



7.3 概念结构设计

概念结构设计是信息世界的表述方式，即使用一种方法对现实世界进行抽象地描述，概念结构设计建立在需求分析阶段所得到的数据流图和数据字典的基础上，为计算机存储数据作准备。

本节内容：

7.3.1 概念结构设计的内容和要求

7.3.2 概念结构设计的策略与方法

7.3.3 设计局部视图

7.3.4 视图集成



7.3.1 概念结构设计的内容和要求

概念结构设计就是将现实事物以不依赖于任何数据模型的方式加以描述，目的在于以符号化的形式正确地反映现实事物及事物与事物间的联系。概念结构设计的内容就是建立概念模型。

对概念模型有如下要求：

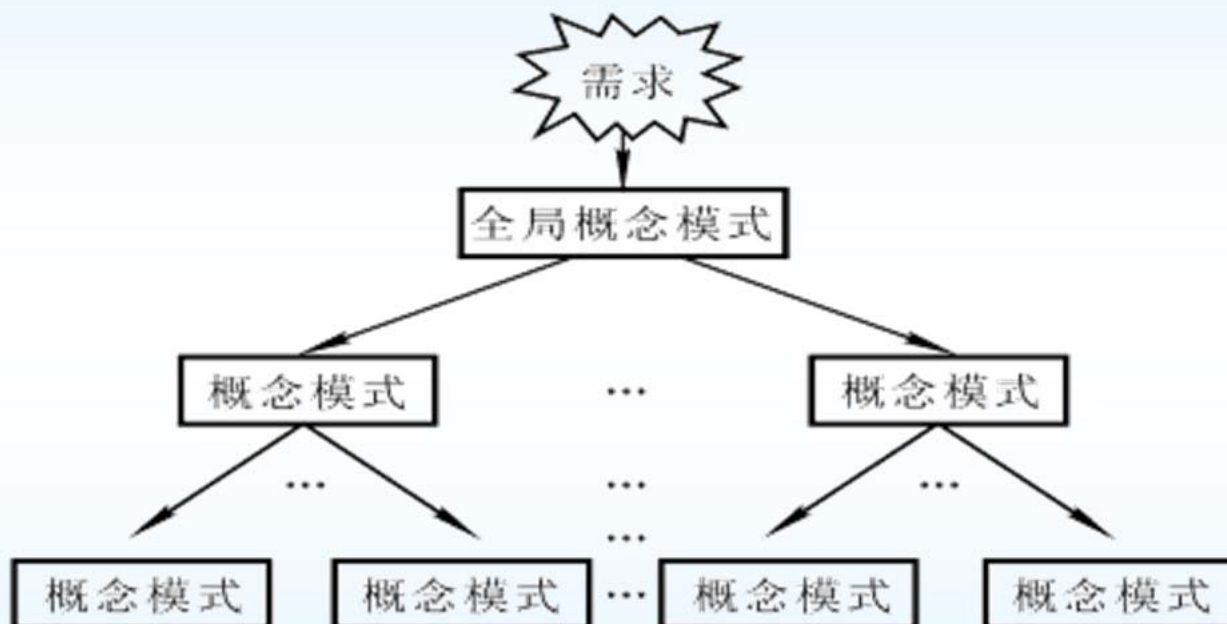
- (1) 概念模型是对现实世界的抽象和概括，它应真实、充分地反映现实世界中的事物和事物之间的联系；
 - (2) 概念模型应简洁、明晰、独立于机器、容易理解，方便数据库设计人员与用户交换意见，让用户能积极参与数据库的设计；
 - (3) 概念模型应易于变动。当应用环境和应用要求改变时，容易对概念模型修改和扩充；
 - (4) 概念模型应容易向关系、层次或网状数据模型转换。
- 描述概念模型的最常用方法是E-R图方法。



7.3.2 概念结构设计的策略与方法

👉 概念结构设计的四种策略：

(1) 自顶向下：即首先定义全局概念结构的框架，然后逐步细化。

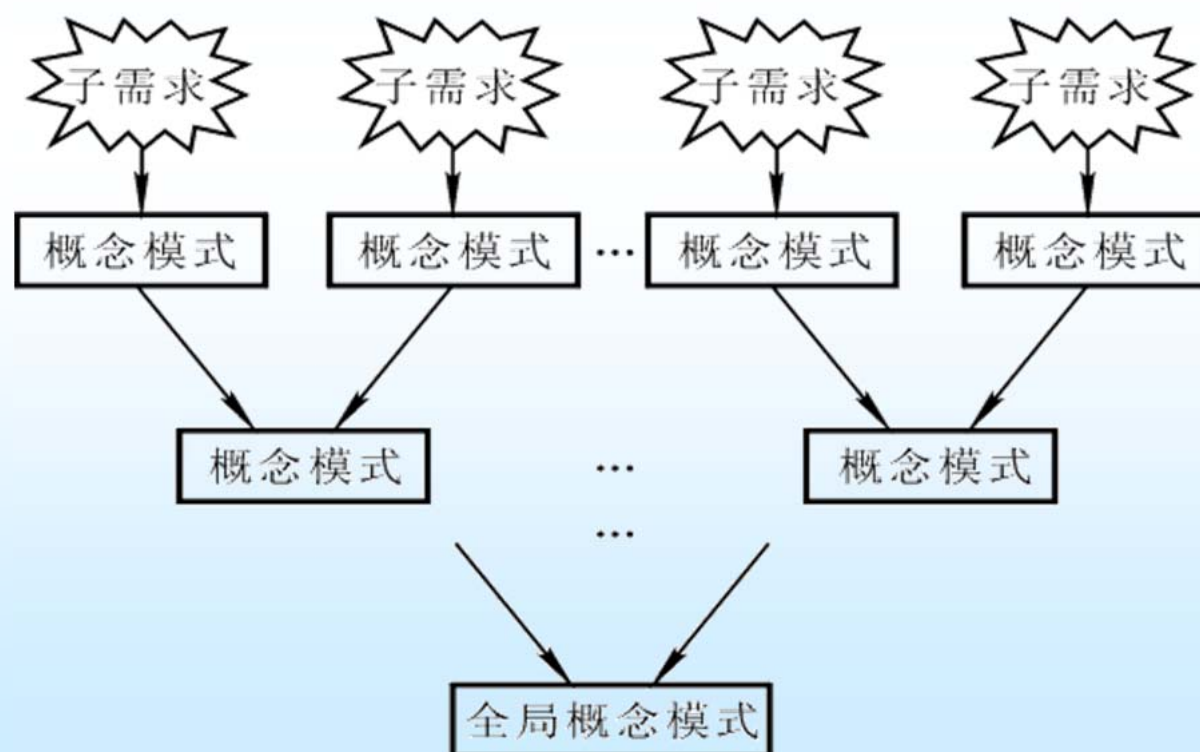




7.3.2 概念结构设计的策略与方法

👉 概念结构设计的四种策略：

- (1) 自顶向下：即首先定义全局概念结构的框架，然后逐步细化。
- (2) 自底向上：即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。

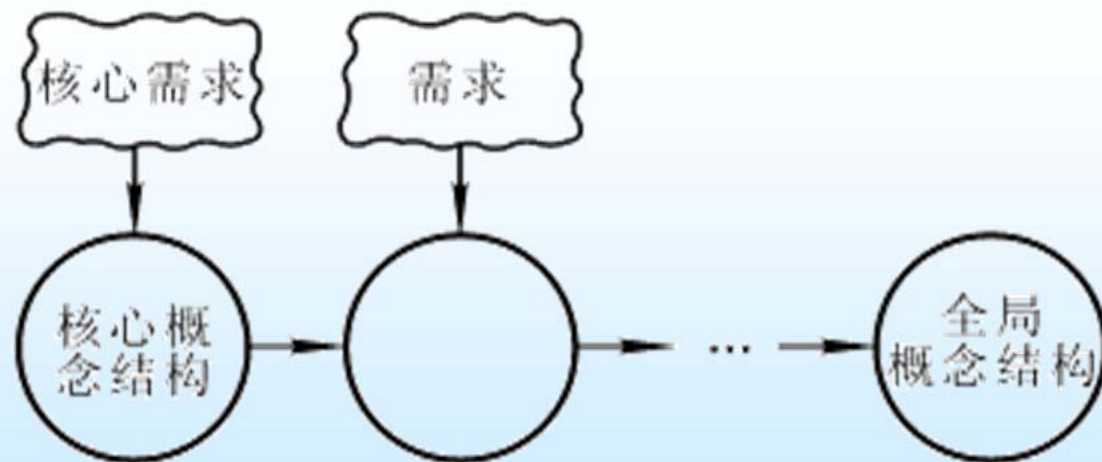




7.3.2 概念结构设计的策略与方法

☞ 概念结构设计的四种策略：

- (1) 自顶向下：即首先定义全局概念结构的框架，然后逐步细化。
- (2) 自底向上：即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。
- (3) **逐步扩张**：即首先确定核心业务的概念结构，然后以此为中心向外扩张，最终实现全局概念结构。





7.3.2 概念结构设计的策略与方法

👉 概念结构设计的四种策略：

- (1) 自顶向下：即首先定义全局概念结构的框架，然后逐步细化。
- (2) 自底向上：即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。
- (3) 逐步扩张：即首先确定核心业务的概念结构，然后以此为中向外扩张，最终实现全局概念结构。
- (4) **混合策略**：即将自顶向下和自底向上两种策略结合使用，首先确定全局框架，划分为若干个局部概念模型，再采取自底向上的策略实现各局部概念模型，加以合并实现全局概念模型。

实际应用中这些策略并没有严格的限定，可以根据具体业务的特点选择，如对于组织机构管理，因其固有的层次结构，可采用自顶向下的策略；对于已实现计算机管理的业务，通常可以此为核心，采取逐步扩张的策略。



7.3.2 概念结构设计的策略与方法

☞ 对现实事物抽象认识的方法分为以下三种：

- (1) **分类(Classification)**: 对现实世界的事物，按照其具有的共同特征和行为，定义一种类型。这在现实生活中很常见的，如学校中的学生和教师就属于不同的类型。在某一类型中，个体是类型的一个成员或实例，即“is member of”，如李娜是学生类型中的一个成员。
- (2) **聚集(Aggregation)**: 定义某一类型所具有的属性。如学生类型具有学号、姓名、性别、班级等共同属性，每一个学生都是这一类型中的个体，通过在这些属性上的不同取值来区分。各个属性是所属类型的一个成份，即“is part of”，如姓名是学生类型的一个成份。
- (3) **概括(Generalization)**: 由一种已知类型定义新的类型。如由学生类型定义研究生类型，在学生类型的属性上增加导师等其它属性就构成研究生类型。通常把已知类型称为超类(Superclass)，新定义的类型称为子类(Subclass)。子类是超类的一个子集，即“is subset of”，如研究生是学生的一个子集。



7.3.3 设计局部视图

👉 设计局部视图的步骤

(1) 选择局部应用

恰当选择各级数据流图中的某一层的某一部分业务，准备设计局部E-R图。通常就选择数据流图的中间级，这一级实体描述准确，实体间的联系也比较清晰，如需求分析阶段的采购业务流程图。

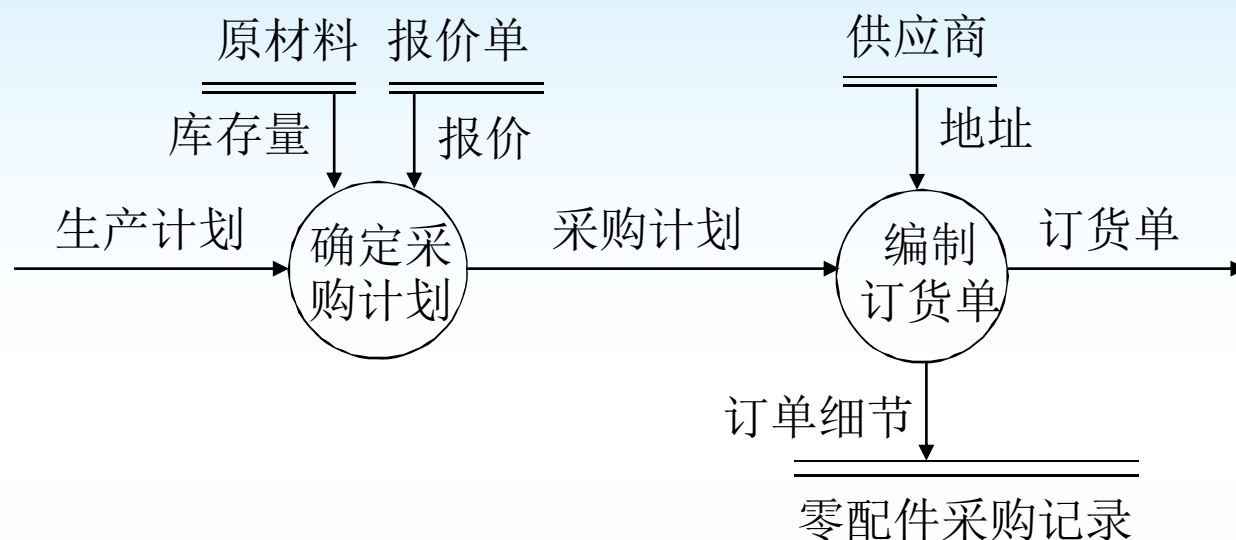
(2) 设计分E-R图

对于每一局部应用，其所用到的数据都应该收集在数据字典中了，依照该局部应用的数据流图，从数据字典中提取出数据，使用抽象机制，确定局部应用中的实体、实体的属性、实体标识符及实体间的联系及其类型。



7.3.3 设计局部视图

针对采购业务流程图和数据字典，可以确定如下数据结构：



生产计划(计划编号, 产品编号, 开始日期, 完成日期, 产量);

原材料(原材料编号, 名称);

库存(原材料编号, 库存量);

供应商(供应商名称, 地址, 电话, 邮编, 负责人)

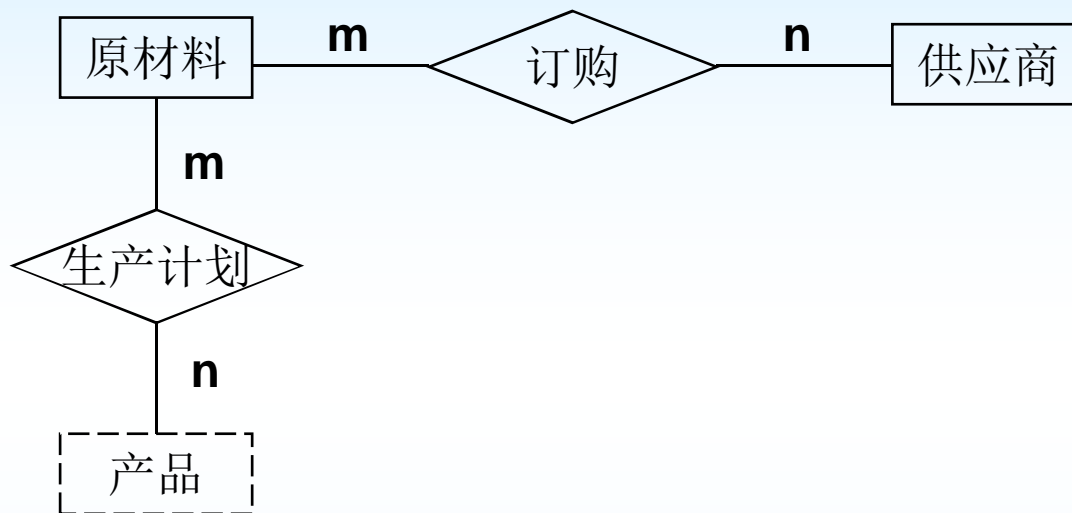
订货单(订货单编号, 供应商名称, 交付日期, 结算日期);

订单细节(订货单编号, 原材料编号, 名称, 单价, 数量)。



7.3.3 设计局部视图

设计如下E-R图：



注：对业务流程中的数据的概括需要经验，不同的人可能会得到不同结果。



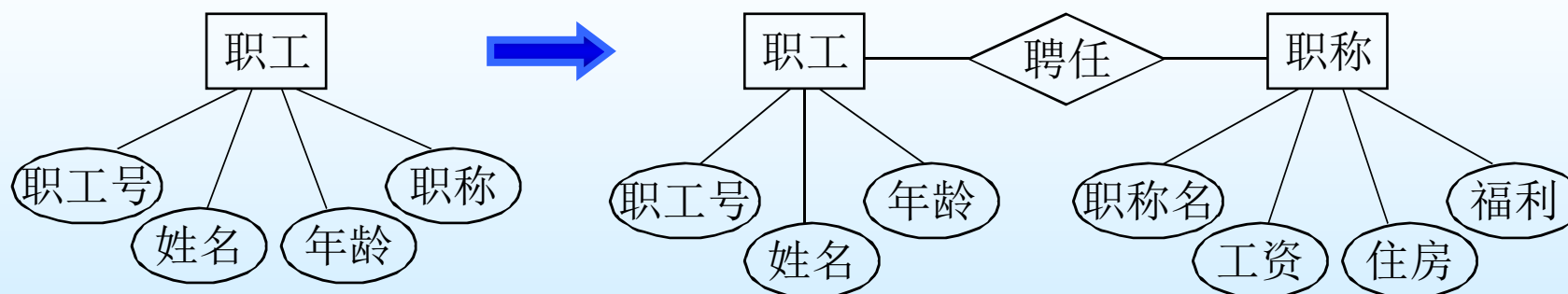
7.3.3 设计局部视图

现实世界中的实体与属性并没有严格的限定，即有些事物既可以作为实体，又可以作为属性。

☞ 确定实体与属性的两条准则：

- (1) 属性是不可再分的数据项，属性不可以再有属性；
- (2) 属性不能与其他实体发生联系，联系只能存在于实体与实体之间。

例：职工(职工号，姓名，年龄，职称)，其中的职称如果与工资、住房和福利挂钩(即有联系)，则应该单独作为实体，而职工与职称间构成联系。



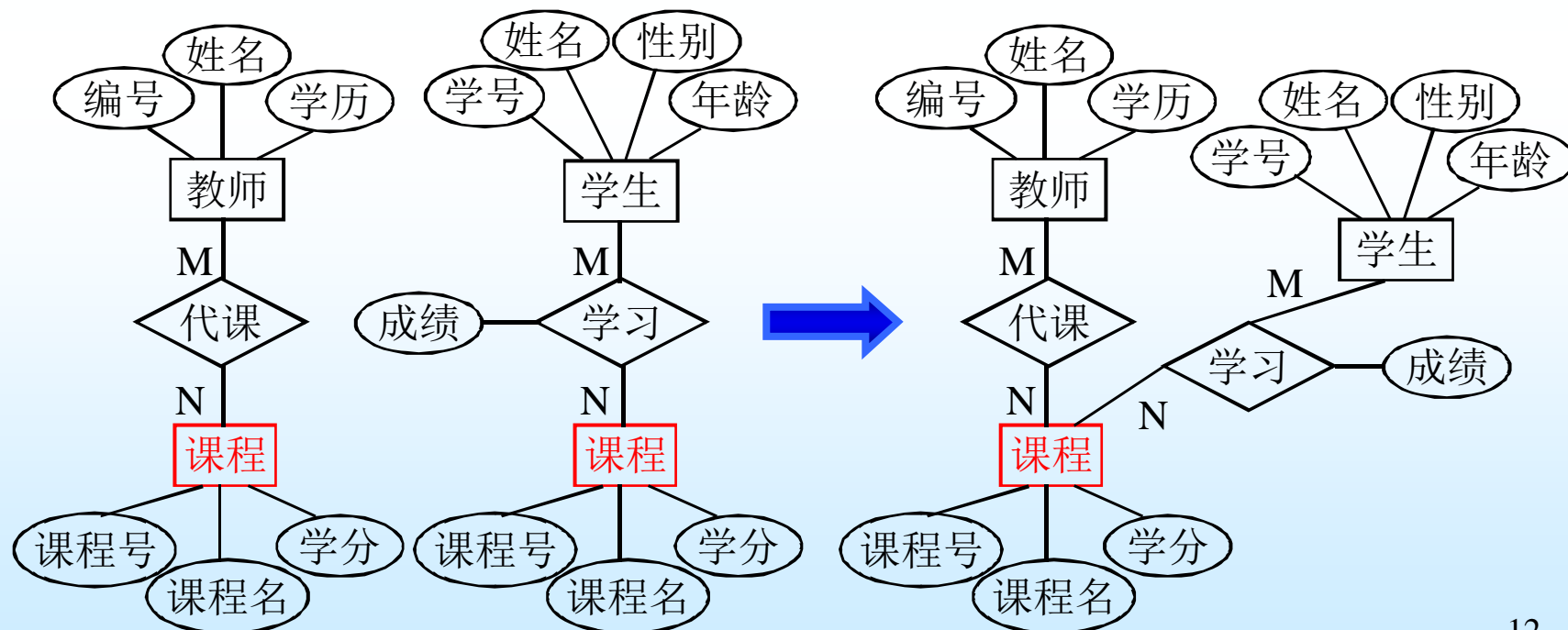


7.3.4 视图集成

局部的E-R图又称为局部视图，将多个局部E-R图合并成一张完整的E-R图的过程称为视图集成。

✧ 视图集成的方法：

选择两个具有相同实体的E-R图，通过相同实体将两个E-R图连结起来构成一个E-R图，然后再与其它E-R图连结直到将所有的局部E-R图全部连结成一个E-R图，称为全局E-R图。





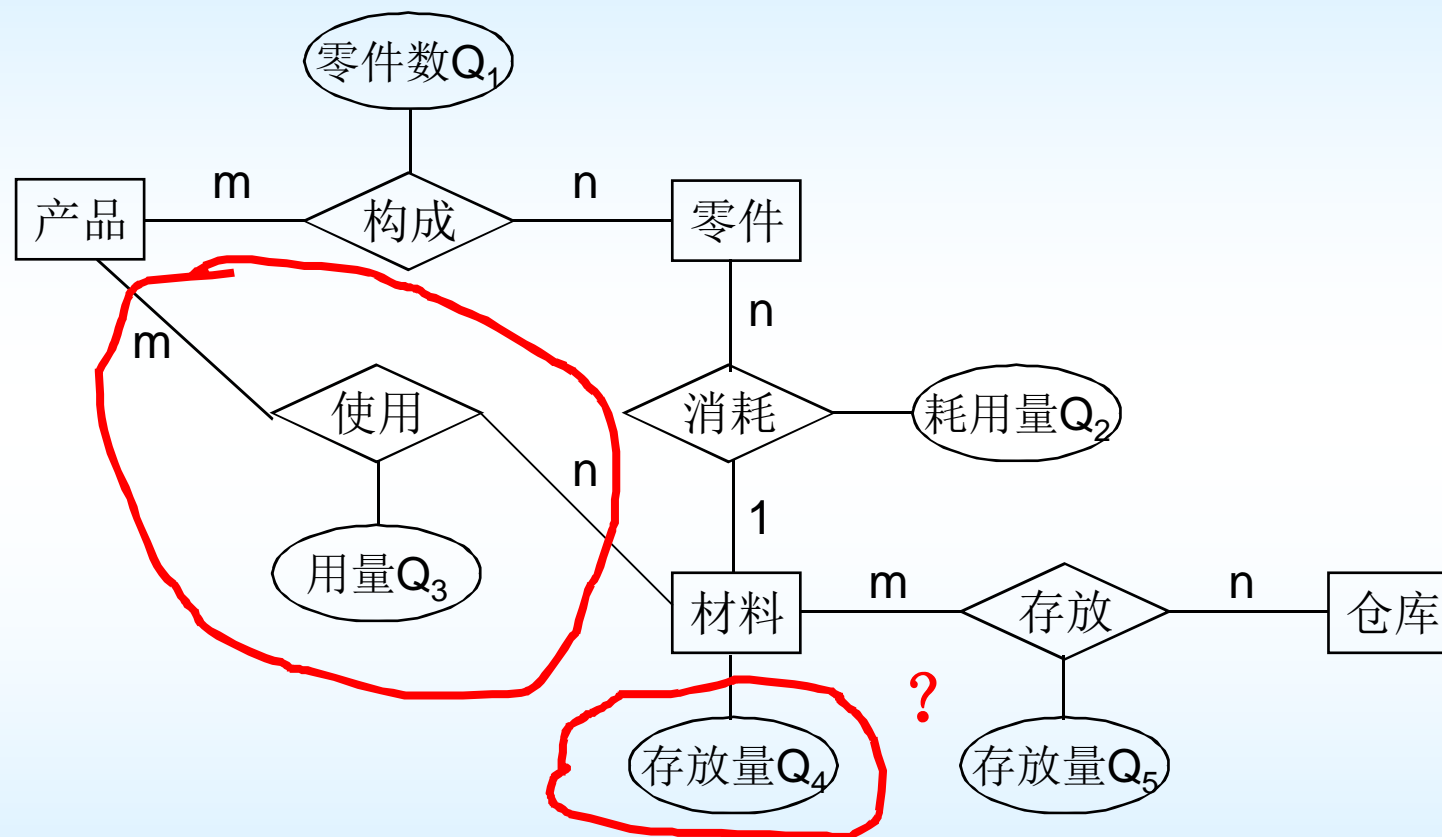
7.3.4 视图集成

- * **视图集成的作用：**一方面可以完整地描述企业的信息及其联系，另一方面在集成过程中可以解决冲突和消除冗余。
- * **分E-R图之间的三类冲突：**
 - (1) **属性冲突：**同一属性可能会存在于不同的分E-R图，由于设计人员不同或是出发点不同，对属性的类型、取值范围、数据单位等可能会不一致，这些属性对应的数据将来只能以一种形式在计算机中存储，这就需要在设计阶段进行统一。
 - (2) **命名冲突：**相同意义的属性，在不同的分E-R图上有着不同的命名，或是名称相同的属性在不同的分E-R图中代表着不同的意义，这些也要进行统一。
 - (3) **结构冲突：**同一实体在不同的分E-R图中有不同的属性，同一对象在某一分E-R图中被抽象为实体而在另一分E-R图中又被抽象为属性，需要统一。



7.3.4 视图集成

* 合并后的E-R图可能存在冗余:

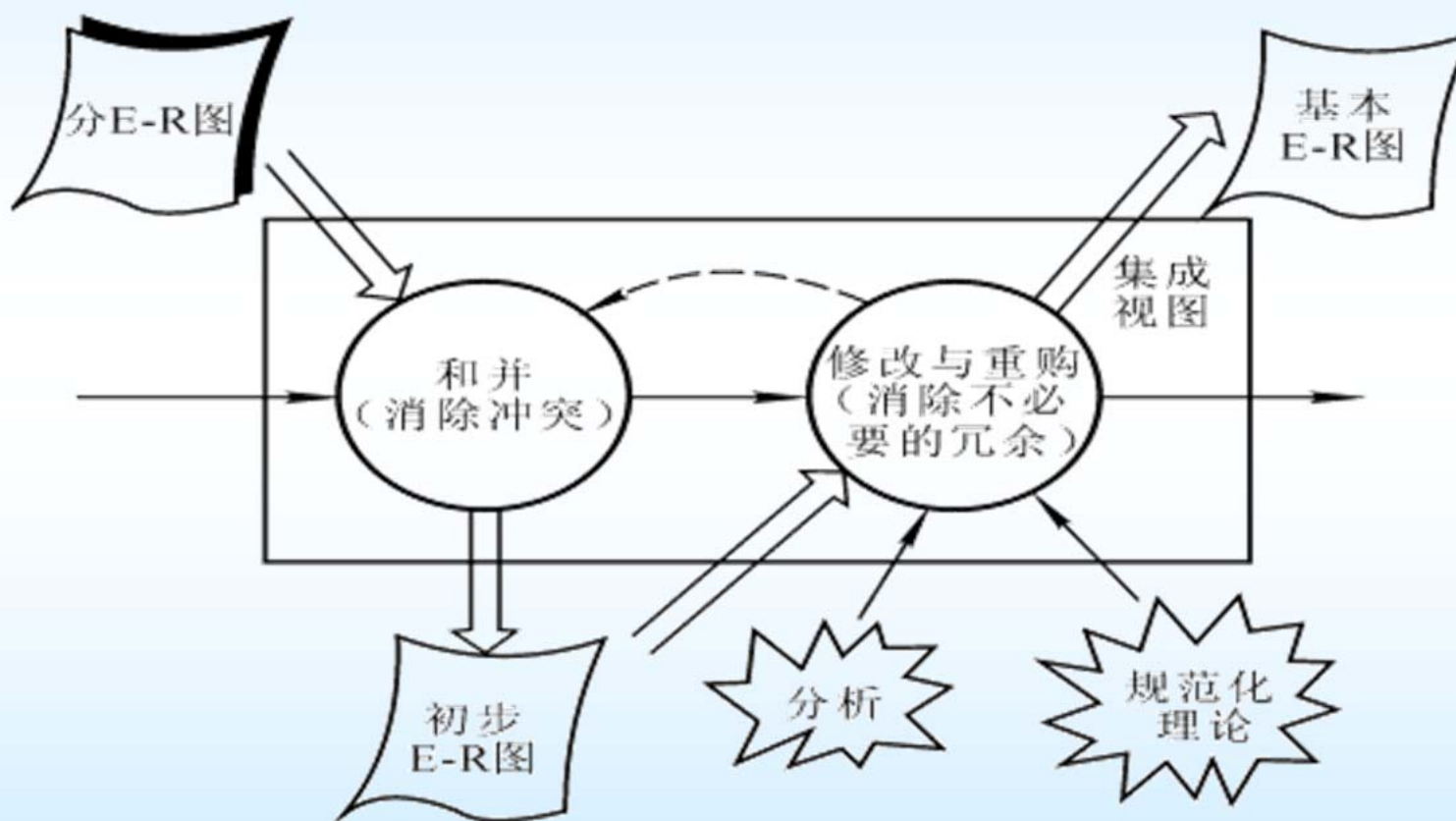


由于 Q_3 可以由 Q_2 和 Q_1 得到, 故 Q_3 多余, 同时“使用”联系也可以由“构成”和“消耗”传递表达, 故“使用”多余, 可以去掉该联系及其属性。另外, 实体中的派生属性也是冗余属性, 应去掉。



7.3.4 视图集成

视图集成过程示意图：





7.3 概念结构设计

概念结构设计总结：

- (1) 概念结构设计是对企业信息的描述，建立在需求分析的基础上；
- (2) 概念结构设计是对需求分析的数据(数据字典)依照业务处理流程(数据流图)抽象出事物之间的联系，以E-R图的形象加以描述；
- (3) 概念结构设计是独立于具体DBMS的，信息世界的范畴，即只考虑正确体现事实世界中的事物及联系的描述，而不考虑如何存储这些信息；
- (4) 概念结构设计所得到的E-R图应该得到用户的认可；
- (5) 进行概念结构设计的同时进行的是系统的总体设计。