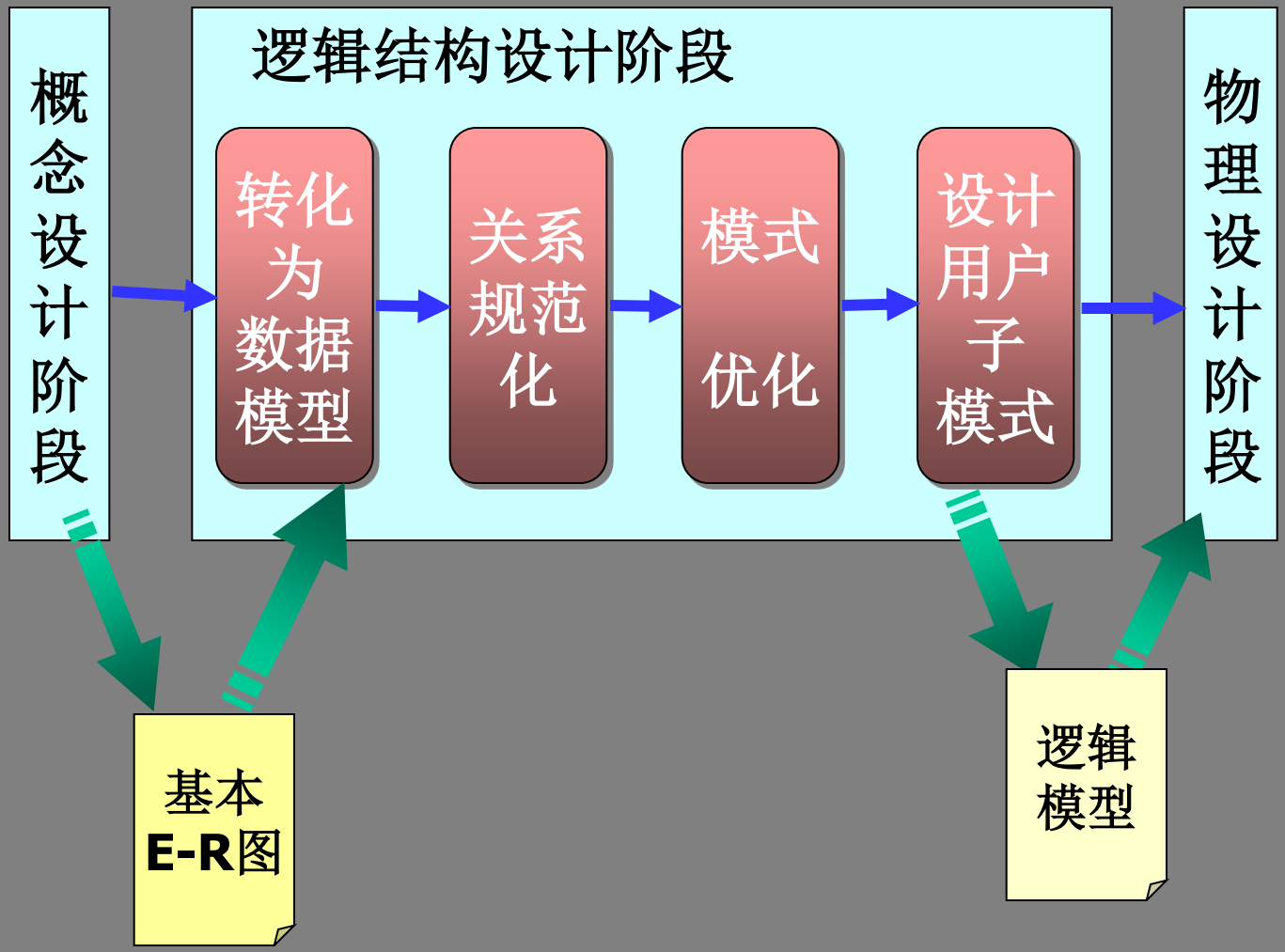
可以在定义用户外模式时注重考虑用户的习惯和方便。这主要包括：

1、使用更符合用户习惯的别名。

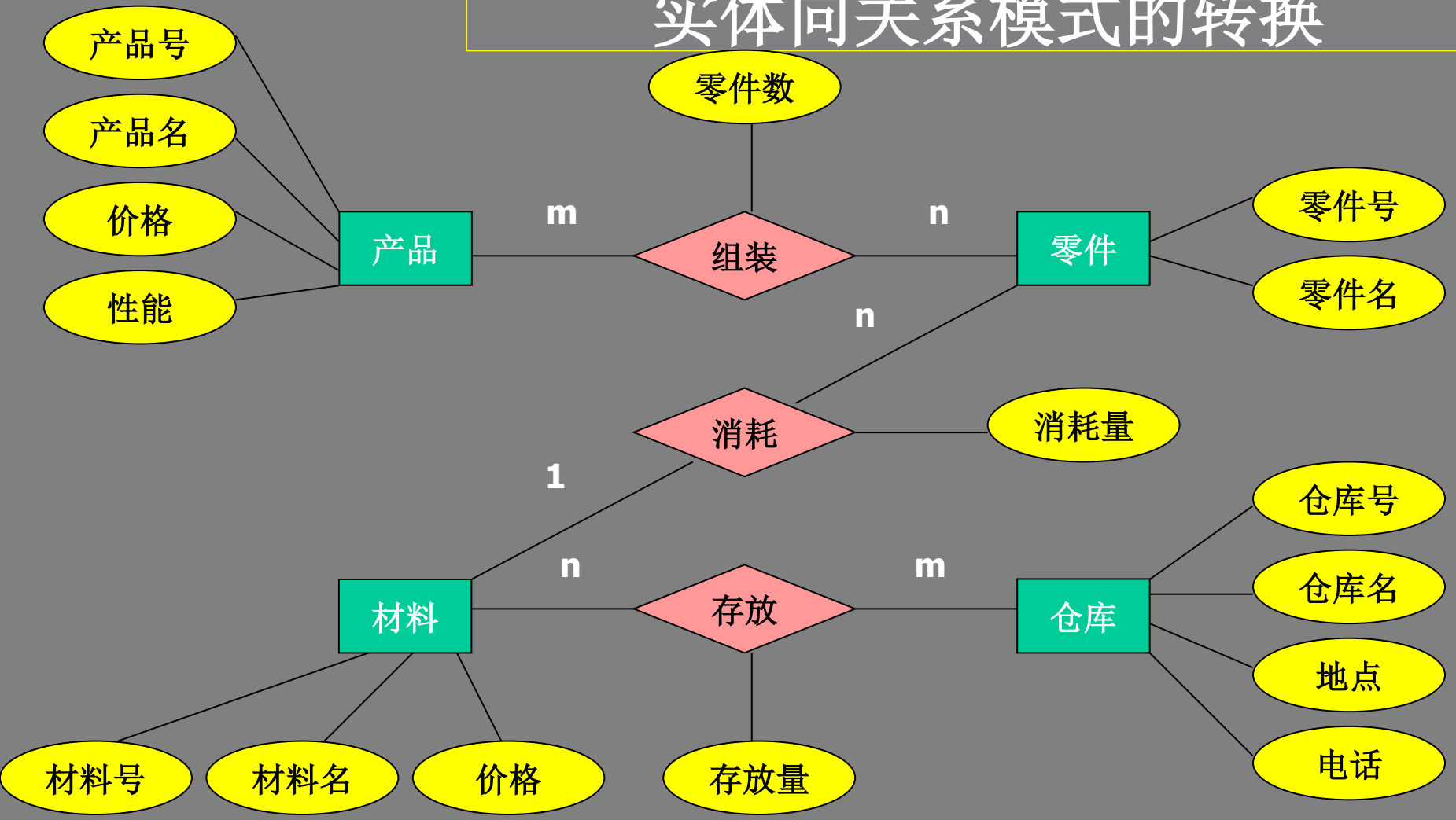
2、可以对不同级别的用户定义不同的视图，以保证系统的安全性。

3、简化用户对系统的使用。

逻辑结构设计阶段小结


概念设计阶段
逻辑结构设计阶段
转化为数据模型
关系规范化
模式优化
设计用户子模式
物理设计阶段
基本E-R图
逻辑模型

实体向关系模式的转换

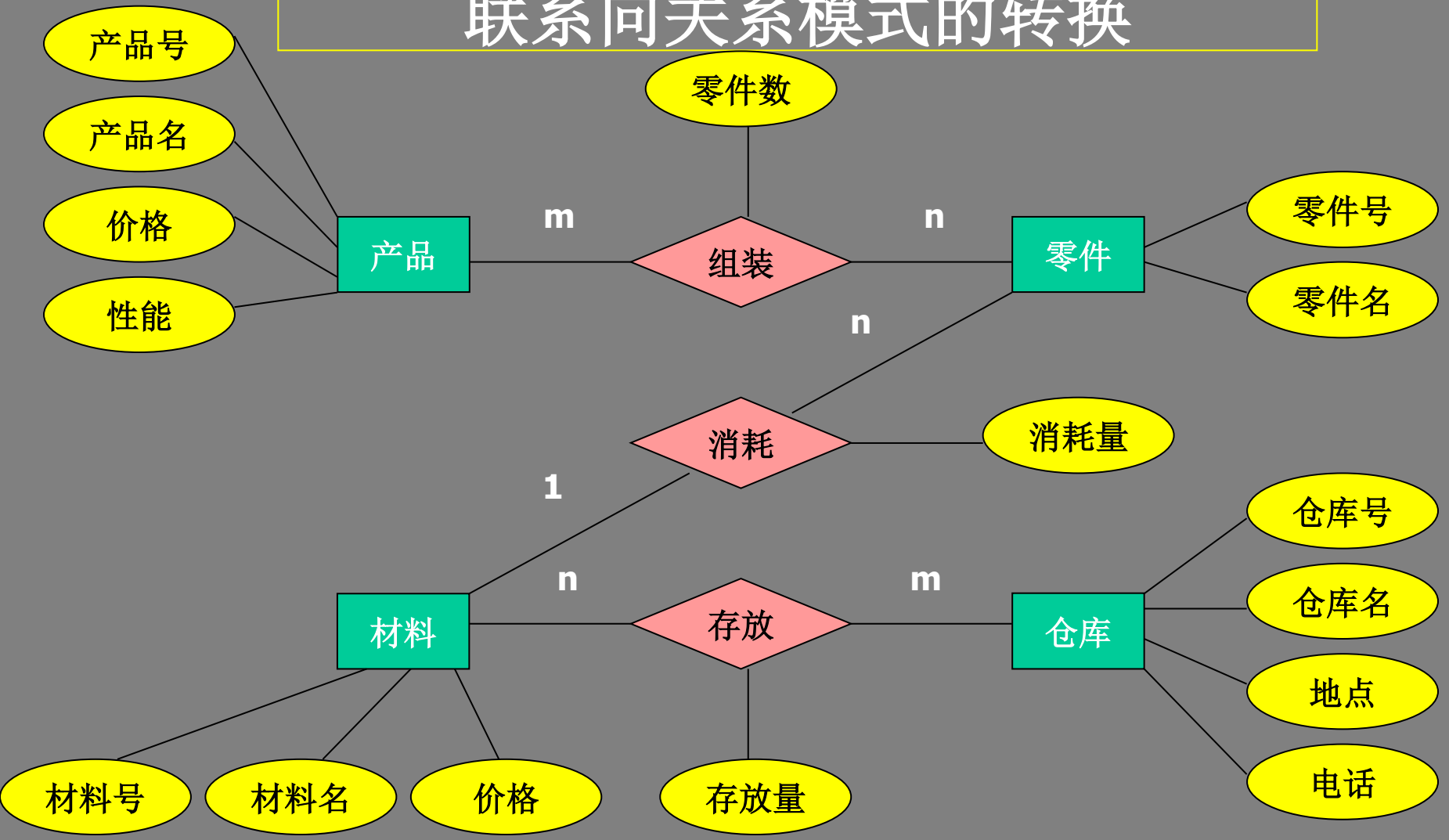


产品(<u>产品号</u>, 产品名, 价格, 性能)

材料(<u>材料号</u>, 材料名, 价格)

仓库(<u>仓库号</u>, 仓库名, 地点, 电话)

联系向关系模式的转换

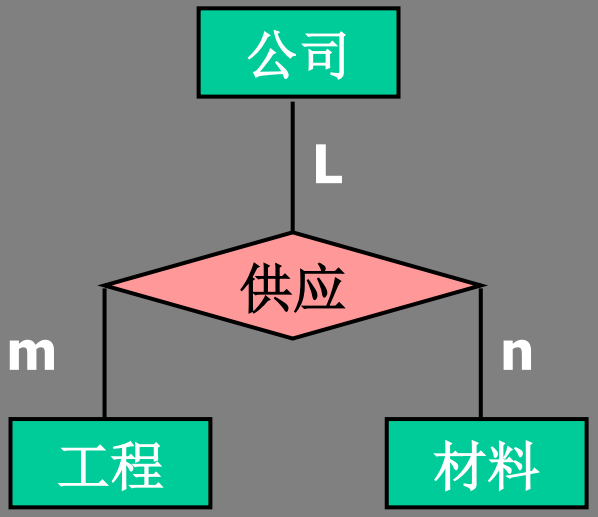


零件（零件号, 零件名, 材料号， 消耗量）

产品零件一览表(产品号, 零件号, 零件数量)

材料存放表(材料号, 仓库号, 存放量)

存在于三个实体间的联系



公司名	工程号	材料名
华都	132	钢管
华兴	215	铝板
向阳	132	水泥
华都	730	水泥

供应(公司名, 工程号, 材料名)

6.5 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要求的物理结构的过程，这就是数据库的物理设计。

数据库的物理设计通常分为两步：

1、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；

2、评价物理结构，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。

6.5.1 数据库的物理设计的内容和方法

由于不同的数据库产品所提供的物理环境、存取方法和存储结构有很大差别，因此没有通用的物理设计方法，只能给出一般的设计内容和原则。

要设计优化的物理数据库结构，使得数据库上运行的各种事务响应时间小、存储空间利用率高、事务吞吐率大，必须做到以下两点：

1、对要允许的事务进行详细分析，获得选择物理数据库设计所需要的参数；

2、要充分了解所选择的RDBMS的内部特征，特别是系统提供的存取方法和存储结构。

数据库物理设计过程中所需的信息

查询事务需要信息：

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；

更新事务需要信息：

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；

除此之外，还需要知道**每个事务在各关系上运行的频率和性能要求。**

6.5.2 关系模式存取方法的选择

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。物理设计的任务就是要确定选择哪些存取方法，建立哪些存取路径。存取方法是快速存取数据库中数据的技术。

数据管理系统一般都提供多种存取方法。常用的存取方法有以下三类：

索引方法——B+树索引方法；

聚簇方法；

Hash方法；

B+树索引方法是数据库中经典的存取方法，使用最为普遍。

索引存取方法的选择

选择索引存取方法：

根据应用要求确定对关系的哪些属性列建立索引、哪些属性列建立组合索引、哪些索引要设计为唯一索引等，建立原则如下：

1、若一个（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（组合索引）；

2、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；

3、若一个（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；

注意：关系上定义的索引数并不是越多越好，因为系统维护索引是要付出代价的。查找索引也要付出代价。

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成为聚簇。

选择聚簇存取方法，即确定需要建立多少个聚簇，每个聚簇中包括哪些关系。

聚簇功能可以大大地提高按照聚簇码进行查询的效率。它不但适用于单个关系，也适用于经常进行连接操作的多个关系。即把多个连接关系的元组按连接属性值聚集存放，聚簇中的连接属性称为聚簇码。一个数据库可以建立多个聚簇，一个关系只能加入一个聚簇。

建立聚簇的方法

一、先设计候选聚簇，原则如下：

（1）经常在一起进行连接操作的关系可以建立聚簇；

（2）若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族；

（3）若一个关系的一个（或一组）属性上的值重复率很高，则该关系可建立聚族。即对应每个聚族码值的平均元组数不是太少，如果太少，则聚族效果不明显。

二、检查候选聚簇中的关系，取消其中不必要的关系：

（1）从聚簇中删除经常进行全表扫描的关系；

（2）从聚簇中删除更新操作远多于连接操作的关系；

（3）不同的聚簇中可能包含相同的关系，要从这多个聚簇方案中选择一个较优的。即在这个聚簇上运行各种事务的总代价最小。

建立聚簇时的注意事项

当通过聚簇码进行访问或连接是该关系的主要应用，与聚簇码无关的其他访问很少或者是次要的，这时可以考虑聚簇。

在**SQL**语句中包含有与聚簇码有关的**ORDER BY**，**GROUP BY**，**UNION**，**DISTINCT**等子句或者短语时，使用聚簇将特别有利，可以省去对结果集的排序操作。

Hash存取方法的选择

选择**Hash存取方法**的规则如下：

如果一个关系的**属性**主要出现在**等值连接**条件中或者主要出现在**相等比较选择条件**中，而且满足下列两个条件之一，则此关系可以选择**Hash**存取方法：

（1）若一个关系的大小可预知，而且不变；

（2）如果关系的大小动态改变，而且DBMS提供了动态Hash存取方法。

SQL Server 2000完全支持上面提到的三种存取方法。

6.5.3 确定数据库的存储结构

确定数据库物理结构主要指确定数据的存放位置和存储结构，包括确定关系、索引、聚簇、日志、备份等的存储安排和存储结构；确定系统配置等。

确定数据的存放位置和存储结构要综合考虑存取时间、存储空间利用率和维护代价三个方面的因素。这三个方面常常是相互矛盾的，因此需进行权衡之后，再选择一个折中方案。

1、确定数据的存放位置

为了提高系统性能，应根据应用情况将数据的易变部分与稳定部分、经常存取部分和存取频率较低部分分开存放。

2、 确定系统配置

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。初始情况下，系统为这些变量赋予了合理的缺省值。但这些值不一定适合每一种应用环境，因此，在物理设计阶段，需要重新对这些变量赋值，以改善系统的性能。

系统配置变量很多，如：同时使用数据库的用户数，同时打开数据库的对象数，内存分配参数，缓冲区分配参数，存储分配参数，物理块的大小，物理块的装填因子，时间片大小，数据库的大小，锁的个数等。

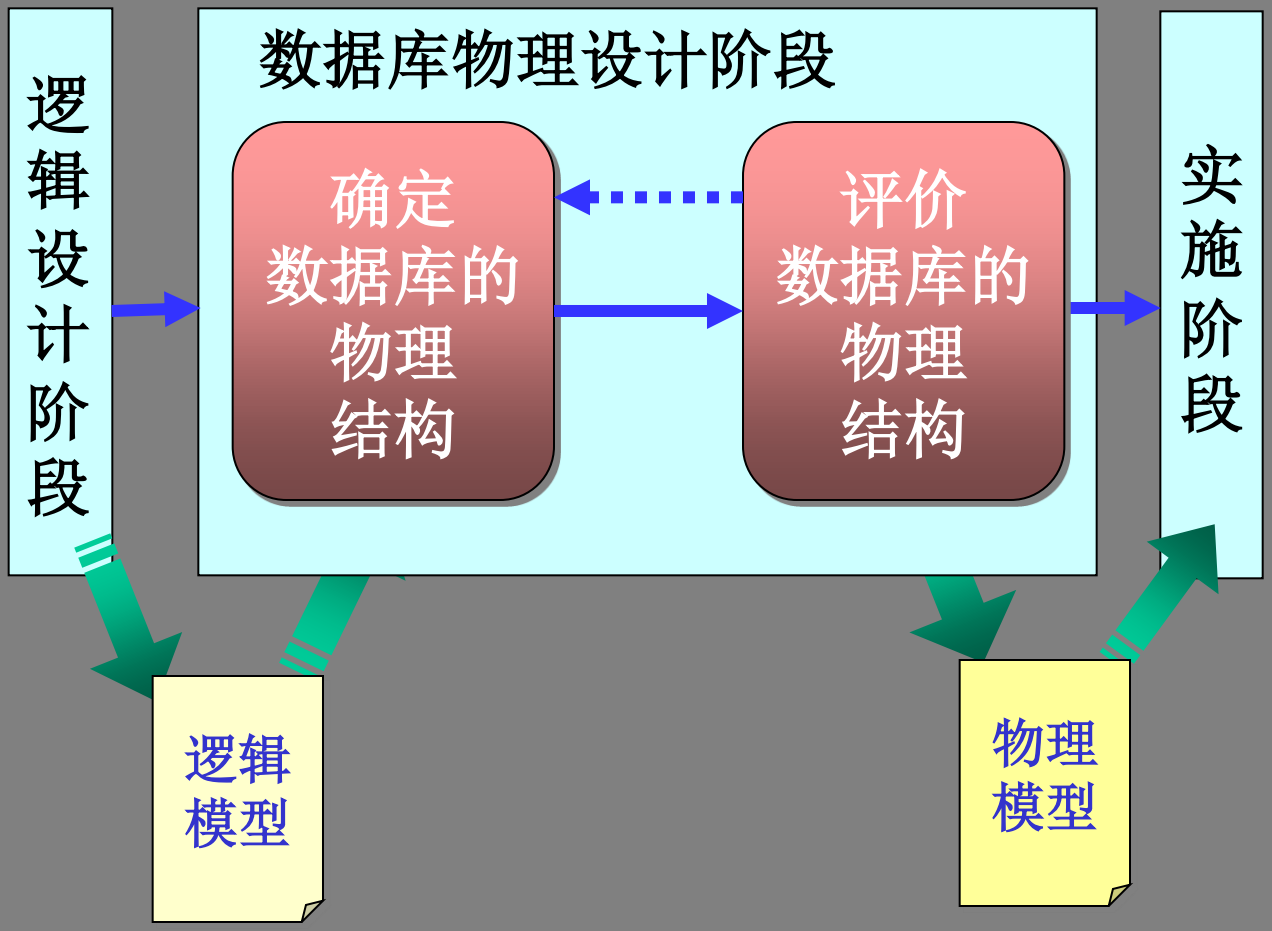
这些参数值影响存取值影响存取时间和存储空间的分配，在物理设计时就要根据应用环境确定这些参数值，以使系统性能最佳。

注意：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，以切实改进系统的性能。

6.5.4　评价物理结构

评价物理数据库的方法完全依赖于所选用的**DBMS**，主要是从定量估算各种方案的存储空间、存取时间和维护代价入手，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。

物理结构设计阶段小结



6.6 数据库的实施和维护

数据库实施阶段的主要任务是：

完成了数据库的物理设计之后，设计人员就要用RDBMS提供的数据定义语言和其他使用程序将数据库逻辑设计和物理设计结果严格描述出来，成为DBMS可以接受的源代码，再经过调试产生目标模式，然后组织数据入库。

6.6.1　数据的载入和应用程序的调试

数据库实施阶段包括两个部分：

数据的载入　和　应用程序的编码和调试

数据的载入：组织数据录入就是要将各类源数据从各个局部应用中抽取出来，输入计算机，再分类转换，最后综合成复合新设计的数据库结构的形式，输入数据库。

这个过程是相当费时的。由于各类数据存在的方式差别很大，所以DBMS一般不提供通用的数据转换工具。此时，需要设计人员自己设计数据录入子系统，由计算机来完成数据入库的任务。（SQL Server 2000提供了DTS工具，它可以将其他多种格式的数据转换到系统中）

当系统不提供数据转换工具时，设计人员需要认真的设计数据录入子系统，以防止不正确的数据入库，同时也方便用户的数据录入工作。

6.6.2 数据库的试运行

数据库的试运行：

是指当数据库中已有部分数据后，对数据库系统进行联合调试的过程。

在这一阶段中，必须实际运行数据库，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，则修改应用程序，直到达到设计要求为止。

在试运行阶段，还要实际测量和评价系统的性能指标，分析其是否达到设计目标。如果测试的结果与设计目标不符，则返回物理设计阶段，重新调整物理结构，修改系统参数，有时需要返回逻辑设计阶段，修改逻辑结构。

6.6.3 数据库的运行和维护

数据库试运行合格后，就可以进入到运行阶段了。由于应用环境和数据库运行过程中物理存储的不断变化，需要对数据库的设计进行评价、调整、修改等维护工作。这部分主要是由**DBA**来完成的。主要任务包括：

1、数据库的转储和恢复：

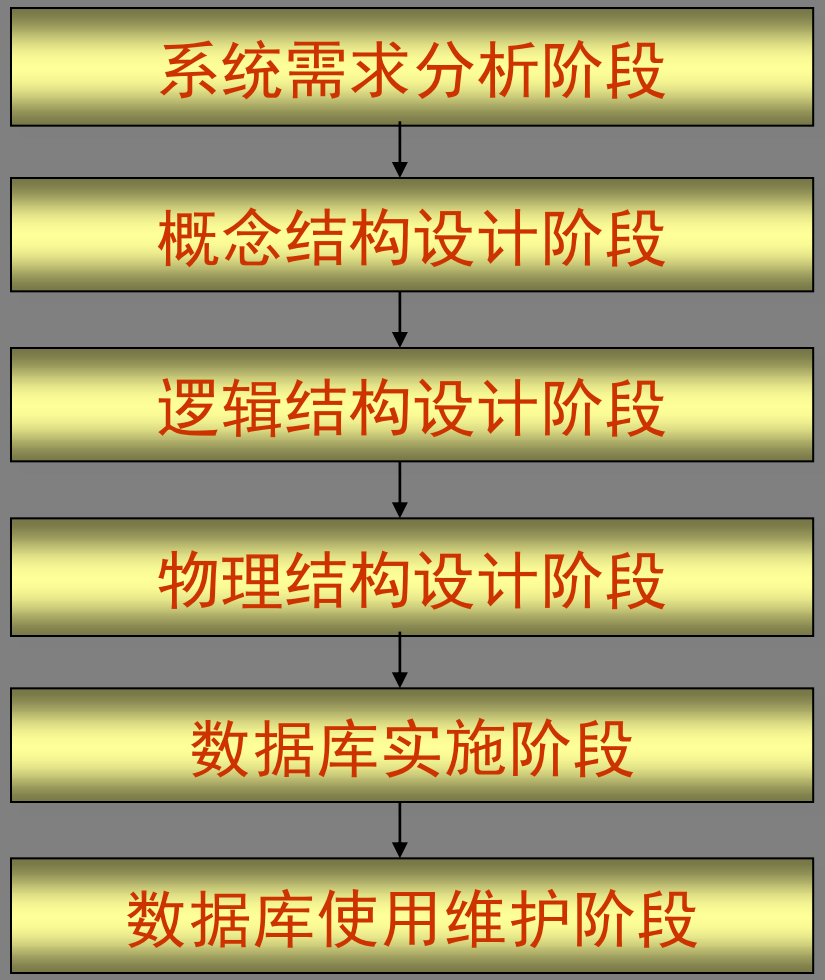
这是最重要的维护工作之一。

2、数据库的安全性、完整性控制；

3、数据库性能的监督、分析和改造；

4、数据库的重组和重构

6.7 小结



综合各个用户的应用需求

形成独立于**DBMS**的概念模型
用**E-R**图描述

将**E-R**图转换成具体关系模式
建立逻辑模型、用户视图

安排物理存储，设计索引

数据入库，编制应用程序

运行、维护数据库

Chapter 06 is over

Any

Question

？ ？ ？

Chapter 06 is over ！ ！

Thanks for my all friends ！ ！

第六章 数据库设计

6.1 数据库设计概述

6.2 需求分析

6.3 概念结构设计

6.4 逻辑结构设计

6.5 数据库的物理设计

6.6 数据库的实施与维护

6.7 小结