

# 第七章 数据库设计

- 本章从**软件工程**的角度来剖析数据库设计的阶段性，比较系统地介绍数据库设计的全过程，主要讨论数据库设计的**方法**和**步骤**，详细介绍数据库设计的各个阶段的目标、方法和应注意的事项。

# 本章主要内容

- 数据库设计概述
- 需求分析
- 概念结构设计
- 逻辑结构设计
- 数据库的物理设计
- 数据库实施和维护

# 一、数据库设计概述

- 数据库已成为现代信息系统的基础与核心部分，已用于各类应用系统，如办公系统、管理信息系统等；
- 通常把使用数据库的各类信息系统都称为**数据库应用系统**；
- 在具备了DBMS、系统软件、操作系统和硬件环境的时候，对数据库应用开发人员来说，就是如何使用这个环境表达用户的要求，转换成有效的数据库结构，构成较优的数据库模式，这个过程称为**数据库设计**。

- 数据库应用系统的开发是一项软件工程，开发过程应遵循软件工程的一般原则和方法，但又应有自身的特点，所以特称为“**数据库工程**”；
- 数据库工程按内容可分为两部分：
  - 作为系统核心的数据库应用系统的设计与实现；
    - **设计数据库的各级模式并建立数据库**
    - 设计整个数据库应用系统
  - 相应的应用软件及其他软件（如通信软件）的设计与实现；

- **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**。
- 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。

## 数据库概述：

- 数据库设计的特点
- 数据库设计的方法
- 数据库设计的基本步骤
- 数据库设计过程中的各级模式

# 1、数据库设计的特点

- **数据库建设**

- 数据库应用系统从设计、实施到运行与维护的全过程；

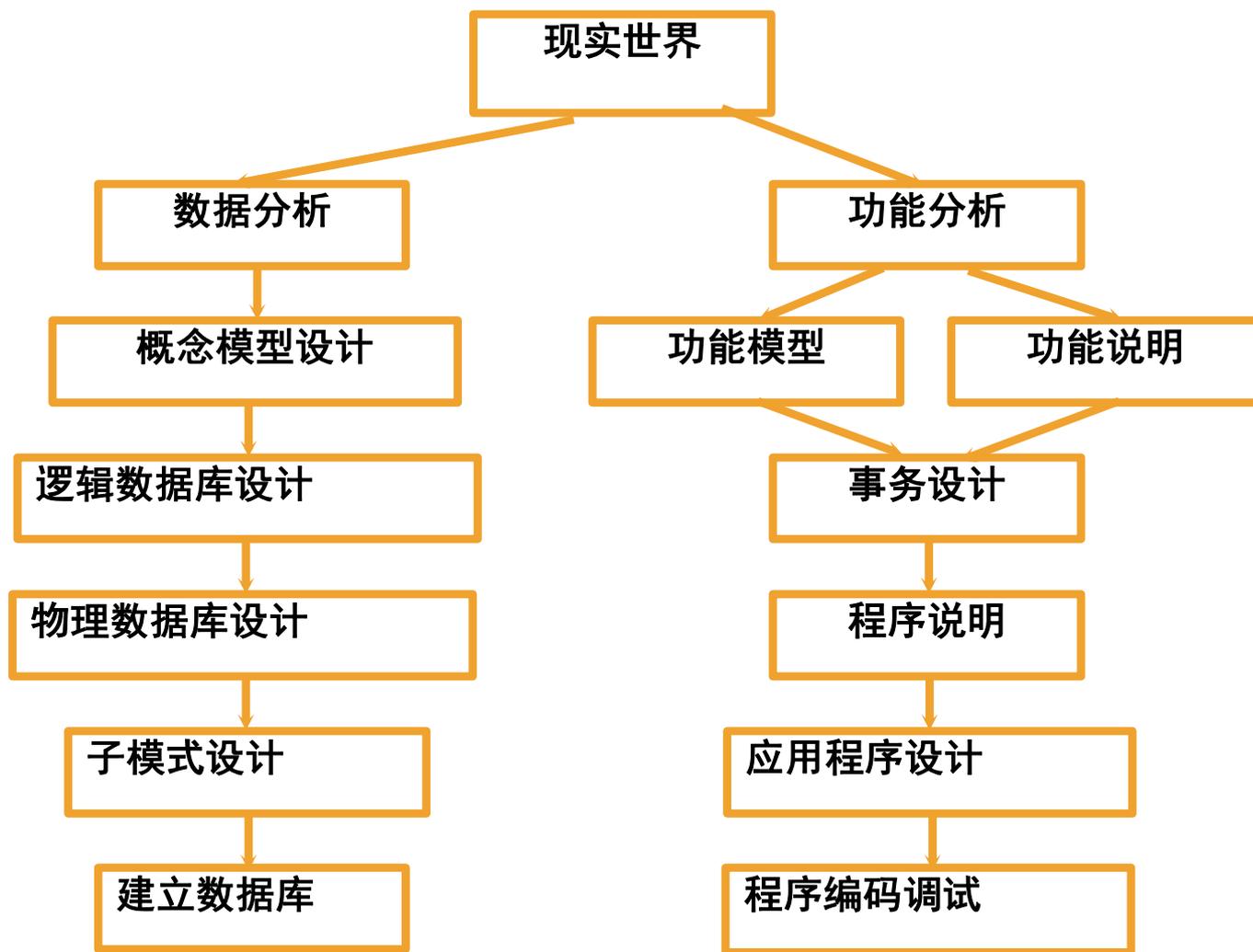
- 数据库建设的基本规律

- 三分技术，七分管理，十二分基础数据

- 整个设计过程中要把数据设计和处理设计密切结合起来

- 结构(数据)设计：设计数据库框架或数据库结构

- 行为(处理)设计：设计应用程序、事务处理等；



## 2、数据库设计的方法

- 大型数据库是涉及多学科的综合性的技术，又是一项庞大的工程项目。它要求从事数据库设计的专业人员具备多方面的技术和知识，主要包括：
  - 计算机的基础知识
  - 程序设计的方法和技巧
  - 软件工程的原理和方法
  - 数据库的基本知识
  - 数据库的设计技术
  - 应用领域的知识

- 早期采用手工与经验相结合方法
  - 设计质量与设计人员的经验和水平有直接关系
  - 数据库运行一段时间后常常不同程度地发现各种问题，需要进行修改甚至重新设计，增加了维护代价
- 规范设计法
  - 以科学理论和工程方法做支持
  - 基本思想：过程迭代和逐步求精

- **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**

- 设计数据库的概念模型，概念设计阶段广泛采用

- **3NF（第三范式）的设计方法**

- 以关系数据理论为指导，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

- 数据库设计工具

- 可辅助设计人员完成数据库设计过程中的很多任务

- Designer ——ORACLE

- PowerDesigner ——Sybase

- ERwin4.1

# 3、数据库设计的基本步骤

- 按照规范设计的方法，考虑数据库及其应用系统开发全过程，将数据库设计分为以下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
- 设计一个完善的数据库应用系统往往是上述六个阶段的不断反复；

## 数据库设计的准备工作：

### (1) 选定参加设计的人员

- **数据库分析及设计人员**：数据库设计的核心人员,自始至终参与数据库设计.其水平决定了数据库系统的质量。
- **用户和数据库管理员**：在数据库设计中也是举足轻重的。主要参加需求分析和数据库的运行维护。用户积极参与带来的好处是可以加速数据库的设计，提高数据库设计的质量。
- **程序员**：在系统实施阶段参与进来，负责编制程序。
- **操作员**：在系统实施阶段参与进来，准备软硬件环境。

## (2) 数据库设计的各个阶段的实施

- 1) 需求分析：
- 2) 概念结构设计
- 3) 逻辑结构设计
- 4) 数据库的物理设计
- 5) 数据库实施和维护

# 1 ) 需求分析阶段

- 数据库设计的“地基”；
- 简单地说，需求分析就是分析用户的需要与要求；
- 需求分析是设计数据库的起点，需求分析的结果是否准确地反映了用户的实际要求，将直接影响到后面各个阶段的设计，并影响到**设计结果是否合理和实用**；
- 最困难、最耗费时间的一步。

## 2) 概念结构设计阶段

- 需求分析阶段描述的用户应用需求是现实世界的具体需求
- 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**，该过程就是概念结构设计；
- 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，从而更加稳定。
- 概念结构设计是整个数据库设计的关键。

## 3 ) 逻辑结构设计阶段

- 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

## 4 ) 物理结构设计阶段

- 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，组织数据入库
  - 编制与调试应用程序
  - 进行试运行

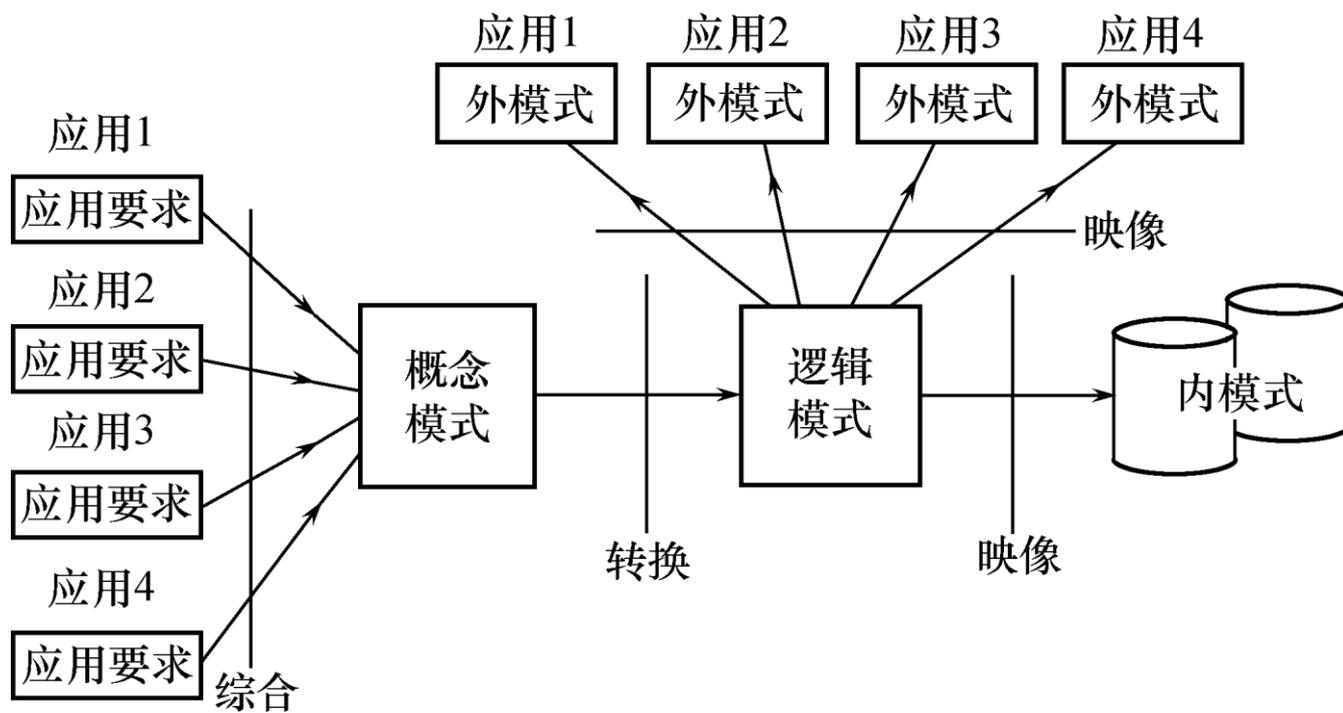
## 6 ) 数据库运行和维护阶段

- 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；

## 4、数据库设计过程中的各级模式

- 需求分析阶段：综合各个用户的应用需求；
- 概念设计阶段：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：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；
- 根据用户处理要求及安全性的考虑，在基本表上再建立必要的视图，形成外模式；
- 物理设计阶段：根据DBMS特点及要求，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；



## 二、需求分析

- 需求分析主要是考虑“做什么”的问题，而不是考虑“怎么做”的问题。
- **需求分析的设计目标**是通过详细调查现实世界要处理的对象（组织、部门、企业等），充分了解原系统的工作概况，明确用户的各种需求，然后在此基础上确定新系统的功能。

- **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

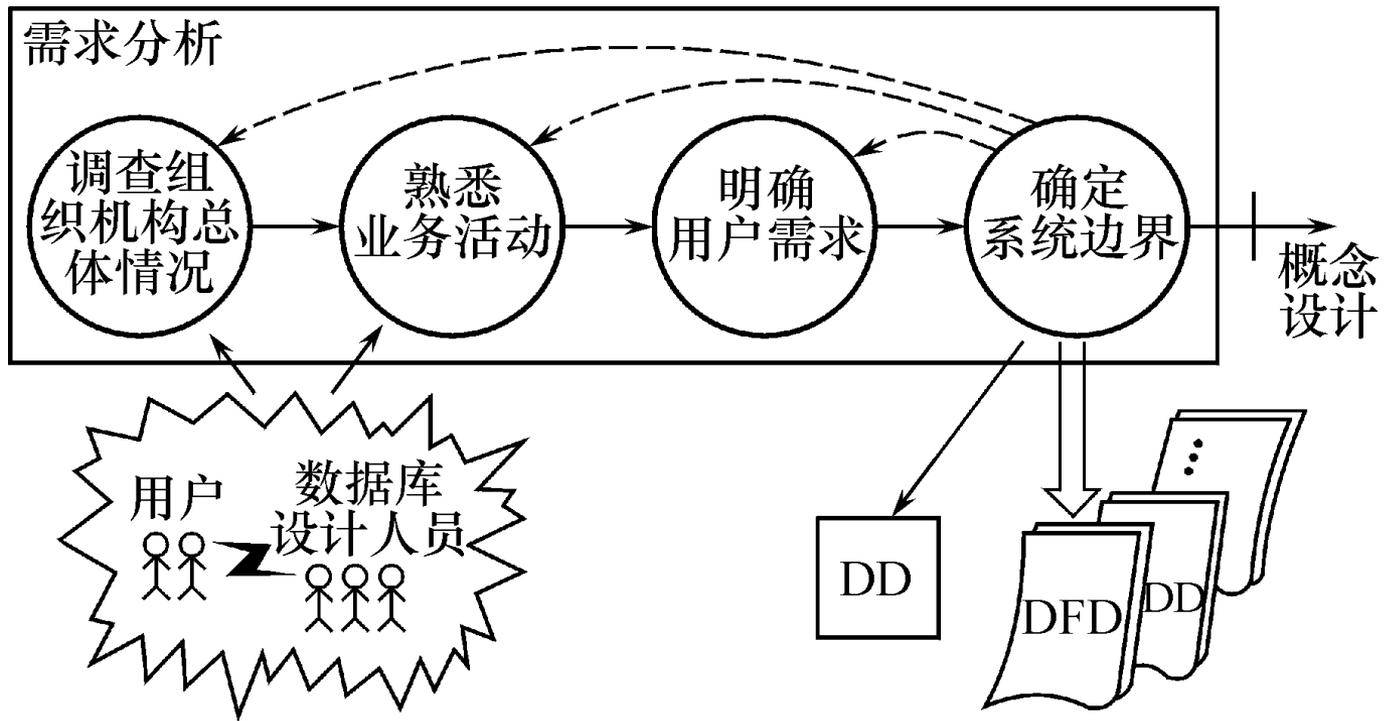
- 需求分析调查的重点是“数据”和“处理”，获得用户对数据库要求

- 信息要求
- 处理要求
- 安全性与完整性要求

## 需求分析的难点：

- 确定用户最终需求
  - 用户缺少计算机知识
  - 设计人员缺少用户的专业知识
- 解决方法
  - 设计人员必须采用有效的方法，与用户不断深入地进行交流，才能逐步得以确定用户的实际需求；

- 需求分析的方法
  - 首先是调查清楚用户的实际需求，与用户达成共识，然后分析与表达这些需求。
  - 调查用户需求的步骤：
    - 调查组织机构情况：部门职责、组成情况等；
    - 调查各部门的业务活动情况：用到的数据、输出信息、信息格式、信息流程等；
    - 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，主要包括信息要求、安全性完整性要求等；
    - 确定系统的边界：确定系统中哪些功能由计算机完成，哪些活动由人工完成；



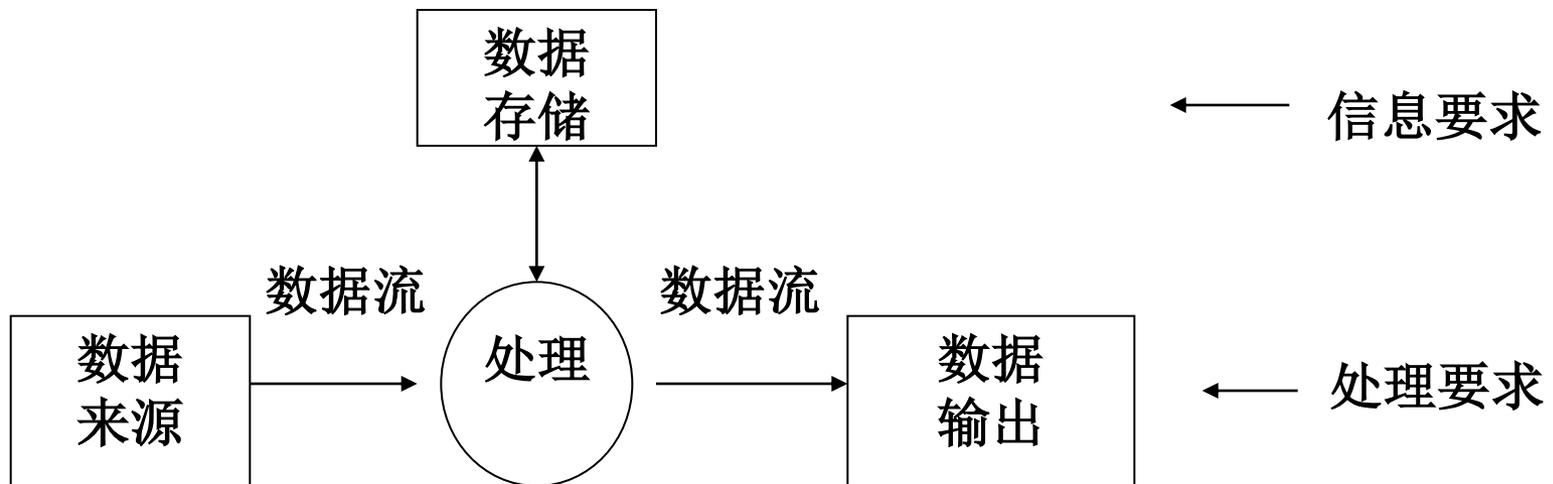
需求分析过程

- 做需求分析调查时，往往需要同时采用以下方法，但无论使用何种调查方法，都需要用户的积极参与和配合；常见的调查方法有：
  - 跟班作业
  - 开调查会
  - 请专人介绍
  - 询问
  - 设计调查表请用户填写
  - 查阅记录

- 分析和表达用户的需求的常用方法
  - 调查了解了用户需求以后，还需要进一步分析和表达用户的需求，一般采用**结构化分析**（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：



## 2. 分解处理功能和数据

- 分解处理功能：将处理功能的具体内容分解为若干子功能
- 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；
- 表达方法
  - 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

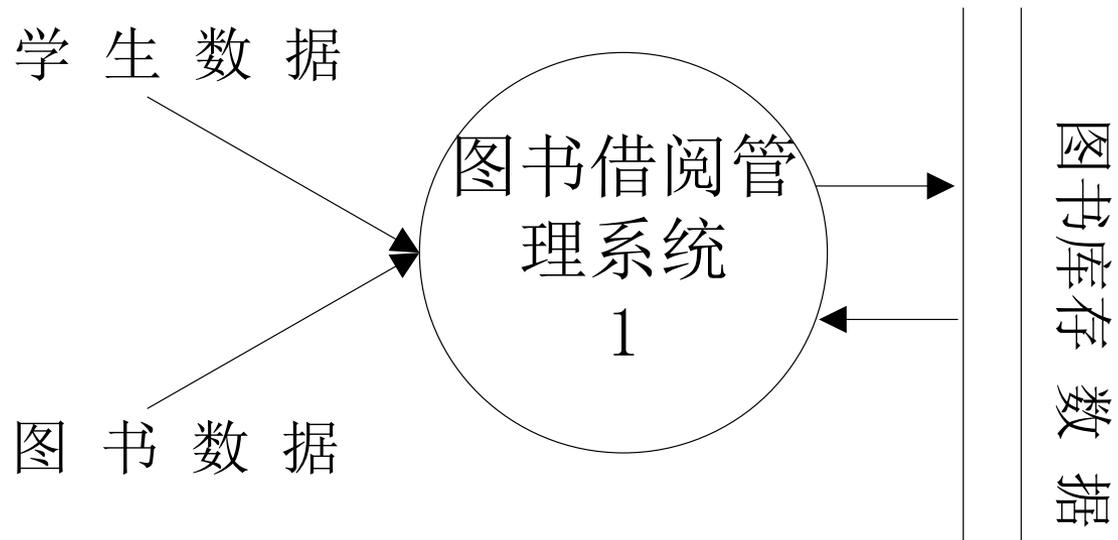
3. 将分析结果再次提交给用户，征得用户的认可。

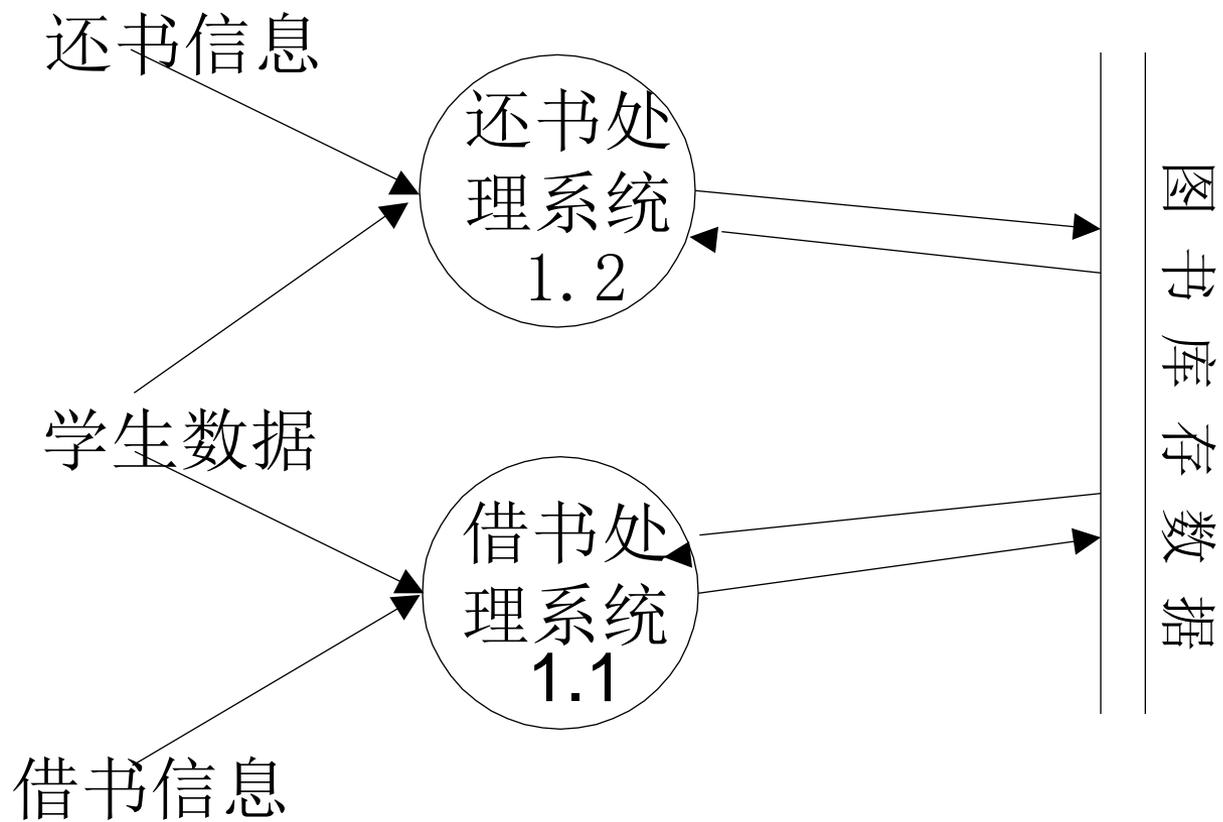
学 生 数 据

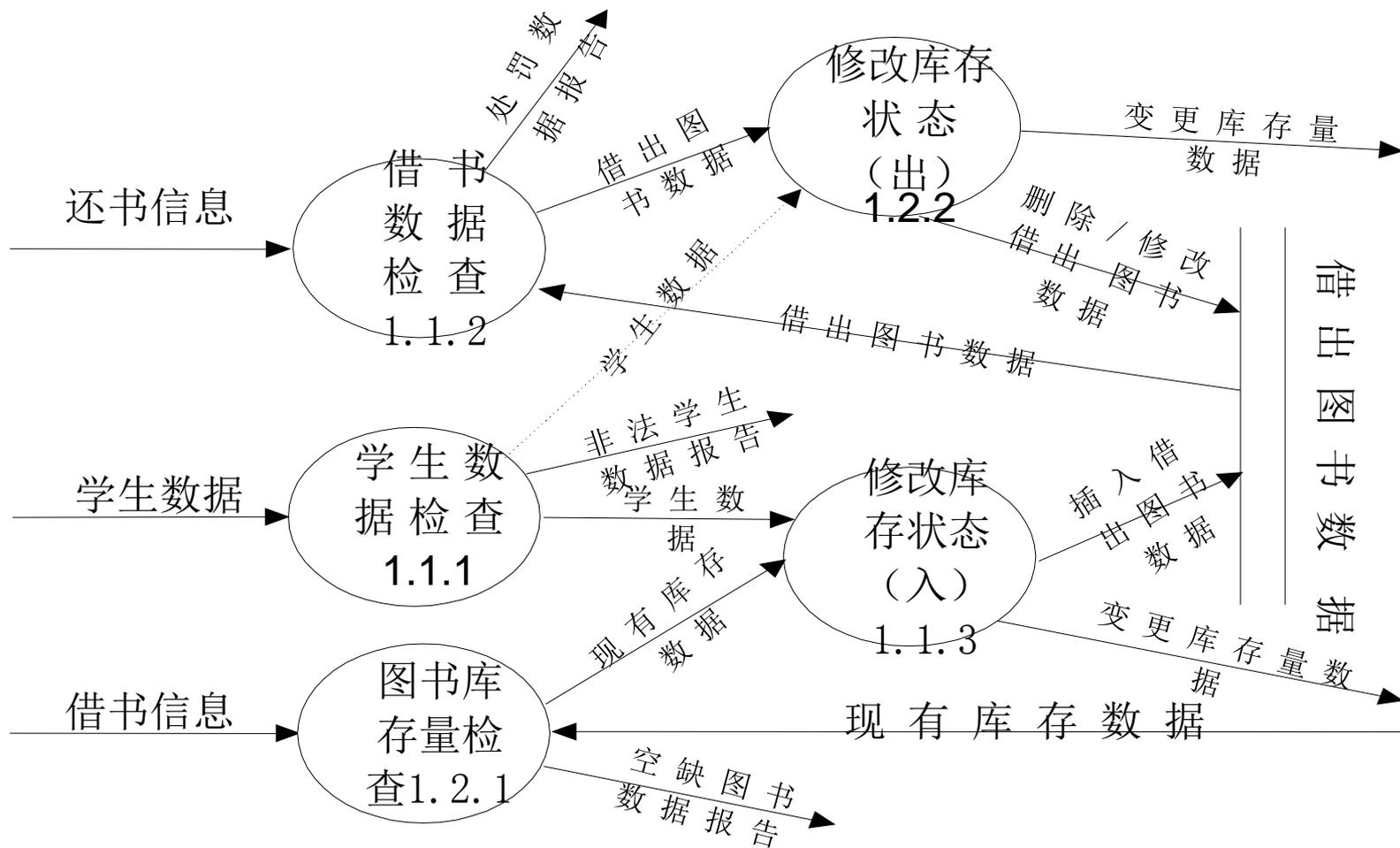
图 书 数 据

图书借阅管  
理系统  
1

图书库存数据







图书馆数据

# 数据字典

- 数据字典的用途
  - 是关于数据库中数据的描述
  - 在需求分析阶段建立，是下一步进行概念设计的基础
- 数据字典的内容
  - 数据项：数据的最小单位
  - 数据结构：若干数据项有意义的集合
  - 数据流：表示某一处理过程的输入或输出
  - 数据存储：处理过程中存取的数据
  - 处理过程：该过程的功能等

## 例：学生学籍管理子系统的数据字典

### 数据项

以“学号”为例：

数据项名：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

与其它数据库的逻辑关系：用于定义数据的完整性约束条件

数据项之间的联系：相关数据依赖

## 数据结构

反映了数据之间的组合关系；可有若干数据项、数据结构组成；

以“学生”为例

“学生”是该系统中的一个核心数据结构：

数据结构名：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：数据结构在系统内传输的途径；**

“体检结果”可如下描述：

数据流名： 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

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；**

“学生登记表”可如下描述：

数据存储名：学生登记表

说明：记录学生的基本情况

流入数据流：来源学生

流出数据流：去向

数据量：每年3000张

存取频度：.....

存取方式：随机存取

## 处理过程：用判定树或判定表来描述；

“分配宿舍”可如下描述：

处理过程名：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

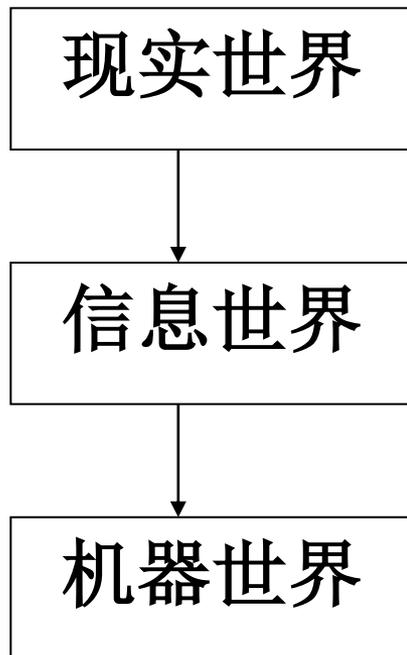
处理：        在新生报到后，为所有新生分配学生宿舍。要求同一间宿舍只能安排同一性别的学生，同一个学生只能安排在一个宿舍中。每个学生的居住面积不小于3平方米。安排新生宿舍其处理时间应不超过15分钟。

- 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。

- 明确地把需求分析阶段作为数据库设计阶段的第一阶段是十分重要的，也是下一步进行概念设计的基础。
- 注意：
  - 需求分析阶段的任务是收集将来应用所涉及的数据，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、易于扩充**；
  - 必须强调**用户的参与**，这是数据库应用系统设计的特点。设计人员应该和用户取得共同的语言，帮助不熟悉计算机的用户建立数据库环境下的共同概念，并对设计工作的最后结果承担共同的责任。

# 三、概念结构设计

- 什么是概念结构设计
  - 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，从而更加稳定；
  - 概念结构设计是整个数据库设计的关键。



需求分析

概念结构设计

- 概念结构设计的特点
  - 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

- **概念结构设计的四类方法**

- **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。

- 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

- 第1步：抽象数据并设计局部视图

- 第2步：集成局部视图，得到全局概念结构

- 概念结构是对现实世界的一种抽象
  - 从实际的人、物、事和概念中抽取所关心的共同特性，忽略非本质的细节；
  - 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。
- 三种常用抽象
  - 分类 (Classification)                      ——语义 is member of
  - 聚集 (Aggregation)                        ——语义 is part of
  - 概括 (Generalization)                      ——语义 is subset of

- 采用E-R方法的概念设计步骤：
  - 设计分E-R模式；
  - 合并分E-R模式，综合生成全局E-R模式；
  - 全局E-R模式的优化；
  
- 设计分E-R图的步骤：
  - 1) 选择局部应用
  - 2) 逐一设计分E-R图

# 1、设计分E-R图

## 1) 选择局部应用

- 通常以中层数据流图作为设计分E-R图的依据。
- 原因：
  - 高层数据流图只能反映系统的概貌；
  - 中层数据流图能较好地反映系统中各局部应用的子系统组成
  - 低层数据流图过细。

## 2) 逐一设计分E-R图

- 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）
- 作为属性的两条准则：
  - 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；
- 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。

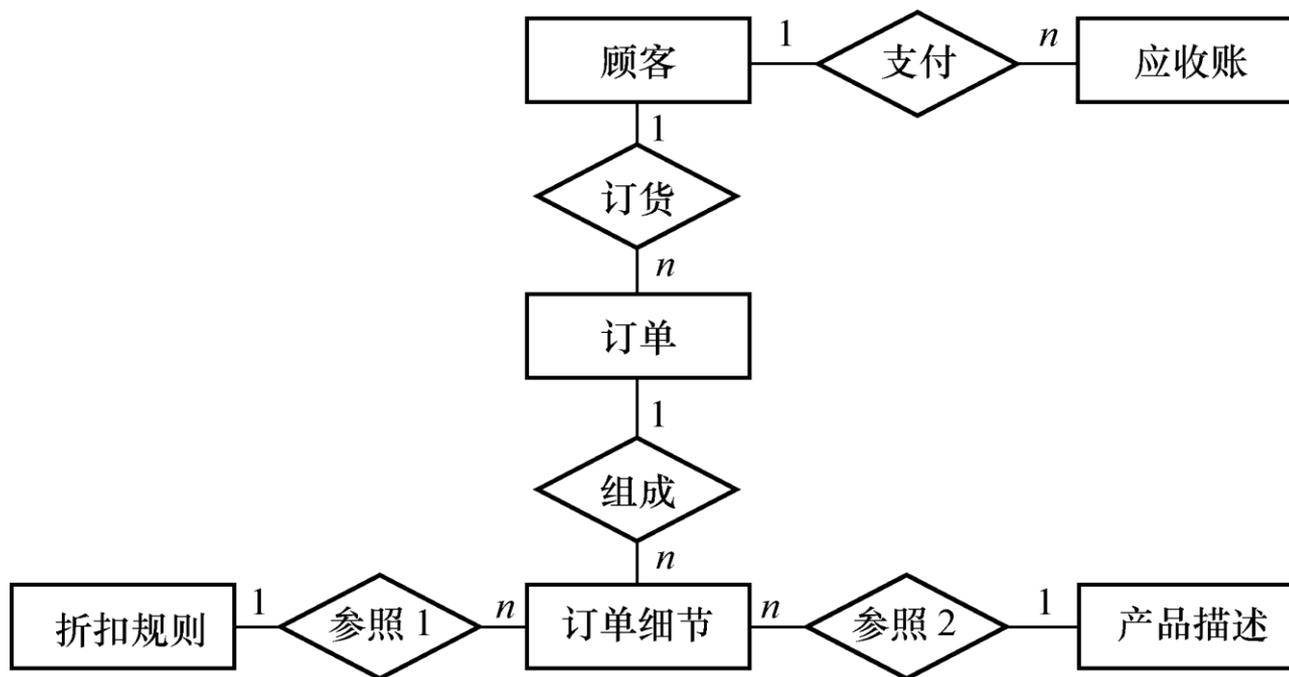
- 例1：“学生”由学号、姓名等属性进一步描述，根据准则 1，“学生”只能作为实体，不能作为属性。
- 例2：职称通常作为教师实体的属性，但在涉及工资、住房分配时，由于工资、分房与职称有关，也就是说职称与工资、住房实体之间有联系，根据准则 2，这时把职称作为实体来处理会更合适些。

## 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：

1. 处理顾客和销售员送来的订单；
2. 工厂是根据订货安排生产的；
3. 交出货物同时开出发票；
4. 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；

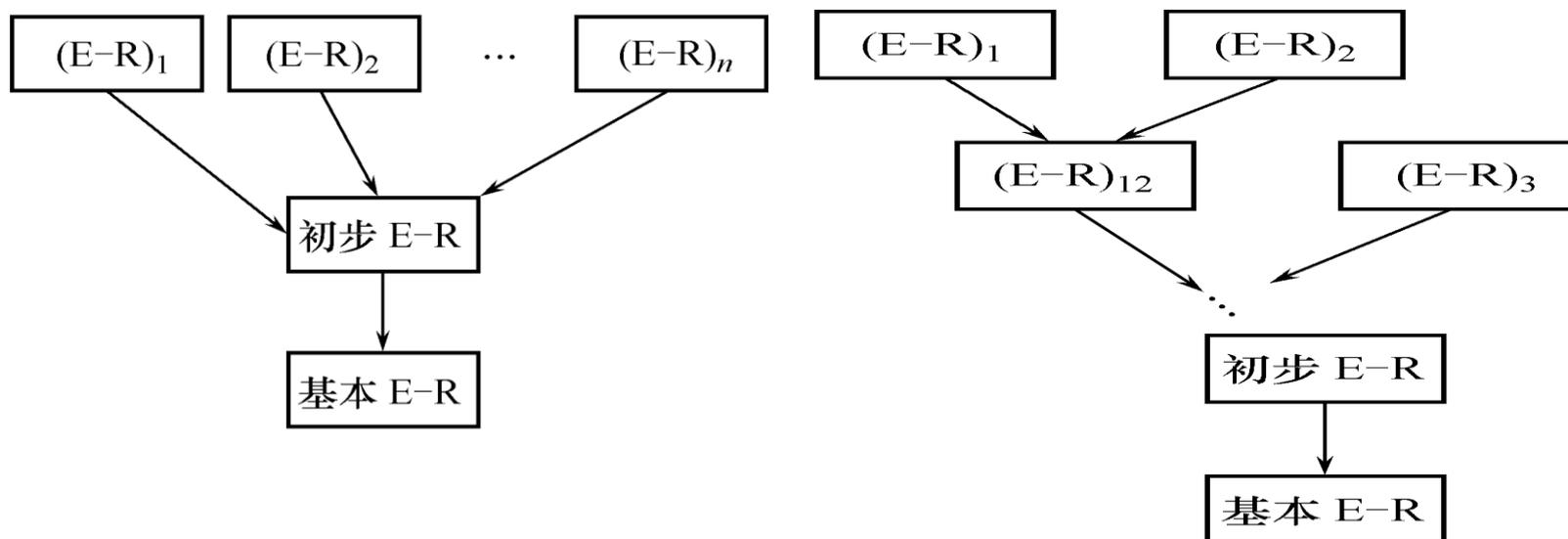
- 最终得到销售管理子系统分E-R图



销售管理子系统的分E-R图

## 2、合并分E-R图，生成全局E-R图

- 各个局部视图即分E-R图建立好后，还需要对它们进行合并，集成为一个整体的数据概念结构即总E-R图。可分为一次集成和逐步累积式；



- 存在问题：
  - 各分E-R图存在冲突
    - 各个局部应用所面向的问题不同；
    - 由不同的设计人员进行设计各个分E-R图之间必定会存在许多不一致的地方；
- 冲突的种类
  - 属性冲突，命名冲突，结构冲突
- 合并分E-R图的主要工作与关键
  - 合理消除各分E-R图的冲突

- **属性冲突**

- 属性域冲突（属性值的类型，取值范围，取值集合不同）
- 属性取值单位冲突

- **命名冲突**

- 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（一义多名）：同一意义的对象在不同的局部应用中具有不同的名字

- **结构冲突**

- 同一对象在不同应用中具有不同的抽象
- 同一实体在不同分E-R图所包含的属性个数和属性排列次序不完全相同
- 实体之间的联系在不同局部视图中呈现不同的类型

### 3、全局E-R模式的优化

- 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；冗余数据和冗余联系容易破坏数据的完整性，给数据库维护增加困难，应当予以消除；
- 消除了冗余后的初步E-R图称为基本E-R图；
  - 冗余的数据：指可由基本数据导出的数据
  - 冗余的联系：指可由其他联系导出的联系

- 消除冗余的方法
  - 分析方法
    - 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
  - 规范化理论
    - 函数依赖的概念提供了消除冗余联系的形式化工具
      - 确定实体间的数据依赖，求其最小覆盖

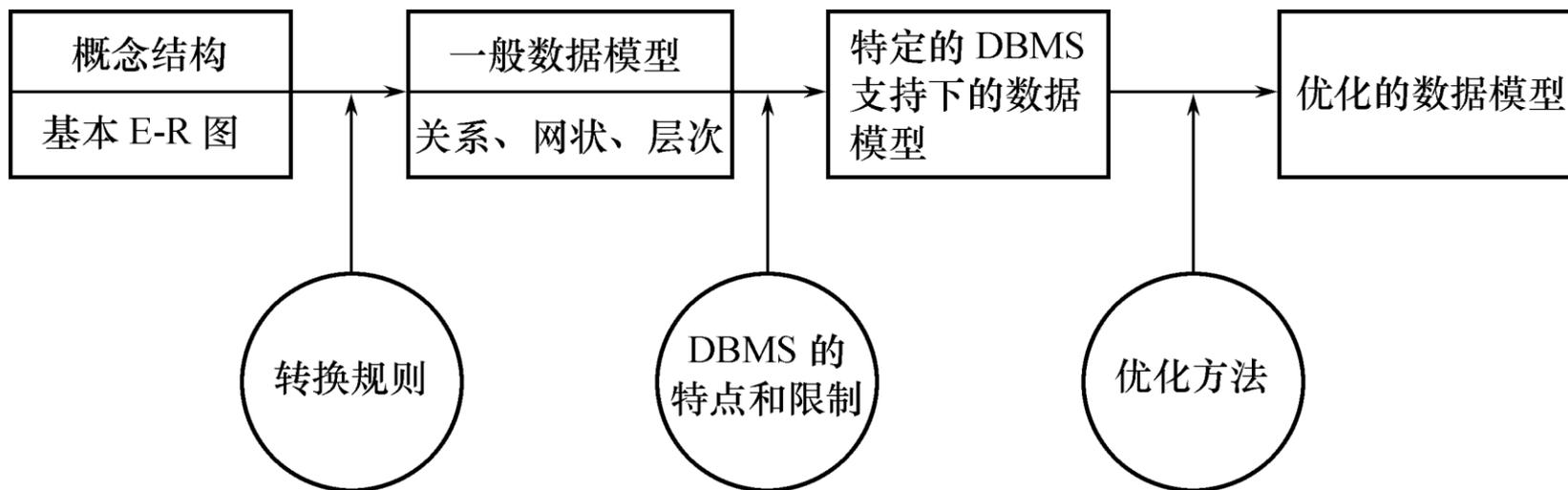
备注：设计数据库概念结构时，哪些冗余信息必须消除，哪些冗余信息允许存在，需要根据用户的整体需求来确定。

- 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。

# 四、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DBMS产品所支持的数据模型相符合的逻辑结构
- 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
  - 对数据模型进行优化



逻辑结构设计时的3个步骤

主要内容：

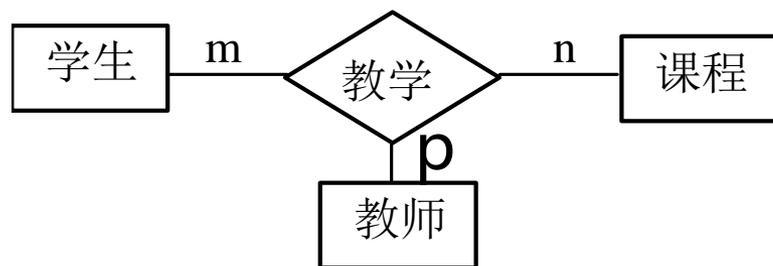
- E-R图向关系模型的转换
- 数据模型的优化
- 设计用户子模式

# 1、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
- 转换内容
  - 将E-R图转换为关系模型：将实体、实体的属性和实体之间的联系转换为关系模式。

- 转换遵循的原则：
  - 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。
    - 转换为一个独立的关系模式
    - 与某一端实体对应的关系模式合并
  - 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。
    - 转换为一个独立的关系模式
    - 与n端对应的关系模式合并
  - 一个m:n联系转换为一个关系模式

- 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;
  - 实体 $\Rightarrow$ 一个关系模式; 联系 $\Rightarrow$ 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, 即教学(XH, KH, JH, CJ)



- 具有相同码的关系模式可合并;
  - 目的：减少系统中的关系个数
  - 合并方法：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：

- 从理论上讲，1:1联系可以与任意一端对应的关系模式合并；
- 但在一些情况下，与不同的关系模式合并效率会大不一样。因此究竟应该与哪端的关系模式合并需要依应用的具体情况而定。
- 由于连接操作是最费时的操作，所以一般应以尽量减少连接操作为目标。例如，如果经常要查询某个班级的班主任姓名，则将管理联系与教师关系合并更好些。

## 2、数据模型的优化

- 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；

- 数据模型的优化方法：
  - 确定数据依赖（各属性之间的数据依赖）
  - 消除冗余的联系（进行极小化处理）
  - 确定所属范式
  - 确定是否要对某些模式进行合并或分解
  - 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；
- 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。

# 3、设计用户子模式

- 将E-R模式转换为逻辑模型后，还应根据局部应用需求，结合具体DBMS的特点，设计用户的外模式；
- 定义用户外模式时应该注重的
  - 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View，以满足系统对安全性的要求；
  - 简化用户对系统的使用。

# 五、数据库的物理设计

- 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；
- 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；

# 确定数据库的存储结构

- 确定数据库的存储结构
  - 数据的存放位置和存储结构
    - 包括关系、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；

# 1 ) 确定数据的存放位置

- 影响数据存放位置和存储结构的因素
  - 硬件环境
  - 应用需求
    - 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。

- 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

例：

- 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；
- 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；

## 2) 确定系统配置

- DBMS产品一般都提供一些系统配置变量、存储分配参数供设计人员和DBA对数据库进行物理优化。在初始情况下，系统为这些变量赋予合理的默认值，在进行物理设计时，需要重新对这些变量赋值，以改善系统性能。
- 系统配置变量包括：
  - 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
  - .....

# 六、数据库的实施和维护

- 数据的载入
- 应用程序的调试
- 数据库的试运行
- 数据库的运行和维护

# 1、数据的载入

- 数据库结构建立好后，就可以向数据库中装载数据了。组织数据入库是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

# 2、应用程序的编码和调试

- 数据库应用程序的设计应该与数据设计并行进行；
- 在组织数据入库的同时还要调试应用程序；

# 3、数据库的试运行

- 向数据库中输入小部分数据，就可以对数据库系统进行联合调试，这也称为数据库的试运行；
- 数据库试运行主要工作包括：
  - 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

## 4、数据库的运行和维护

- 数据库试运行合格后，数据库即可投入正式运行；
- 数据库投入运行标志着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

- 在数据库运行阶段，对数据库经常性的维护工作主要是由DBA完成的，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、分析和改进
  - 数据库的重组织和重构造

# 总结

- 学习本章，了解各阶段的基本方法，在实际工作中运用这些思想，设计符合应用需求的数据库应用系统。
- 数据库的设计过程
  - **需求分析**：准确了解与分析用户需求
  - **概念结构设计**：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
  - **逻辑结构设计**：将概念模型转化为DBMS所支持的数据模型，并对其优化
  - **物理设计**：为逻辑数据模型选取一个最合适应用环境的物理结构
  - **实施和维护**：在数据库运行过程中对其进行评价、调整与修改

- 本章讲解数据库设计方法和技術，內容的實踐性較強；
- 需要了解的：数据库設計的特點，数据库物理設計及数据库的實施和維護；
- 需要牢固掌握的：数据库設計的基本步驟，数据库設計各個階段的具體設計內容、描述及方法；
- 需要舉一反三的：E-R圖的設計， E-R圖向關係模型的轉換；
- 難點： E-R圖的設計及數據模型的優化