



# 数据库设计过程和方法



PART 01  
数据库应用  
设计概述



PART 02  
需求分析



PART 03  
概念结构设计



PART 04  
逻辑设计  
及优化



PART 05  
物理结构设计



PART 06  
实施和运  
行维护



## 学习目标：

- ✓ 了解数据库的设计方法
- ✓ 掌握软件和数据库开发生命周期
- ✓ 掌握数据库设计各阶段的主要工作内容
- ✓ 了解和掌握数据库设计的基本概念，设计原则



### 数据库应用设计：

- 是指根据用户需求研制数据库结构的过程。
- 也就是把现实世界中的数据，根据各种应用处理的要求，加以合理地组织，满足硬件和操作系统的特性，利用已有的DBMS来建立能够实现系统目标的数据库。
- 数据库设计包括：数据库的结构设计和数据库的行为设计两方面的内容。



### 直观设计法

依赖于设计者的经验和技巧，缺乏科学理论和工程原则的支持，设计质量难以保证。

### 新奥尔良设计法

新奥尔良法将数据库设计分成：需求分析、概念设计，逻辑设计和物理设计。

### 面向对象设计法

使用面向对象的概念和术语来描述和完成数据库的结构设计，并可方便地转换为面向对象的数据库。

### 计算机辅助设计工具

计算机辅助设计工具的开发和实现为更方便快捷地数据库开发提供了良好的条件。



- ◆ 软件危机：花费在软件维护上的资源以惊人的速度增长，导致许多软件工程项目延期、超过预算、可靠性低并且难于维护。
- ◆ 软件开发生命周期（SDLC）是解决软件危机的一个方法。



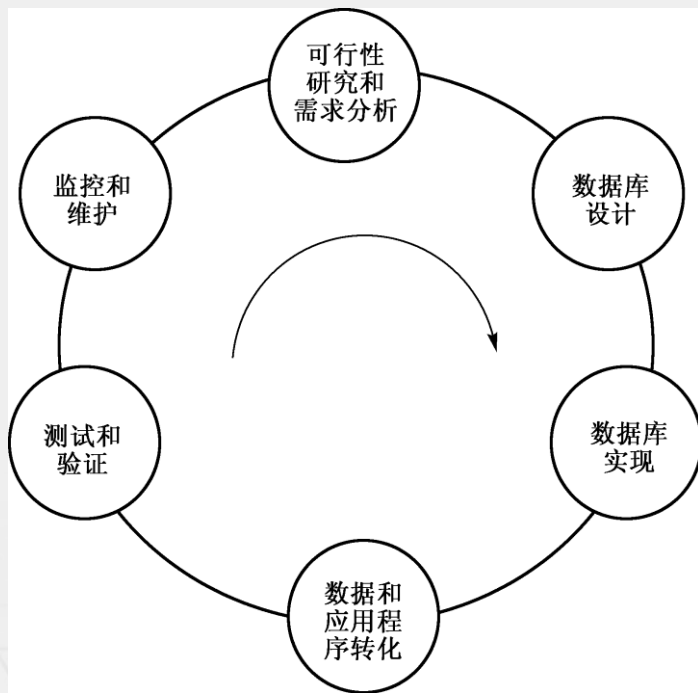
软件生命周期大体上可以分为以下几个阶段：

- 需要（或概念）阶段
- 规格说明阶段
- 计划阶段
- 设计阶段
- 编程（编码或实现）阶段
- 集成（测试）阶段
- 维护阶段
- 衰退阶段



数据库开发生命周期（DDLC）：是一个设计、实现和维护数据库系统的过程，需要符合组织战略和操作信息的需求。

从数据库系统的需求提出，到设计与实现，再到运行和维护，最终会随着应用的发展而走向终结。



数据库开发生命周期（DDLC）



## ■ 需求分析阶段

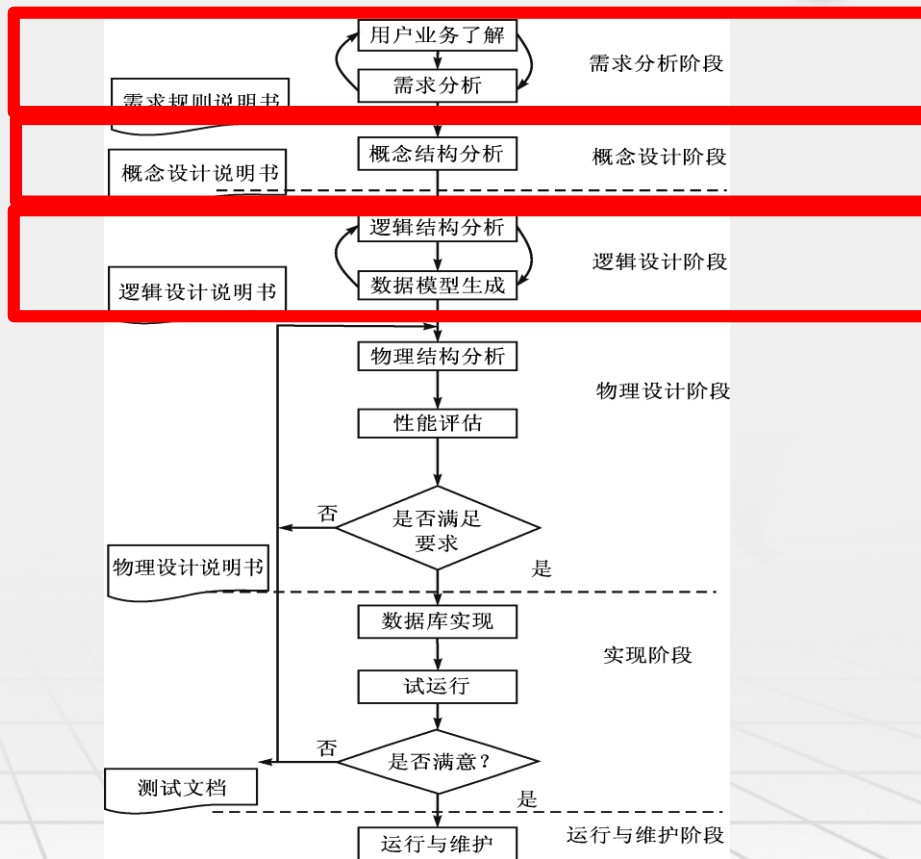
明确系统需要完成何种工作任务

## ■ 概念设计阶段

数据库概念设计将用户的需求抽象为用户与开发人员都能接受的概念模型，是用户现实需求与数据库产品之间的纽带。

## ■ 逻辑设计阶段

该阶段把抽象的概念结构进一步转换为可以被具体的DBMS产品所能支持的数据模型。







### ■ 物理设计阶段

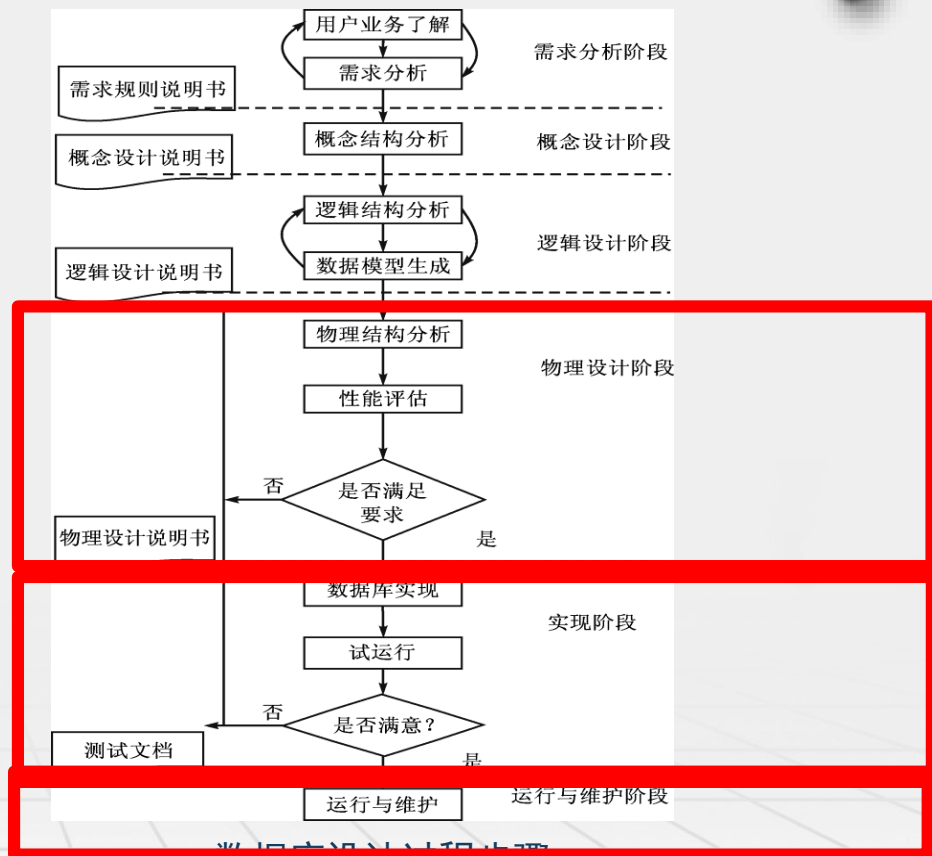
确定的物理存储结构，以及存取方法方法能否满足用户最终的需求。

### ■ 实现阶段

进行数据库的构建工作。包括针对数据库的应用程序开发和调试，以及现实数据的录入和试运行等基本工作。

### ■ 运行与维护阶段

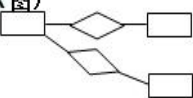
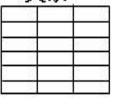
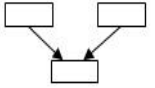

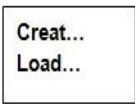
■ 保证数据库系统的效率，以及根据实际运行情况和用户的需求变动进行调整。





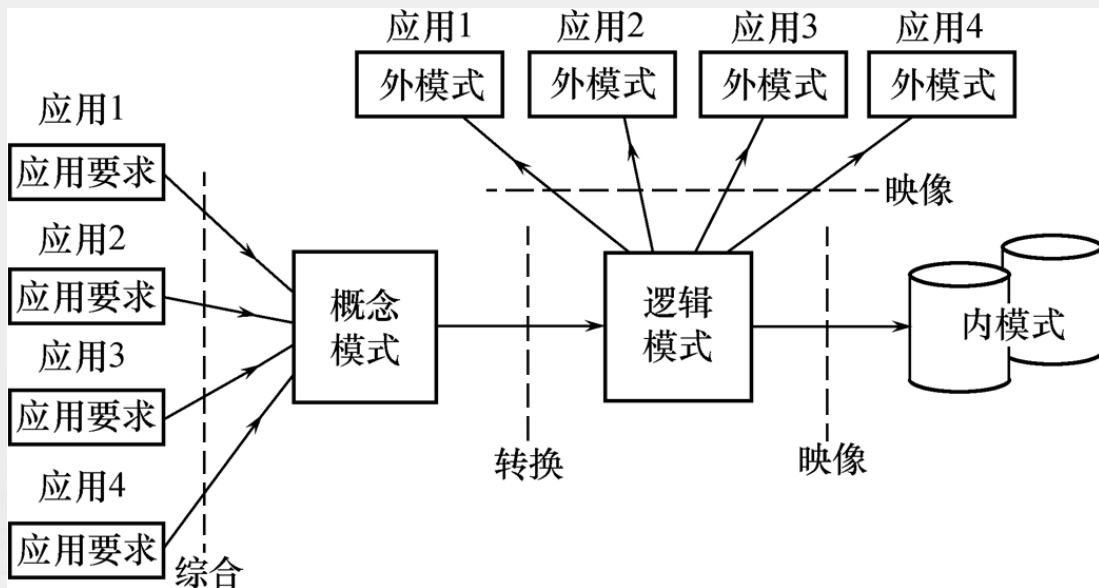
## 各阶段工作：

- ❑ 需求分析阶段综合各个**用户的应用需求**（**现实世界的需求**）；
- ❑ **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（信息世界模型）**，用**E-R图**来描述；
- ❑ **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**，
- ❑ 然后根据用户处理的要求，安全性的考虑，在基本表的基础上再建立必要的视图形成数据的**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，设计索引，形成数据库**内模式**。

设计阶段	设计描述
需求分析	数据字典、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



## — 数据库设计过程中的各级模式（对应阶段？）





- 在软件的生命周期中，需求分析是数据库设计的起点，是最为重要的一个阶段。
- 需求分析中最基本的一项原则就是必须要**正确理解客户的需求**。
- 需求分析包含：

需求的获取、分析、规则说明、变更、验证及管理等内容。

- 需求分析的基本任务：
  - 完成对产品开发的可行性研究
  - 调查应用领域
  - 对各种应用的信息和操作要求进行详细分析
  - 列出系统中所有的输入流、输出流和数据存储, 得到完整的数据流图、数据字典和数据加工的描述
  - 形成需求分析说明书



- 需求分析就是分析**用户的要求**
  - 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**（业务过程）
  - 明确用户的各种需求, 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
- 调查重点是“数据”和“处理”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
    - 由信息要求可以导出数据要求，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**

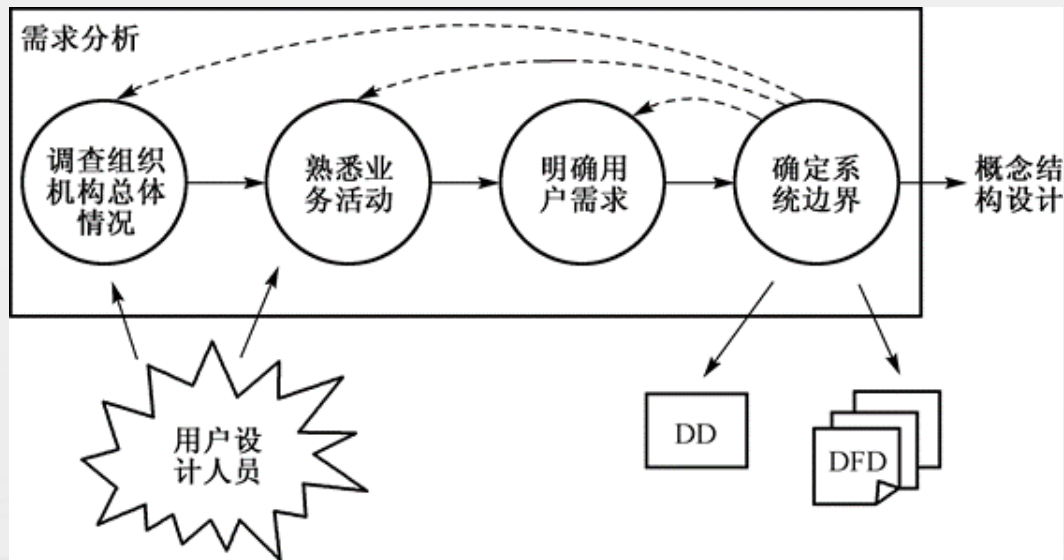


- 确定用户最终需求的难点
  - 用户缺少计算机知识，不能准确地表达自己的需求，他们所提出的需求往往不断地变化。
  - 设计人员缺少用户的专业知识，不易理解用户的真正需求，甚至误解用户的需求
- 解决方法
  - 设计人员必须不断深入地与用户进行交流，才能逐步确定用户的实际需求



需求分析就是从用户的需求角度出发，对用户所期望的产品进行详细描述。

需求分析的过程如下：



需求分析的过程



# 需求分析

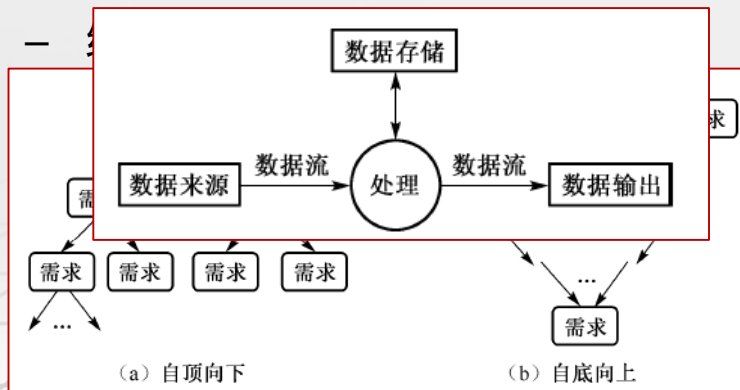
## 需求分析的步骤



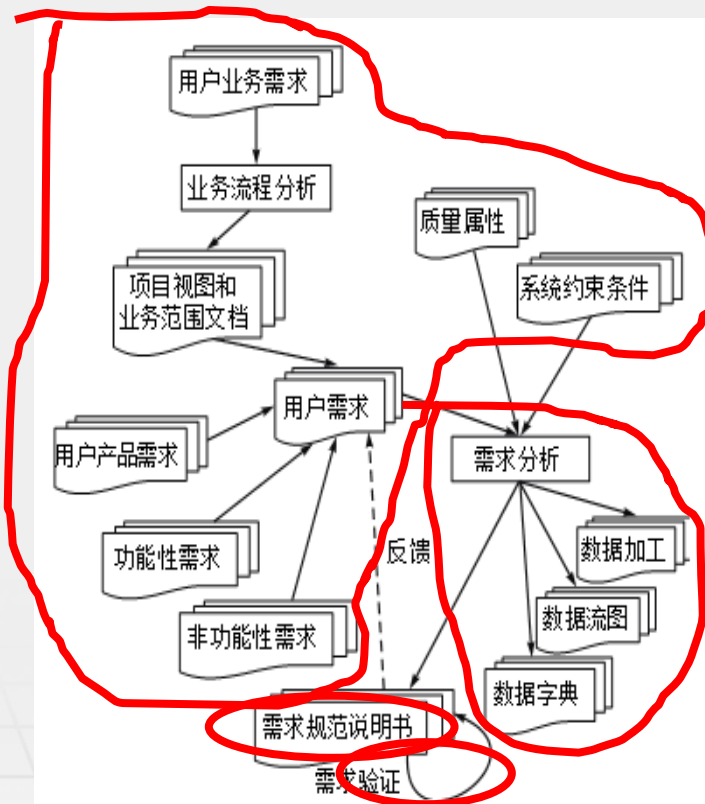
### 一 用户活动的调查分析

编写的目的、背景和定义。  
用户的特点和系统的目标。  
系统的概况。  
系统的总体结构和子系统结构划分。  
通信接口定义。  
系统功能需求说明。  
系统非功能需求说明。  
数据处理流程的描述，数据管理能力的要求。  
系统方案的可行性论证。  
运行环境和故障处理的要求。

结构化方法、原型化方法、  
数据流分析方法等。



• 完备性验证







### 进行需求分析

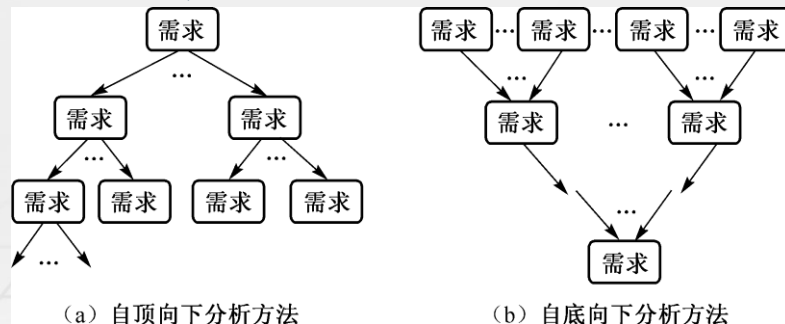
常用的需求分析方法有：结构化方法、原型化方法、数据流分析方法等。

- 原型化方法 (Prototype)

一种可快速、形象地描述用户产品需求的分析方法。

- 结构化分析方法 (Structured Analysis)

一般使用自顶向下，或者是自底向上的分析方式。

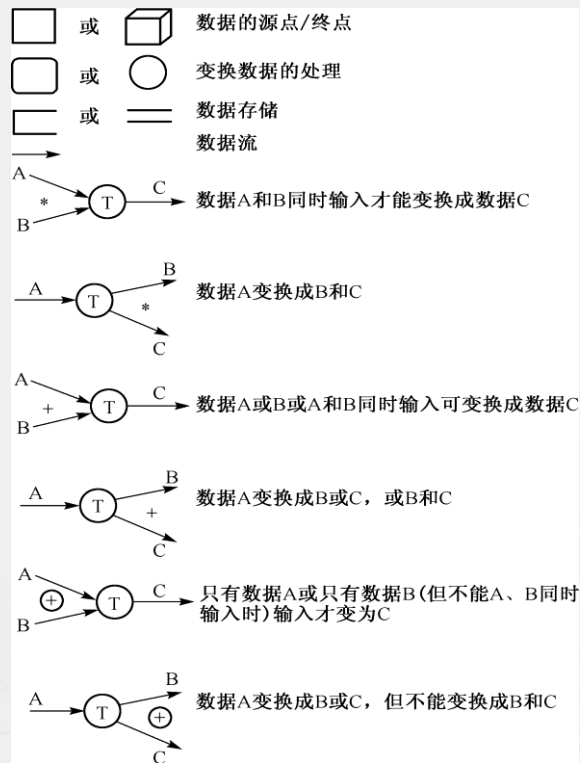


### 结构化分析方法

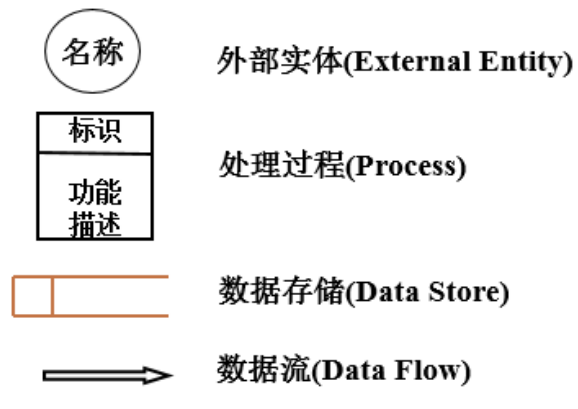
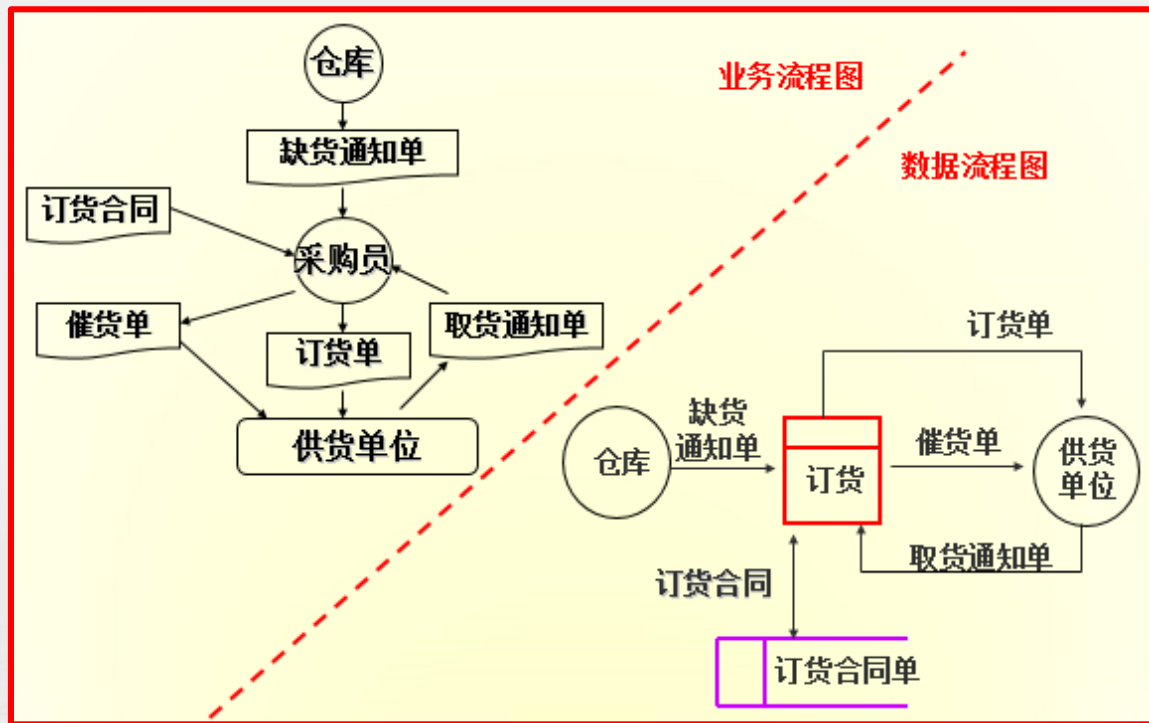


SA方法采用**自顶向下**，**逐层分解**的方式分析系统，  
用**数据流图**（Data Flow Diagram, DFD）、**数据字典**  
（Data Dictionary, DD）描述系统。

数据流图：是软件工程中专门描绘信息在系统中流  
动和处理过程的图形化工具。图中给出了数据流图中  
所使用的符号及其含义。



数据流图的符号





数据字典存放了系统所用到的所有数据信息，包含了5个基本组成部分：



## 数据项

记录了数据对象的基本信息，是不可再分的基本数据单位，描述了数据的静态特性。



## 数据结构

反映了数据之间的组合关系，也可以是由多个数据结构的复合。



## 数据流

是对数据动态特性的描述，表示了数据结构沿着系统的事务和处理过程中的传输流向。



## 数据存储

是在事务和处理过程中，数据所停留和保存过的地方。



## 数据处理过程

仅是对处理相关信息的简要描述。



- 思考：经过需求分析学习后，你现在会怎样做“教务管理数据库”的需求分析呢？



- 思考：经过需求分析学习后，你现在会怎样做“图书管理数据库”的需求分析呢？





- 概念设计：系统的开发者将现实世界中存在的具体要求，抽象成信息结构的表达方式的转换过程，以方便选择具体的DBMS进行实现。
- 概念结构设计是以用户的观点，对用户信息进行抽象和描述。仅从用户的需求角度来抽象概念模型。
- 概念设计是从现实世界到信息世界的第一次抽象，并不考虑具体的数据库管理系统。
- 最常用的模式就是**实体-关系模型（E-R）**。



## 概念设计方法

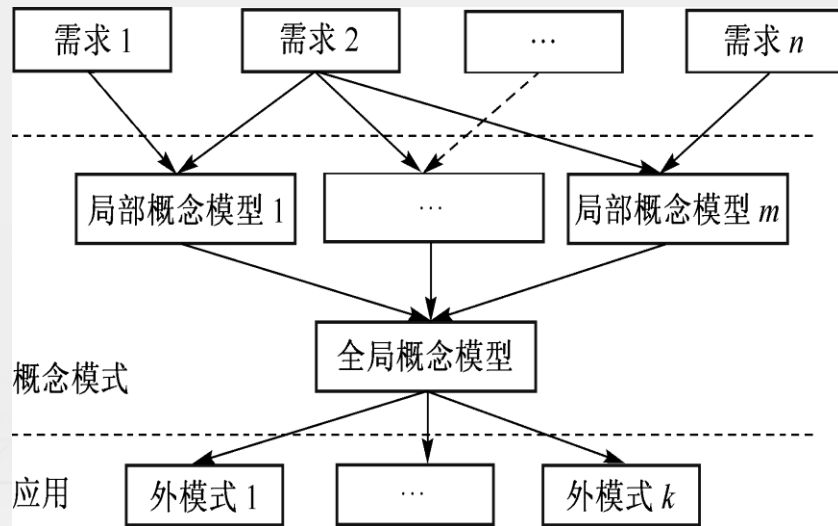
常用的概念设计有4类方法：

- 自底向上
- 自顶向下
- 逐步扩张
- 混合策略



### 自底向上

通过分析用户的子需求，首先构建起局部概念模式，然后再向上组合成全局模式。用户子需求相对较为具体和确定，只需要把有关联的需求模块进行合并，就能较为方便地建立起局部概念模式。

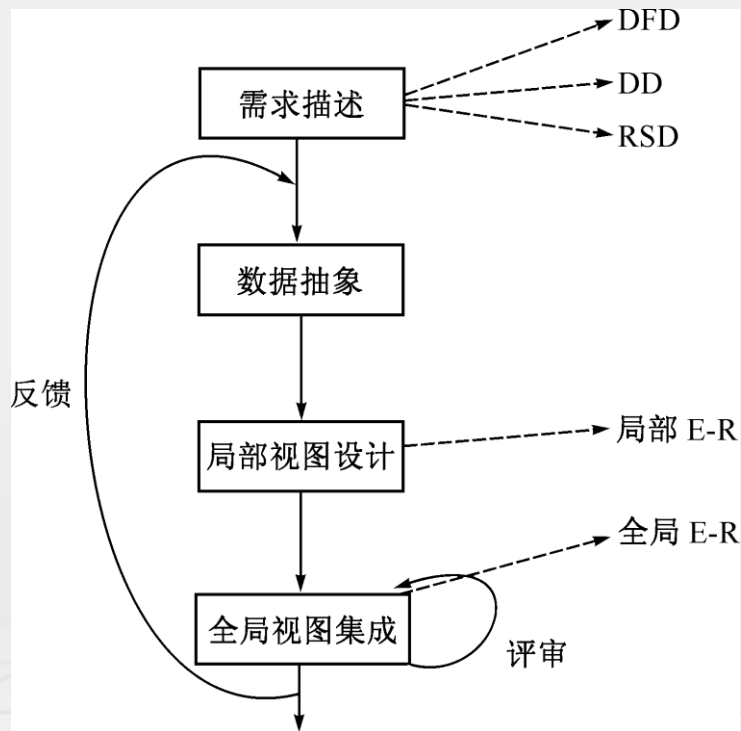


自底向上概念模式生成





采用典型的自底向上设计方式时，概念模式的设计过程可分为3个基本步骤完成：



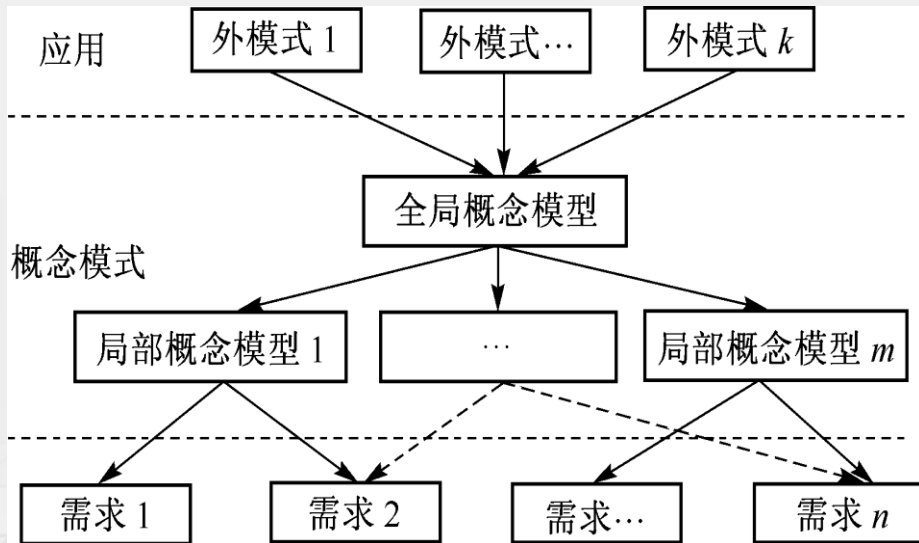
- 数据的抽象，设计局部概念模式
- 视图的集成，将局部概念模式综合成全局概念模式
- 对生成的概念模式进行评审

自底向上的概念模式设计步骤



### 自顶向下

首先要有对系统全局概貌的框架，其次再采用总分方式将大的概念模式逐步分解为更详细的较小划分模式。



### 自顶向下概念模式生成



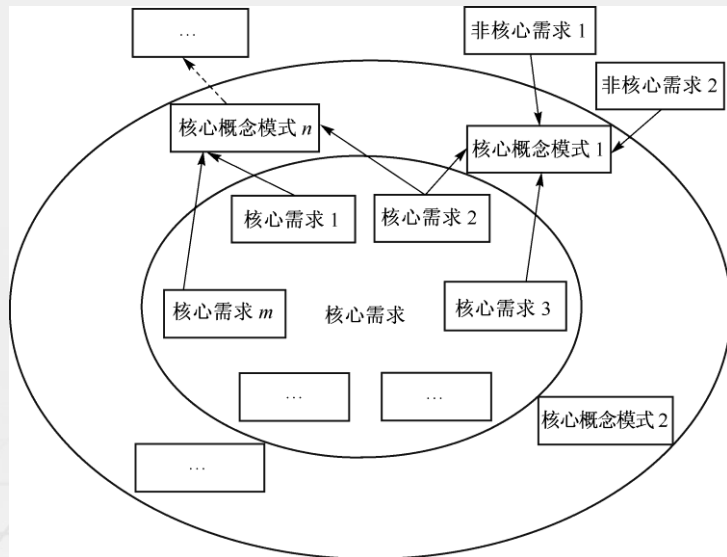
自顶向下的概念模式构建方式相比自底向上的方式更加难以控制

- ❑ 直接抽象出全局的概念模式，再分解为局部的概念模式，并通过局部模式投影到各个子需求模块上
- ❑ 这是一个从抽象到具体的逆向思维的过程



### 逐步扩张

采用了层状扩展的方式，先定义出用户需求中核心的概念结构，然后在此基础上向外扩展，逐步将非核心的需求融入到模式中，最终完成系统的概念结构设计。



逐步扩张设计方法



## 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，以它为骨架集成自底向上策略中涉及的各局部概念结构。

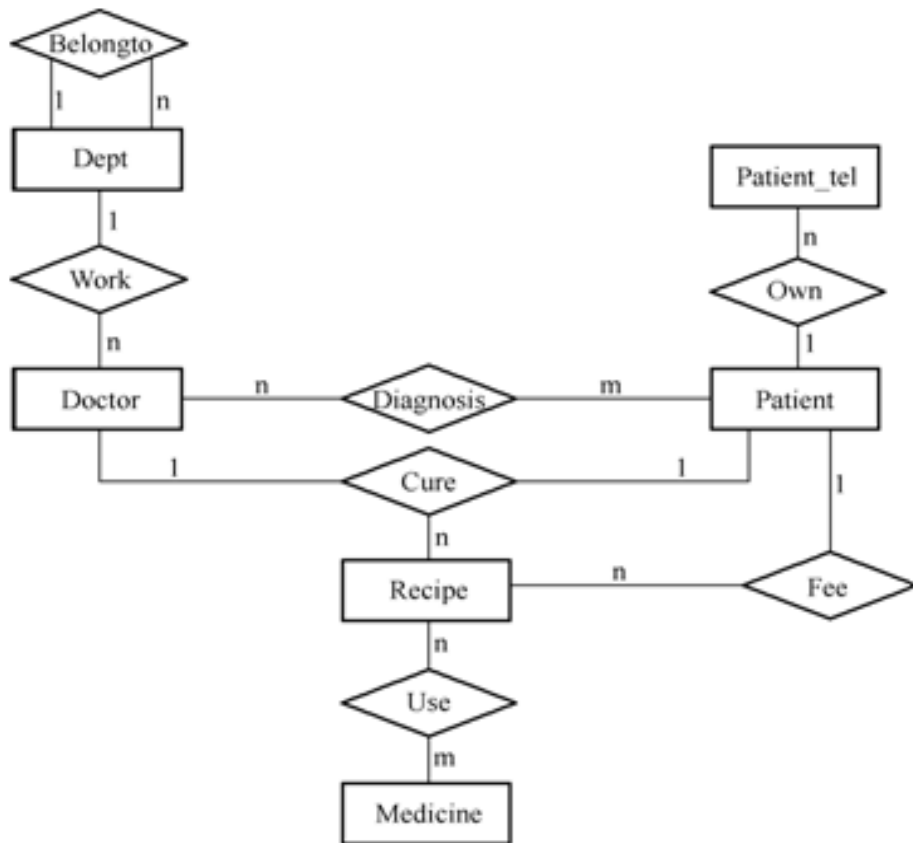


这些概念模型的设计方法也可以混合地使用，例如经常采用的方法：

- 自顶向下地分解用户的需求
- 自底向上地完成核心概念模式的设计
- 再围绕核心模式逐步扩展完成整个数据库的概念设计工作



E-R 图方法是“实体-联系方法”的简称，是一种非常典型的数据库概念模式设计方法。能够很好地抽象出现实世界的基本内容，并以图形化的表现方式为概念设计人员与用户提供对数据库系统的直观表达。





## 数据库逻辑结构设计：

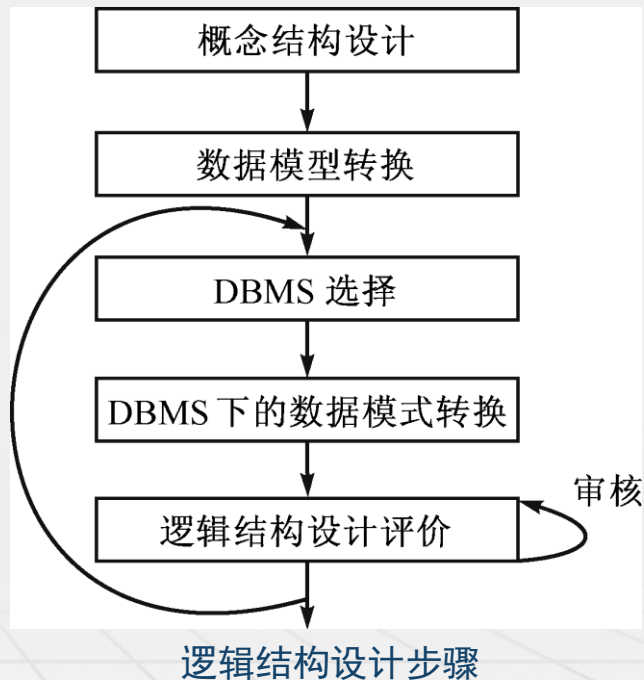
- 从数据库逻辑设计所导出的数据库结构则是特定DBMS支持下的数据库定义，因此逻辑设计依赖于实现的DBMS基础。
- 逻辑结构设计的任务就是为概念结构设计阶段生成的全局E-R视图，转换为特定DBMS产品所能支持的数据模型。





### 逻辑设计的3个步骤：

1. 将概念模式转换为适合的3类常用关系、网状、层状数据模型中的一种；
2. 转换后的数据模型确定DBMS产品，并在该环境下实现数据模型向DBMS所支持的数据模型转换；
3. 对完成转换后的数据模型进行优化和评估。





- E-R模式

- 描述的是现实世界中数据对象，以及数据对象之间的联系。

- 关系模型

- 关系模型的逻辑结构则是一组关系模式的集合。

- E-R模式向关系模式的转换

- 最主要的工作就是要确定如何将E-R中的3个基本要素（实体、联系、键），转换为关系模式集合的表示。



E-R模式向关系模式的转换可遵循下述的转换规则：

### ● 实体转换规则

将一个实体转换为一个关系模式，实体的属性就是关系的属性，而实体的键就是关系的键。

### ● 联系转换规则

实体之间的联系转换为关系模式，联系的属性直接转换为关系的属性。与联系相连的实体的键转化为关系模式时，则分下述3种情况考虑：

- 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。
- 一个1:n的联系：可以转换为一个独立的关系模式，并与n端所对应的关系模式合并。
- 一个m:n联系：转换为一个关系模式。



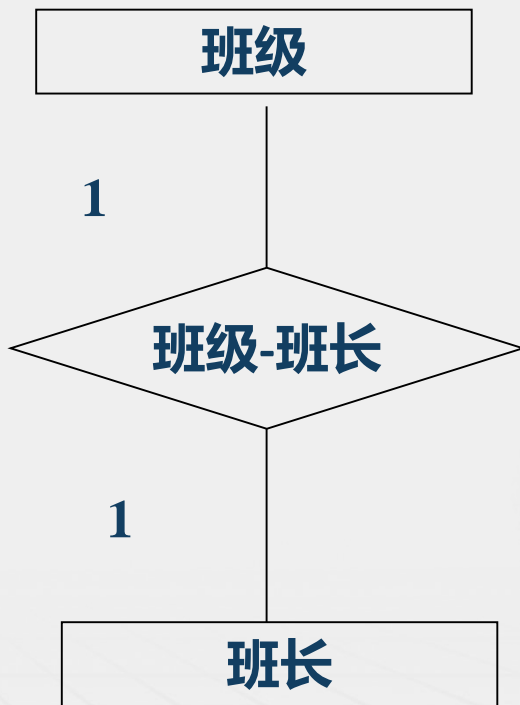
### ➤ 两个实体型之间的联系

#### ◆ 一对一联系(1:1)

- ◆ 如果对于实体集A中的每一个实体，实体集B中至多有一个实体与之联系，反之亦然，则称实体集A与实体集B具有一对一联系。记为1:1。

#### ◆ 实例：班级与班长之间的联系：

- ◆ 一个班级只有一个正班长
- ◆ 一个班长只在一个班中任职



**1:1联系**



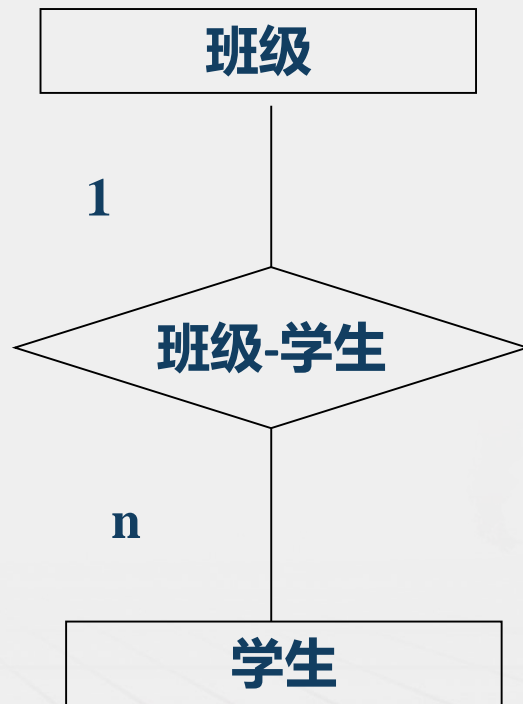
### ➤ 两个实体型之间的联系

#### ◆ 一对多联系(1:n)

- ◆ 如果对于实体集A中的每一个实体，实体集B中有n个实体 ( $n \geq 0$ ) 与之联系，反之，对于实体集B中的每一个实体，实体集A中至多只有一个实体与之联系，则称实体集A与实体集B有一对多联系记为1:n

#### ◆ 实例：班级与学生之间的联系

- ◆ 一个班级中有若干名学生，
- ◆ 每个学生只在一个班级中学习



**1:n联系**



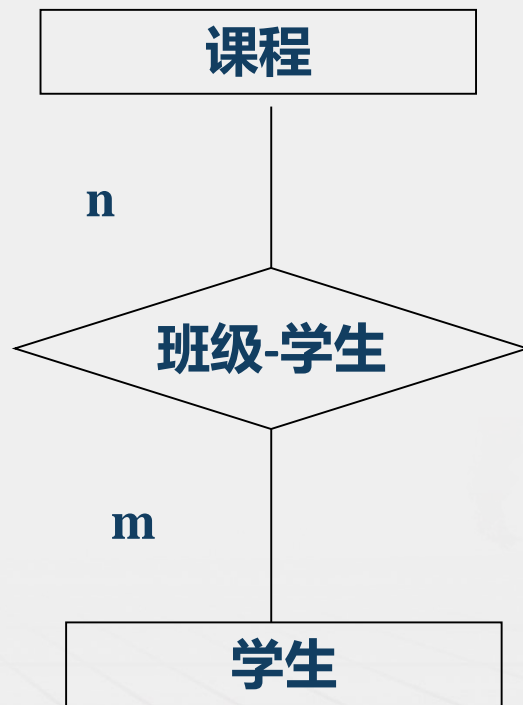
### ➤ 两个实体型之间的联系

#### ◆ 多对多联系 ( $m:n$ )

- ◆ 如果对于实体集A中的每一个实体，实体集B中有 $n$ 个实体 ( $n \geq 0$ ) 与之联系，反之，对于实体集B中的每一个实体，实体集A中也有 $m$ 个实体 ( $m \geq 0$ ) 与之联系，则称实体集A与实体B具有多对多联系。记为 $m:n$

#### ◆ 实例: 课程与学生之间的联系

- ◆ 一门课程同时有若干个学生选修
- ◆ 一个学生可以同时选修多门课程

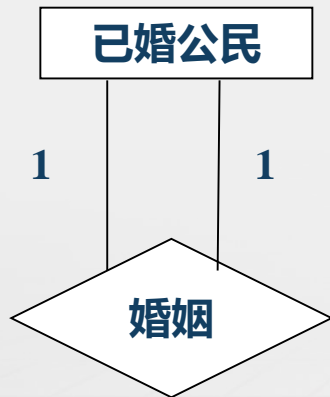


**$n:m$ 联系**

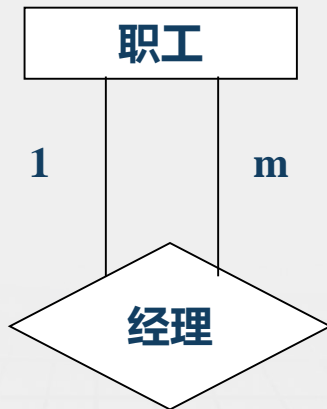


### ● 同一实体型之中的联系

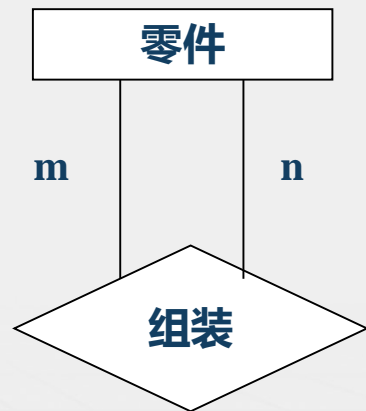
- 一对一联系
- 一对多联系
- 多对多联系



同一实体型内部的1:1联系



同一实体型内部的1:m联系

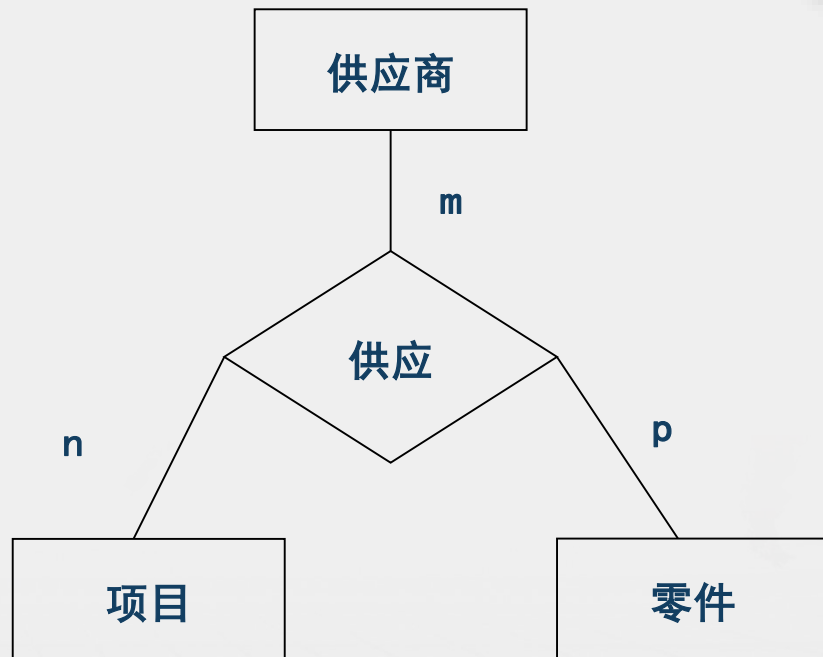


同一实体型内部的m:n联系



### ● 多个实体型之间的联系

- 一对多联系
- 一对一联系
- 多对多联系

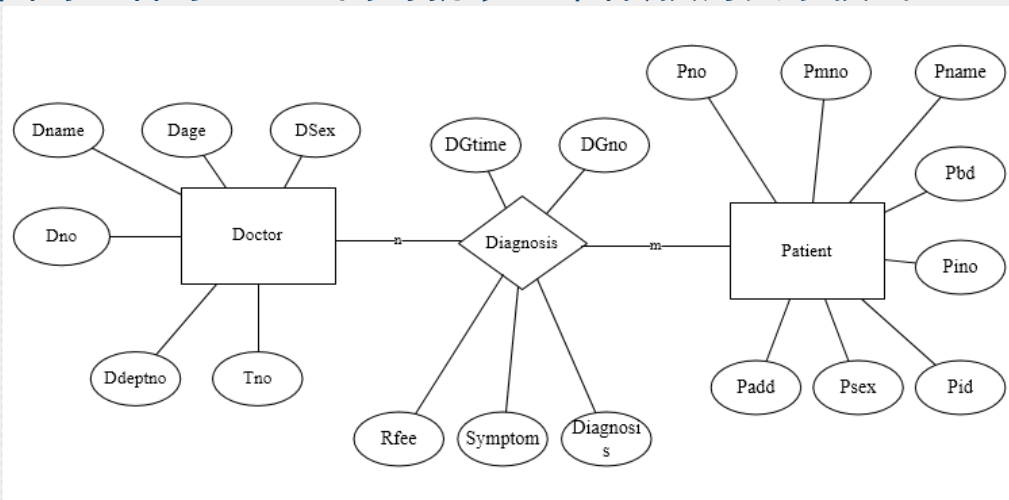


多个实体型之间  
多对多的联系





【举例】将下图的患者与医生之间的就诊E-R图转换为关系模式。



解:

① 将Patient（患者）实体转换为Patient模式如下:

Patient (Pno, Pmno, Pname, Pbd, Pino, Pid, Psex, Padd)

② 将Doctor（医生）实体转换为Doctor模式如下:

Doctor (Dname, Dage, Dsex, Dno, Ddeptno, Tno)

③ 将Diagnosis（就诊）联系转换为Diagnosis模式如下:

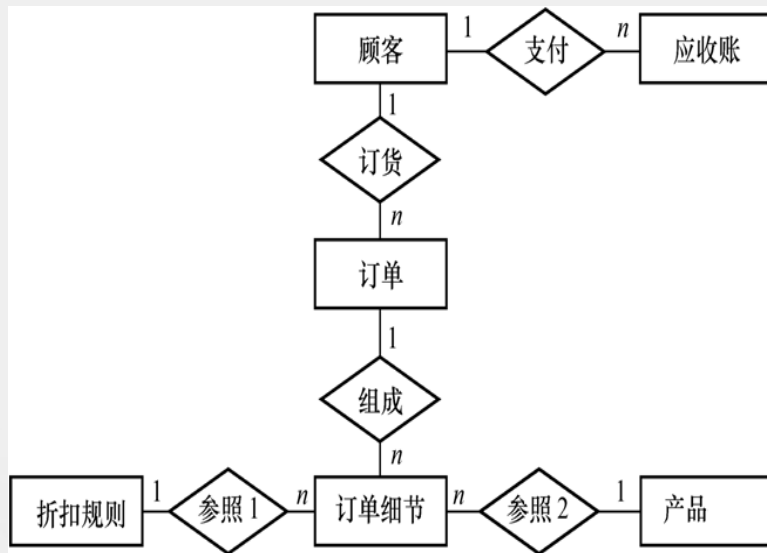
Diagnosis (DGno, Diagnosis, Symptom, DGtime, Rfee)



### – 练习：销售管理子系统E-R图转化为逻辑模式结构

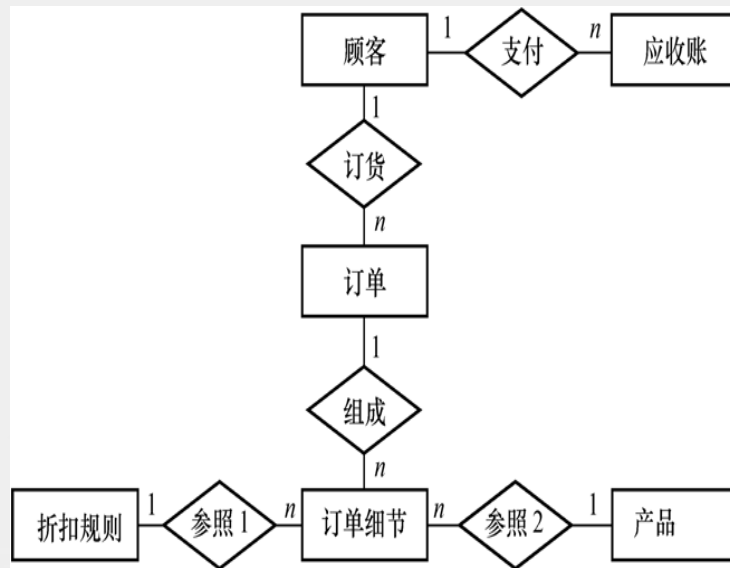
#### • 需求描述

- 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





- 每个实体的属性可以如下：
  - 顾客：{顾客号，顾客名，地址，电话，信贷状况，账目余额}
  - 订单：{订单号，顾客号，订货项数，订货日期，交货日期，工种号，生产地点}
  - 订单细则：{订单号，细则号，零件号，订货数，金额}
  - 应收账款：{顾客号，订单号，发票号，应收金额，支付日期，支付金额，当前余额，贷款限额}
  - 产品：{产品号，产品名，单价，重量}
  - 折扣规则：{产品号，订货量，折扣}
- 讨论：
  - 如果产品调整单价，会产生哪些影响？
  - 如果一个订单可以包含多个产品？





用关系规范化理论对关系数据模型进行优化：

- 确定范式的使用

- 采用各级范式来确定关系模式中的各种关系是否符合规范，检测依赖关系

- 实施规范化

- 根据需求规则说明书和实际应用环境调整改进关系模式



## 物理设计内容

数据库的物理设计即为确定的逻辑数据模型制定出合适的物理结构

数据库的物理设计包含两个方面的内容：

- 为逻辑数据模型确定物理结构，即存储结构和存取方法
- 对整体物理结构的时间和空间性能进行评价



## 存储结构的设计

存储结构设计时需要考虑两个方面的考虑因素：

- 数据的存放位置

数据根据操作或存取的性质进行区分存放。

- 确定系统配置

决定使用的DBMS产品后，需要对相应产品的系统配置变量、存储分配参数进行分析。



## 存取方法的设计

存取方法是为存取物理设备上的数据而提供存储和检索的方法，较为常用的方法有：

- 聚簇

把属性（聚簇键）按照具有相同键值的元组集中存放的原则，组织存储在连续的物理块中。

聚簇可以极大地提高这类属性的查询效率：

- 节省存储空间
- 提高查询速度



## 存取方法的设计

存取方法是为存取物理设备上的数据而提供存储和检索的方法，较为常用的方法有：

- 索引

- 索引是指根据应用要求，确定对关系的哪些属性列建立索引、组合索引及唯一索引等。
- 能够提高检索速度，还能够避免关系键重复值的录入，保证了数据的完整性。

- HASH存取方法

一种直接存取方法，通过对HASH值的计算来获得存取地址。





### 实际数据库结构的建立

- 当数据库物理结构设计完成之后，我们需要创建数据库，用DDL 来创建表、索引、视图等对象。
- 开发人员编写SQL语句用以创建数据库、创建表和定义视图。
- 也可以使用数据库系统提供的管理工具来完成数据库的实施。
- 例如，可以在SQL Server数据库中使用企业管理器的可视化工具来完成创建库和表等操作。



### 装入数据

- 数据库建立起来后，在进入实施运行阶段之前，将真实数据录入到数据库，这个过程称之为数据的加载。
- 数据全部装载入数据库中，需要很大的工作量，与此同时，在导入数据的过程中还要很好地和新建立的数据库系统融合。



### 应用程序开发测试

- 在软件开发过程中，数据库应用程序要考虑批量数据的处理能力，外围的数据使用和展示，以及数据完整性检测的要求。
- 数据库应用程序的开发过程与软件开发过程一样，需要经历设计、编码、调试和测试等步骤。



### 数据库试运行

- 功能性测试
  - 实际运行数据库应用程序，执行数据库的各种操作，检测应用程序能否正确完成需求的各项功能。
- 性能测试
  - 磨合期阶段测试系统的各项性能指标，检测是否能够达到设计的要求。并根据检测结果，调整物理设计中所设定的各类运行参数，以达到运行时最优。
- 非功能性测试
  - 系统为用户定制，试运行过程中充分让用户参与进来，让用户对人机交互友好性、操作合理性，以及稳定性等方面提出自己的意见和建议，以便及时进行改进。



## 数据库运行和维护

- 数据库的备份和恢复
  - DBA应针对不同的应用需求，制定相应的数据备份和转储计划，以达到尽可能降低故障损失的目标。
- 数据库的完整性和安全性
  - DBA应及时根据实际情况监控和调整数据库的安全性，以保证用户的资料和信息不会受到损害。
  - 变动也可能会导致对数据的完整性约束条件发生更改，需要DBA进行修正，以确保各类数据库应用程序的正常运行。



### 数据库运行和维护

- 数据库性能满足既定的要求
  - 数据库系统发生变动后，DBA需要对变动进行追踪，除了保证变动后系统的功能应保证正常以外，还应检测系统的性能是否仍然能满足用户的要求。
  - 当DBA检测到系统的性能下降时，应分析下降原因并及时调整系统参数，以及进行数据库重组或重构的工作。



### 数据库运行和维护

- 数据库的重组和重构
  - DBA应制定周期性的重组织计划，重新安排和整理数据的存储结构，处理存储碎片的合并与回收，以保证数据的存储效率和存取性能。
  - 重组织并不会改变原设计的逻辑和物理结构，然而数据库的重构却会改变数据库的模式和内模式。



### 1、数据库的设计方法目前可分为四类：

直观设计法、规范设计法、面向对象设计法和计算机辅助设计法。

### 2、数据库的设计需要经过六个阶段：

需求分析、概念设计、逻辑设计、物理设计、实现和运行与维护阶段。

### 3、需求分析的步骤分为：

需求调研、进行需求分析、编写需求分析说明书和需求分析说明书的验证。

### 4、概念设计的方法有4种方法：

自底向上、自顶向下、逐步扩张和混合策略。

### 5、数据库实现阶段主要工作包括：

数据库结构的建立、装入数据、应用程序开发测试和数据库试运行。

### 6、运行与维护阶段的主要工作包括：

数据库的备份和恢复、数据库的完整性和安全性、数据库性能满足既定的要求和数据库的重组和重构。