第11章 数据库设计

北京理工大学 计算机学院 张文耀

zhwenyao@bit.edu.cn

主要内容

- 1 数据库设计方法
 - 设计方法概述
 - 需求分析
 - 概念设计
 - 逻辑设计
 - ■物理设计
 - 数据库实施
 - 数据库运行和维护
- 2 数据模型与数据建模
- 3 IDEF1X数据建模方法



- 什么是数据库设计
 - 广义:设计数据库应用系统
 - 狭义:设计数据库本身
- 数据库设计是指对于一个给定的应用环境,构造 (设计)优化的数据库逻辑模式和物理结构,并 据此建立数据库及其应用系统,使之能够有效地 存储和管理数据,满足各种用户的应用需求,包 括信息管理要求和数据操作要求。



- 信息管理要求:在数据库中应该存储和管理哪些数据对象。
- 数据操作要求: 对数据对象需要进行哪些操作,如查询、增、删、改、 统计等操作。
- 在数据库领域内,常常把使用数据库的各类系统 统称为数据库应用系统。例如:
 - 管理信息系统
 - 办公自动化系统
 - 电子商务系统
 - • • •

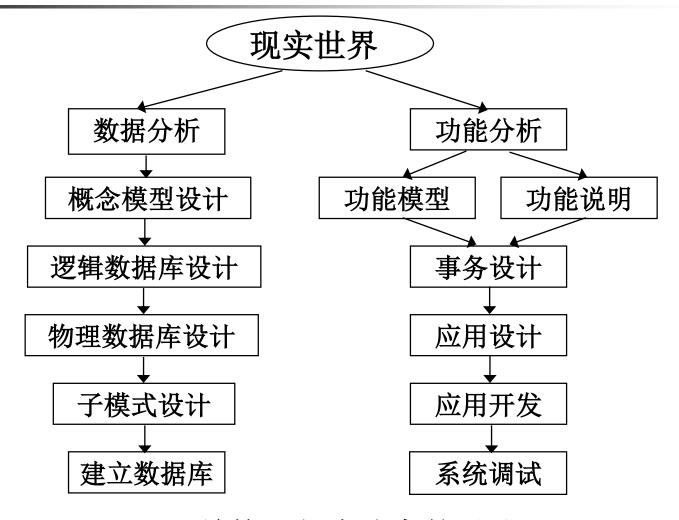
- 4
 - 大型数据库应用系统的设计和开发是一项庞大的工程,涉及多学科的综合性技术。
 - 数据库设计人员应该具备多方面的技术和知识:
 - 计算机的基本知识
 - 软件工程的原理和方法
 - 程序设计的方法和技巧
 - 数据库的基本知识
 - 数据库设计技术
 - 应用领域的知识



- 数据库设计的目标
 - 为用户和各种应用系统提供一个信息基础设施和高效率 的运行环境。
 - ■高效率的运行环境
 - 数据库数据的存取效率高
 - 数据库存储空间的利用率高
 - 数据库系统运行管理的效率高

- 4
 - 数据库设计的特点
 - 数据库建设有"三分技术,七分管理,十二分基础数据"的说法。
 - 数据库的设计应该和应用系统设计相结合,即:将结构 (数据)设计和行为(处理)设计紧密结合起来。
 - 传统的软件工程: 重"行为设计"
 - 忽视对应用中数据语义的分析和抽象,只要有可能就尽量 推迟数据结构设计的决策
 - 早期的数据库设计: 重"结构设计"
 - 致力于数据模型和数据库建模方法研究,忽视了行为设计 对结构设计的影响





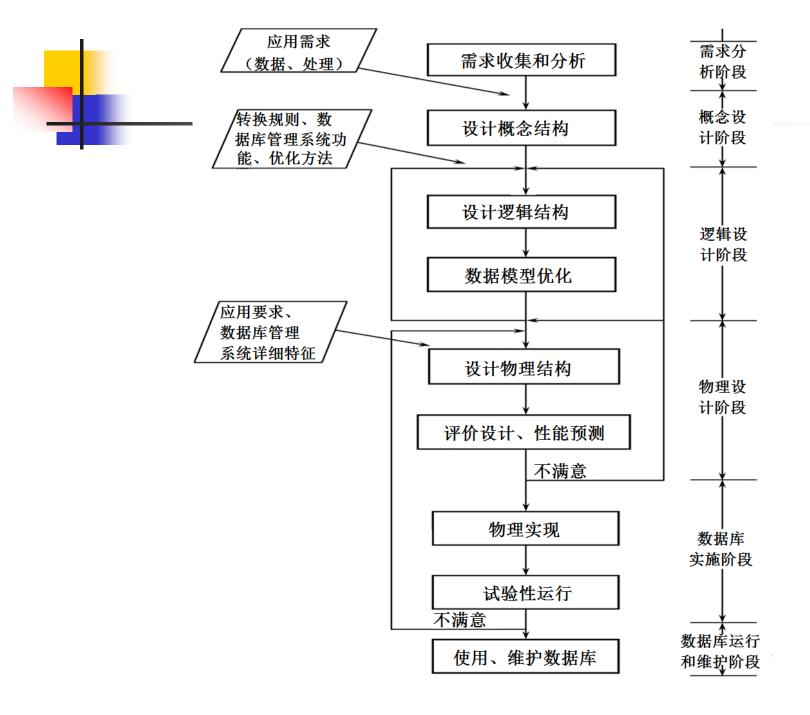
结构和行为分离的设计



- 数据库设计的方法
 - 手工试凑法凭经验设计,缺乏科学理论和工程方法的支持,设计的 质量难以保证
 - 规范化的设计方法: 按照一套规范和流程开展设计工作
 - 新奥尔良(New Orleans) 方法
 - 基于E-R模型的数据库设计方法
 - 基于3NF的设计方法
 - ■面向对象的数据库设计方法
 - 统一建模语言(UML)方法
 - 计算机辅助的自动设计方法



- 数据库设计的基本步骤——按照软件工程的思想,数据库设计可以分为6个阶段:
 - 需求分析
 - 概念设计
 - ■逻辑设计
 - ■物理设计
 - 数据库实施
 - 数据库运行和维护
 - ——一个完善的数据库应用系统的设计往往是上 述**6**个阶段不断反复的结果





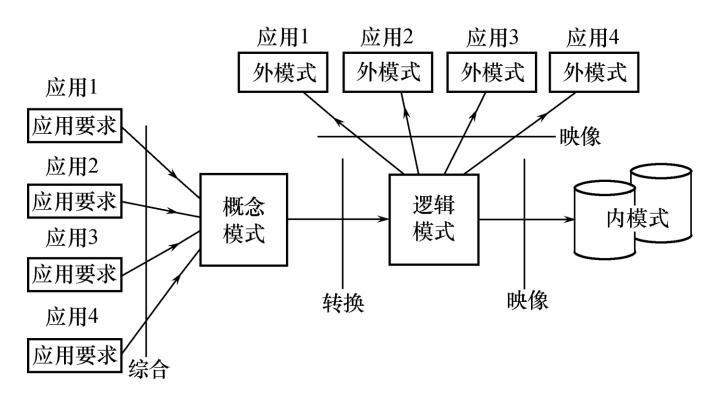
- (1)需求分析
 - 准确了解与分析用户需求(包括数据与处理);
 - 是整个设计过程的基础,是最困难、最耗时的一步。
- (2)概念设计 (概念结构设计)
 - 是整个数据库设计的关键;
 - 通过对用户需求进行综合、归纳与抽象,形成一个独立 于具体DBMS的概念模型
- (3)逻辑设计(逻辑结构设计)
 - 将概念模型转换为某个DBMS所支持的数据模型;
 - 并对其进行优化。

- 4
 - (4) 物理设计
 - 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)。
 - (5) 数据库实施阶段
 - 运用DBMS提供的数据语言、工具及宿主语言,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、并进行试运行测试。
 - (6) 数据库运行与维护阶段
 - 数据库应用系统经过试运行后即可投入正式运行。
 - 在运行和维护阶段,需要不断地完善系统性能、改进系 统功能,通过适当的修改和调整来延长系统的使用时间。

	设计阶 段	设 计 描 述	
-		数 据	处理
	需求分 析	数据字典、全系统中数据项、 数据流、数据存储的描述	数据流图和判定表(判定树)、数 据字典中处理过程的描述
	概念结构设计	概念模型(E-R图) 数据字典	系统要求、方案和概图; 反映新系统信息流的 数据流图
	逻辑结构设计	某种数据模型	系统结构图 (模块结构) 一
	物理设计	存储安排 方法选择 存取路径建立	模块设计 IPO表
	实施阶段	编写模式 装入数据 数据库试运行	程序编码、编译联结、 Main() if then end
	运行维护	性能监测、转储/恢复 数据库 重组和重构	系统转换、运行、维护(修正性、 适应性、改善性维护)

- 数据库设计中各级模式的形成过程
 - 需求分析阶段: 综合各个用户的应用需求
 - 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(E-R图)
 - 逻辑设计阶段
 - 首先将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,建立必要的视图(View),形成数据的外模式
 - 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, 建立索引,形成数据库内模式

数据库设计不同阶段形成的数据库各级模式



1.2 数据库设计一一需求分析

- 需求分析,简单的说就是分析用户的要求。
 - 设计数据库的起点。
 - 结果是否准确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。
- 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)工作概况;
 - 明确用户的各种需求;
 - 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变,不能仅按当前需求来设计数据 库)。

- 需求分析的重点是"数据"和"处理",获得用户对数据库的要求
 - 信息要求
 - 用户需要从数据库中获得信息的内容与性质;
 - 由信息要求可导出数据要求,即在数据库中需要存储哪些数据。
 - 处理要求
 - 用户要完成的处理功能
 - 对处理性能的要求
 - 安全性与完整性要求



- 需求分析的难点
 - 用户缺少计算机知识,不能准确地表达自己的需求,他们所提出的需求往往不断地变化。
 - 设计人员缺少用户的专业知识,不易理解用户的真正需求,甚至误解用户的需求。
- 难点解决方法
 - 设计人员必须不断深入地与用户进行交流,才能逐步确定用户的实际需求。



- 需求分析的方法
 - ■调查清楚用户的实际要求
 - 与用户达成共识
 - 分析和表达用户需求
 - 借助一些工具、模型或方法来分析描述用户的需求
 - 结构化分析方法(Structured Analysis, SA方法) 一种简单实用的分析和表达用户需求的方法

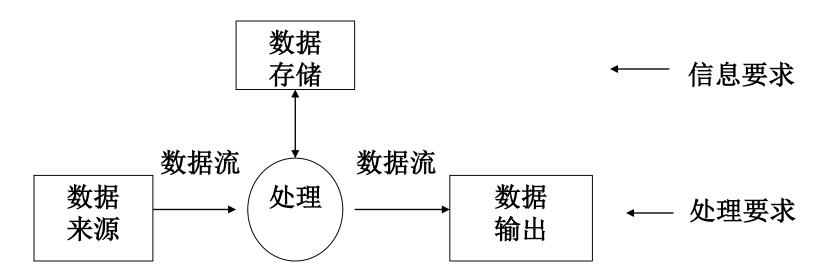


- ■调查用户需求的步骤
 - 1. 调查组织机构情况;
 - 2. 调查各部门的业务活动情况,包括输入和使用什么样的数据,如何加工数据,输出什么信息,输出到什么部门,输出结果的格式是什么;
 - 3. 在熟悉业务的基础上,协助用户明确对新系统的各种 要求;
 - 4. 确定新系统的<mark>边界</mark>,明确哪些功能由计算机完成,哪 些活动由人工完成。



- 常用调查方法
 - 跟班作业 亲身参加业务工作了解业务活动的情况
 - 开调查会 通过与用户座谈来了解业务活动情况及用户需求
 - ■请专人介绍
 - ■询问
 - ■问卷调查
 - 查阅记录,查阅与原系统有关的记录

- 4
 - 结构化分析方法(SA方法)基本过程
 - 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,并用数据流图和数据字典描述系统。
 - 1) 把任何一个系统都抽象为:

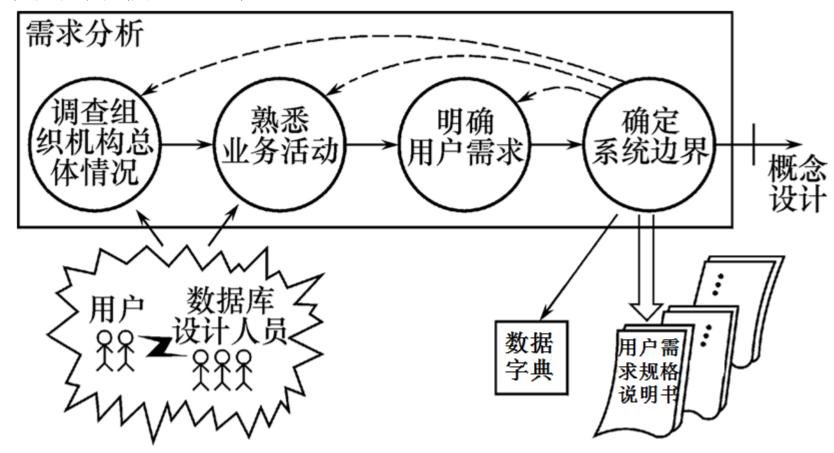




- 2) 对处理功能和数据进行分解
 - 将处理功能的具体内容分解为若干子功能,再将每个子功能继续分解,直到把系统的工作过程表达清楚为止。(分解处理功能)
 - 在处理功能逐步分解的同时,其所用的数据也逐级 分解,形成若干层次的数据流图。(分解数据)
 - 数据流图表达了数据和处理过程的关系。
 - 处理过程用判定表或判定树来描述。
 - 数据用数据字典来描述。
- 3)将分析结果再次提交给用户,征得用户的认可。



■ 需求分析的过程



数据字典

- 数据字典
 - 系统中各类数据描述的集合;
 - 是关于数据库中数据的描述(元数据),不是数据本身。
 - 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。
 - 数据字典是进行详细的数据收集和数据分析所获得的主要结果。



- 数据字典的内容
 - ■数据项
 - 数据结构
 - 数据流
 - ■数据存储
 - 处理过程



- 数据项
 - ——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}}



- 数据流
 - ——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}



- 数据存储 ——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}



■ 处理过程

具体的处理逻辑,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

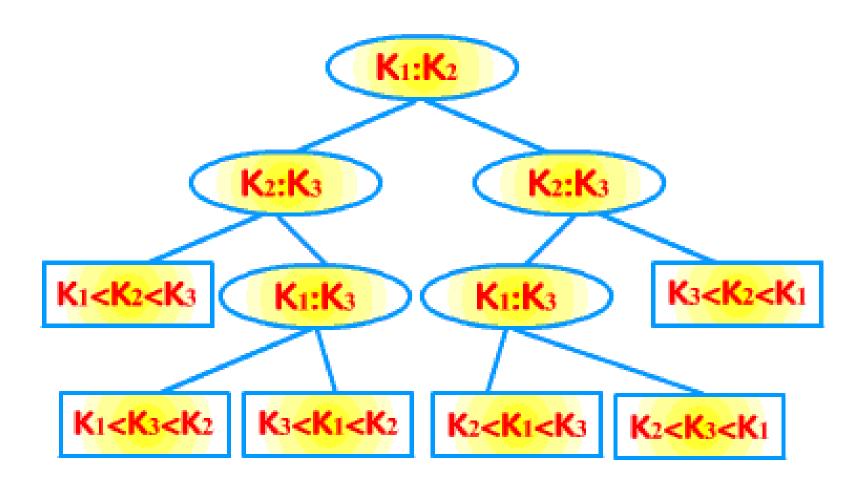
处理过程描述={处理过程名,说明,

输入:{数据流},

输出:{数据流},

处理:{简要说明}}





数据字典示例

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: 0000000至99999999

取值含义:前两位表示该学生所在年级,

后六位按顺序编号

与其他数据项的逻辑关系:



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: 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:

高峰期流量:

4

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



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。



- 注意: 数据字典
 - 关于数据库中数据的描述,是元数据,而不是数据本身;
 - 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。



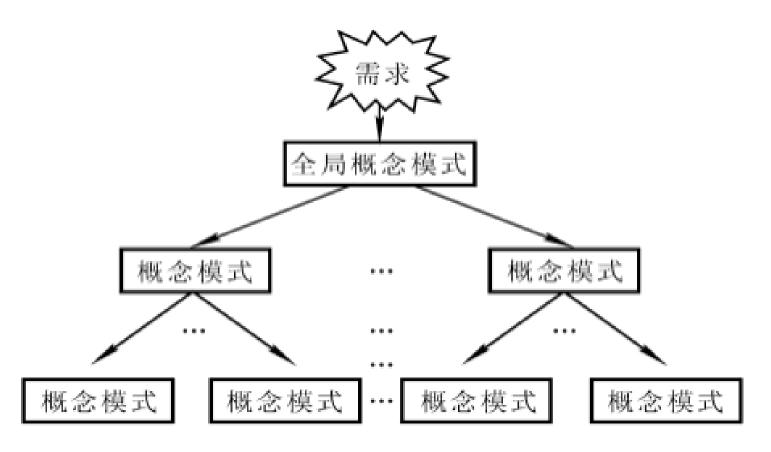
- 需求收集和分析是数据库设计的第一阶段,十分 重要。
- 需求分析收集的基础数据(用数据字典来表达) 是下一步进行概念设计的基础。
- 设计人员应充分考虑到可能的扩充和改变,使设计易于更改,系统易于扩充。
- 必须强调用户的参与。

1.3 数据库设计一一概念设计

- 一什么是概念结构设计(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)的过程,就是概念结构设计。
 - 概念模型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,从而更加稳定。
 - 概念模型独立于具体的数据模型(层次、网状、关系), 与DBMS无关。
 - 概念设计的目的是建立概念模型。
 - 描述概念模型的工具: E-R模型。
- 概念设计是整个数据库设计的关键。

- 4
 - 概念结构设计的要求
 - 能真实、充分地反映现实世界;
 - 易于理解;
 - 易于更改;
 - 易于向关系、网状、层次等各种数据模型转换。
 - 描述概念模型的工具
 - E-R模型
 - E-R图

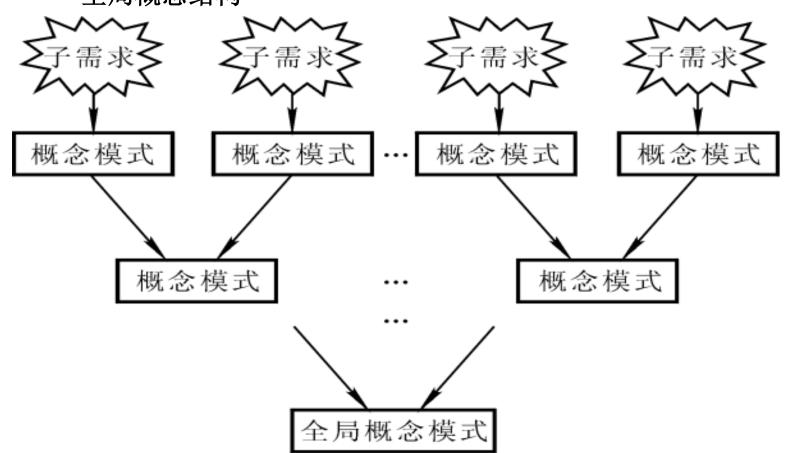
- 4
 - 设计概念结构的方法(4种)
 - 自顶向下
 - 首先定义全局概念结构的框架,然后逐步细化





■ 自底向上

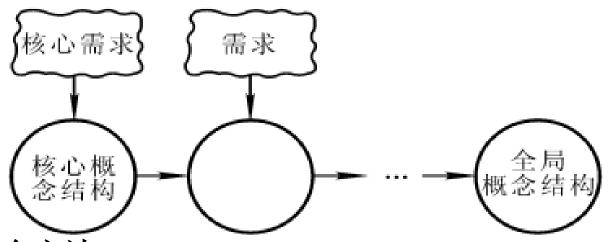
首先定义各局部应用的概念结构,然后将它们集成起来,得到 全局概念结构





■ 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。

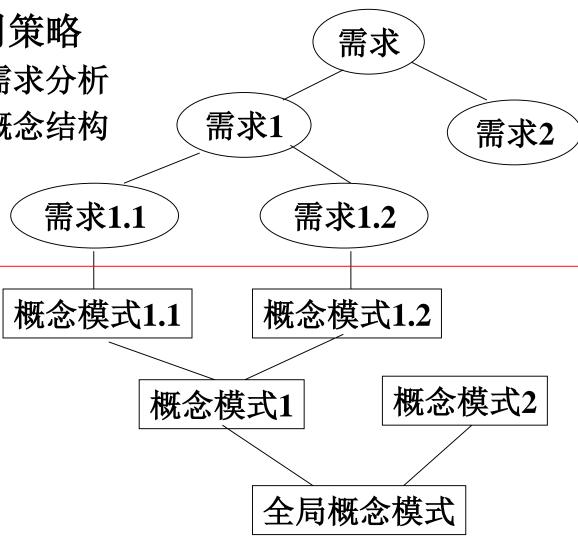


■ 混合方法

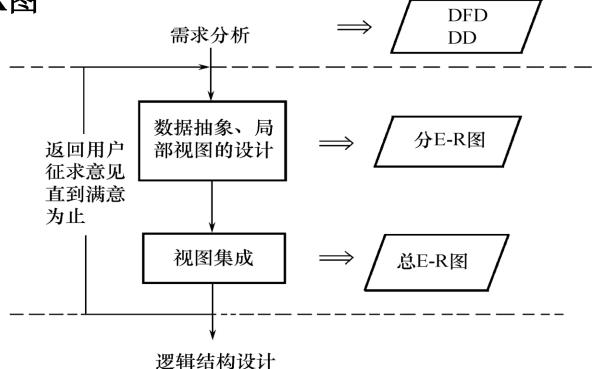
将自顶向下和自底向上相结合,用自顶向下策略设计一个全局概念结构的框架,以它为骨架集成由自底向上策略中设计的各局部概念结构。



- 概念设计的常用策略
 - 自顶向下进行需求分析
 - 自底向上设计概念结构



- 4
 - 自底向上设计概念结构的步骤
 - 1)选择局部应用,抽象数据,逐一设计局部视图,即局部E-R图
 - 2) 将各个局部视图,集成为全局概念结构, 即总的E-R图



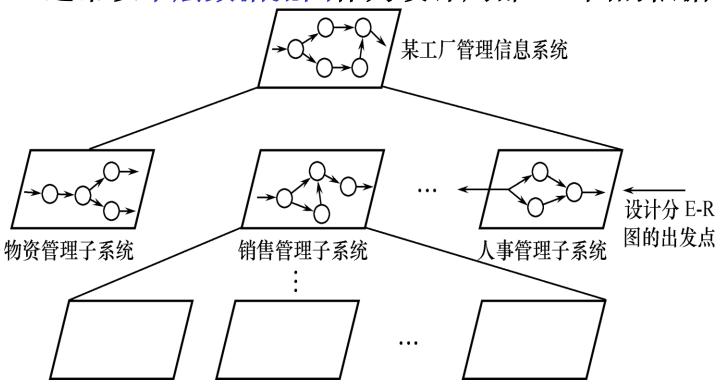


设计局部E-R图

- 设计局部E-R图要解决的问题和步骤:
 - 1)选择局部应用
 - 2)逐一设计局部E-R图



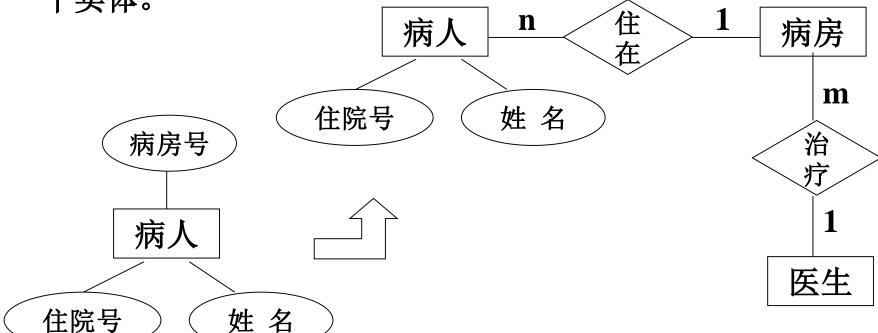
- 选择局部应用
 - 在多层的数据流图中选择一个适当层次的数据流图,作为设计局部E-R图的出发点
 - 通常以中层数据流图作为设计局部E-R图的依据



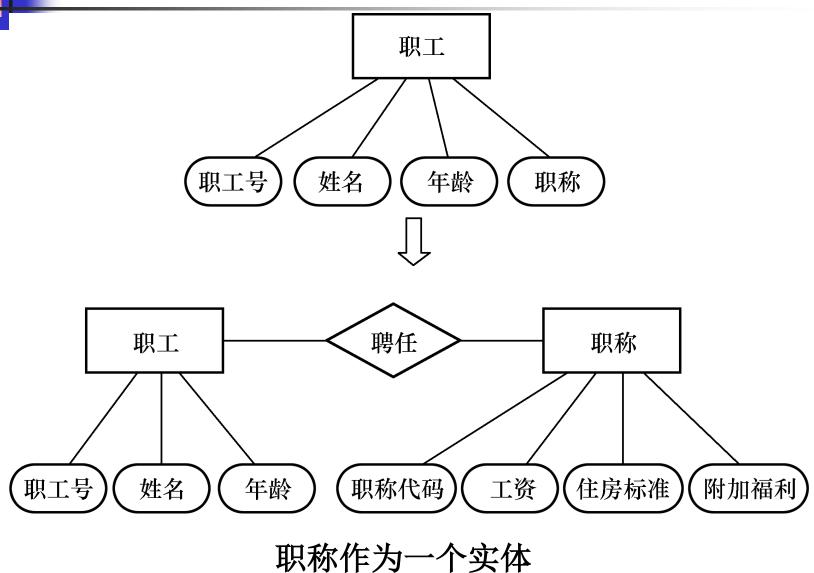
- 4
 - 设计局部E-R图(局部概念设计)
 - 将各局部应用涉及的数据分别从数据字典中抽取出来;
 - 参照数据流图,标定各局部应用中的实体、实体的属性、 标识实体的码;
 - 确定实体之间的联系及其类型(1:1, 1:n, m:n)。
 - 如何抽象实体和属性?
 - 实体:现实世界中一组具有某些共同特性和行为的对象 就可以抽象为一个实体。
 - 属性:对象类型的组成成分可以抽象为实体的属性。
 - 实体与属性是相对而言的。同一事物,在一种应用环境中作为"属性",在另一种应用环境中可能须作为"实体"。

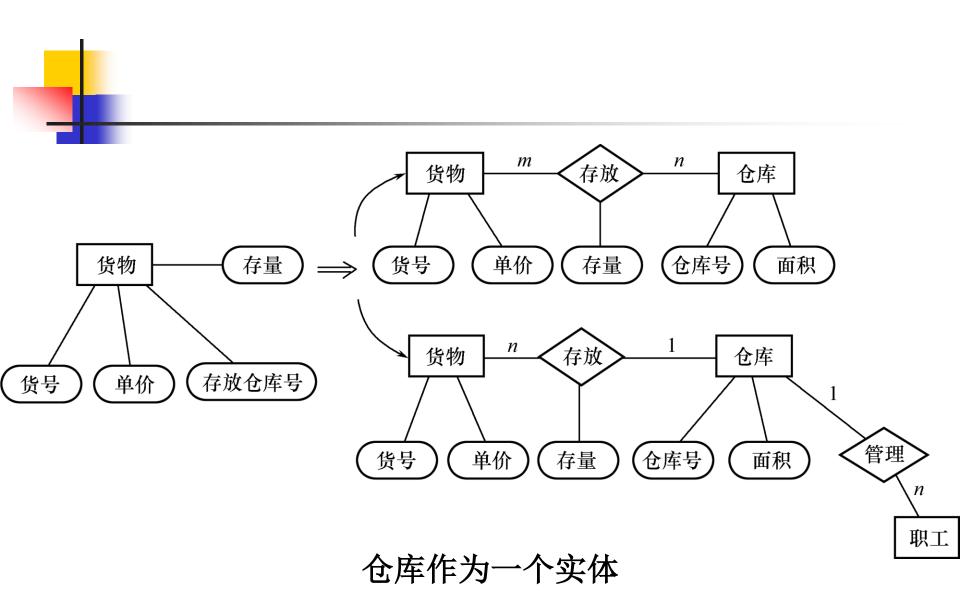
■ 局部E-R图示例

在医院,一个病人只能住在一个病房,病房号可以作为病人实体的属性。如果病房还与医生实体有联系,即一个医生负责几个病房的病人的治疗工作,那么病房应该作为一个实体。











- 区分实体和属性的2条原则
 - 属性不能再具有需要描述的性质属性必须是不可分的数据项,不能再由另一些属性组成。
 - 属性不能与其他实体具有联系 联系只发生在实体之间。

实体与属性是相对的,符合上述两条特性的事物一般作为属性对待。

为了简化E-R图的处置,现实世界中的事物凡能够作为属性对待的,应尽量作为属性。

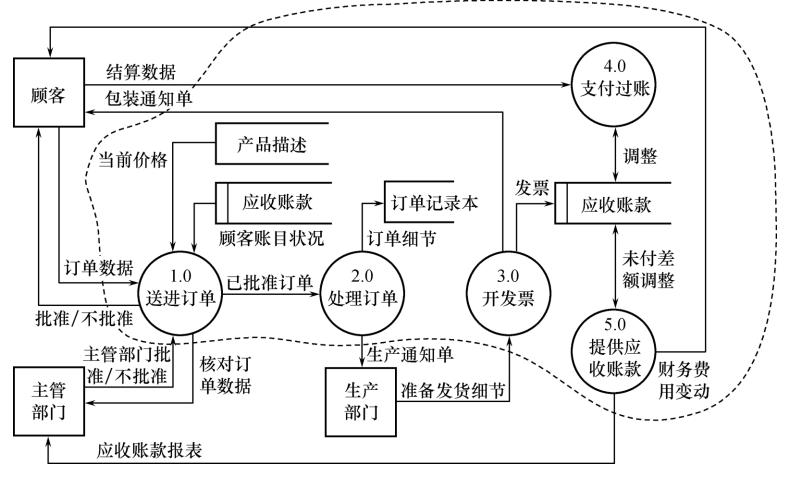
属性仅与一个实体关联,不会与别的实体或属性关联。



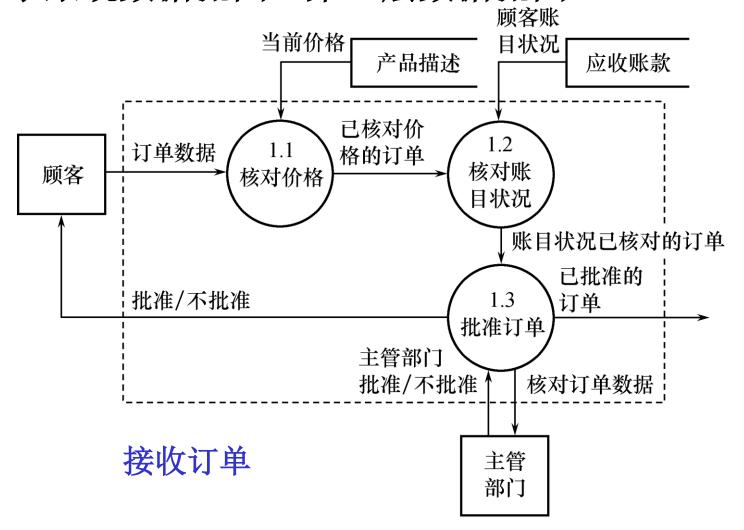
【实例】销售管理子系统E-R图的设计

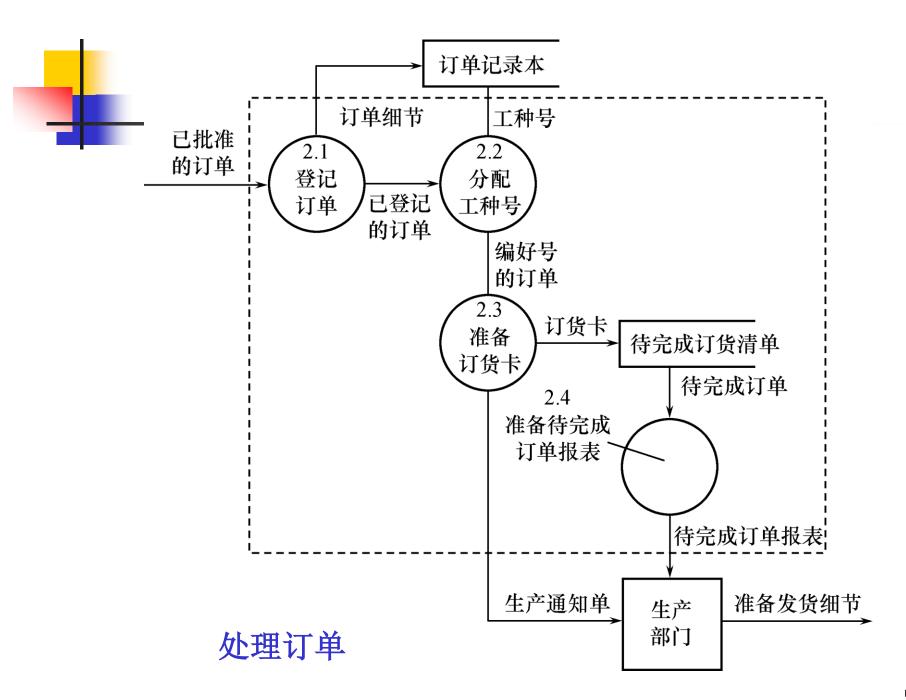
- 销售管理子系统的主要功能:
 - 处理顾客和销售员送来的订单
 - 工厂是根据订货安排生产的
 - 交出货物同时开出发票
 - 收到顾客付款后,根据发票存根和信贷情况进行应收款 处理

销售管理子系统第一层数据流图,虚线部分划出 了系统边界

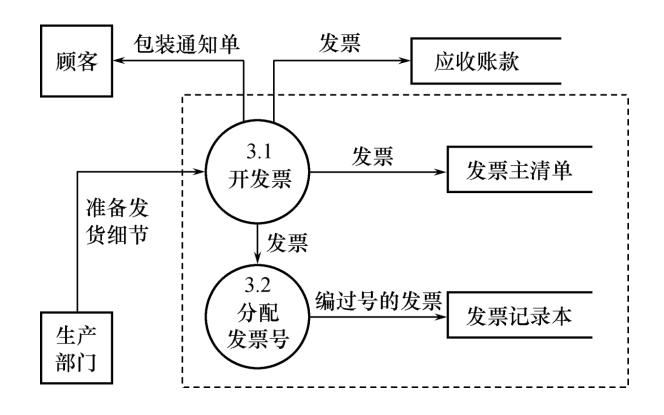


子系统数据流图(第二层数据流图)



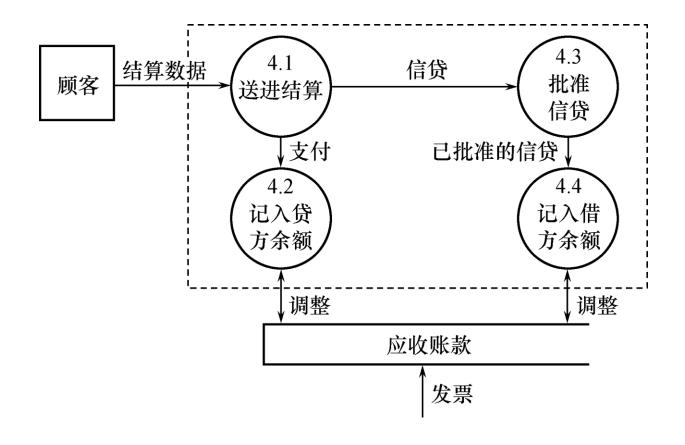






开发票

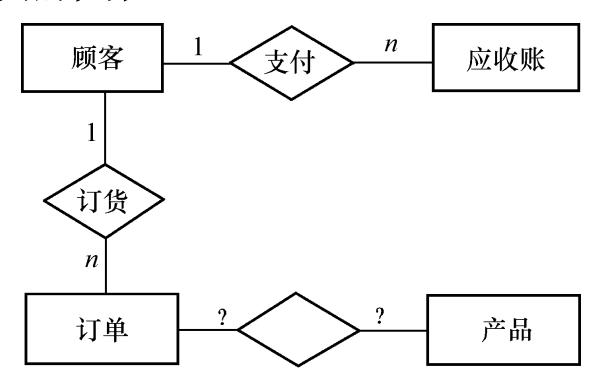




支付过帐

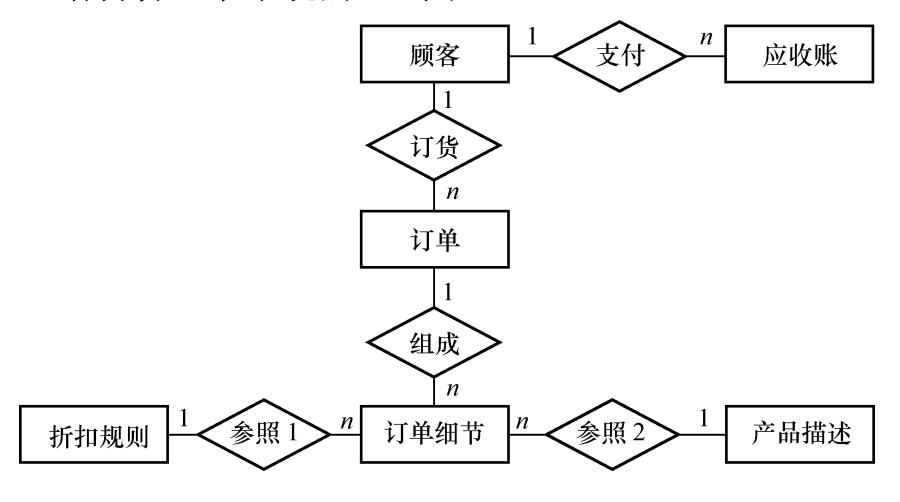


■ E-R图的框架



- 4
 - 参照第二层数据流图和数据字典,进行如下调整:
 - 订单和产品的联系实际上是订单细节和产品的联系。
 - 订单与订单细节是1:n的联系
 - "发票主清单"是一个数据存储 不必作为实体加入E-R图
 - 工厂对大宗订货给予优惠 (新需求),增加一个折扣 规则

销售管理子系统的E-R图

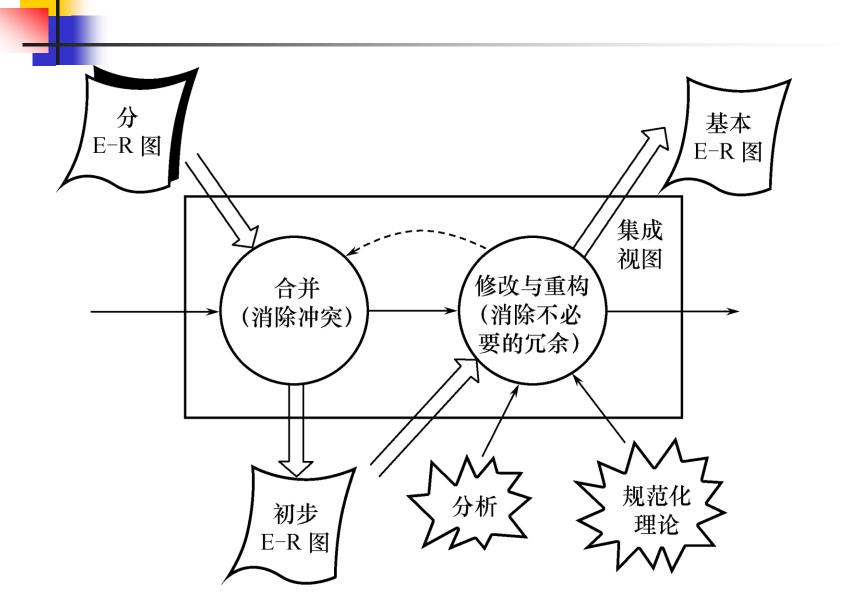


- - 各个实体定义的属性如下:
 - 顾客: {<u>顾客号</u>,顾客名,地址,电话,信贷状况,账目余额}
 - 订单: {<u>订单号</u>, 顾客号, 订货项数, 订货日期, 交货日期, 工种号, 生产地点}
 - 订单细节: {订单号,细节号,产品号,订货量,金额}
 - 应收账款: <u>{顾客号,订单号</u>,发票号,应收金额,支付日期,支付金额,当前余额,货款限额}
 - 产品描述: <u>{产品号</u>,产品名,单价,重量}
 - 折扣规则: {产品号, 订货量, 折扣}



视图集成(局部E-R图的综合)

- 各个局部E-R图需要集成为一个总E-R图
- 如何综合集成局部E-R图?
 - 一般首先集成两个局部视图(通常是比较关键的两个局部视图),以后每次将一个新的局部视图集成进来
 - 集成局部E-R图的步骤
 - 1) 合并分E-R图,消除冲突,生成初步E-R图;
 - 2) 通过修改与重构,消除不必要的冗余, 得到基本E-R图。





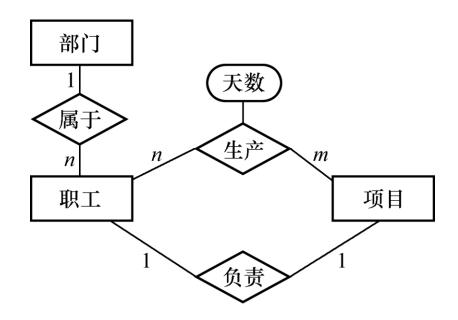
- 1)合并分E-R图,生成初步E-R图
 - 各个分E-R图之间必定会存在许多不一致的地方
 - 合并分E-R图的主要工作与关键是: 合理消除各分E-R图的冲突
 - ■常见的冲突有
 - 属性冲突例如类型不一样,取值范围不一样
 - 命名冲突(实体、属性和联系的名)同名异义、异名同义
 - 结构冲突例如:属性和实体的转换,实体间的联系不一样,同一实体的属性不一样

- - 2) 通过修改与重构,消除不必要的冗余, 得到基本E-R图
 - 初步合并得到的E-R图,可能存在冗余的数据和冗余的 实体间联系
 - 冗余数据和冗余联系容易破坏数据库的完整性,给数据库维护增加困难
 - 消除不必要的冗余后的初步E-R图称为基本E-R图
 - 冗余的消除方法
 - 通过分析数据项之间逻辑关系来消除冗余数据;
 - 通过关系数据库规范化理论中的函数依赖的概念来 消除冗余的联系。

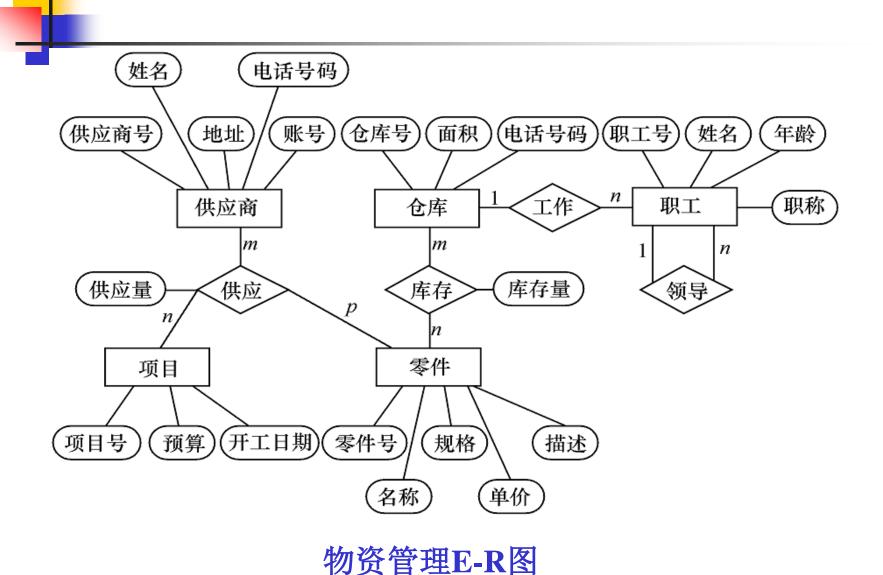


局部E-R图综合实例

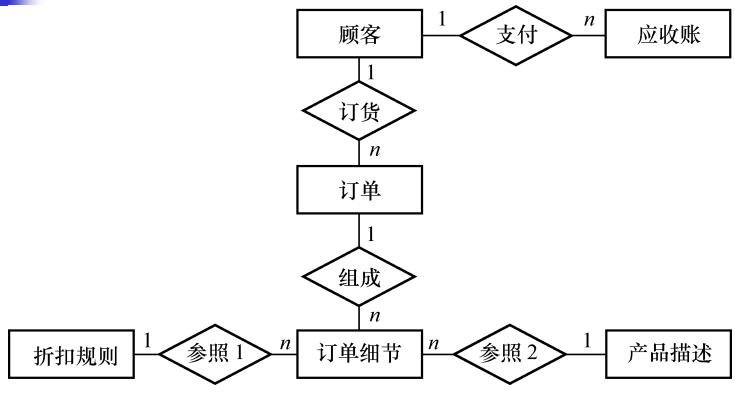
- 某工厂管理信息系统的视图集成
 - 已有该厂物资、销售和劳动人事管理的分E-R图,将其 综合为一个基本的E-R图



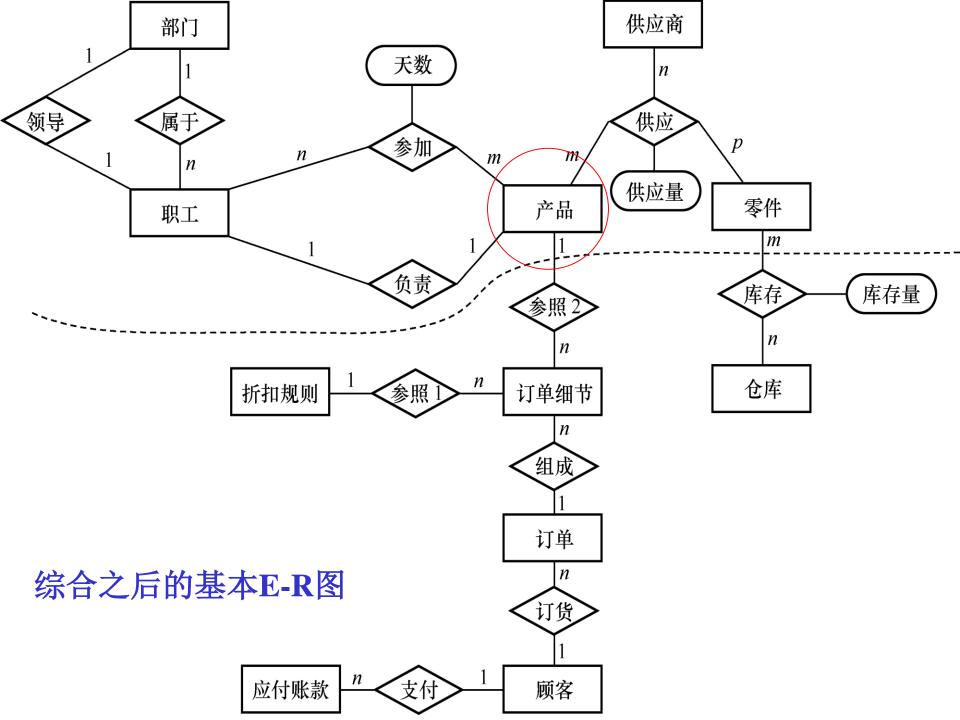
劳动人事管理E-R图







销售管理E-R图





- 集成过程,解决了以下问题:
 - 异名同义,项目和产品含义相同;
 - 库存管理中职工与仓库的工作关系已包含在劳动人事管理的部门与职工之间的联系之中,所以可以取消;
 - 库存管理中职工之间领导与被领导关系可由部门与职工 (经理)之间的领导关系、部门与职工之间的从属关系 两者导出,所以也可以取消。

- 视图集成后形成的整体概念结构,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局部概念结构,包括 属性、实体及实体间的联系。
 - 整体概念结构能满足需要分析阶段所确定的所有要求。
- 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,然后才能作为进一步设计数据库的依据。



自底向上设计概念结构的步骤(小结)

- 1. 选择局部应用,抽象数据,逐一设计局部视图;
- 2. 将各个局部视图(E-R图),集成为全局概念结构,即总的E-R图:
 - 1. 合并分E-R图,消除冲突,生成初步E-R图;
 - 2. 通过修改与重构,消除不必要的冗余,得到基本E-R图;
- 3. 验证整体概念结构;
- 4. 提交用户评审。

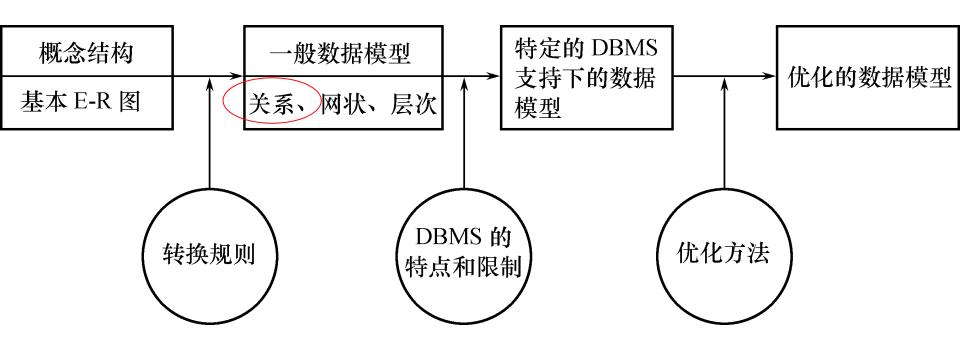


1.4 数据库设计一一逻辑设计

- 逻辑结构设计的任务
 - 概念结构是各种数据模型的共同基础,是独立于任何一种数据模型的信息结构;
 - 为了能够用某一DBMS实现用户需求,还必须将概念结构进一步转化为相应的数据模型,这正是数据库逻辑结构设计所要完成的任务。
- 逻辑结构设计的步骤
 - 1)将概念结构转化为一般的关系、网状、层次模型;
 - 2)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;
 - 3) 对数据模型进行优化。



■ 逻辑结构设计时的3个步骤





1) E-R图向关系模型的转换

- 转换的内容
 - 将实体、实体的属性和实体之间的联系转换为关系 模式
- 要解决的问题
 - 如何将实体型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



■ 转换原则

- 一个实体型转换为一个关系模式
- 一个m:n联系转换为一个关系模式,例如 选修(学号,课程号,成绩)
- 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。
- 一个1:1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
- 三个或三个以上实体间的一个多元联系转换为一个关系模式,例如:讲授(课程号,职工号,书号)

- - 同一实体集的实体间的联系,即自联系,也可按上述 1:1、1:n和m:n三种情况分别处理,例如: 教师(职工号,姓名,性别,职称,系主任)
 - 具有相同码的关系模式可合并,以减少系统中的关系个数,具体方法是: 将其中一个关系模式的全部属性加入到另一个关系模式中,然后去掉其中的同义属性(可能同名也可能不同名),并适当调整属性的次序。
 - 注意: 1:1联系可以与任意一端的关系模式合并,但与不同的关系模式合并效率可能会大不一样,应视具体情况而定。由于连接操作是最费时的操作,一般应以尽量减少连接操作为目标



2)向特定DBMS规定的模型进行转换

- 一般的数据模型还需要向特定DBMS规定的模型进行转 换;
- 转换的主要依据是所选用的DBMS的功能及其限制, 没有通用规则;
- 对于关系模型来说,这种转换通常都比较简单。

3)数据模型的优化

- 数据库逻辑设计的结果不是唯一的;
- 得到初步数据模型后,还应该适当地修改、调整数据模型的结构,以进一步提高数据库应用系统的性能,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。



■ 优化关系数据模型的方法

- **1.**确定数据依赖。按照需求分析得到的语义,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.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。
- **4.**按照需求分析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。

- 优化注意: 并不是规范化程度越高就越优。
 - 当一个应用的查询中经常涉及到两个或多个关系模式的属性时,系统必须经常地进行连接运算,而连接运算的代价是相当高的,可以说关系模型低效的主要原因就是做连接运算引起的,因此在这种情况下,第二范式甚至第一范式也许是最好的。
 - 非BCNF的关系模式虽然从理论上分析会存在不同程度的 更新异常,但如果在实际应用中对此关系模式只是查询, 并不执行更新操作,就不会产生实际影响。
 - 对于一个具体应用来说,到底规范化进行到什么程度,需要权衡响应时间和潜在问题两者的利弊才能决定。一般说来,第三范式就足够了。



■ 4)设计用户子模式

(设计外模式,确定逻辑模式之后做)

- 使用更符合用户习惯的别名;
- 对不同级别的用户定义不同的视图;
- 简化用户对系统的使用。



逻辑设计示例

■ 某公司销售业务中使用的订单格式如下

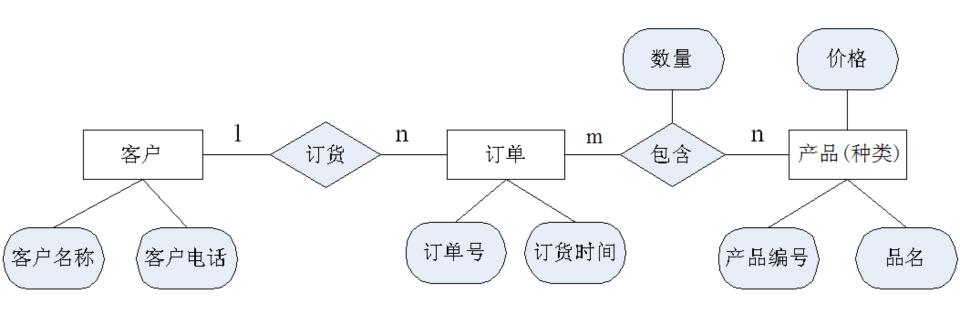
订单号: 1145	订货日期	09/15/2009	客户名称	:ABC 电话: 8141763
产品编号	商品名称	价格	数量	金额
AK0001	电源	100	2	200
HY0022	电表	200	3	600
BM0231	卡尺	40	5	200
			总计	1000



- 公司的业务规定:
 - 订单号是唯一的,每张订单对应一个订单号;
 - 一张订单可订购多种产品,每种产品可在多个订单中出现;
 - 一张订单有一个客户,且一个客户可以有多张订单;
 - 每一个产品编号对应一种产品的品名和价格;
 - ■每一个客户有一个确定的名称和电话号码。



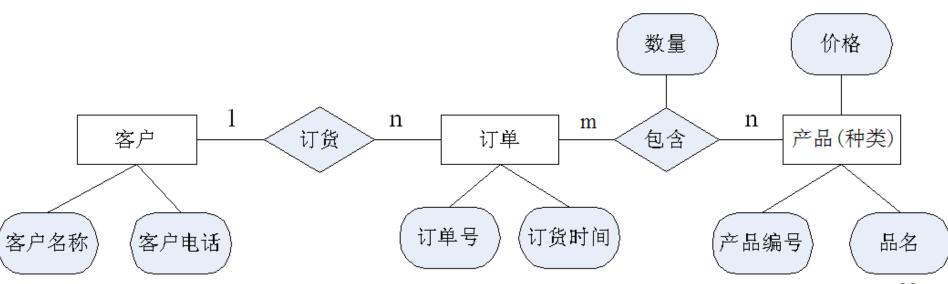
■ 根据上述表格和业务规则设计出E-R图

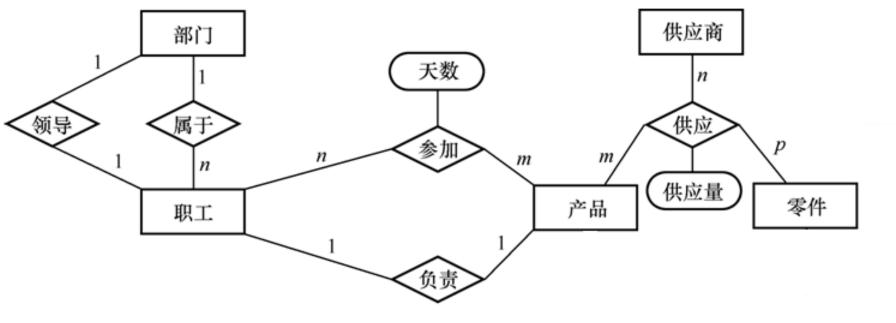


属性标示不太严格



- 根据E-R图转换得到的关系模式
 - 客户(客户编号,客户名称,客户电话)
 - 订单(订单号,订货日期,客户编号,......)
 - 产品(产品编号,商品名称,单价)
 - 订单明细(订单号,产品编号,数量,金额)





- 部门(部门号,部门名,经理的职工号,...)
- 职工(职工号、部门号,职工名,职务,...)
- 产品(产品号,产品名,产品组长的职工号,...)
- 供应商(供应商号,姓名,...)
- 零件(零件号,零件名,...)
- 职工工作(职工号,产品号,工作天数,...)
- 供应(产品号,供应商号,零件号,供应量)

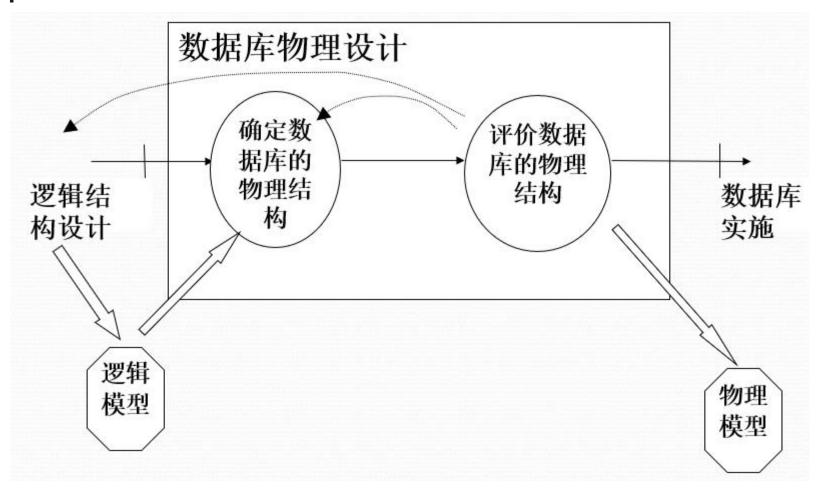
逻辑结构设计小结

- 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(外模式)

1.5 数据库设计一一物理设计

- 什么是数据库的物理设计(物理结构设计)
 - 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环境的 物理结构的过程,就是数据库的物理设计。
- 数据库物理设计的步骤
 - 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) 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。







- 设计数据库物理结构的准备工作
 - 充分了解应用环境,详细分析要运行的事务,以获得设计物理数据库时所需的参数;
 - 充分了解所用DBMS的内部特征,特别是系统提供的存取方法和存储结构。

- - 设计物理数据库所需的参数(也是确定关系存取 方法的依据)
 - 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
 - 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
 - 每个事务在各关系上运行的频率和性能要求

- - 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。
 - DBMS常用的存取方法
 - 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
 - 聚集 (Cluster) 方法
 - 散列(HASH)方法



- 设计索引需要考虑的问题
 - 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



- 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

- 4
 - 确定数据库的物理存储结构
 - 1. 确定数据的存放位置和存储结构
 - 关系
 - 索引
 - 日志
 - 备份
 - 2. 确定系统配置(存储分配参数)



- 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。

- 4
 - 存放数据的基本原则:

根据应用情况将易变部分与稳定部分、存取频率较高部分与存取频率较低部分分开存放,以提高系统性能。

- 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。



- 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
 - 使用的缓冲区长度、个数
 - 时间片大小
 - 数据库的大小
 - 装填因子
 - 锁的数目
 -



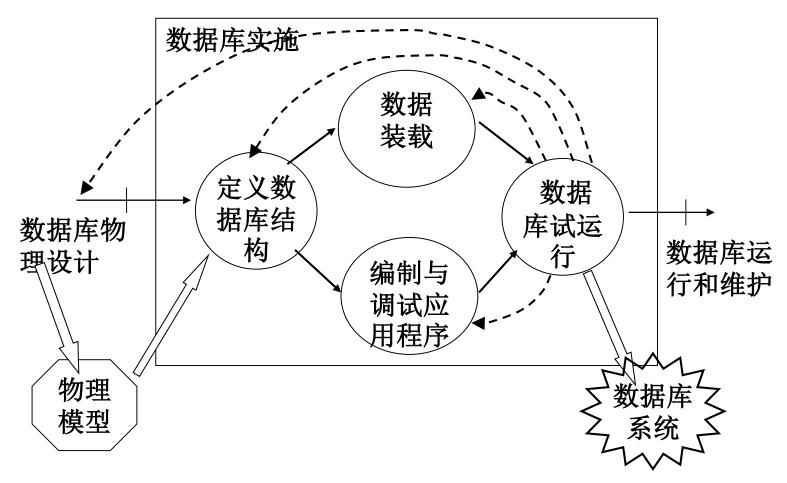
- 数据库物理结构的评价
 - 对数据库物理设计过程中产生的多种方案进行细致的评价,从中选择一个较优的方案作为数据库的物理结构。
- ■物理结构的评价方法(依赖于选用的DBMS)
 - 定量估算各种方案
 - 存储空间
 - > 存取时间
 - **维护代价**
 - 对估算结果进行权衡、比较,选择出一个较优的合理的物理结构;
 - 如果该结构不符合用户需求,则需要修改设计。



1.6 数据库的实施

- 数据库实施的工作内容
 - ■用DDL定义数据库结构
 - 组织数据入库(数据装载)
 - 编制与调试应用程序
 - 数据库试运行







■ 数据装载

- 数据库实施阶段最主要的工作
- 数据装载方法
 - 人工方法
 - 计算机辅助数据入库

■ 注意事项

- 分期分批组织数据入库,先输入小批量数据供调试用,待试运行基本合格后再大批量输入数据;
- 做好数据库的转储与恢复工作,预防随时可能发生 各种故障。

- 4
 - 数据库试运行

装载一小部分数据后,即可对数据库系统进行联合调试

- 1)功能测试
- 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、调整,直到达到设计要求。

2) 性能测试

- 测量系统的性能指标,分析是否达到设计目标;
- 如果测试的结果与设计目标不符,则要返回物理设计阶段,重新调整物理结构,修改系统参数,某些情况下甚至要返回逻辑设计阶段,修改逻辑结构。



- 数据库试运行结果符合设计目标后,数据库可以 真正投入运行。
- 数据库投入运行标志开发任务的基本完成和维护工作的开始。
- 对数据库设计进行评价、调整、修改等维护工作 是一个长期的任务,也是设计工作的继续和提高。
 - 应用环境在不断变化
 - 数据库运行过程中物理存储会不断变化

- 4
 - 在数据库运行阶段,对数据库的日常维护工作主要是由DBA来完成,内容包括:
 - 数据库的转储和恢复
 - 数据库的安全性、完整性控制
 - 数据库性能的监督、分析和改造
 - 数据库的重组织与重构造



- 数据库的重组织
 - 为什么要重组织数据库
 - 数据库运行一段时间后,由于记录的不断增、删、 改,会使数据库的物理存储变坏,从而降低数据库 存储空间的利用率和数据的存取效率,使数据库的 性能下降。
 - 重组织的工作内容和目标
 - 按原设计要求重新安排存储位置、回收垃圾、减少 指针链等,以提高系统性能。
 - 数据库的重组织不会改变原设计的逻辑结构和物理结构。



- 数据库的重构造
 - 为什么要进行数据库的重构造
 - 数据库应用环境发生变化,会导致实体及实体间的 联系也发生相应的变化,使原有的数据库设计不能 很好地满足新的需求
 - 增加新的应用或新的实体
 - 取消某些已有应用
 - 改变某些已有应用



- 数据库重构造的主要工作
 - 根据新环境调整数据库的模式和内模式
 - 増加或删除某些数据项
 - 改变数据项的类型
 - ■増加或删除某个表
 - 改变数据库的容量
 - 增加或删除某些索引



- 重构造数据库的程度是有限的
 - 若应用变化太大,已无法通过重构数据库来满足新的需求,或重构数据库的代价太大,则表明现有数据库应用系统的生命周期已经结束,应该重新设计新的数据库系统,开始新数据库应用系统的生命周期了。

本章小结

- 数据库设计方法
 - ■设计方法概述
 - 需求分析
 - 概念设计
 - 逻辑设计
 - ■物理设计
 - 数据库实施
 - 数据库运行和维护

- 4
 - 数据库设计包含的艺术成分多于科学成分;
 - 有一些理论和方法可以用;
 - 仅有理论和方法还不够,还需要大量的经验和创造性;
 - 理论可以学习,经验源于实践!

作业

- P270(e288) 习题12. 给出概念设计和逻辑设计的结果(采用关系模型)
- 附加题,参见课程作业文档

课程设计文档怎么写? 可以参考绿本11.7节"合同管理系统数据建模"。