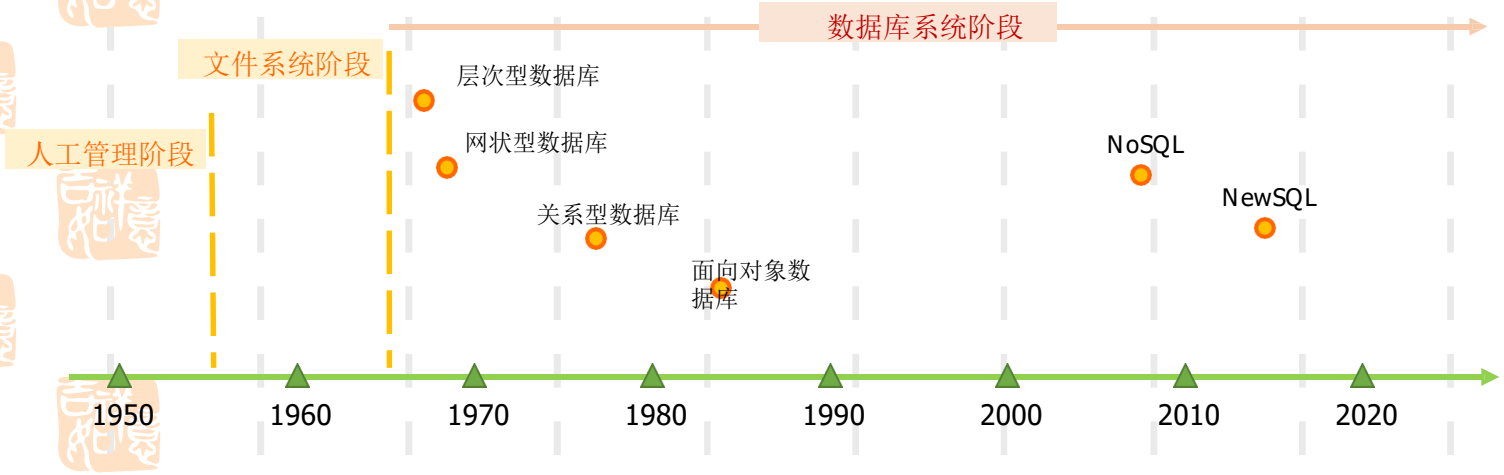
第1章 绪论

- 1.1 数据管理技术的发展
- 1.2 数据库基本概念
- 1.3 数据模型
- 1.4 数据库模式
- 1.5 数据库管理系统

数据管理技术的发展

- 数据管理就是对数据进行分类、组织、编码、存储、查询和维护。
- 其发展经历了三个阶段，即：人工管理阶段、文件系统阶段和数据库系统阶段。
- 每一个阶段都是以减小数据冗余、增强数据独立性和方便操作为目的进行发展。



人工管理阶段

- 时期
  - 20世纪40年代中--50年代中
- 背景
  - 应用需求：科学计算
  - 硬件条件：无直接存取存储设备
  - 软件水平：没有操作系统
  - 处理方式：批处理

人工管理阶段

- 特点
  - 数据管理者：用户(程序员)
  - 数据应用范围：单一应用程序
  - 数据共享程度：无共享、冗余度极大
  - 数据独立性：不独立，完全依赖于程序
  - 数据结构化：无结构
  - 数据控制：应用程序负责

人工管理阶段

- 程序与数据之间的关系

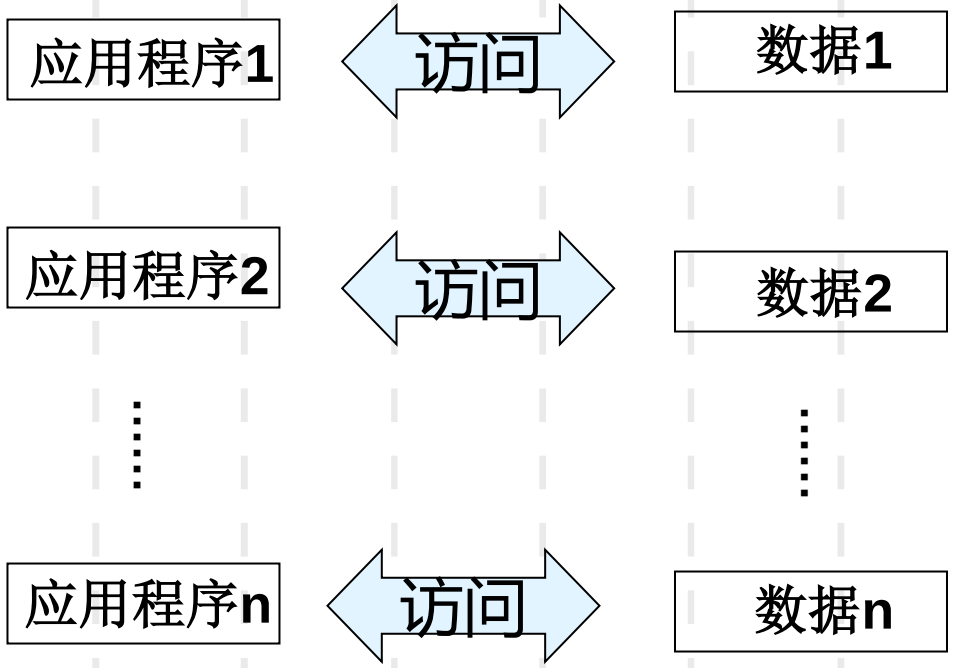


图 程序与数据之间的关系(人工管理阶段)

文件系统阶段

- 时期
  - 20世纪50年代末--60年代中
- 背景
  - 应用需求：科学计算+数据管理
  - 硬件条件：磁盘、磁鼓(直接存取设备)
  - 软件水平：有文件系统
  - 处理方式：联机分时处理+批处理

文件系统阶段

特点

- 数据管理者：文件系统，数据可长期保存
- 数据应用范围：某些应用程序
- 数据共享程度：共享性差、冗余度大
- 数据结构化：记录内有结构，无整体结构
- 数据独立性：较差，数据的逻辑结构改变必须修改应用程序
- 数据控制：应用程序负责

文件系统阶段

- 程序与数据之间的关系

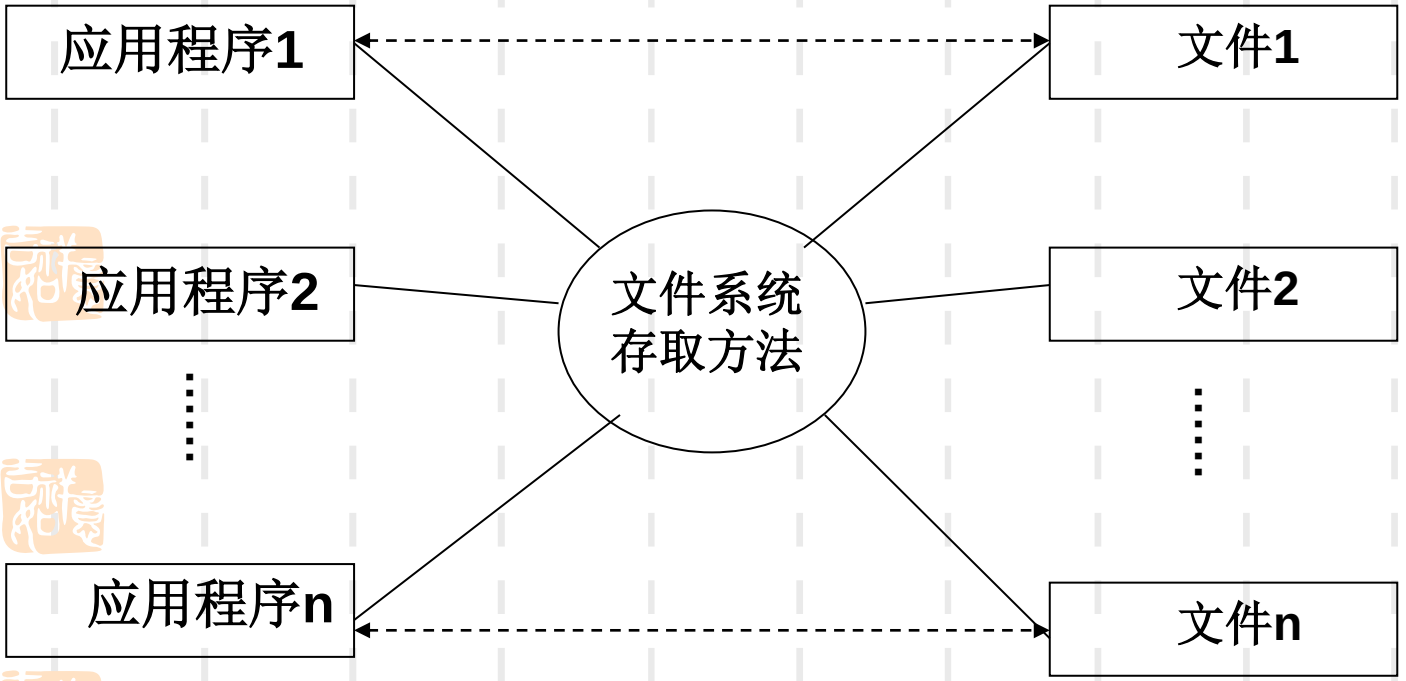


图 程序与数据之间的关系(文件系统阶段)

文件系统

- 文件系统是操作系统(OS)用于管理辅存(磁盘/磁带)数据的子系统，提供数据的物理存储和存取方法。
- 在文件系统中，一个命名的数据集合称为文件(file)。
- 文件是OS管理数据的基本单位。

文件系统

- 文件管理方式本质上是把数据组织成文件形式存储在磁盘上。
- 用户通过编程，定义数据的逻辑结构和输入输出(I/O)格式。
- 应用程序由于必须直接访问所使用的数据文件，所以完全依赖于数据文件的存储结构。
- 当数据文件修改时，应用程序必须作相应修改。

文件系统的数据结构

- 记录内有结构。
- 数据结构由应用程序定义和解释。
- 文件是孤立的，数据整体无结构。
  - 可以间接实现数据整体结构，但必须在应用程序中描述文件之间的联系
- 数据的最小存取单位是记录。

数据库系统阶段

- 时期
  - 20世纪60年代末之后
- 背景
  - 应用需求：大规模数据管理
  - 硬件条件：大容量磁盘、磁盘阵列
  - 软件水平：数据库管理系统
  - 处理方式：联机实时处理+分布处理

为什么要使用数据库？

- 使用数据库可以高效且条理分明地组织和存储数据，它使人们能够更加迅速和方便地管理数据，主要体现在以下几个方面：
  - 数据库可存储大量的结构化数据信息，方便用户进行有效的检索和访问。
  - 数据库可有效地保持数据信息的一致性、完整性、降低数据冗余。
  - 数据库可满足数据共享和安全方面的要求，把数据放在数据库中在很多情况下也是出于安全的考虑。
  - 数据库技术使得人们能够更加方便、智能化地获取和处理数据，产生新的有用信息。

数据库系统阶段

- 程序与数据之间的关系

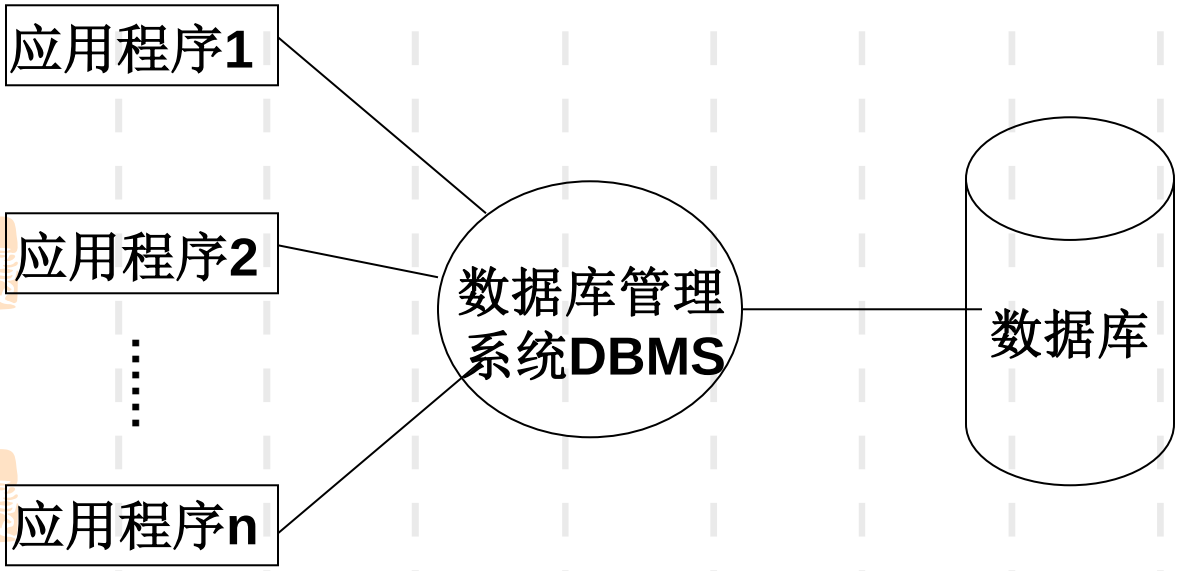


图 程序与数据之间的关系(数据库系统阶段)

数据库系统阶段

- 特点
- 面向全组织
- 面向现实世界
- 独立性较强
- 数据的由数据库管理系统(DBMS)统一存取和维护

数据管理各阶段的比较

		人工管理阶段	文件系统阶段	数据库系统阶段
产生背景	应用需求	科学计算	科学计算机/数据管理	大规模数据管理
	硬件水平	无直接存储设备	磁盘/磁鼓	大容量磁盘
	软件水平	没有操作系统和数据管理软件	高级语言+操作系统(文件系统)	数据库管理系统(DBMS)
	处理方式	批处理	联机实时处理+批处理	联机实时处理+分布式处理
特点	数据的管理者	应用程序	文件系统	数据库管理系统
	应用范围	单个应用程序	单个或少量应用程序	多个应用 整个企业

数据管理各阶段的比较

特点	数据共享程度	无共享	共享性差	共享性好
	数据独立性	不独立，完全依赖于程序	记录内有结构，整体无结构，独立性差	高度的物理独立性和一定的逻辑独立性
	数据结构化	无结构	若数据的逻辑结构改变则必须修改应用程序	采用数据模型表示复杂的数据结构，整体结构化
	数据控制	完全由应用程序控制	基本由应用程序控制	由***DBMS***统一管理和控制，提供数据库的并发控制，数据库的恢复，数据的完整性和数据安全性等支持
	用户接口	无	物理存取	数据库系统为用户提供方便的用户接口

第1章 绪论

- 1.1 数据管理技术的发展
- 1.2 数据库基本概念
- 1.3 数据模型
- 1.4 数据库模式
- 1.5 数据库管理系统

数据(Data)

- 数据是对事实或观察的记录，是对客观事物的逻辑归纳
- 数据是信息的表现形式和载体，可以是符号、文字、数字、语音、图像、视频等。
- 在计算机系统中，各种字母、数字符号的组合、语音、图形、图像等统称为数据，数据以二进制信息单元0/1(bit)形式表示。
- 表现形式：
  - **数字数据**：由离散的符号、文字、数字符号构成。如各种统计或测量数据。
  - **模拟数据**：是指在某个区间产生的连续值，如视频、图像、文字、声音等。

数据的语义

- 数据的含义或解释称为数据的语义(Semantics)，数据与其语义是不可分的。
  - 例如：“93”

    语义1：某学生某门课的成绩

    语义2：某人的体重KG

    语义3：软件学院2021级学生人数

    语义4：…
- 人工智能技术的发展使得机器也可以对数据的语义进行理解和处理。

数据举例

- 学生档案中的学生记录：

  （李明，男，200305，陕西省西安市，软件学院，2021）

- 语义：学生姓名、性别、出生年月、出生地、所在院系、入学时间

- 解释：李明是大学生，2003年5月出生，陕西省西安市人， 2021年考入软件学院

数据库 (Database)

- 数据库是指 持 久 存储在计算机内、有组织、可共享的大量的数据的集合。

  数据库中的数据按照一　　定的数据模型组织、描述和存储，具有较小的冗余度、较高的数据独立性和易扩展性，并可为各 种用户共享。

持久存储

有组织

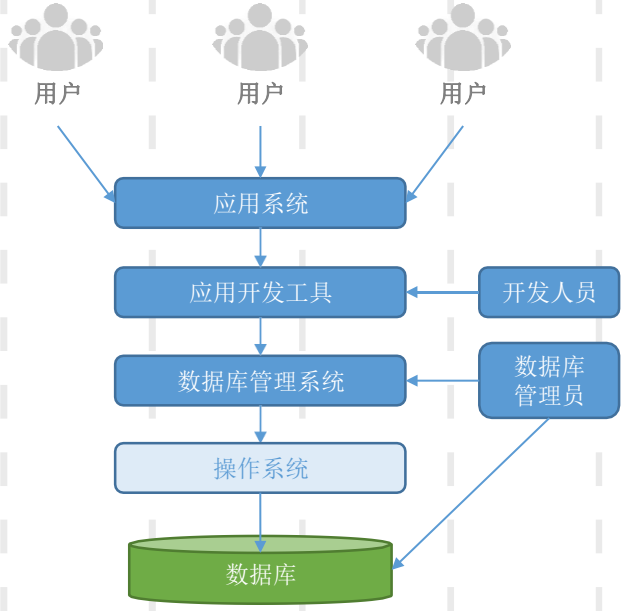
可共享

数据库管理系统 (DBMS)

- 数据库管理系统（DataBase-Management System，DBMS）由一个互相关联的数据集合和一组用以访问这些数据的程序组成。
- **DBMS**是位于用户应用与操作系统之间的一层数据管理软件，充当数据库与其用户或程序之间的接口，允许用户检索、更新和管理信息的组织和优化方式。
- DBMS的主要目的是为用户提供一种可以方便、高效地存取数据库信息的途径。此外，DBMS还有助于监督和控制数据库，提供各种管理 功能，例如性能监视、调优、备份和恢复。
- 具有代表性的数据库管理系统有：Oracle、IBM DB2 、 Microsoft SQL Server、MySQL及PostgreSQL等。

数据库系统 DBS

- 数据库系统是指在计算机系统中引入数据库后的系统。
- 一个完整的数据库系统一般由数据库、数据库管理系统、应用开发工具、应用系统、数据库管理员和用户组成。



数据库系统 DBS

- 数据库系统（Database System，DBS）由硬件和软件共同构成。
- 硬件主要用于存储、处理数据库中的数据，包括计算机、存储设备等。
- 软件部分主要包括数据库管理系统、支持数据库管理系统运行的操作系统，以及支持多种语言进行应用开发的访问接口等。

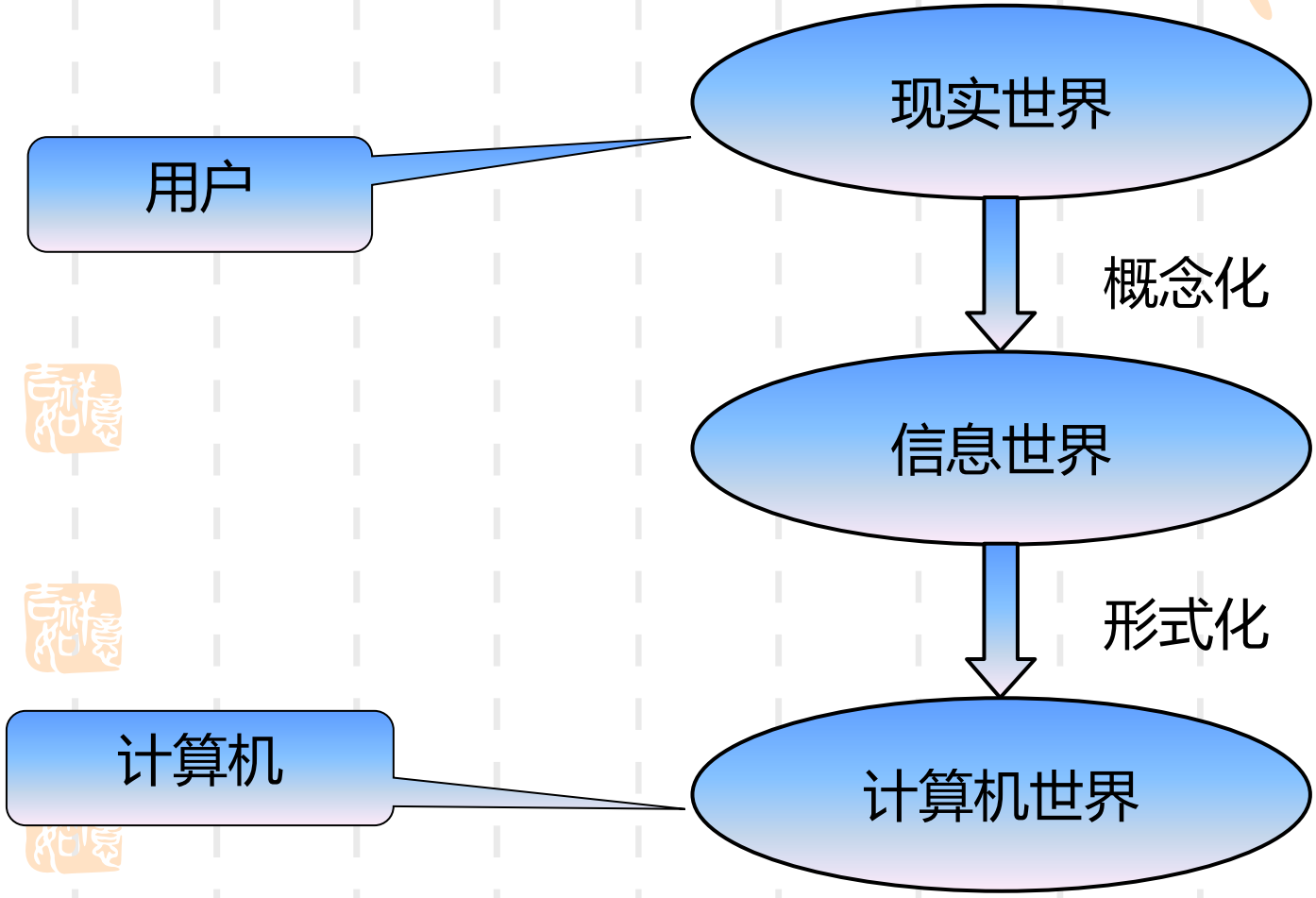
第1章 绪论

- 1.1 数据管理技术的发展
- 1.2 数据库系统
- 1.3 数据模型
- 1.4 数据库模式
- 1.5 数据库管理系统

什么是数据模型

- 数据模型是对现实世界数据特征的抽象。
- 通俗地讲数据模型就是对现实世界的模拟。
- 数据模型应满足三方面要求
  - 真实地模拟现实世界
  - 容易为人所理解
  - 便于在计算机上实现
- 数据模型是数据库系统的核心和基础

数据抽象



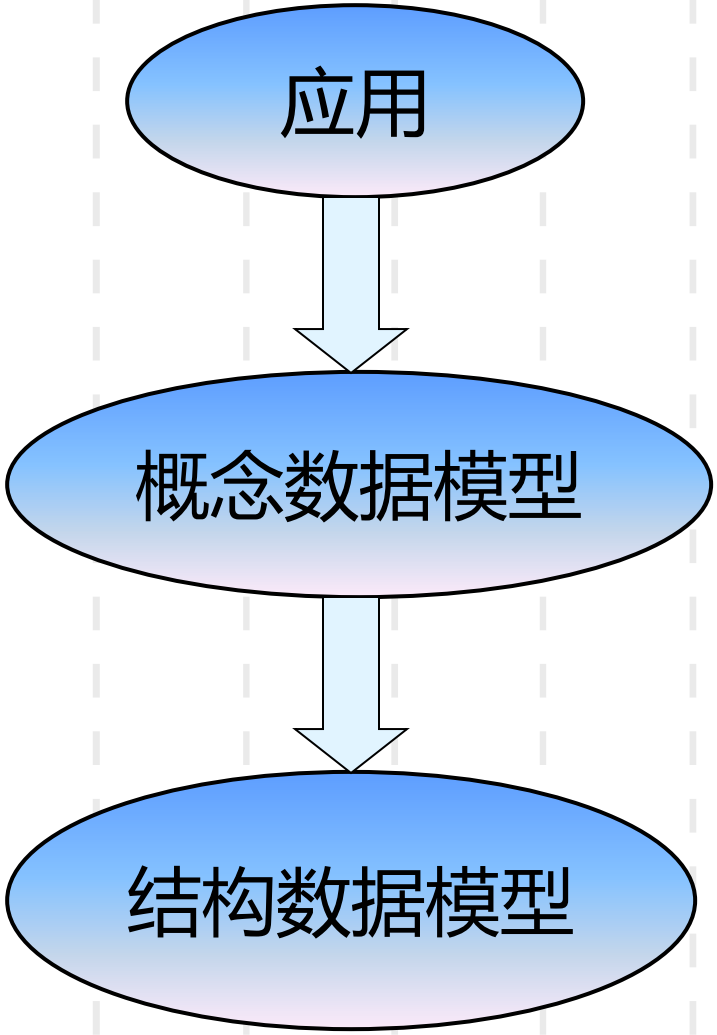
信息世界

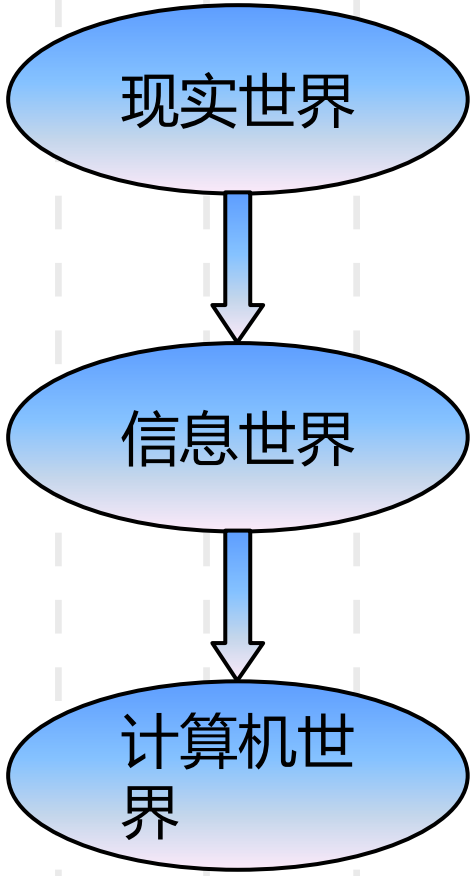
- 数据库系统是面向计算机的，而应用是面向现实世界的，两个世界存在着很大差异。
- 要直接将现实世界中的语义映射到计算机世界十分困难。
- 因此,需要引入一个信息世界作为现实世界通向计算机实现的桥梁。

数据模型的定义(1)

数据模型

是数据库系统中用于提供信息表示和操作手段的形式构架





数据模型的定义(2)

- 数据模型是规定现实世界数据特征的抽象，是用来描述数据的语法、语义和操作的一组概念的集合。
- 数据模型是对现实世界某方面的抽象和模拟。

三类数据模型

- **概念模型**
  - 也称信息模型，它是按用户的观点对数据和信息建模，用于数据库概念设计和建模。
- **逻辑模型**
  - 也称数据模型(狭义)，它是按计算机系统的观点对数据建模，主要包括网状模型、层次模型、关系模型、面向对象模型等，用于数据库系统实现。
- **物理模型**
  - 也称存储模型，它是对数据最底层的抽象，描述数据在系统内部的表示方式和存取方法，在磁盘或磁带上的存储方式和存取方法。

数据模型三要素

- 数据结构
- 数据操作
- 完整性约束条件

数据结构

- 描述数据库的组成对象，以及对象之间的联系
- 描述内容
  - 与数据类型、内容、性质有关的对象
  - 与数据之间联系有关的对象
- 数据结构是对系统静态特性的描述

数据操作

- 对数据库中各种对象(型)的实例(值)允许执行的操作及有关的操作规则
- 数据操作的类型
  - 查询
  - 更新(包括插入、删除、修改)

数据操作

- 数据模型对操作的定义
  - 操作的确切含义
  - 操作符号
  - 操作规则(如优先级)
  - 实现操作的语言
- 数据操作是对系统动态特性的描述

完整性约束条件

- 一组完整性规则的集合。
- 完整性规则：给定的数据模型中数据及其联系所具有的制约和储存规则。
- 用以限定符合数据模型的数据库状态以及状态的变化，以保证数据的正确、有效、相容。

概念模型

- 概念模型用于信息世界的建模。
  - 是现实世界到机器世界的一个中间层次。
  - 是数据库设计的工具。
  - 是数据库设计人员和用户之间交流的语言工具。

实体－联系(ER)模型

- 最常用的概念模型的表示方法是实体-联系方法(Entity-Relationship Approach)。
- 1976年，P.P.S.Chen提出E-R模型，用E-R图来描述概念模型。
- 观点：客观世界是由一组称作实体的基本对象和这些对象之间的联系构成的。
- ER模型与DBMS所支持的数据模型相独立，是各种数据模型的共同基础。

信息世界中的基本概念

(1) 实体(Entity)

- 客观存在并可相互区别的事物称为实体，可以是具体的人、事、物或抽象的概念。
- 实体集(entity set)：具有共同属性的所有实体的集合；
- 实体实例(entity instance): 实体集中的单个实体；
- 例：学生，职工，部门，课程，…都是实体(集)，而“王英”，“张凡”都是“学生”实体集中的实体实例(值)
- 实体的表示:

  E={e1,…,en}(外延法)

  E(A1,…,An)（内涵法)

信息世界中的基本概念

(2) 属性(Attribute)

- 实体(或联系)所具有的某方面特征称为属性
- 一个实体可以由若干个属性来刻画
- 值集：属性的取值范围，也称为域(Domain)
- 根据属性的取值方式，可以将属性进一步区分为：：
  - 简单属性与复合属性
  - 单值属性与多值属性
  - 标识性属性与描述性属性
  - 固有属性与导出属性
- 属性的表示：
  - 一般属性： A：$E \rightarrow \rho(V1) \times \rho(V2) \ldots\ldots \times \rho(Vn)$
  - 简单属性： A：$E \rightarrow \rho(V)$

(3) 码/键/关键字(Key)

- 可唯一标识实体实例的属性或属性集

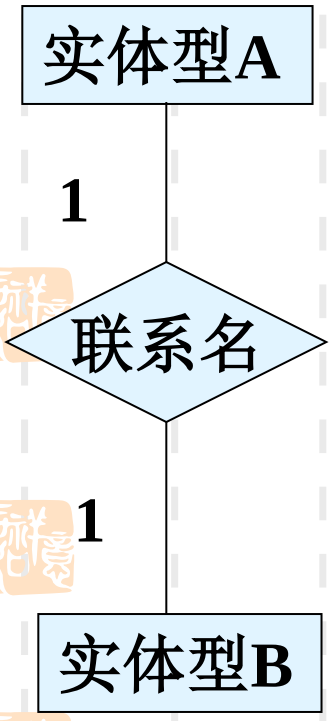
信息世界中的基本概念

(4) 联系(Relationship)

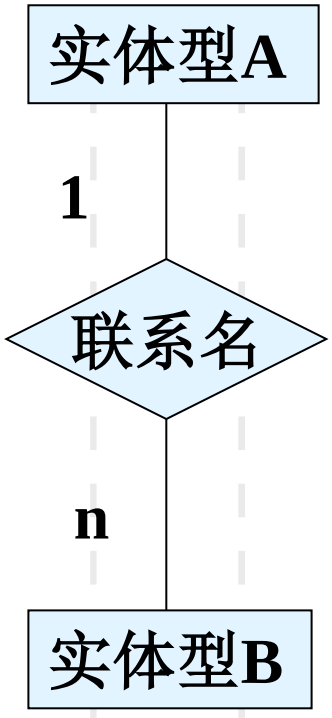
- 现实世界中事物内部以及事物之间的联系在信息世界中反映为实体内部的联系和实体之间的联系。
- 将发生在实体之间具有特定含义的对应关系称为联系。
- 注意，与实体一样，也存在联系集(型)与联系值的区别。

两个实体型之间的联系

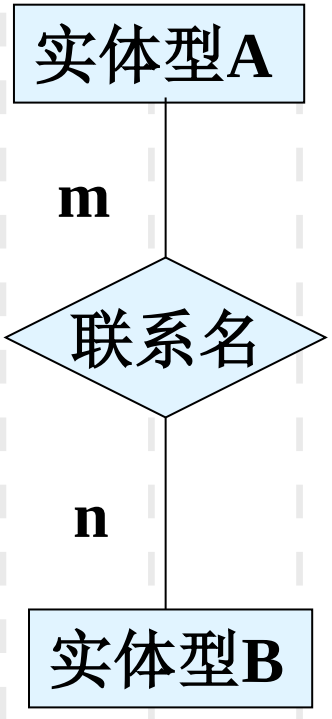
- 用图形来表示两个实体型之间的三类联系



1:1联系



1:n联系



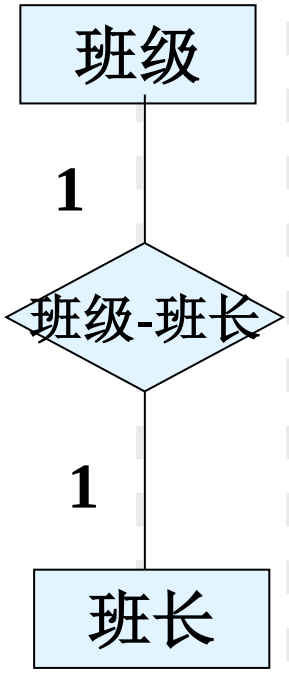
m:n联系

两个实体型之间的联系

- 一对一联系(1:1)
  - 定义：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(也可以没有)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：

    一个班级只有一个班长

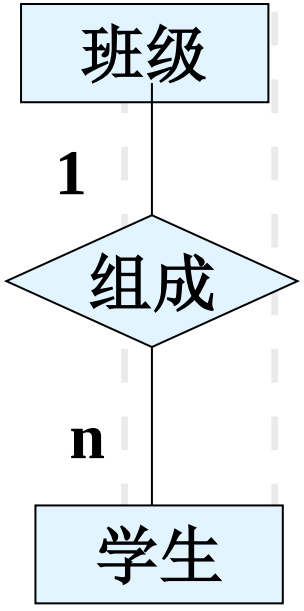
    一个班长只在一个班中任职



1:1联系

两个实体型之间的联系

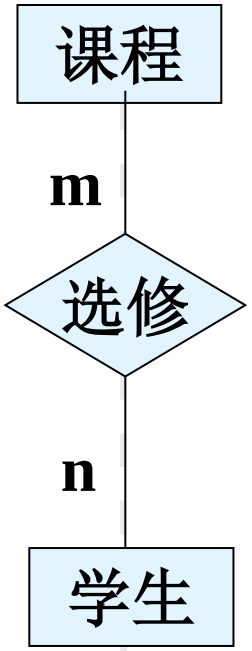
- 一对多联系(1：n)
  - 定义：
    - 如果对于实体集A中的每一个实体，实体集B中有n个实体(n≥0)与之联系，反之，对于实体集B中的每一个实体，实体集A中至多只有一个实体与之联系，则称实体集A与实体集B有一对多联系，记为1:n
  - 实例：
    - 一个班级中有若干名学生，
    - 每个学生只在一个班级中学习



1:n联系

两个实体型之间的联系

- 多对多联系(m:n)
  - 定义：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≥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≥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：
    - 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



m:n联系

两个以上实体型之间的联系

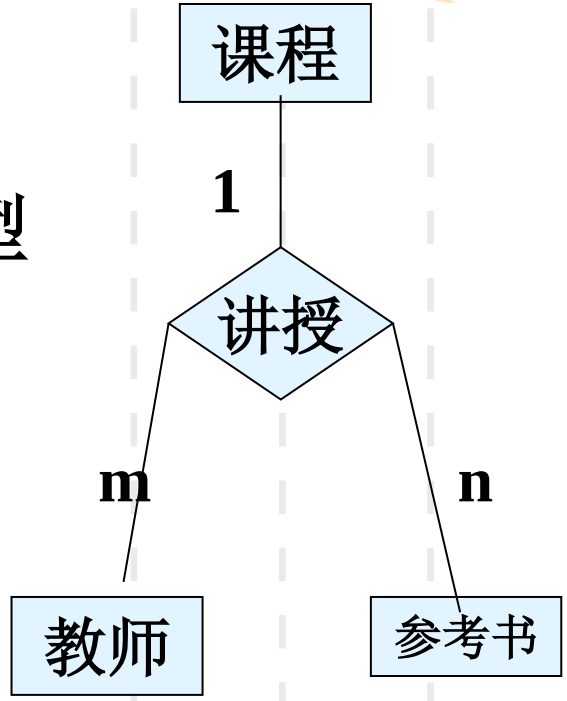
- 两个以上实体型之间一对多联系
  - 若实体集E_1，E_2，…，E_n存在联系，对于实体集E_j(j=1，2，…，i-1，i+1，…，n)中的给定实体，最多只和E_i中的一个实体相联系，则我们说E_i与E_1，…，E_n之间的联系是一对多的。

两个以上实体型之间的联系

- 实例

课程、教师与参考书三个实体型

- 一门课程可以有若干个教师讲授
- 使用若干本参考书
- 每一个教师只讲授一门课程
- 每一本参考书只供一门课程使用

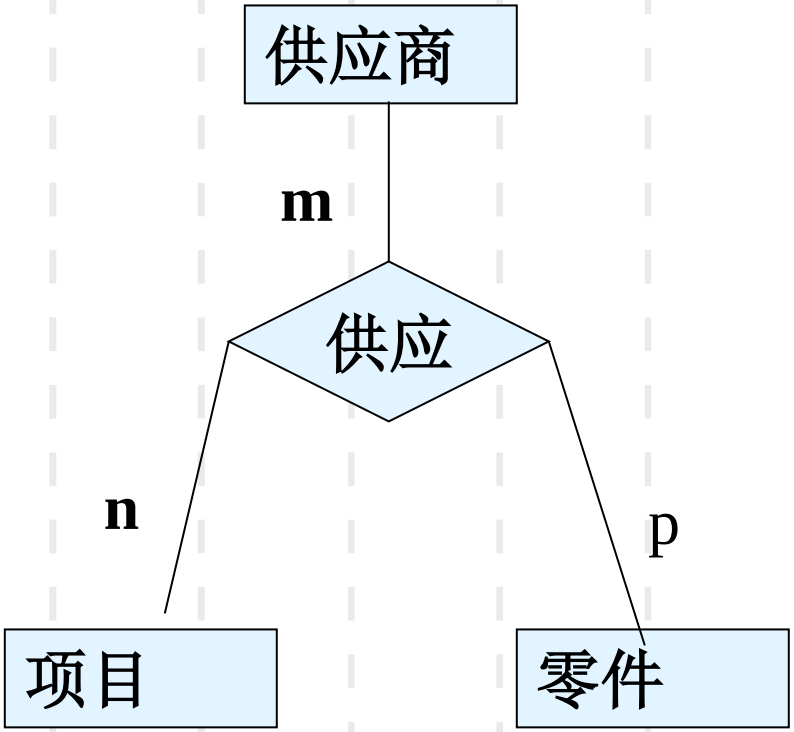


两个以上实体型间1:n联系

两个以上实体型之间的联系

- 两个以上实体型间的多对多联系
- 例：
  - 供应商、项目、零件三个实体型
  - 一个供应商可以供给多个项目多种零件
  - 每个项目可以使用多个供应商供应的零件
  - 每种零件可由不同供应商供给

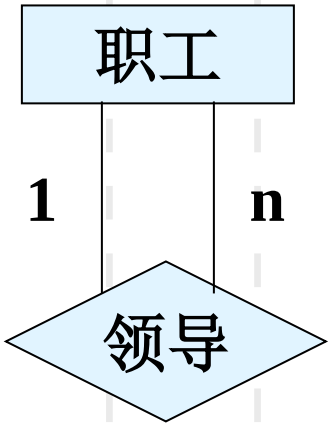
两个以上实体型之间的联系



两个以上实体型间m:n联系

单个实体型内的联系

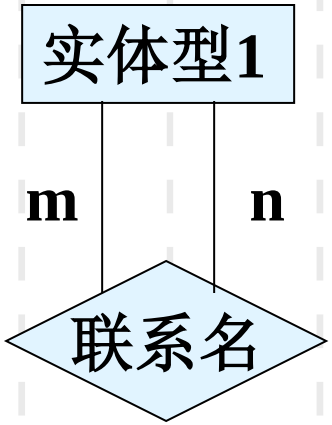
- 一对多联系
- 实例
  - 职工实体型内部具有领导与被领导的联系
  - 某一职工(干部)“领导”若干名职工
  - 一个职工仅被另外一个职工直接领导
  - 这是一对多的联系
- 一对一联系
- 请举例



单个实体型内部1:n联系

单个实体型内的联系

■多对多联系

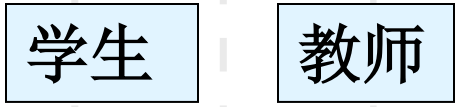


单个实体型内的
m:n联系

E-R图

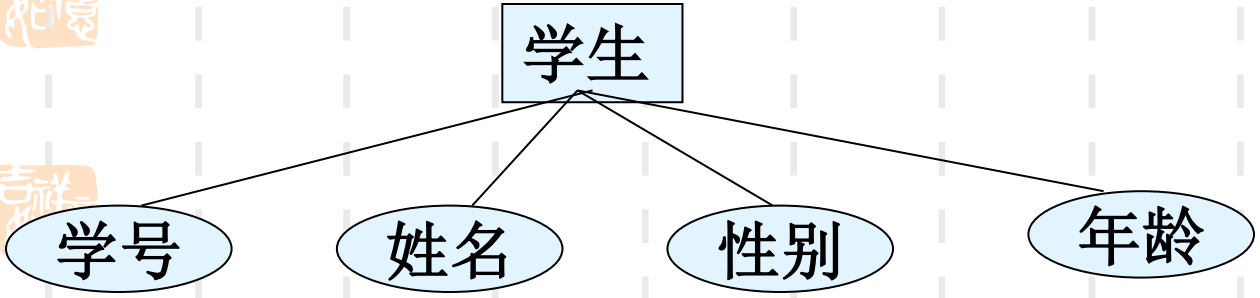
- 实体型

  用矩形表示，矩形框内写明实体名



- 属性

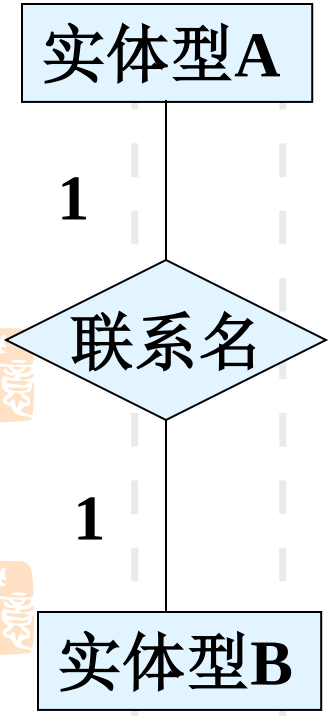
  用椭圆形表示，并用无向边将其与相应的实体连接起来



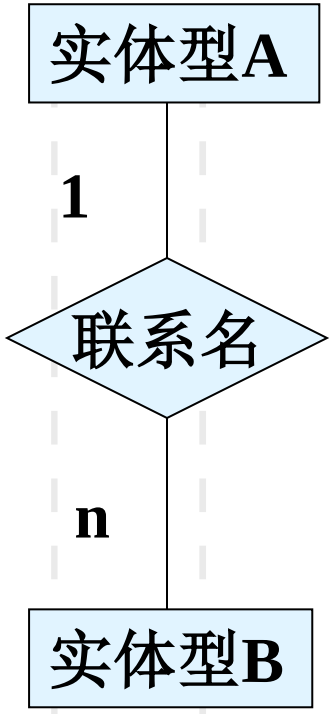
E-R图

- 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)

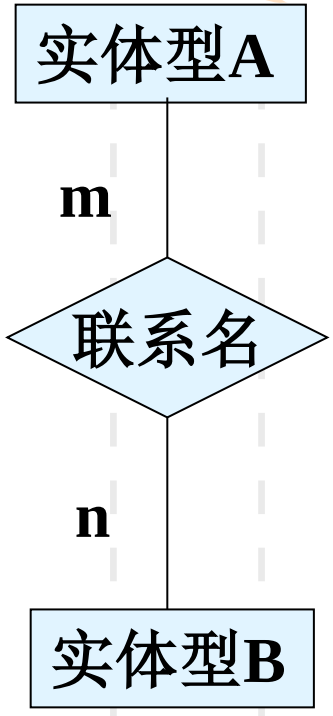
联系的表示方法



1:1联系

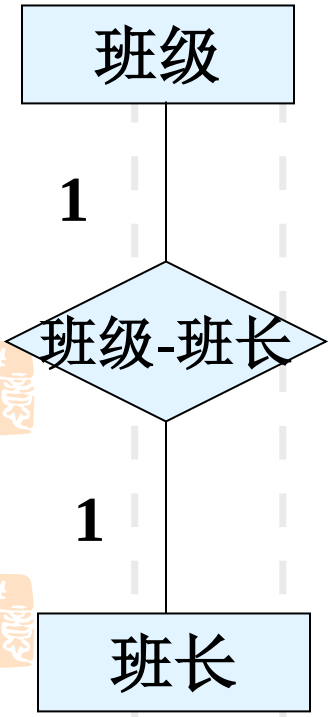


1:n联系

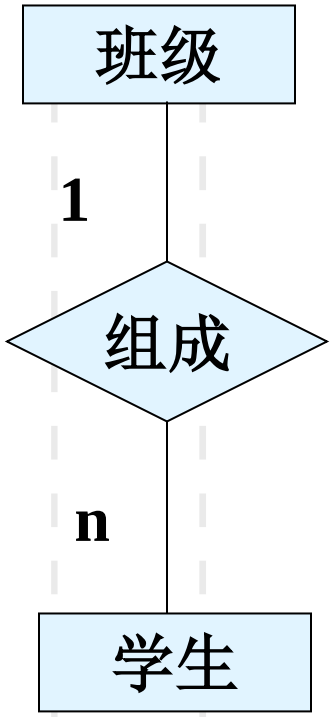


m:n联系

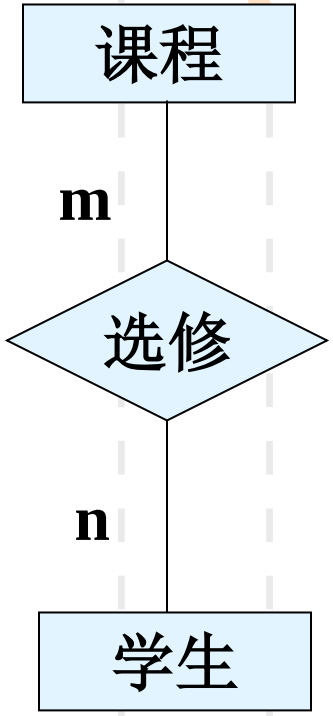
联系的表示方法示:例



1:1联系



1:n联系



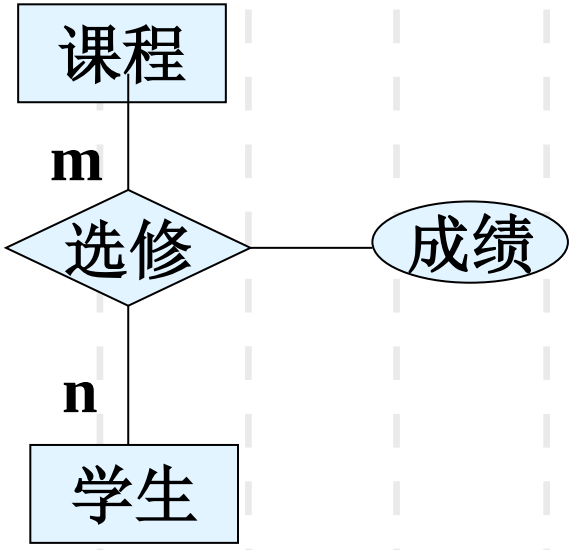
m:n联系

联系的属性

■联系的属性：

- 联系本身也是一种实体型，也　可以有属性
- 如果一个联系具有属性，则这些属性也要用无向边与该联系连接起来

➢例：



ER图:实例

- 用E-R图表示某个工厂物资管理的概念模型
- 实体
  - 仓库： 仓库号、面积、电话号码
  - 零件 ：零件号、名称、规格、单价、描述
  - 供应商：供应商号、姓名、地址、电话号码、帐号
  - 项目：项目号、预算、开工日期
  - 职工：职工号、姓名、年龄、职称

ER图:实例

- 实体之间的联系如下：

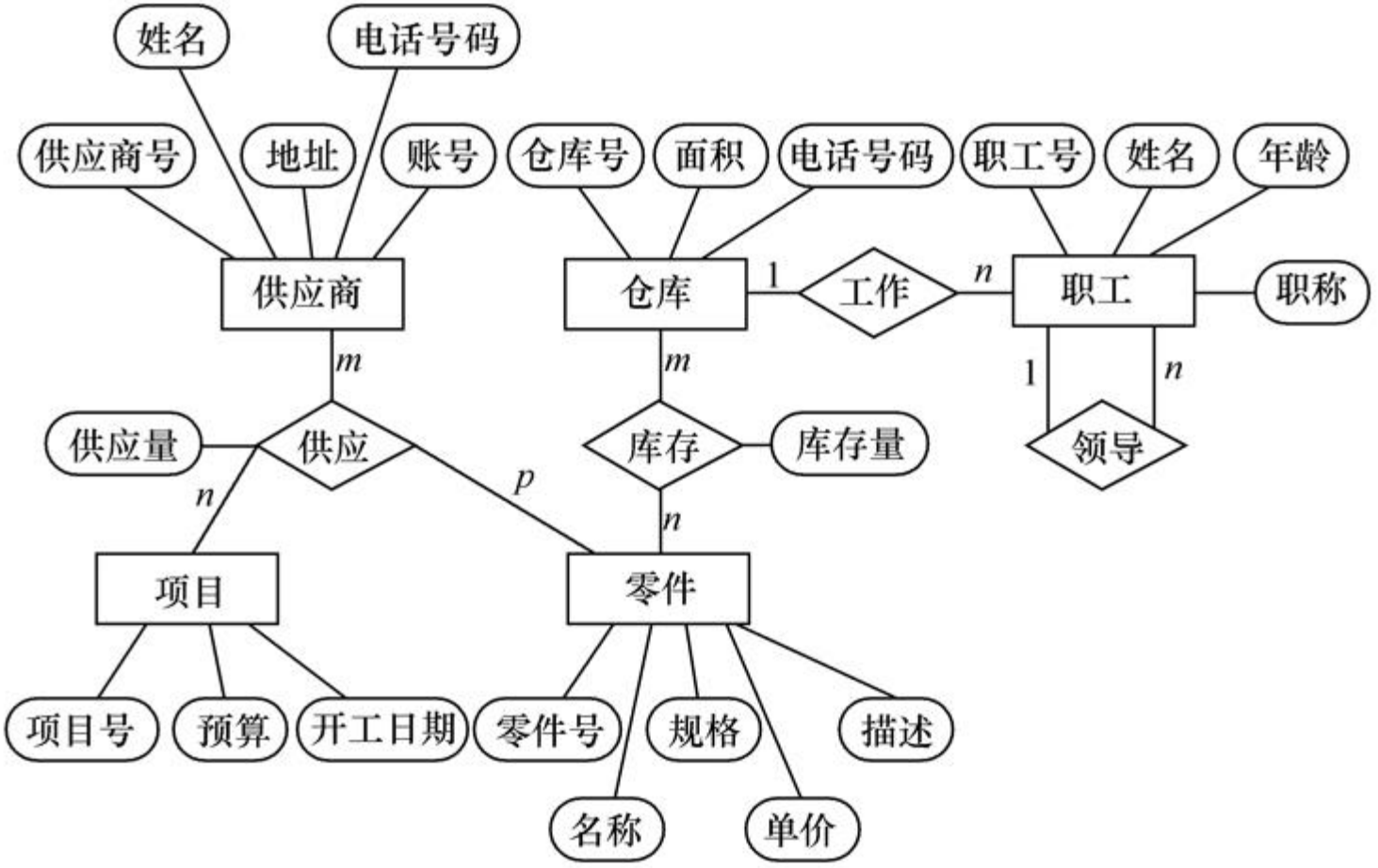
(1)一个仓库可以存放多种零件，一种零件可以存放在多个仓库中。仓库和零件具有多对多的联系。用库存量来表示某种零件在某个仓库中的数量

(2)一个仓库有多个职工当仓库保管员，一个职工只能在一个仓库工作，仓库和职工之间是一对多的联系。职工实体型中具有一对多的联系

(3)职工之间具有领导-被领导关系。即仓库主任领导若干保管员

(4)供应商、项目和零件三者之间具有多对多的联系

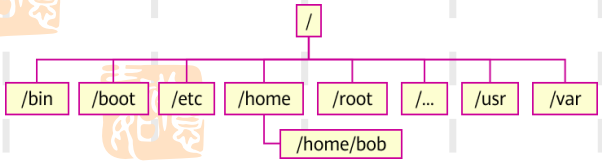
ER图:实例



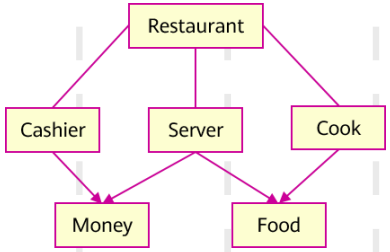
(c) 完整的实体-联系图

经典逻辑数据模型

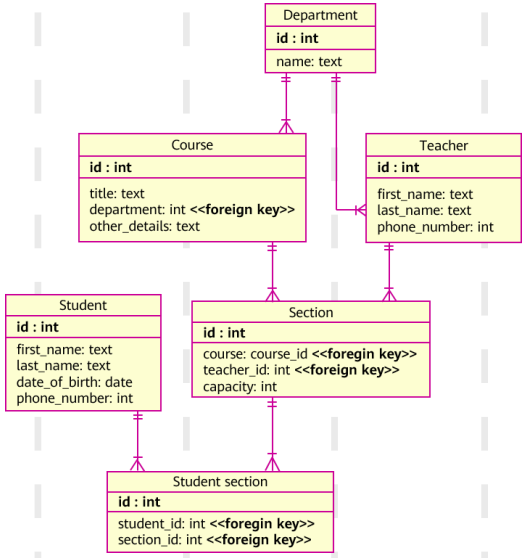
- 数据库自二十世纪中期以来，诞生了层次模型(Hierarchical Model)、网状模型(Network Model)、关系模型(Relational Model)等很多种经典数据模型，有些数据库模型发挥了其作用后，已退出历史舞台,如层次模型、网状模型；
- 有些数据库模型经受住了历史地考验，如今依然站立在时代的潮头，如关系模型。



层次模型
（Hierarchical Model）



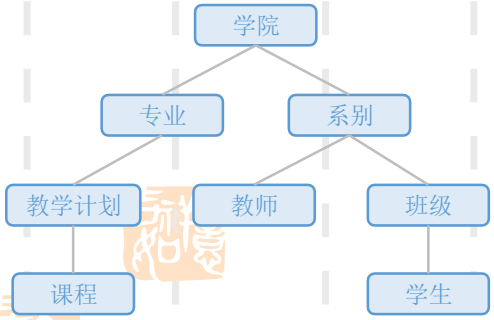
网状模型
（Network Model）



关系模型
（Relational Model）

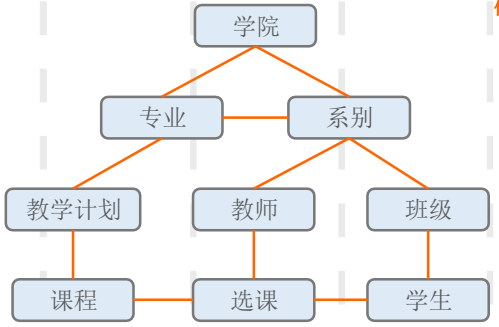
数据库模型 – 层次、网状、关系模型

层次数据模型



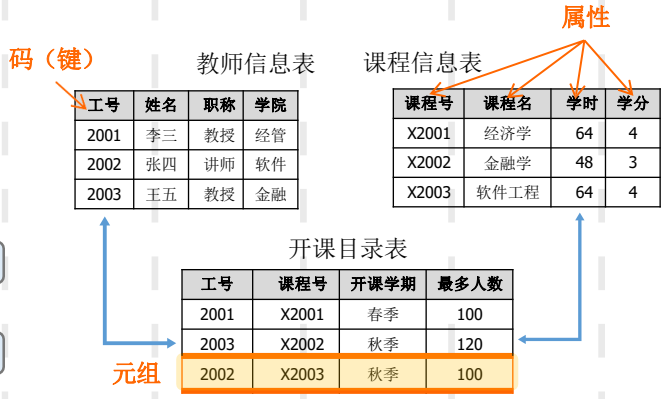
- 层次模型
  - 有且只有一个节点没有双亲，该节点被称为根节点（root）。
  - 根节点以外的其他节点有且只有一个双亲节点。

网状数据模型



- 网状模型
  - 允许一个以上的节点无双亲。
  - 一个节点可以有多于一个的双亲。

关系数据模型



- 关系模型
  - 建立在严格的数据概念基础上。
  - 关系必须是规范化的。
  - 关系的分量必须是一个不可分的数据项。

其它数据模型

- 面向对象数据模型（**OO Data Model**）
- 对象关系数据模型（**OR Data Model**）
- 半结构化数据模型（**Semistruture Data Model**）
- 语义数据模型（**Semantics Data Model** ）
- 图数据模型（**Graph Data Model**）

第1章 绪论

- 1.1 数据管理技术的发展
- 1.2 数据库系统
- 1.3 数据模型
- 1.4 数据库模式
- 1.5 数据库管理系统

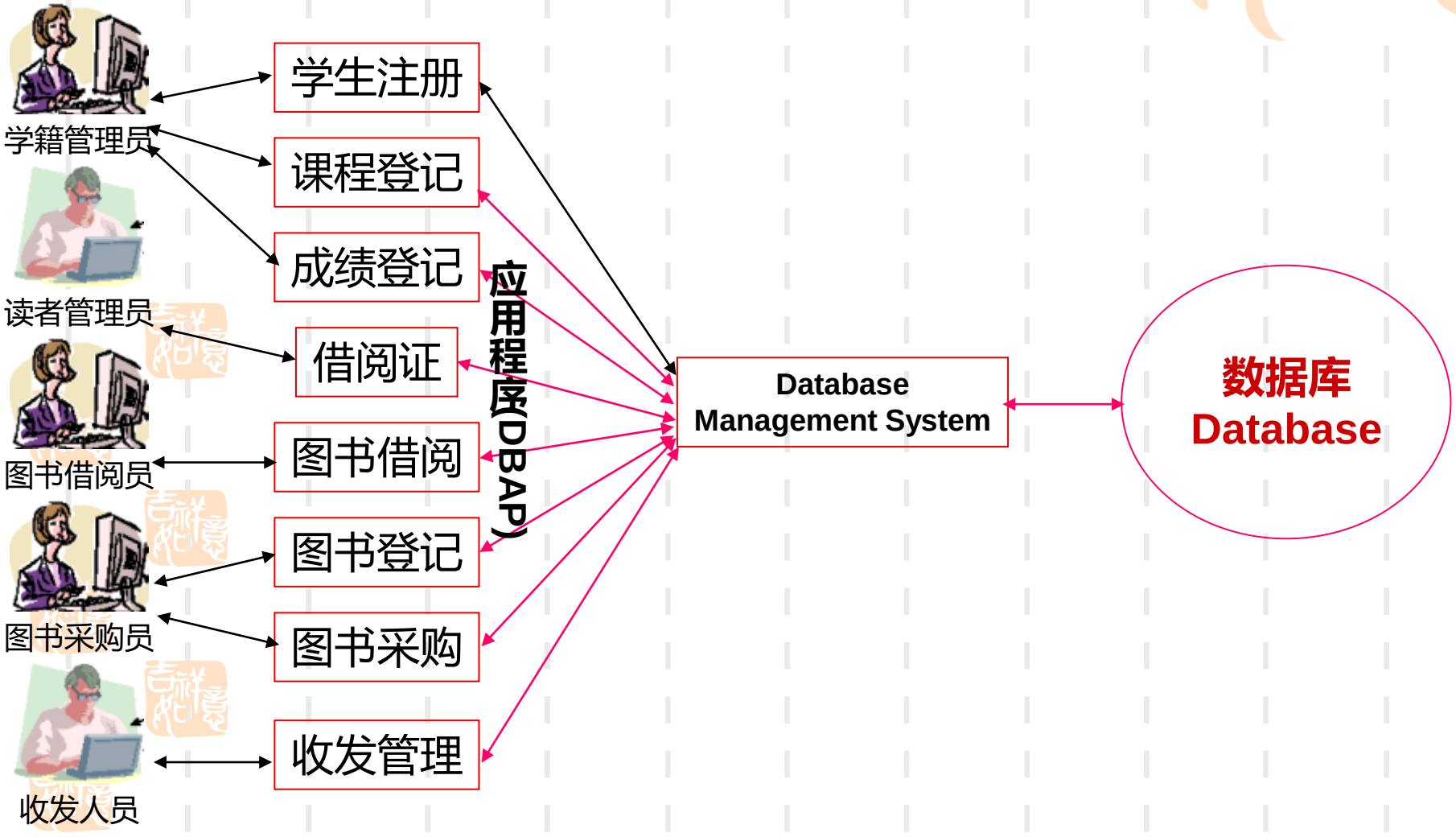
数据库系统的结构抽象

怎样抽象一个数据库系统?

数据库系统的抽象结构

(1)数据库系统的分层抽象

- 一个典型数据库系统的结构抽象



数据库系统的抽象结构

(1)数据库系统的分层抽象

DBMS管理数据的三个层次

External Level = User Level

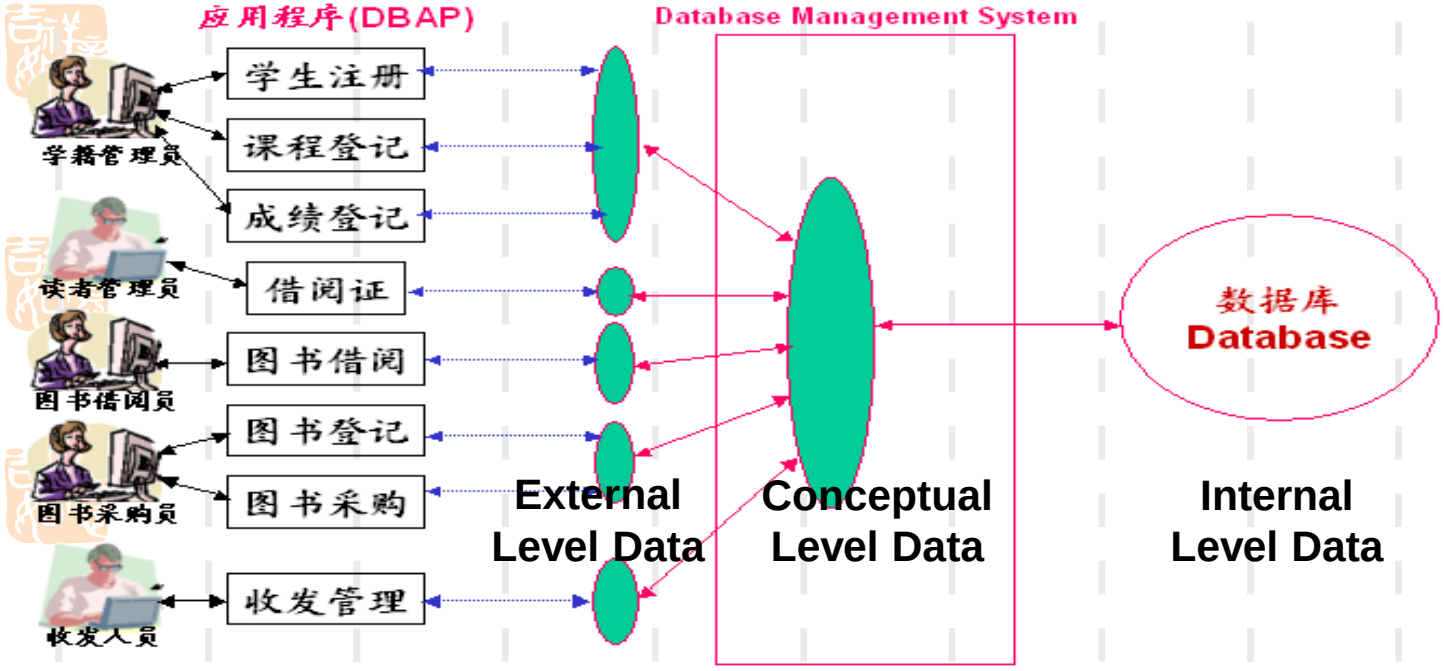
某一用户能够看到与处理的数据, 全局数据中的某一部分

Conceptual Level = Logic level

从全局角度理解/管理的数据, 含相应的关联约束

Internal Level = Physical level

存储在介质上的数据，含存储路径、存储方式 、索引方式等



数据库系统的抽象结构

(2)数据(视图)与模式

数据 与 数据的结构--模式

- **模式(Schema)**

  对数据库中数据所进行的一种结构性的描述

  所观察到数据的结构信息

- **视图(View)/数据(Data)**

  某一种表现形式下表现出来的数据库中的数据

学生登记表(学号 char(8), 姓名 char(10), 性别 Char(2), 出生年月 datetime, 入学日期 Datetime, 家庭住址 Char(40))

数据的结构----模式

学生登记表

学号	姓名	性别	出生年月	入学日期	家庭住址
98110101	张三	男	1980.10	1998.09	黑龙江省哈尔滨市
98110102	张四	女	1980.04	1998.09	吉林省长春市
98110103	张五	男	1981.02	1998.09	黑龙江省齐齐哈尔市
98110201	王三	男	1980.06	1998.09	辽宁省沈阳市
98110202	王四	男	1979.01	1998.09	山东省青岛市
98110203	王武	女	1981.06	1998.09	河南省郑州市

展现的数据----视图

“型”和“值”的概念

- 型（Type）
  - 对某一类数据的结构和属性的说明
- 值（Value）
  - 是型的一个具体赋值

例：

学生记录型（Type）：

（学号，姓名，性别，系别，年龄，籍贯）

一个记录值（Value）：

（202015130，李明，男，软件学院，20，陕西西安市）

模式的概念

- 模式（Schema）
  - 数据库逻辑结构和特征的描述
  - 是型的描述，不涉及具体值
  - 反映的是数据的结构及其联系
  - 模式相对稳定
- 实例（Instance）
  - 模式的一个具体值
  - 反映数据库某一时刻的状态
  - 同一个模式可以有很多实例
  - 实例随数据库中的数据的更新而变动

数据库系统的抽象结构

(3)三级模式两层映像

三级模式(三级视图)

External Schema ---- (External) View

某一用户能够看到与处理的数据的结构描述

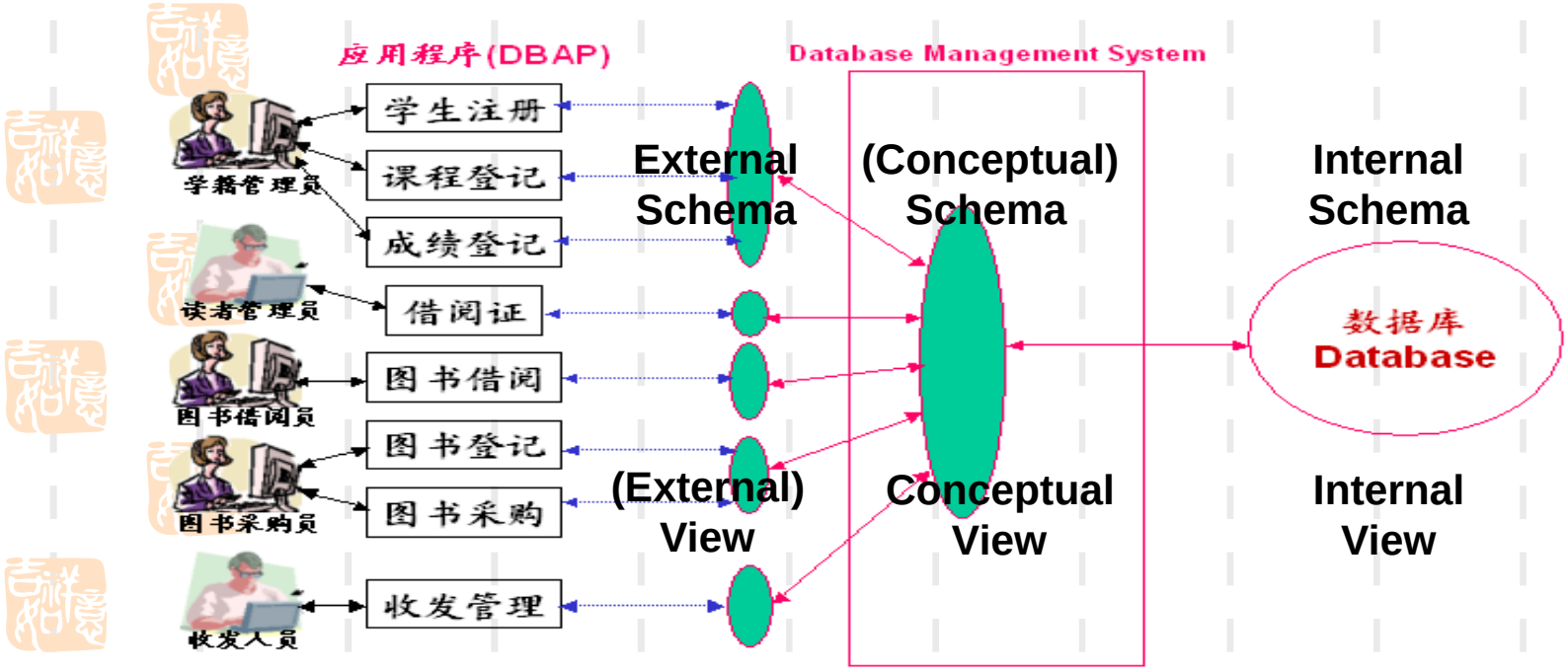
(Conceptual) Schema ---- Conceptual View

从全局角度理解/管理的数据的结构描述, 含相应的关联约束

体现在数据之间的内在本质联系

Internal Schema ---- Internal View

存储在介质上的数据的结构描述，含存储路径、存储方式 、索引方式等



数据库系统的抽象结构

(3)三级模式两层映像

两层映像

E-C Mapping：External Schema-Conceptual Schema Mapping

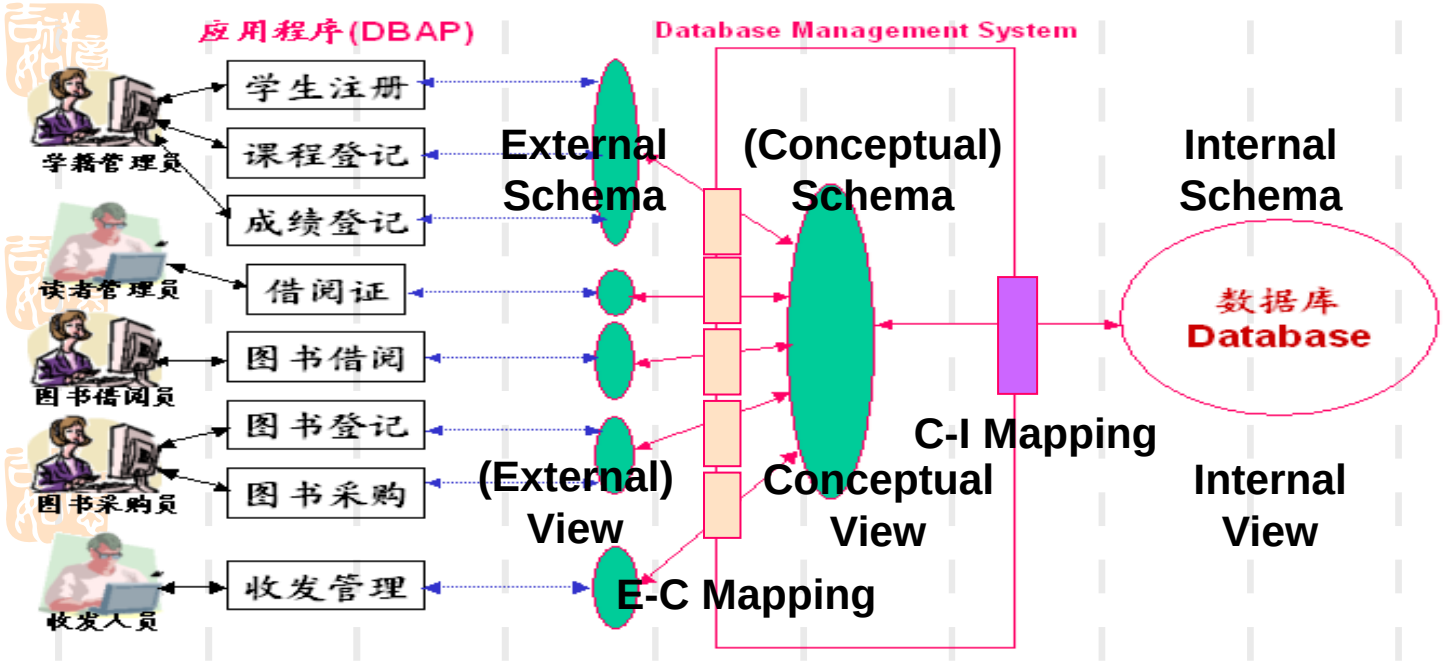
----将外模式映射为概念模式，从而支持实现数据概念视图向外部视图的转换

----便于用户观察和使用

C-I Mapping：Conceptual Schema-Internal Schema Mapping

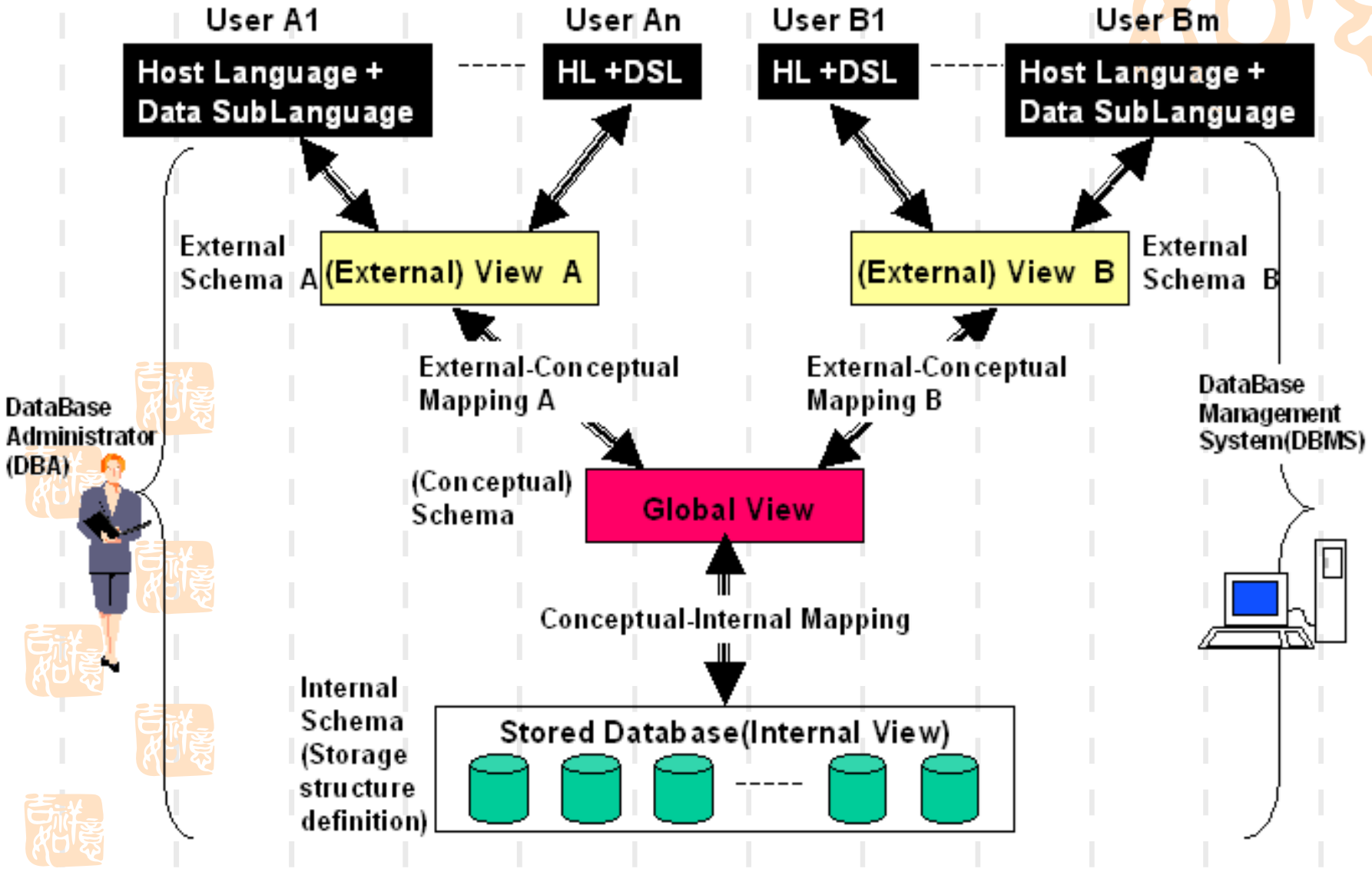
----将概念模式映射为内模式，从而支持实现数据概念视图向内部视图的转换

----便于计算机进行存储和处理



数据库系统的抽象结构

(4)数据库系统的标准结构



为什么要按照标准结构进行数据库系统的抽象？

数据库系统的标准结构

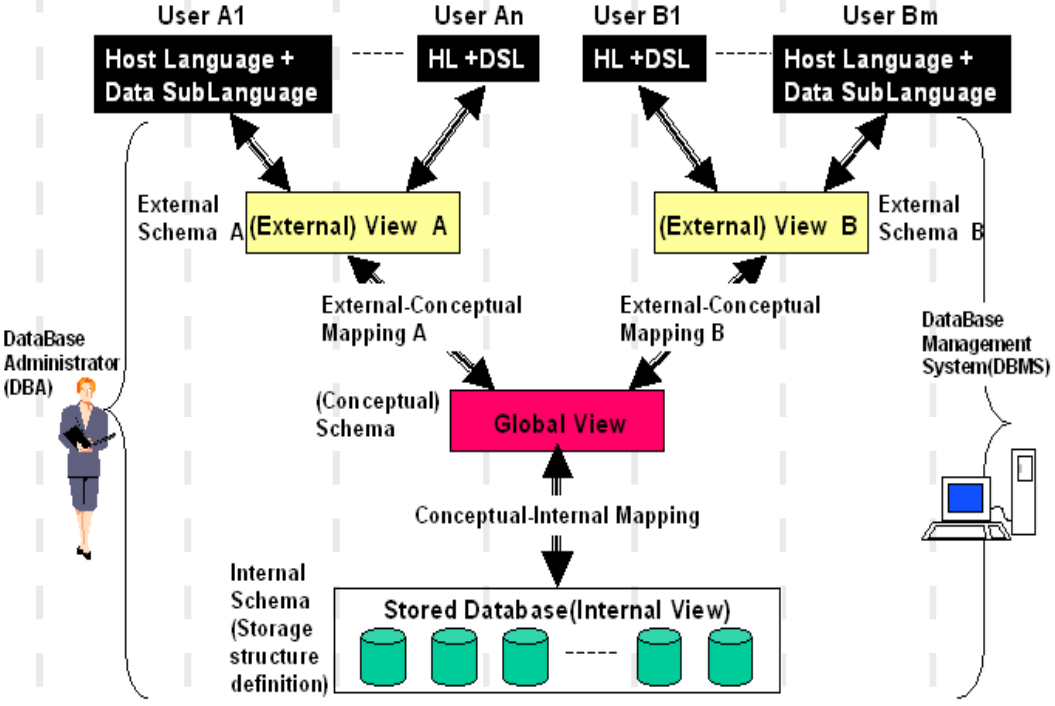
(5)两级独立性

- **逻辑数据独立性**

当概念模式变化时，可以不改变外部模式(只需改变E-C Mapping)，从而无需改变应用程序。

- **物理数据独立性**

当内部模式变化时，可以不改变概念模式(只需改变C-I Mapping)，从而不改变外部模式。



第1章 绪论

- 1.1 数据管理技术的发展
- 1.2 数据库系统
- 1.3 数据模型
- 1.4 数据库模式
- 1.5 数据库管理系统

什么是数据库管理系统？

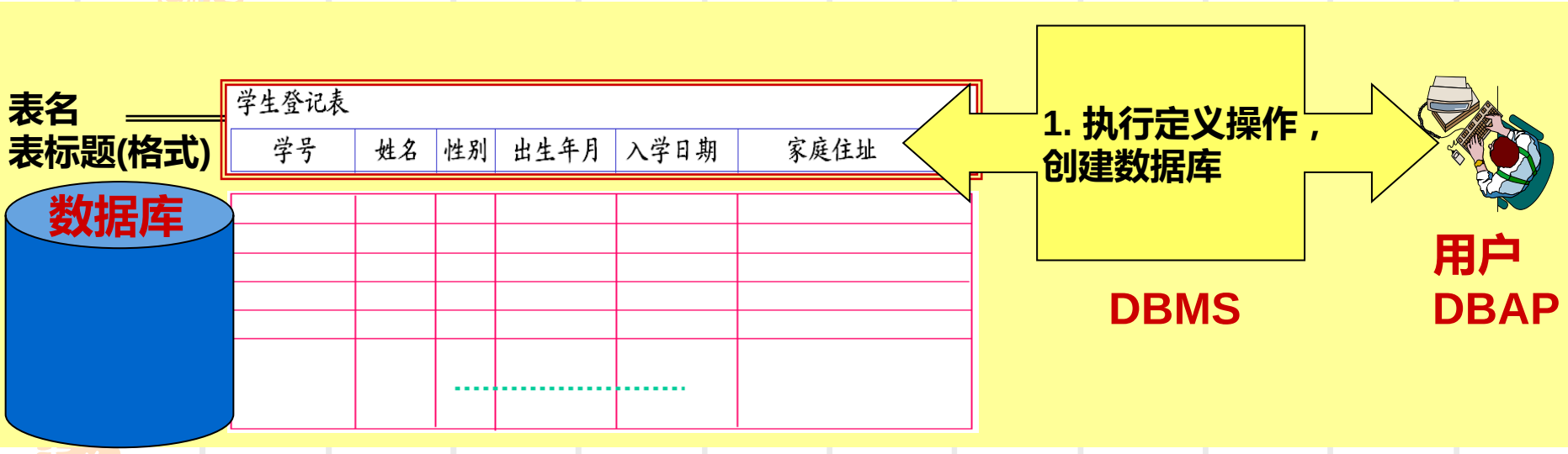
从用户角度看

怎样利用DBMS管理数据库

从用户角度看数据库管理系统DBMS的功能

➢数据库定义: 定义数据库中Table的名称、标题(内含的属性名称及对该属性的值的要求)等

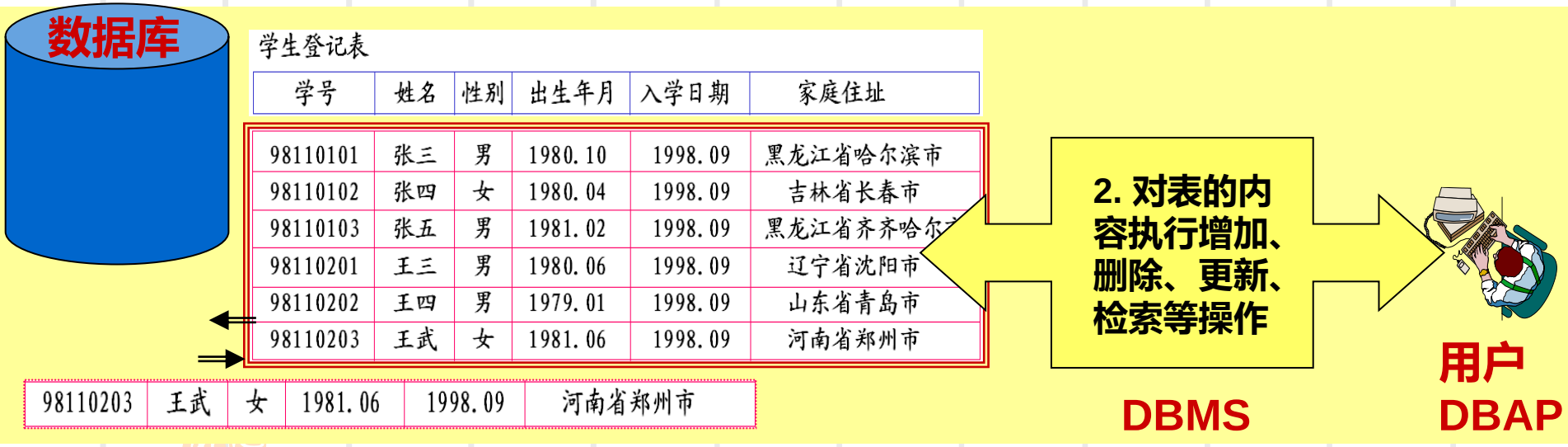
- ❑ DBMS提供一套数据定义语言(DDL:Data Definition Language)给用户
- ❑ 用户使用DDL描述其所要建立表的格式
- ❑ DBMS依照用户的定义，创建数据库及其中的Table



从用户角度看数据库管理系统DBMS的功能

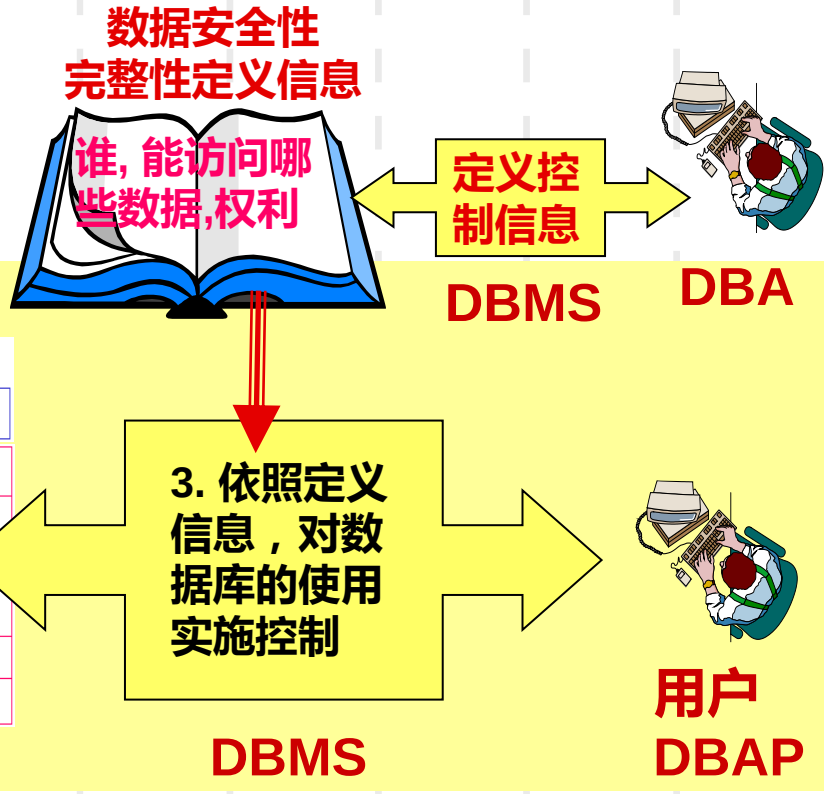
➢数据库操纵：向数据库的Table中增加/删除/更新数据及对数据进行查询、检索、统计等。

- ❑ DBMS提供一套数据操纵语言(DML:Data Manipulation Language)给用户。
- ❑ 用户使用DML描述其所要进行的增、删、改、查等操作。
- ❑ DBMS依照用户的操作描述，实际执行这些操作。



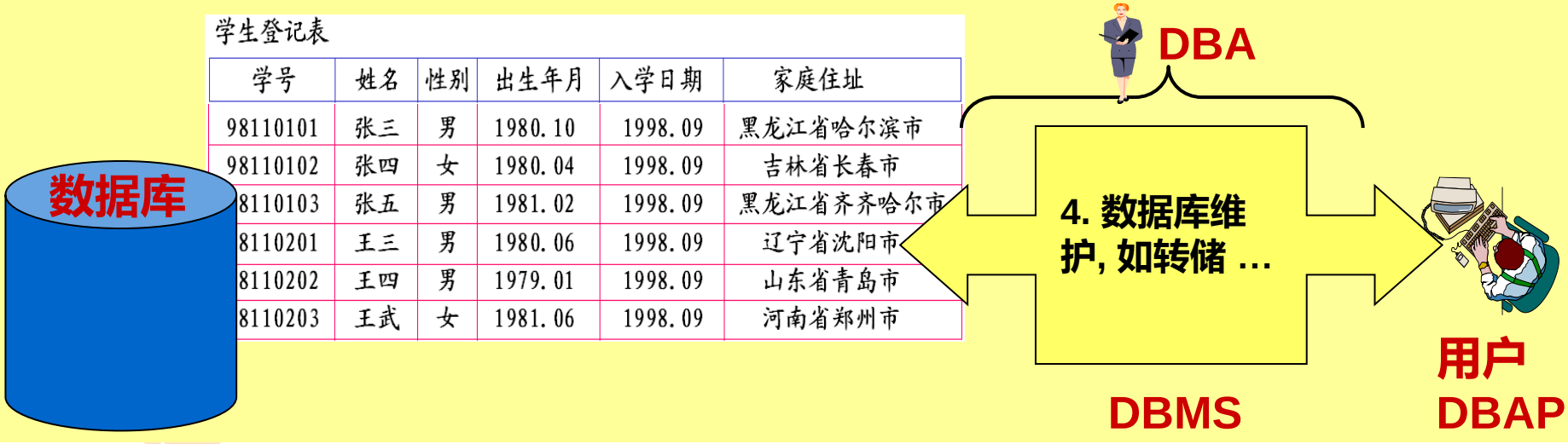
从用户角度看数据库管理系统DBMS的功能

- 数据库控制: 控制数据库中数据的使用---哪些用户可以使用,哪些不可以
  - DBMS提供一套数据控制语言(DCL:Data Control Language)给用户
  - 用户使用DCL描述其对数据库所要实施的控制
  - DBMS依照用户的描述，实际进行控制



从用户角度看数据库管理系统DBMS的功能

- 数据库维护: 转储/恢复/重组/性能监测/分析…
  - DBMS提供一系列程序(实用程序/例行程序)_给用户
  - 在这些程序中提供了对数据库维护的各种功能
  - 用户使用这些程序进行各种数据库维护操作
- 数据库维护的实用程序，一般由数据库管理员(DBA)来使用



从用户角度看数据库管理系统DBMS的功能

数据库语言：使用者通过数据库语言利用DBMS操作数据库

数据定义语言(DDL:Data Definition Language)
----DBMS提供给用户,以便用户定义数据格式

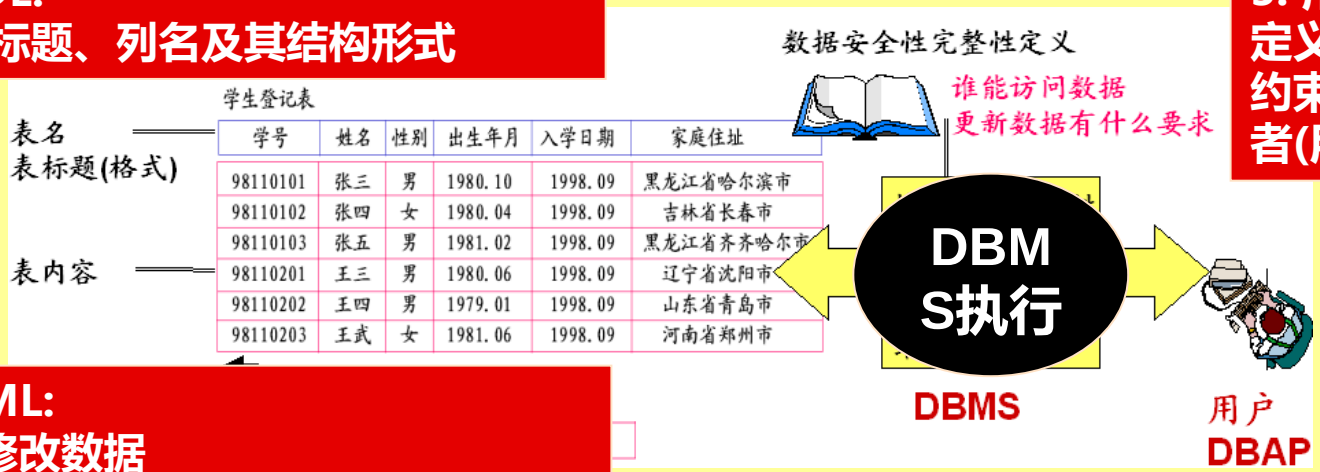
数据操纵语言(DML:Data Manipulation Language)
----DBMS提供给用户,以便用户对数据进行操作

数据控制语言(DCL:Data Control Language)
----DBMS提供给用户,以便用户对数据进行控制

数据库各种操作的执行
----DBMS按用户要求进行定义、操纵、控制和维护

SQL语言：结构化的数据库语言

1. 用户使用DDL:
定义表名，表标题、列名及其结构形式



2. 用户使用DML:
增加、删除、修改数据
查询数据、检索数据、统计数据

3. 用户使用DCL:
定义对不同操作的约束、对不同操作者(用户)的约束

从用户角度看数据库管理系统DBMS的功能

数据库语言与高级语言:

一条数据库语言语句相当于高级语言的一个或多个循环程序

数据库语言(标准的：SQL语言)

Select 学号，姓名 From 学生登记表 Where 性别 = ‘男’；

For K=1 to 最后一条记录
读第K条记录
If 性别 = ‘男’ then
显示第K条记录
Endif
Next K

学生登记表

学号	姓名	性别	出生年月	入学日期	家庭住址
98110101	张三	男	1980.10	1998.09	黑龙江省哈尔滨市
98110102	张四	女	1980.04	1998.09	吉林省长春市
98110103	张五	男	1981.02	1998.09	黑龙江省齐齐哈尔市
98110201	王三	男	1980.06	1998.09	辽宁省沈阳市
98110202	王四	男	1979.01	1998.09	山东省青岛市
98110203	王武	女	1981.06	1998.09	河南省郑州市

什么是数据库管理系统?

数据库语言可以嵌入到高级语言(宿主语言)中使用

交互式数据库语言(标准的：SQL语言)

Select 学号，姓名
From 学生登记表
Where 性别 = ‘男’ ；

嵌入式数据库语言(SQL语句嵌入到某一种高级语言中)

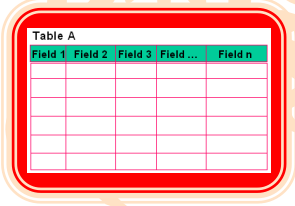
```
exec sql include sqlca;
exec sql begin declar section;
        char cust_id[5], agent_id[14];
        double dollar_sum;
exec sql end declare section;
int main()
{  char cid_prompt[ ]="Please enter customer ID:";
    exec sql declare agent_dollars cursor for select aid,sum(dollars) from orders where cid = :cust_id group by aid;
    exec sql whenever sqlerror goto report_error;
    exec sql connect to testdb;
    exec sql whenever not found goto finish;
    while((prompt(cid_prompt,1,cust_id,4)) >=0) {
       exec sql open agent_dollars;
       while(TRUE) {
          exec sql fetch agent_dollars into :agent_id, :dollar_sum;
          printf("%s %11.2f\n",agent_id, dollar_sum);
                    }
    finish: exec sql close agent_dollars;
           exec sql commit work; }
    exec sql disconnect current;
}
```

什么是数据库管理系统?

从系统实现角度看到的DBMS功能模型

系统：数据库管理系统应具有什么功能?

从系统实现角度看DBMS的功能

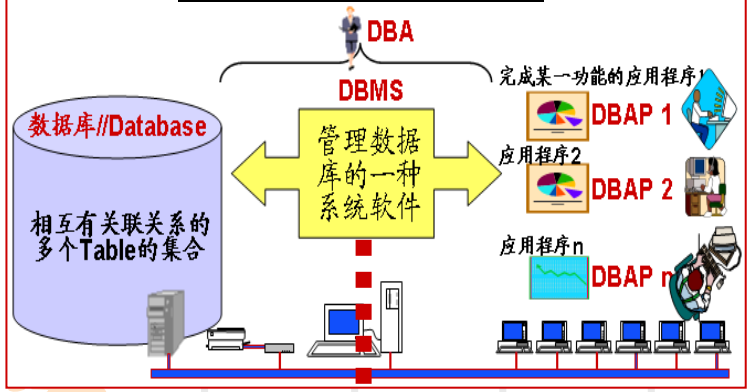


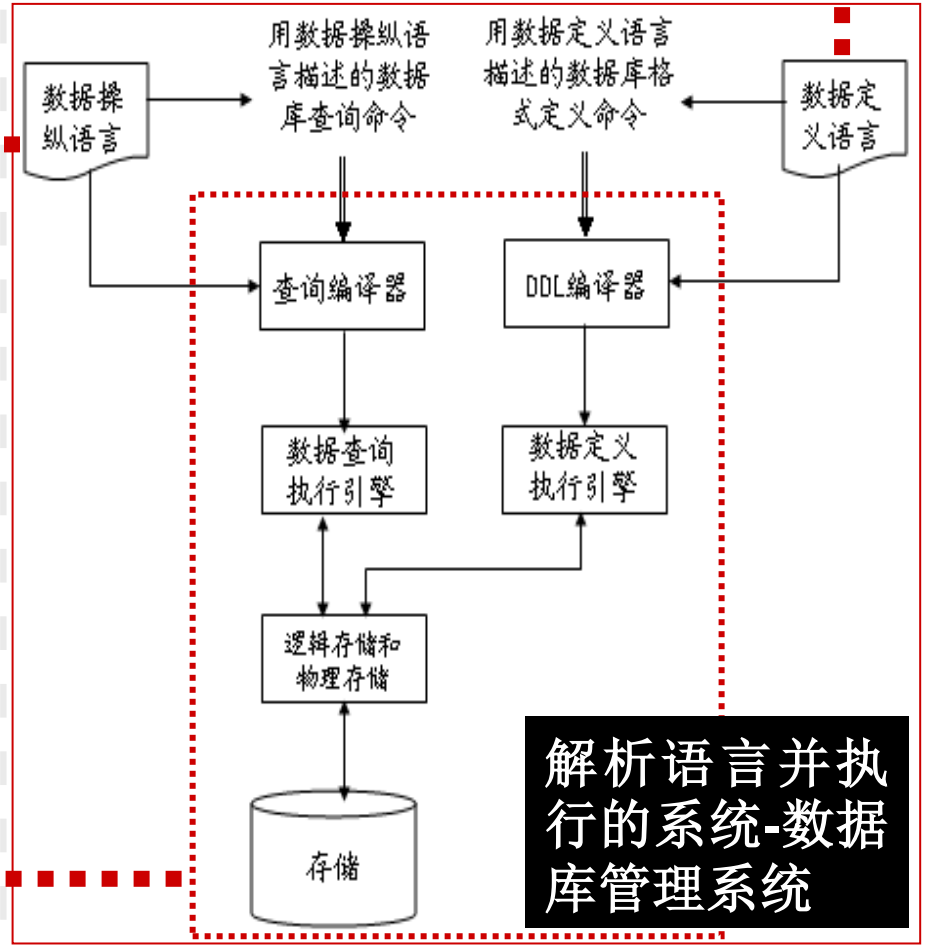
“形式➔构造➔自动化” --数据库管理系统的实现

数据库语言

```
CREATE  TABLE 表名(列名1  类型  [NOT NULL]
        [, 列名2  类型  [NOT NULL]].....);
SELECT  [DISTINCT]  列名1[, 列名2…]
FROM 表名1[, 表名2…]
[WHERE 条件1]
[GROUP BY 列名i1  [, 列名i2 …] [HAVING 条件2]]
[ORDER BY 表达式1  [ASC / DESC]...]
```

数据库系统

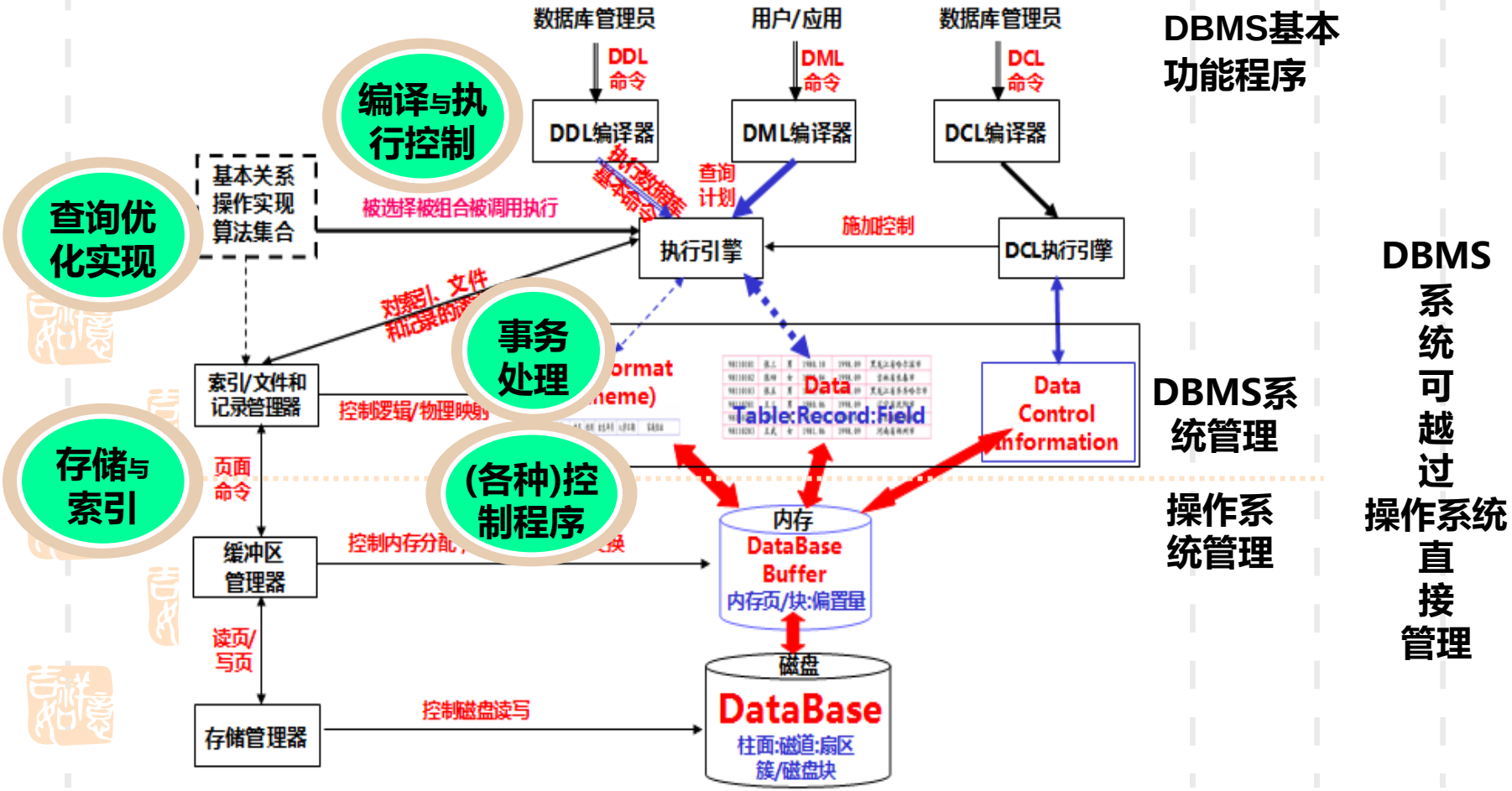




从系统实现角度看DBMS的功能

数据库管理系统(系统软件):从系统角度看DBMS的功能

- **DBMS为完成DB管理，在后台运行着一系列程序…**

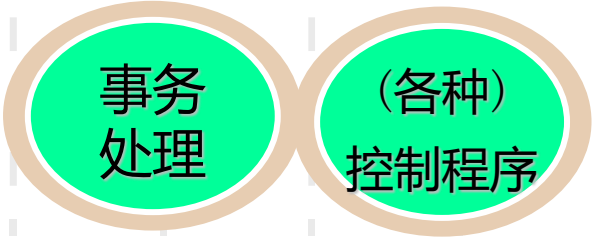


从系统实现角度看DBMS的功能

- DBMS为完成DB管理，在后台运行着一系列程序…
  - 语言编译器：将用数据库语言书写的内容，翻译成DBMS可执行的命令。例如: DDL编译器, DML编译器, DCL编译器等;
  - 查询优化(执行引擎)与查询实现(基本命令的不同执行算法)：提高数据库检索速度的手段；例如贯穿于数据存取各个阶段的优化程序;
  - 数据存取与索引：提供数据在磁盘、磁带等上的高效存取手段。例如: 存储管理器，缓冲区管理器，索引/文件和记录管理器等;
  - 通信控制：提供网络环境下数据库操作与数据传输的手段。

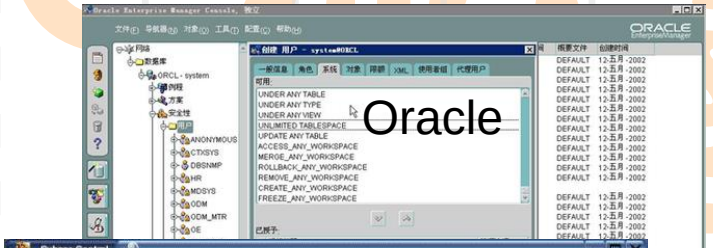
从系统实现角度看DBMS的功能

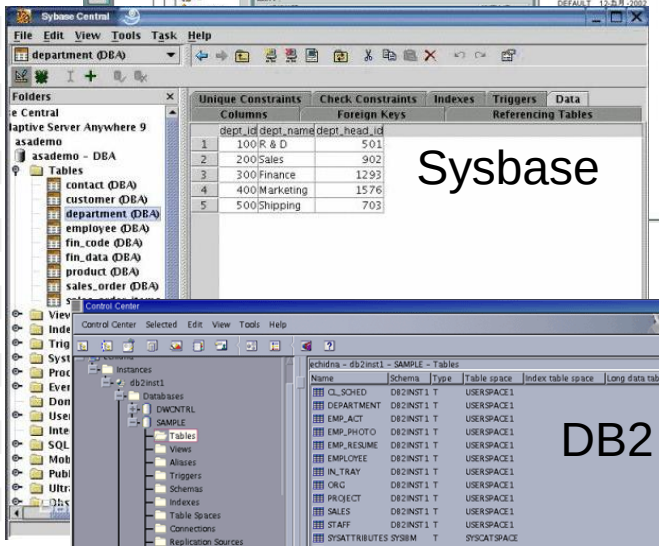
- DBMS为完成DB管理，在后台运行着一系列程序…
  - 事务管理：提供提高可靠性并避免并发操作错误的手段；
  - 故障恢复：使数据库自动恢复到故障发生前正确状态的手段，例如提供了备份、运行日志操控等实用程序；
  - 安全性控制：提供合法性检验，避免非授权非法用户访问数据库的手段
  - 完整性控制：提供数据及数据操作正确性检查的手段；
  - 数据字典管理：管理用户已经定义的信息；
  - 应用程序接口(API)：提供应用程序使用DBMS特定功能的手段；
  - 数据库数据装载、重组等实用程序；
  - 数据库性能分析：统计在运行过程中数据库的各种性能数据，便于优化运行；
  - …

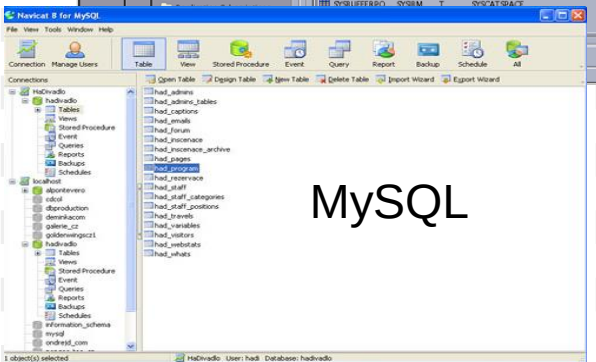


典型的数据库管理系统(DBMS)

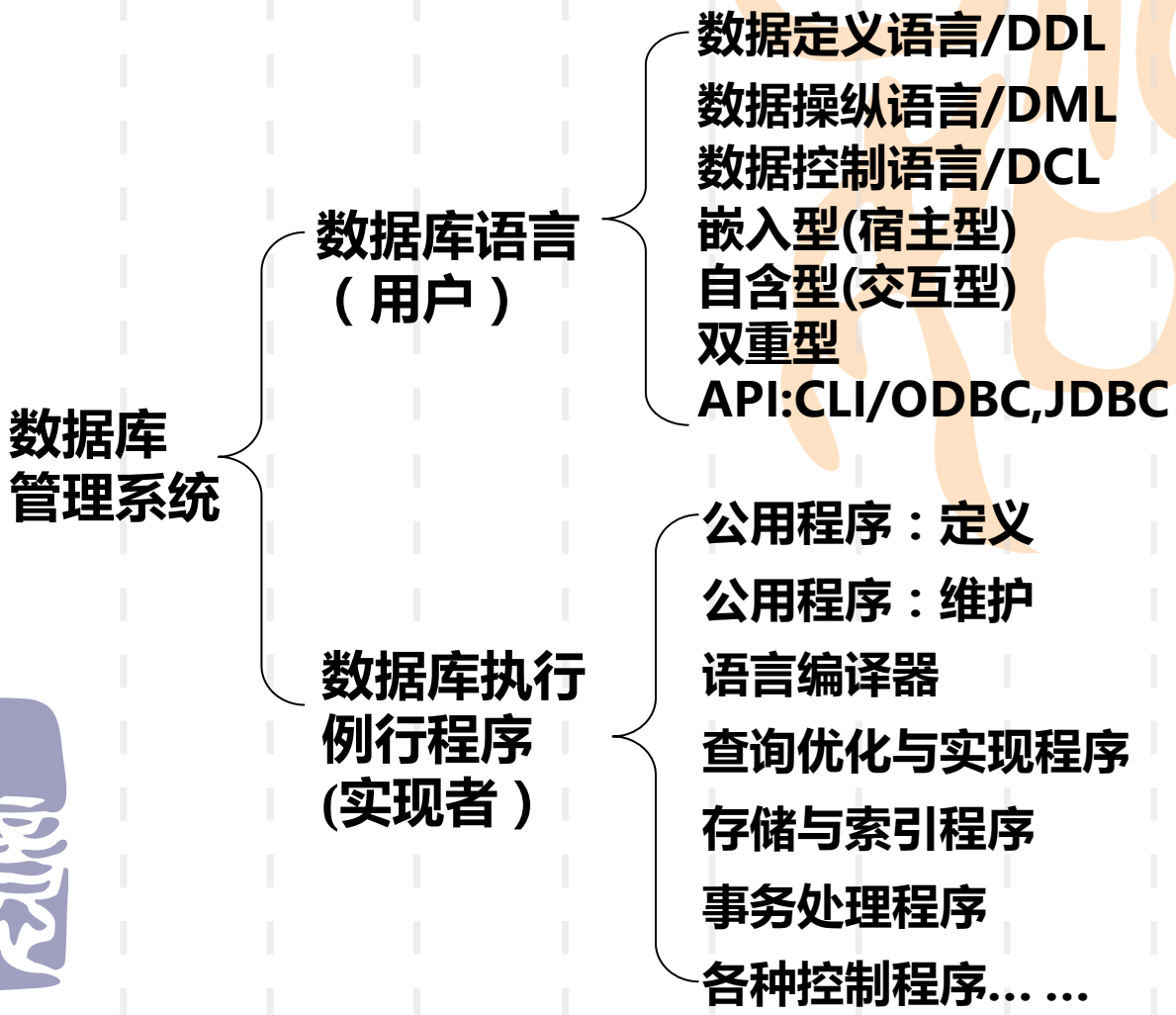
--Oracle
--DB 2 (IBM)
--Sybase
--Informix
--Ingres
--MySQL
--PostgreSQL
--MS SQL Server
--…







数据库管理系统的两个方面



作业

- 1.3
- 1.6
- 1.9
- 1.15
- 1.17
- 1.18