数据库系统原理

第7章　数据库设计

分级通关平台，QQ交流

基础篇

第1章 绪论
第2章 关系数据库*3
第3章 标准语言SQL*3
第4章 数据库安全性
第5章 数据库完整性
实验1

设计与应用开发篇

第6章 关系数据理论*2
第7章 数据库设计
实验2
第8章 数据库编程

系统篇

第9章 *关系查询处理和优化
实验3
第10章 数据库恢复技术
第11章 并发控制
实验4

内容提要

- ✓ 数据库设计概述
- ✓ 需求分析
- ✓ 概念结构设计
- ✓ 逻辑结构设计
- ✓ 物理结构设计
- ✓ 数据库的实施和维护

数据库设计概述

- 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。
- 数据库设计：对于一个给定的应用环境，构造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。
  - 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。

- ✓ 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

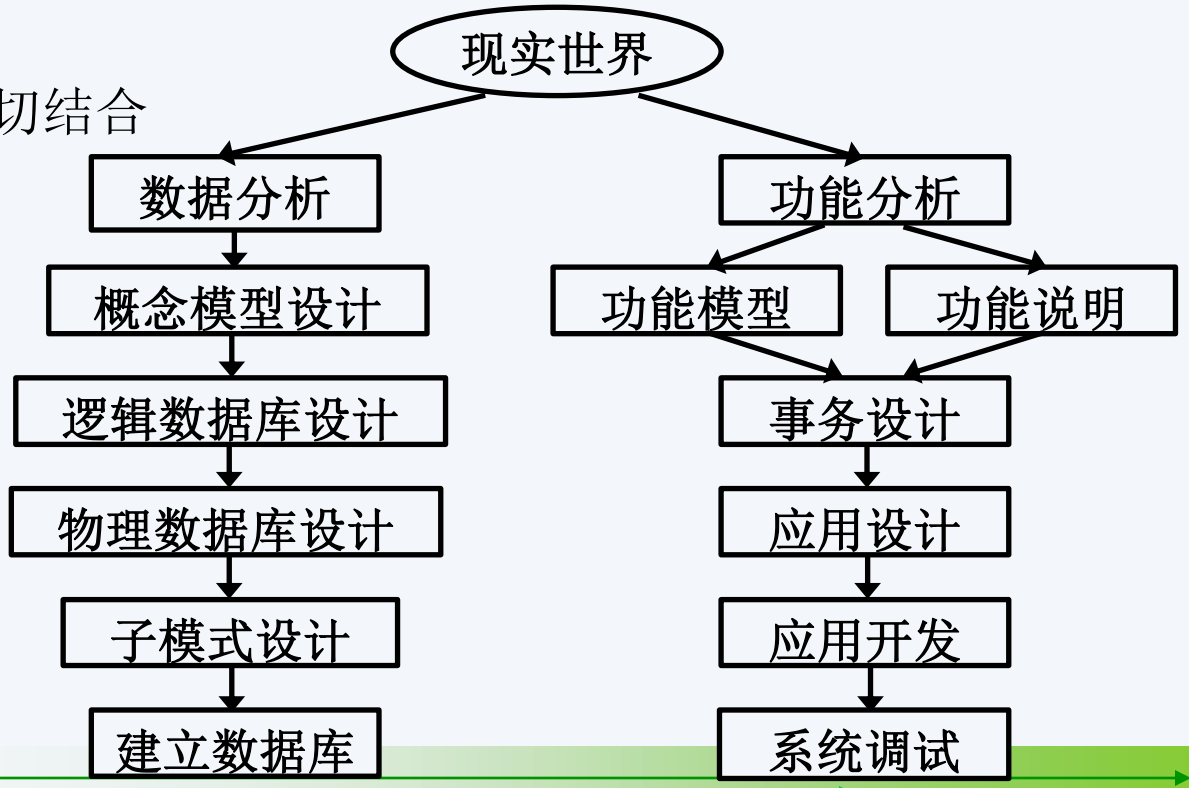
数据库设计概述

1. 数据库建设的基本规律

- 三分技术，七分管理，十二分基础数据

2. 结构（数据）设计和行为（处理）设计相结合

- 将数据库结构设计
  和数据处理设计密切结合



数据库设计概述

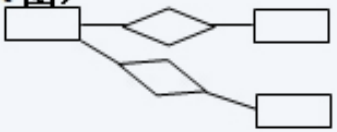
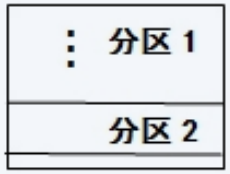
- 大型数据库设计是涉及多学科的综合性技术，是工程项目
  - ✓ 计算机的基础知识
  - ✓ 软件工程的原理和方法
  - ✓ 程序设计的方法和技巧
  - ✓ 数据库的基本知识
  - ✓ 数据库设计技术
  - ✓ 应用领域的知识
- 基本思想
  - 过程迭代和逐步求精
- 典型方法
  - ✓ 新奥尔良（New Orleans）方法
  - ✓ 基于E-R模型的数据库设计方法
  - ✓ 3NF（第三范式）的设计方法
  - ✓ 面向对象的数据库设计方法
  - ✓ 统一建模语言（UML）方法

数据库设计概述

- 数据库设计分6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
- 需求分析和概念设计独立于任何数据库管理系统
- 逻辑设计和物理设计与选用的数据库管理系统密切相关

✓ **这个设计步骤既是数据库设计的过程，也包括了数据库应用系统的设计过程**

数据库设计概述

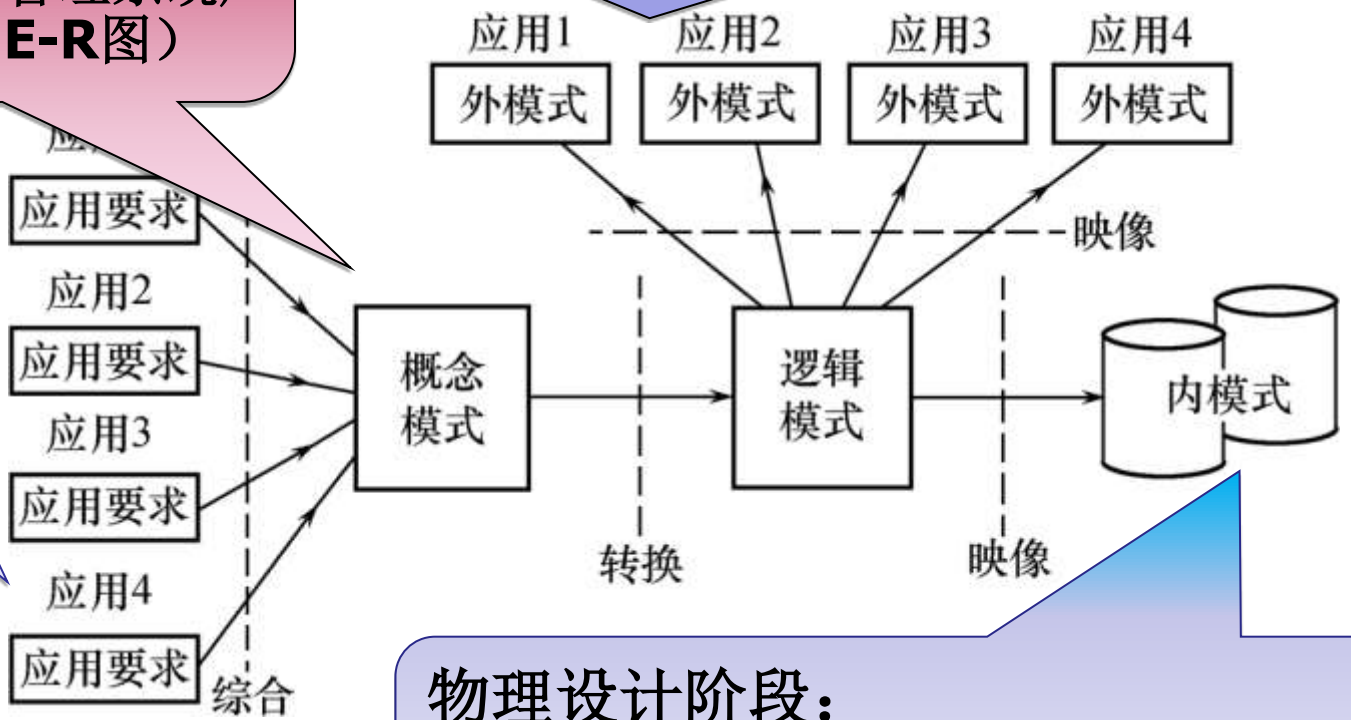
设计阶段	设计描述
需求分析	数字字典、全系统中数据项、数据结构、数据流、数据存储的描述
概念结构设计	概念模型（E-R 图）  数据字典
逻辑结构设计	某种数据模型 关系 非关系 
物理结构设计	存储安排 存取方法选择 存取路径建立 
数据库实施	创建数据库模式 装入数据 数据库试运行 
数据库运行和维护	性能监测、转储/恢复、数据库重组和重构

数据库设计概述

概念设计阶段：
形成独立于机器特点，独立于各个数据库管理系统产品的概念模式（E-R图）

逻辑设计阶段：
1. 将E-R图转换成数据库逻辑模式
2. 建立必要的视图，形成数据的外模式

需求分析阶段：
综合各个用户的应用需求



物理设计阶段：
进行物理存储安排，建立索引，形成数据库内模式

需求分析

- 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，确定新系统的功能（充分考虑今后可能的扩充和改变）
- 调查的重点是“数据”和“处理”

  （1）信息要求

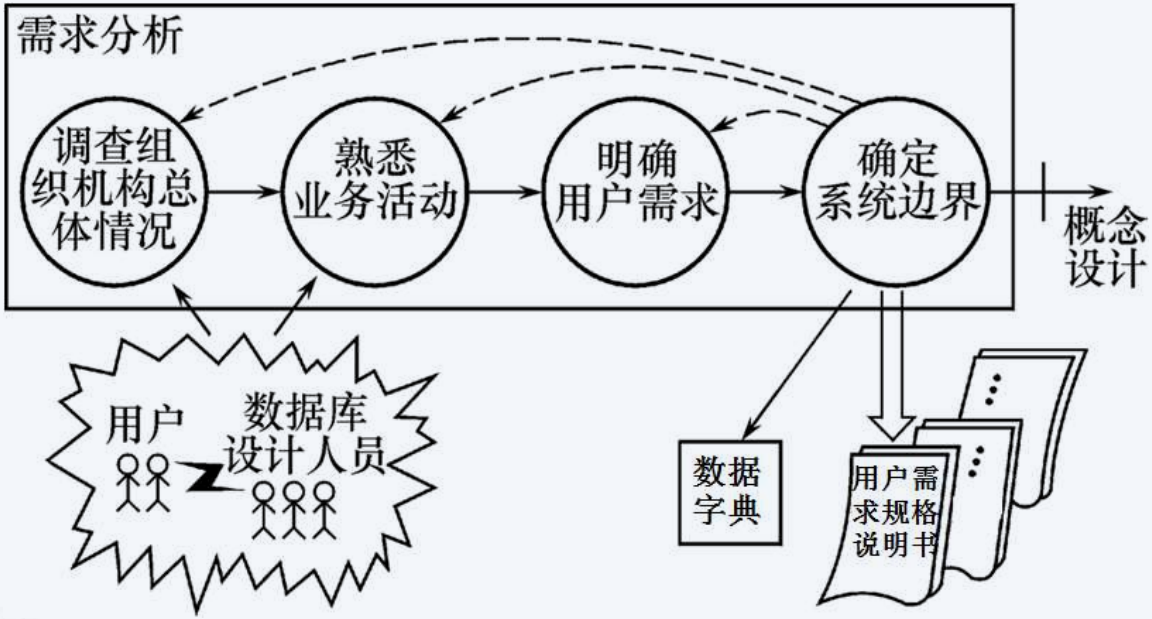
  （2）处理要求（处理功能+处理性能）

  （3）安全性与完整性要求

➢ 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

需求分析

- 结构化分析方法（Structured Analysis，简称SA方法）
  - SA方法从最上层的系统组织机构入手
  - 采用自顶向下、逐层分解的方式分析系统



需求分析—数据字典

- 数据字典是数据收集和数据分析所获得的主要结果
  - 数据字典是关于数据库中数据的描述，即元数据，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
- 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

➢ 数据项是不可再分的数据单位

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}

需求分析—数据字典

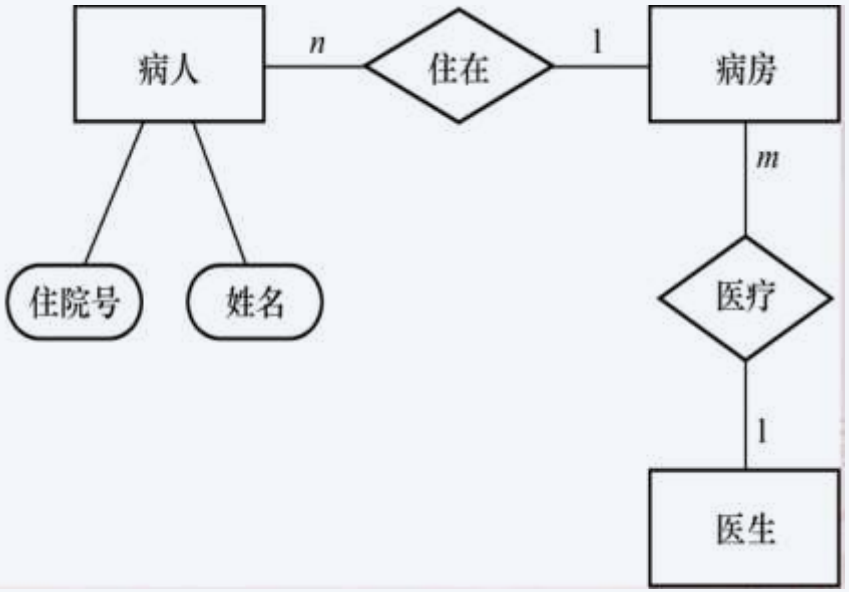
- 数据结构反映了数据之间的组合关系。
  - 一个数据结构可以由若干个数据项组成，也可以由若干个数据结构组成，或由若干个数据项和数据结构混合组成。
  - **数据结构描述=**
    {数据结构名，含义说明，组成:{数据项或数据结构}}
- 数据流是数据结构在系统内传输的路径。
  - **数据流描述=**
    {数据流名,说明,数据流来源,数据流去向,
    组成:{数据结构}, 平均流量,高峰期流量}

需求分析—数据字典

- 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。
  - 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}
- 处理过程一般用判定表或判定树来描述。描述处理的说明
  - 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}}

概念结构设计

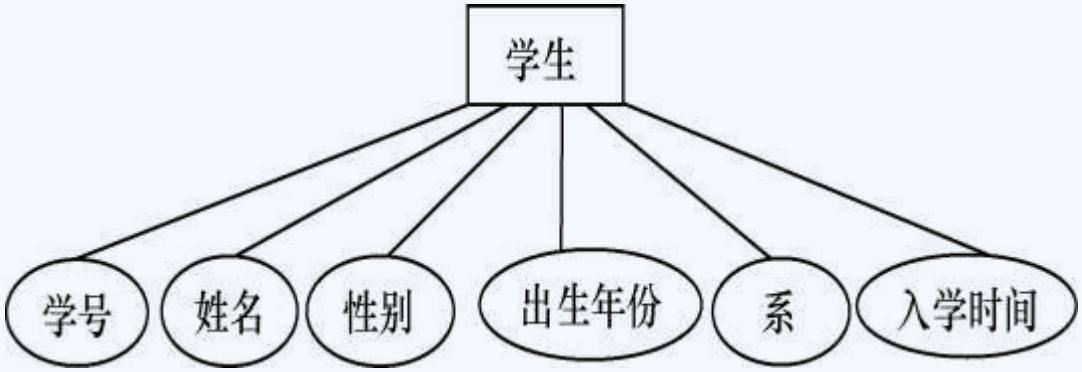
- 将需求分析得到的用户需求抽象为概念模型的过程
- 概念模型的特点
  - 1）是现实世界的一个真实模型，映射
  - 2）易于理解，可以与用户交换意见
  - 3）概念模型容易修改和扩充
  - 4）易于向关系、网状、层次等各种数据模型转换
- 描述概念模型的工具
  - E-R模型



概念结构设计—E-R模型

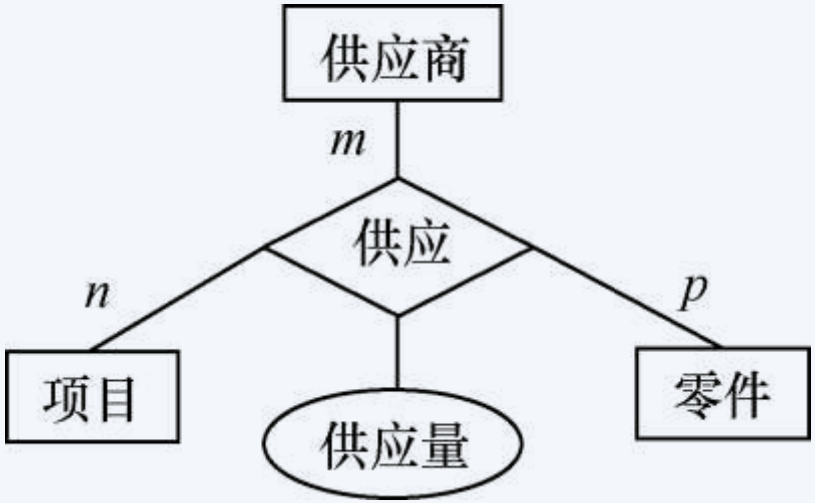
- E-R图提供了表示实体型、属性和联系的方法
  - 实体型：用矩形表示，矩形框内写明实体名。
  - 属性：用椭圆形表示，并用无向边将其与相应的实体型连接起来

✓ 学生实体具有学号、姓名、性别、出生年份、系、入学时间等属性，用E-R图表示



概念结构设计—E-R模型

- 联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体型连接起来，同时在无向边旁标上联系的类型（1:1，1:*n*或*m*:*n*）。
- 联系可以具有属性

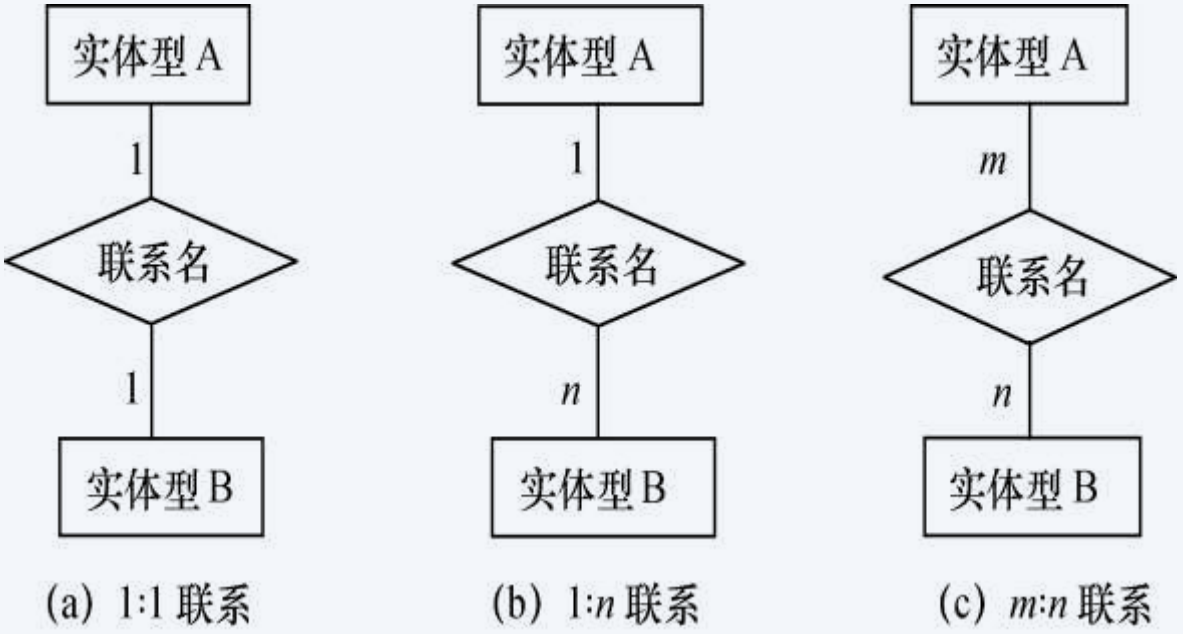


概念结构设计—E-R模型

- 实体之间的联系

（1）两个实体型之间的联系：

①一对一联系（1:1） ②一对多联系（1:*n*） ③多对多联系（*m*:*n*）

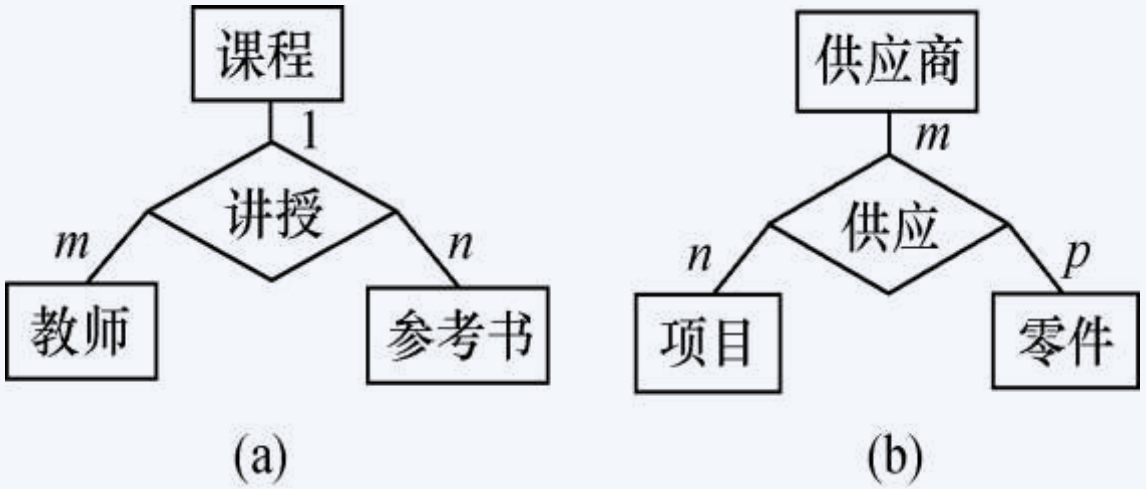


概念结构设计—E-R模型

- 实体之间的联系

（2）两个以上的实体型之间的联系

- 一般地，两个以上的实体型之间也存在着一对一、一对多、多对多联系。



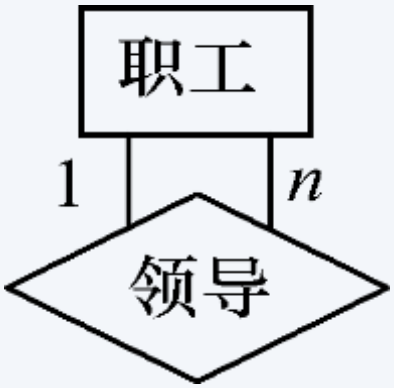
概念结构设计—E-R模型

- 实体之间的联系

（3）单个实体型内的联系

- 同一个实体集内的各实体之间也可以存在一对一、一对多、多对多的联系。

➢ 例如，职工实体型内部具有领导与被领导的联系，即某一职工（干部）“领导”若干名职工，而一个职工仅被另外一个职工直接领导，因此这是一对多的联系。



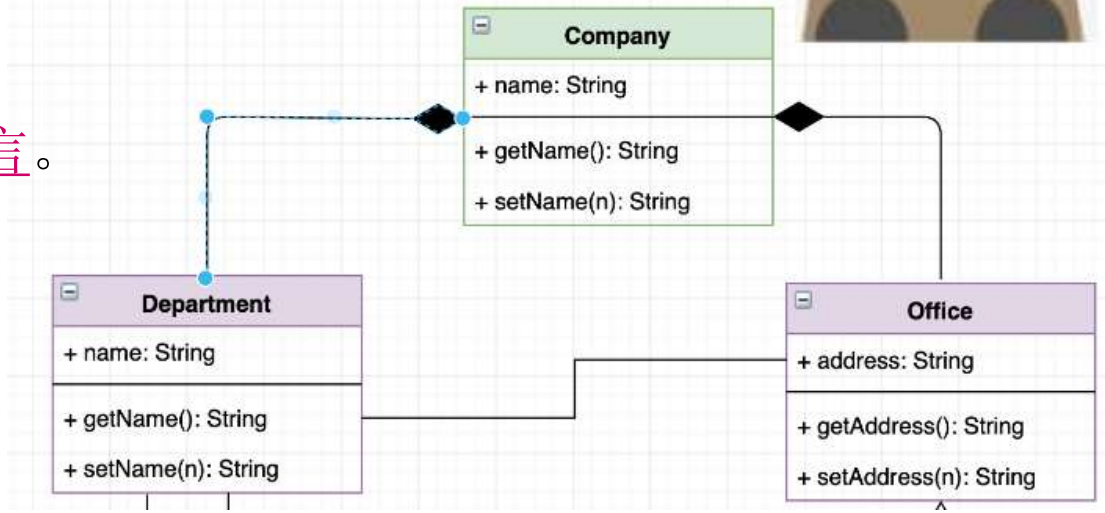
概念结构设计—E-R模型

- 实体与属性的划分原则
  - 为了简化E-R图的处置，能作为属性对待的，尽量作为属性对待。
  - 两条准则：

    1）属性必须是不可分的数据项，不能包含其他属性。

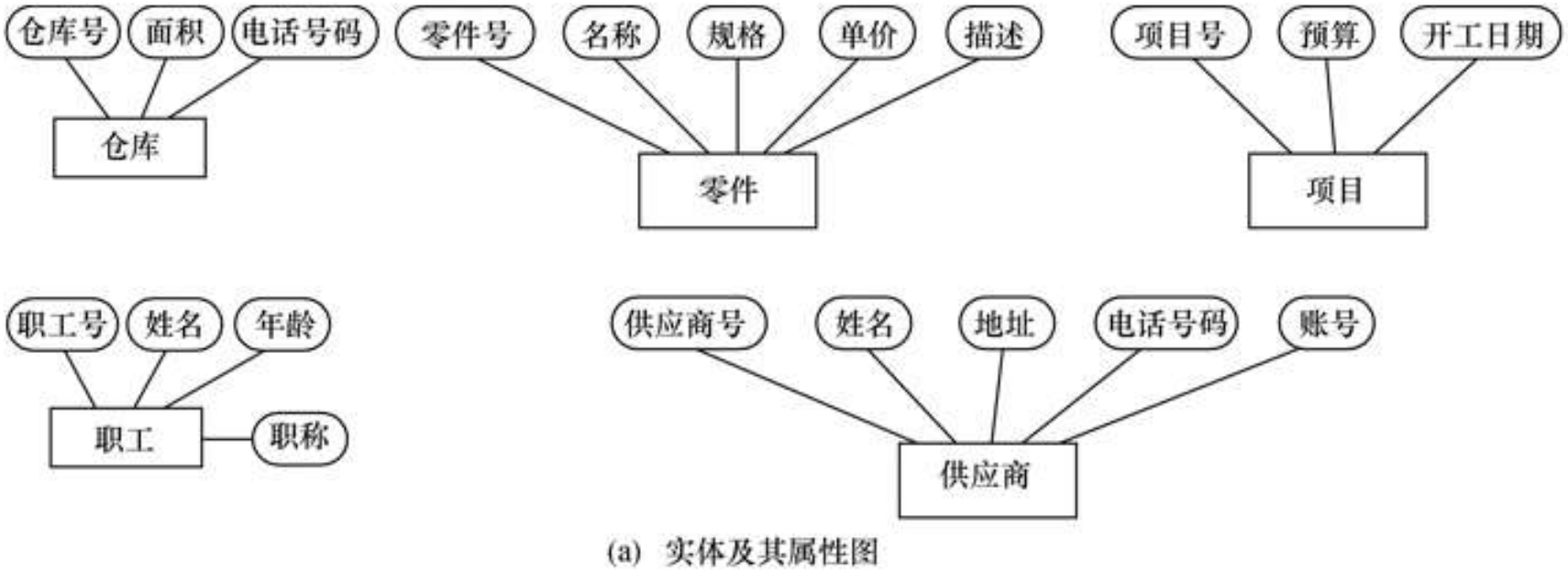
    2）E-R图中所表示的联系是实体之间的联系。

- UML-Unified Modeling Language 统一建模语言。



概念结构设计—E-R模型

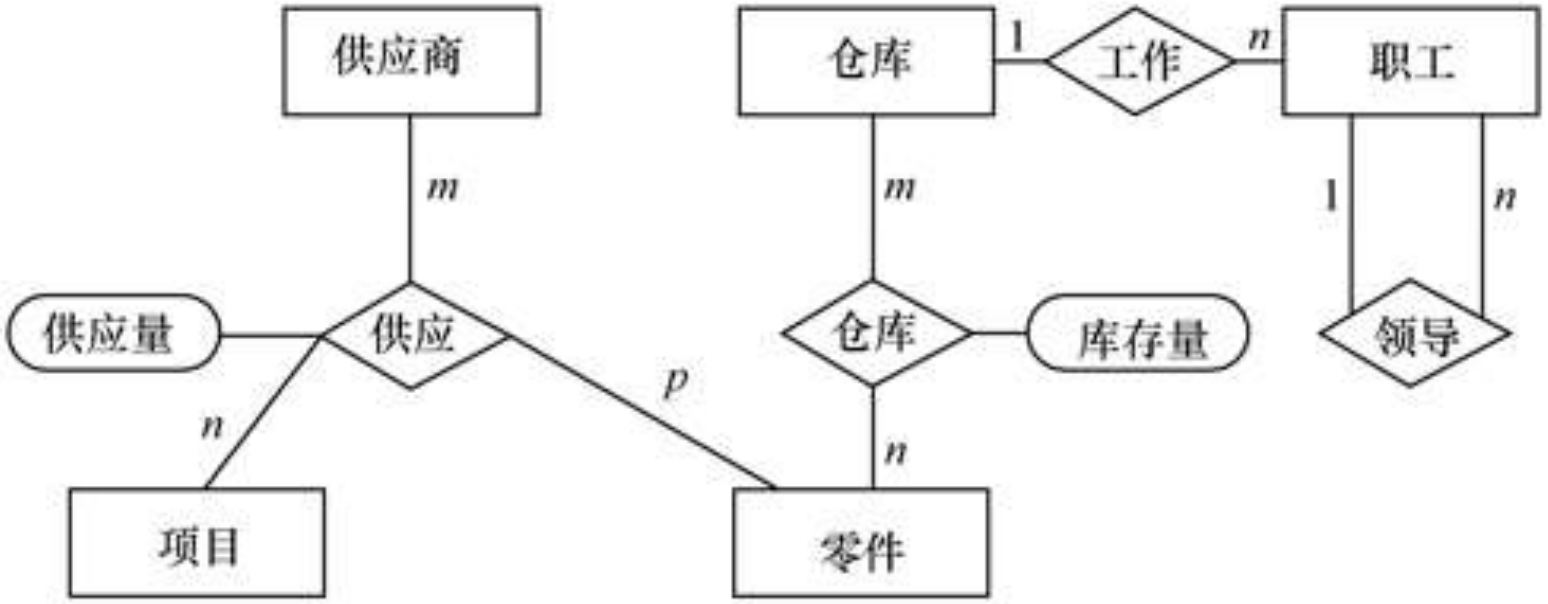
- 某个工厂物资管理的概念模型。物资管理涉及的实体有：



(a) 实体及其属性图

概念结构设计—E-R模型

- 某个工厂物资管理的概念模型。

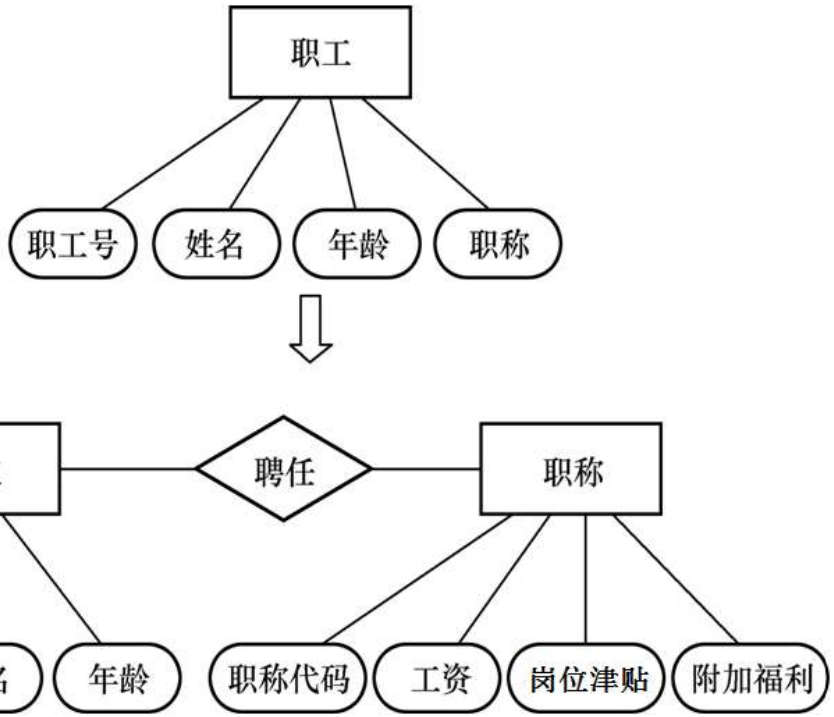


(b) 实体及其联系图

概念结构设计

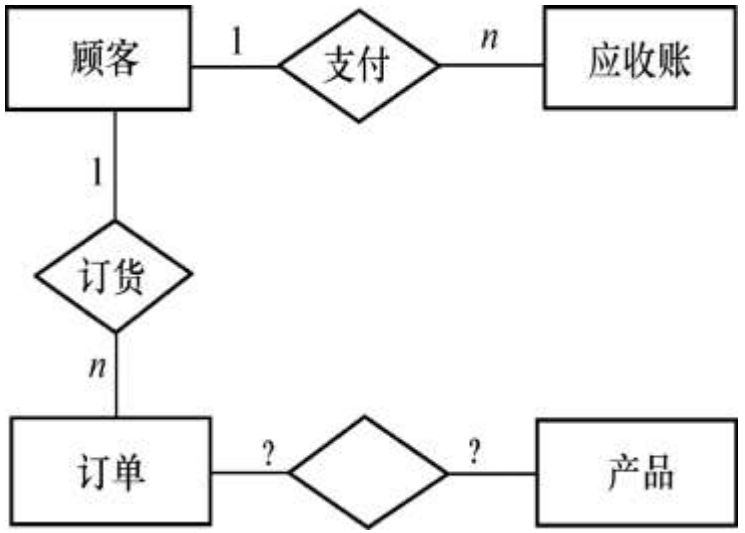
[例1] 职工是一个实体，职工号、姓名、年龄是职工的属。

- 职称如果没有与工资、福利挂钩，可以作为职工实体的属性
- 如果不同的职称有不同的工资、住房标准和不同的附加福利，则职称作为一个实体更恰当



概念结构设计

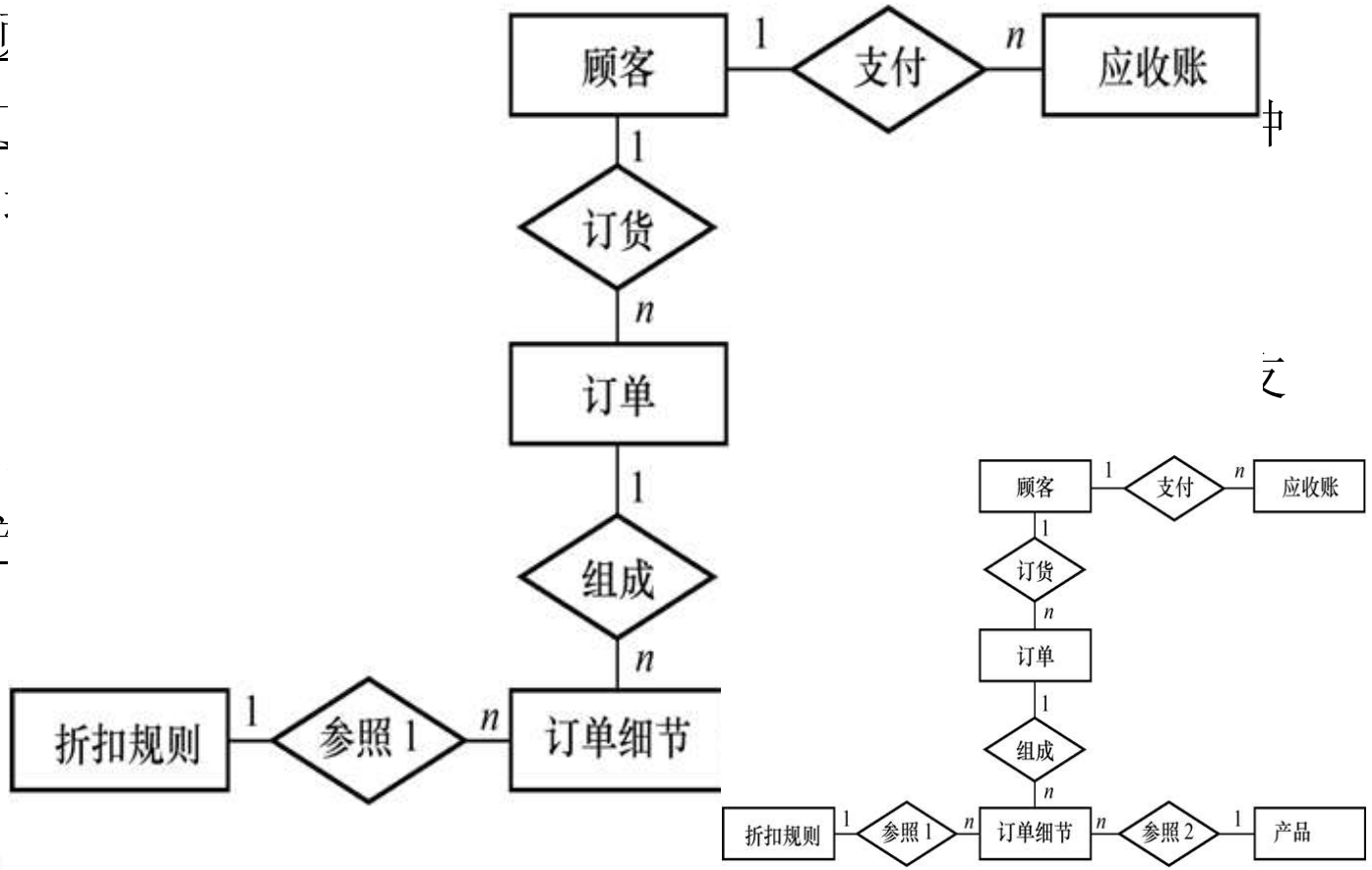
- [例7.1] 销售管理子系统E-R图的设计。
  - 该子系统的主要功能是：
    - 处理顾客和销售员送来的订单
    - 工厂是根据订货安排生产的
    - 交出货物同时开出发票
    - 收到顾客付款后，根据发票存根和信贷情况进行应收款处理



✓ 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。

概念结构设计

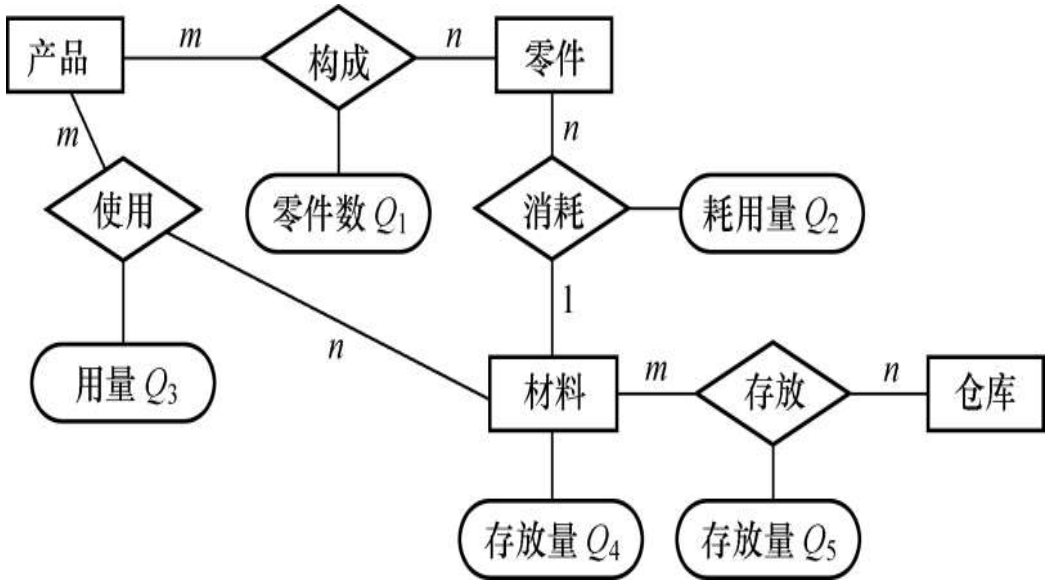
- [例7.1] 销售管理子系统E-R图的设计。
  - 顾客：{顾
  - 订单：{订号，生产
  - 订单细则
  - 应收账款付金额，
  - 产品：{产
  - 折扣规则



概念结构设计

- E-R图的集成
  - 合并。解决各分E-R图之间的冲突，将分E-R图合并起来生成初步E-R图。
  - 修改和重构。消除不必要的冗余，生成基本E-R图。

✓ **并不是所有的冗余数据与冗余联系都必须加以消除，有时为了提高效率，不得不以冗余信息作为代价。**



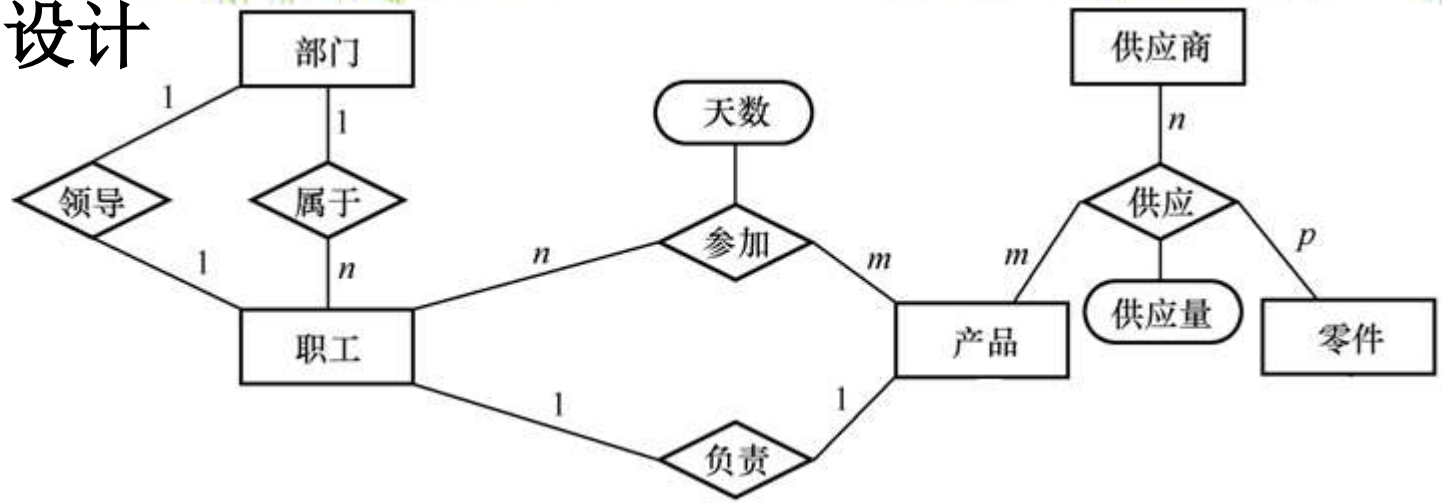
逻辑结构设计

- 逻辑结构设计的任务
  - 把概念结构设计阶段设计好的基本E-R图转换为与选用数据库管理系统产品所支持的数据模型相符合的逻辑结构

✓ 将E-R图转换为关系模型：将实体型、实体的属性和实体型之间的联系转化为关系模式

- 转换原则
  - 一个实体型转换为一个关系模式，属性和码；
  - 实体型间的联系处理（1:1, 1:n, m:n）
  - 三个或三个以上实体间的一个多元联系
  - 具有相同码的关系模式可合并

逻辑结构设计



- 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—优化

- 数据库逻辑设计的结果不是唯一的。
- 以规范化理论为指导，进一步提高数据库应用系统的性能
  - 按照数据依赖的理论对关系模式进行分析，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。
  - 非BCNF的关系模式虽然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

逻辑结构设计—优化

- 对关系模式进行必要分解，提高数据操作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

➢ 水平分解

✓对符合80/20的，把经常被使用的数据（约20%）水平分解出来，形成一个子关系。

✓水平分解为若干子关系，存取的数据需要对应一个子关系

➢ 垂直分解

➢经常在一起使用的属性从R中分解出来形成一个子关系模式

➢可能使另一些事务不得不执行连接操作，降低了效率

逻辑结构设计—用户子模式

- 定义数据库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。
  - 使用更符合用户习惯的别名
  - 针对不同级别的用户定义不同的视图，以保证系统的安全性。
  - 简化用户对系统的使用

✓ 设计视图时重新定义某些属性名，使其与用户习惯一致

✓ 为不同的使用者设计不同的视图。

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）

✓ 将一些复杂查询定义为视图

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要求的物理结构的过程，就是数据库的物理设计。
  - 对物理结构进行评价
    - 评价的重点是时间和空间效率，还有维护的代价

- 数据库查询事务，数据更新事务，每个事务在各关系上运行的频率和性能要求

物理结构设计

- 为关系模式选择存取方法（建立存取路径）
- 设计关系、索引等数据库文件的物理存储结构

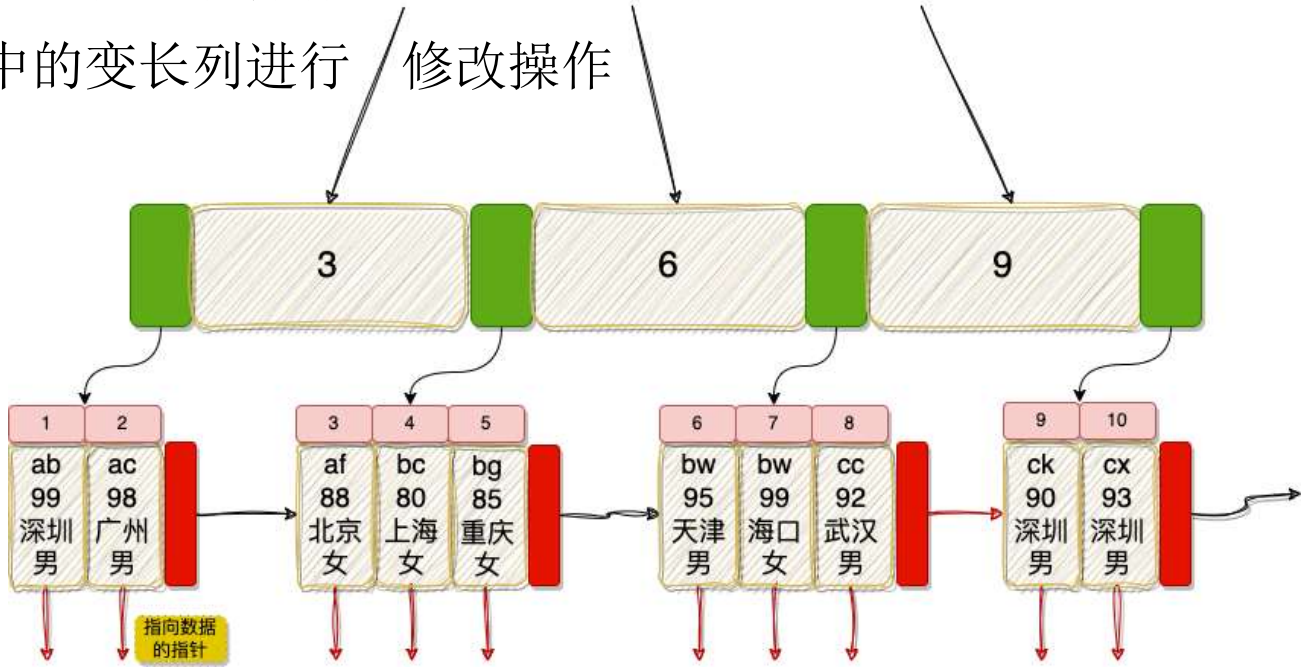
- 数据库管理系统常用存取方法
  1. B+树索引存取方法
  2. Hash索引存取方法
  3. 聚簇存取方法
- 选择索引存取方法的一般规则
  - 如果一个（或一组）属性经常在查询条件中出现
  - 如果一个属性经常作为聚集函数的参数
  - 如果一个（或一组）属性经常在连接操作的连接条件中出现

物理结构设计

- 选择Hash存取方法的规则
  - 如果一个关系的属性主要出现在等值连接条件中或主要出现在等值比较选择条件中
- 什么是聚簇
  - 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中称为聚簇。
  - 该属性（或属性组）称为聚簇码（cluster key）
  - 聚簇存放与聚簇索引的区别

物理结构设计

- 聚簇索引
  - 在一个基本表上最多只能建立一个聚簇索引
- 聚簇索引的适用条件
  - 很少对基表进行增删操作
  - 很少对其中的变长列进行修改操作



物理结构设计—聚簇

- 对于某些类型的查询，可以提高查询效率
  - ✓ 大大提高按聚簇属性进行查询的效率
  - ✓ 聚簇码值不必在每个元组中重复存储，节省存储空间
- [例] 假设学生关系按所在系建有索引，查询信息系的所有学生名单。
  - 计算机系的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。

- 聚簇的局限性
  - ✓ 聚簇只能提高某些特定应用的性能
  - ✓ 建立与维护聚簇的开销相当大

物理结构设计

- 为关系模式选择存取方法（建立存取路径）
- 设计关系、索引等数据库文件的物理存储结构

- 存储结构的设计
  - 确定每个表中有哪些字段，每个字段用什么数据类型。
- 易变部分与稳定部分分开存放，经常存取部分与存取频率较低部分分开存放
  - 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。

物理结构设计

- 数据库管理系统的**系统配置**
  - 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
  - 物理块的大小
  - 物理块装填因子
  - 时间片大小
  - 数据库的大小
  - 锁的数目等

➢ 要根据系统实际运行情况做进一步的调整，以切实改进系统性能。

数据库的实施和维护

- 数据库结构建立好后，组织数据入库是数据库实施阶段最主要的工作。
- 应用程序调试完成，且小部分数据入库后，就可以开始对数据库系统进行联合调试，也称数据库的试运行。
- 数据库试运行就是要实际测量系统的各种性能指标（不仅是时间、空间指标），如果结果不符合设计目标，则需要返回物理设计阶段，调整物理结构，修改参数；有时甚至需要返回逻辑设计阶段，调整逻辑结构。

数据库的实施和维护

- 数据库的转储和恢复
  - 在数据库试运行阶段，系统还不稳定，误操作也不可避免
  - 因此必须做好数据库的**转储和恢复**工作，尽量减少对数据库的破坏

➢ 数据库管理员完成：

1. 数据库的转储和恢复
2. 数据库的安全性、完整性控制
3. 数据库性能的监督、分析和改进
4. **数据库的重组织与重构造**

✓ 调整数据库的模式和内模式

- **增加或删除某些数据项**
- **改变数据项的类型**
- **增加或删除某个表**
- 改变数据库的容量
- 增加或删除某些索引

小结

- ✓ 数据库的6大设计过程，反复迭代
- ✓ 应用需求--概念模式， E-R图；
  --逻辑模式、外模式；--内模式
- ✓ 适当地修改、调整数据模型的结构，确定是否要对关系模式进行合并或分解
- ✓ 对数据库经常性的维护