



数据库系统概论

第一章 绪论

❖ 为什么要开设该课程？

- ◆ 其重要性以及应用的广泛性在日常生活中的体现
 - 学分制系统
 - 医院的挂号等系统
 - 银行的各种业务系统
 - 火车票的查询和订票系统
 - . . .

❖ 学会这么课程之后你能做什么？

- ◆ DBA
- ◆ 开发动态网站
- ◆ 其他与数据库有关的应用
- ◆

- 第一节 数据库系统概述
- ❖ 第二节 数据模型
- ❖ 第三节 数据库系统结构
- ❖ 第四节 数据库系统的组成



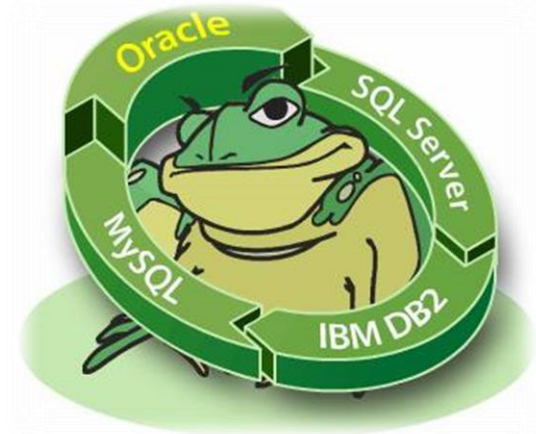
❖ 教学目标：

- ◆ 掌握数据、数据库、数据库管理系统、数据库系统等基本概念；数据库系统四大特点
- ◆ 了解数据管理技术的产生和发展的三个阶段及各阶段的特点

❖ 重点

- ◆ 数据库
- ◆ 数据库管理系统
- ◆ 数据库系统的特点和组成

- ❖ 数据库技术产生于六十年代末，是数据管理的最新技术，是计算机科学的重要分支。
- ❖ 数据库技术是信息系统的核心和基础，它的出现极大地促进了计算机应用向各行各业的渗透。
- ❖ 数据库的建设规模、数据库信息量的大小和使用频度已成为衡量一个国家信息化程度的重要标志



❖ 数据(Data)是数据库中存储的基本对象

❖ 数据的定义

- ◆ 描述事物的符号记录

❖ 数据的种类

- ◆ 文字、图形、图象、声音

❖ 数据的特点

- ◆ 数据与其语义是不可分的



❖ 学生档案中的学生记录

- ◆ (李明, 男, 1972, 江苏, 计算机系, 1990)

❖ 数据的形式不能完全表达其内容

❖ 数据的解释

- ◆ 语义：学生姓名，性别，出生年月，籍贯，所在系别，入学时间
- ◆ 解释：李明是个大学生，1972年出生，江苏人，1990年考入计算机系

❖ 请给出另一个解释和语义

❖ 数据库的定义

- ◆ 数据库(Database,简称DB)是长期储存在计算机内、有组织的、可共享的大量数据的集合。

❖ 数据库的基本特征

- ◆ 数据按一定的数据模型组织、描述和储存
- ◆ 可为各种用户共享
- ◆ 冗余度较小
- ◆ 数据独立性较高
- ◆ 易扩展



❖ 什么是DBMS

- ◆ 位于用户与操作系统之间的一层数据管理软件。
- ◆ 是基础软件，是一个大型复杂的软件系统



- ◆ ORACLE
- ◆ **SQL SERVER**
- ◆ SYBASE
- ◆ INFORMIX
- ◆ DB/2
- ◆ COBASE
- ◆ **MySQL**
- ◆ PBASE
- ◆ EasyBase
- ◆ OpenBase



❖ 数据定义功能

- ◆ 提供数据定义语言(DDL)
- ◆ 定义数据库中的数据对象

❖ 数据组织、存储和管理

- ◆ 分类组织、存储和管理各种数据
- ◆ 确定组织数据的文件结构和存取方式
- ◆ 实现数据之间的联系
- ◆ 提供多种存取方法提高存取效率

❖ 数据操纵功能

- ◆ 提供数据操纵语言(DML)
- ◆ 实现对数据库的基本操作（查询、插入、删除和修改）

❖ 数据库的事务管理和运行管理

- ◆ 数据库在建立、运行和维护时由DBMS统一管理和控制保证数据的安全性、完整性、多用户对数据的并发使用发生故障后的系统恢复

❖ 数据库的建立和维护功能

- ◆ 数据库初始数据装载转换
- ◆ 数据库转储
- ◆ 介质故障恢复
- ◆ 数据库的重组织
- ◆ 性能监视分析等

❖ 其它功能

- ◆ DBMS与网络中其它软件系统的通信
- ◆ 两个DBMS系统的数据转换
- ◆ 异构数据库之间的互访和互操作



❖ 什么是数据库系统 (Database System, DBS)

- ◆ 在计算机系统中引入数据库后的系统
- ◆ 是由数据库、数据库管理系统应用程序和数据库管理员组成的存储、管理、处理和维持数据的系统。

❖ 数据库系统的构成

- ◆ 硬件平台及数据库
- ◆ 软件
- ◆ 人员





应用系统

开发工具

DBMS

OS

数据库

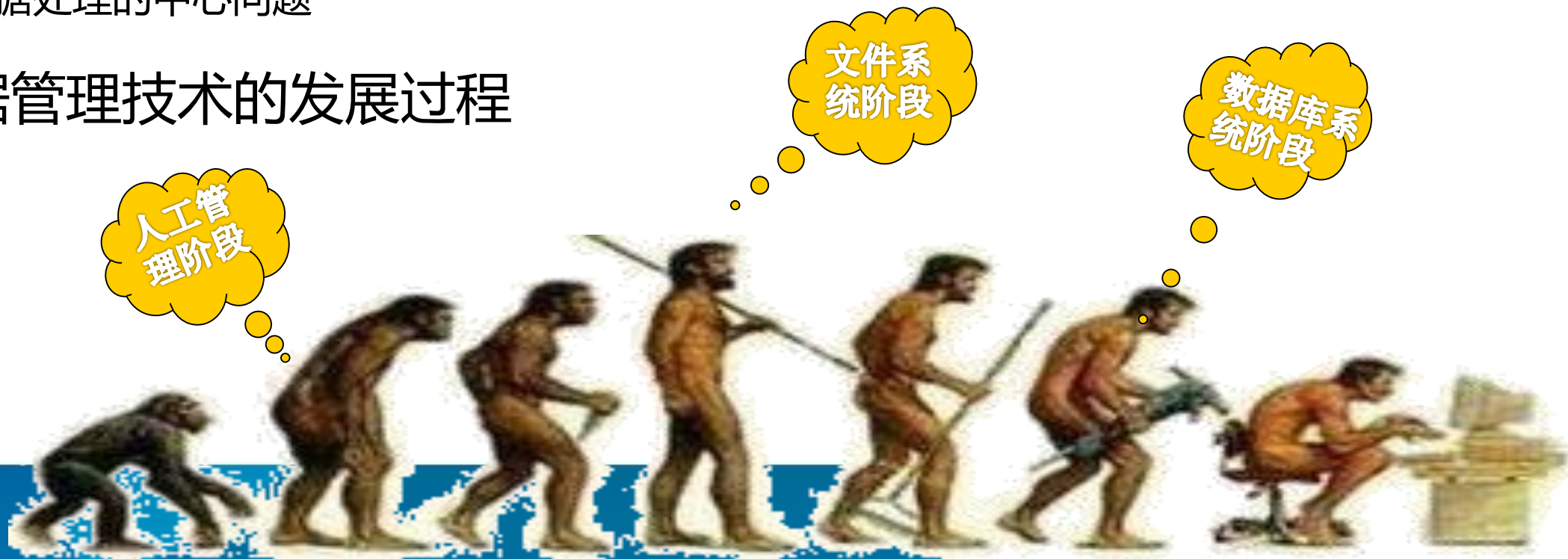


DBA

❖ 什么是数据管理

- ◆ 对数据进行分类、组织、编码、存储、检索和维护
- ◆ 数据处理的中心问题

❖ 数据管理技术的发展过程



❖ 时期

- ◆ 20世纪40年代中--50年代中

❖ 产生的背景

- ◆ 应用需求科学计算
- ◆ 硬件水平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



世界上第一台电子计算机ENIAC

ENIAC是一个庞然大物，体积大约90立方米，占地170平方米，总重量达到30吨。它拥有电子管18000个，继电器1500个，耗电150千瓦，每秒运算5000次，比机械计算机快几百倍到1000倍，比人运算快1000倍到几千倍。并且计算过程是按照编好的程序自动进行的。

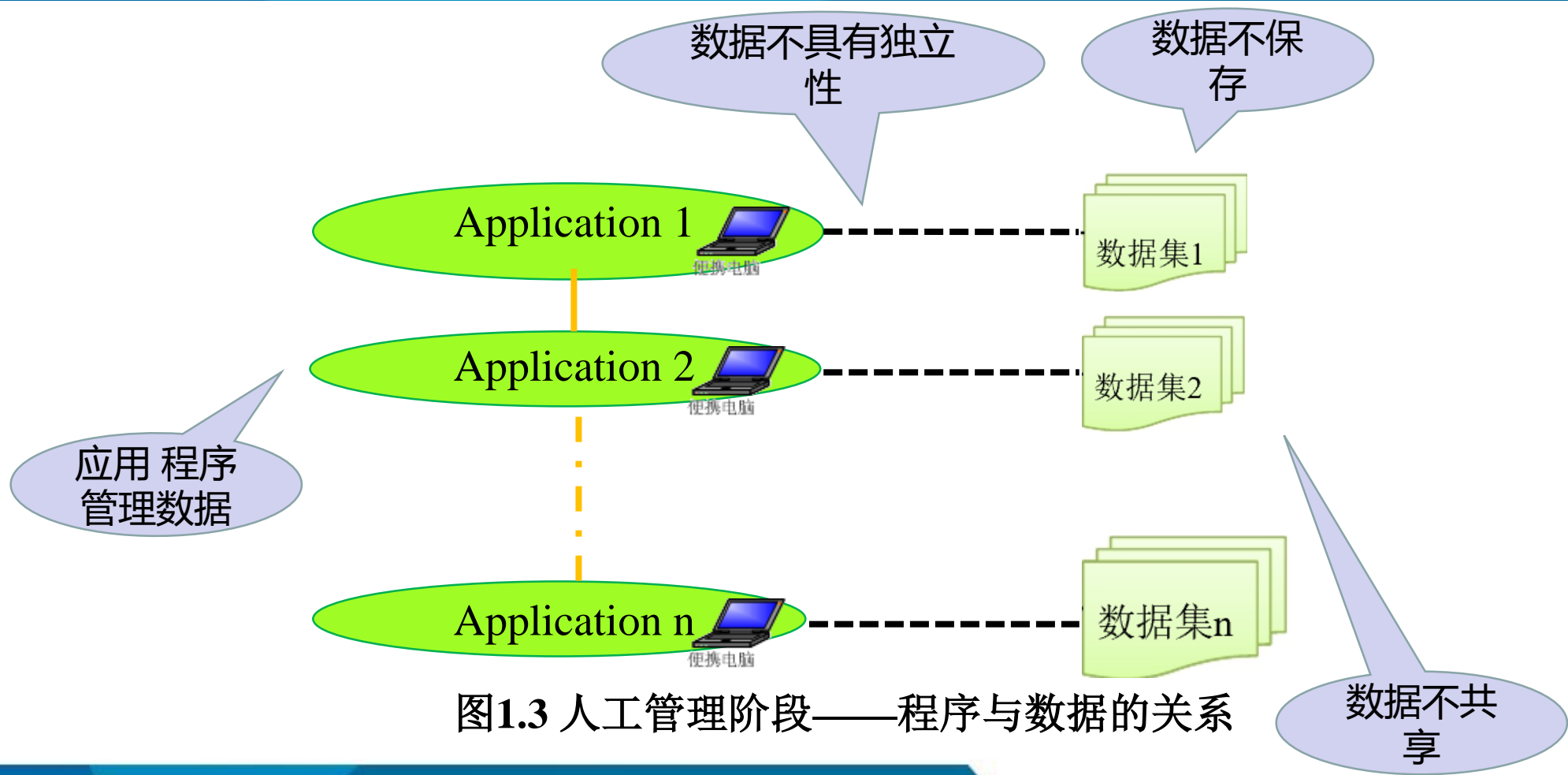


图1.3 人工管理阶段——程序与数据的关系

❖ 特点

- ◆ 数据的管理者：用户（程序员），数据不保存
- ◆ 数据面向的对象：某一应用程序
- ◆ 数据的共享程度：无共享、冗余度极大
- ◆ 数据的独立性：不独立，完全依赖于程序
- ◆ 数据的结构化：无结构
- ◆ 数据控制能力：应用程序自己控制



❖ 时期

- ◆ 20世纪50年代末--60年代中

❖ 产生的背景

- ◆ 应用需求 科学计算、管理
- ◆ 硬件水平 磁盘、磁鼓
- ◆ 软件水平 有文件系统
- ◆ 处理方式 联机实时处理、批处理



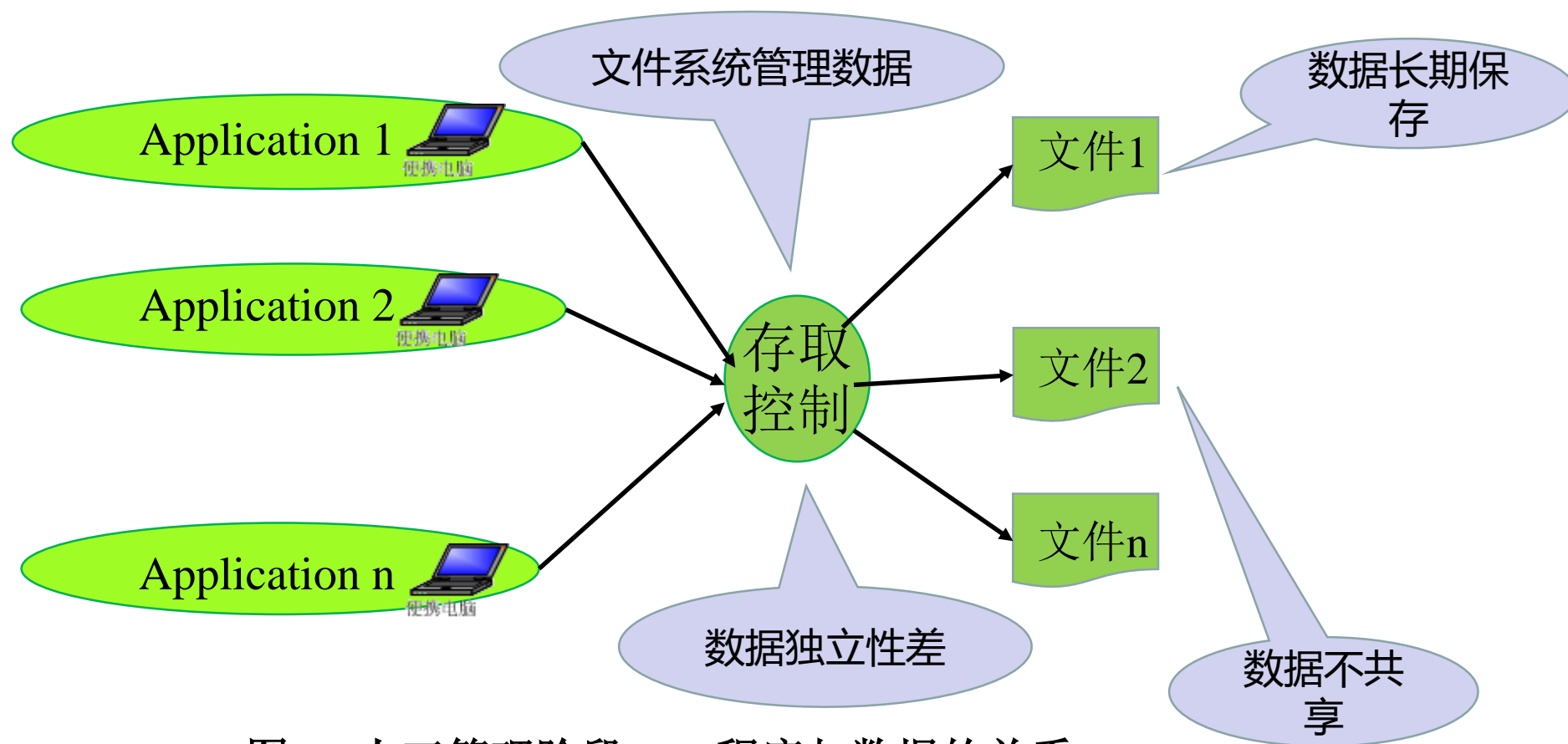


图1.4 人工管理阶段——程序与数据的关系

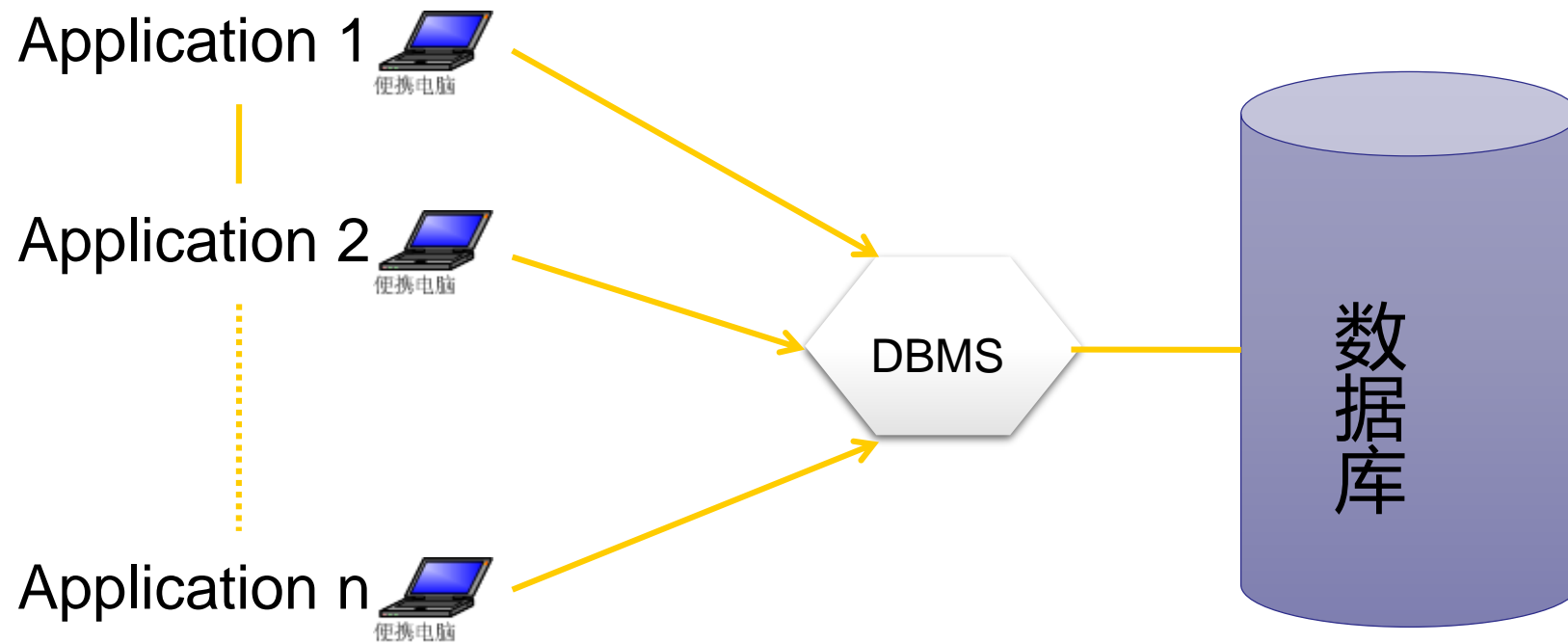
❖ 时期

- ◆ 20世纪60年代末以来

❖ 产生的背景

- ◆ 应用背景 大规模管理
- ◆ 硬件背景 大容量磁盘、磁盘阵列
- ◆ 软件背景 有数据库管理系统
- ◆ 处理方式 联机实时处理,分布处理,批处理





- ❖ 数据结构化
- ❖ 数据的共享性高，冗余度低，易扩充
- ❖ 数据独立性高
- ❖ 数据由DBMS统一管理和控制



- ❖ 整体数据的结构化是数据库的主要特征之一
- ❖ 整体结构化
 - ◆ 不再仅仅针对某一个应用，而是面向全组织
 - ◆ 不仅数据内部结构化，整体是结构化的，数据之间具有联系
- ❖ 数据库中实现的是数据的真正结构化
 - ◆ 数据的结构用数据模型描述，无需程序定义和解释
 - ◆ 数据可以变长
 - ◆ 数据的最小存取单位是数据项

数据的共享性高，冗余度低，易扩充

- ❖ 数据库系统从整体角度看待和描述数据，数据面向整个系统，可以被多个用户、多个应用共享使用
- ❖ 数据共享的好处
 - ◆ 减少数据冗余，节约存储空间
 - ◆ 避免数据之间的不相容性与不一致性
 - ◆ 使系统易于扩充



❖ 物理独立性

- ◆ 指用户的应用程序与存储在磁盘上的数据库中数据是相互独立的。当数据的物理存储改变了，应用程序不用改变。

❖ 逻辑独立性

- ◆ 指用户的应用程序与数据库的逻辑结构是相互独立的。数据的逻辑结构改变了，用户程序也可以不变。

❖ 数据独立性是由DBMS的二级映像功能来保证的

❖ DBMS提供的**数据控制功能**

- ◆ 数据的**安全性** (Security) 保护
保护数据，以防止不合法的使用造成的数据的泄密和破坏
- ◆ 数据的**完整性** (Integrity) 检查
将数据控制在有效的范围内，或保证数据之间满足一定的关系
- ◆ **并发** (Concurrency) 控制
对多用户的并发操作加以控制和协调，防止相互干扰而得到错误的结果
- ◆ **数据库恢复** (Recovery)
将数据库从错误状态恢复到某一已知的正确状态

- ❖ 第一节 数据库系统概述
- 第二节 数据模型
- ❖ 第三节 数据库系统结构
- ❖ 第四节 数据库系统的组成



现实世界



学生



课程

?



机器世界



数据库

- ❖ 在数据库中用数据模型这个工具来抽象、表示和处理现实世界中的数据和信息
- ❖ 通俗地讲数据模型就是现实世界的模拟
- ❖ 数据模型应满足三方面要求
 - ◆ 能比较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



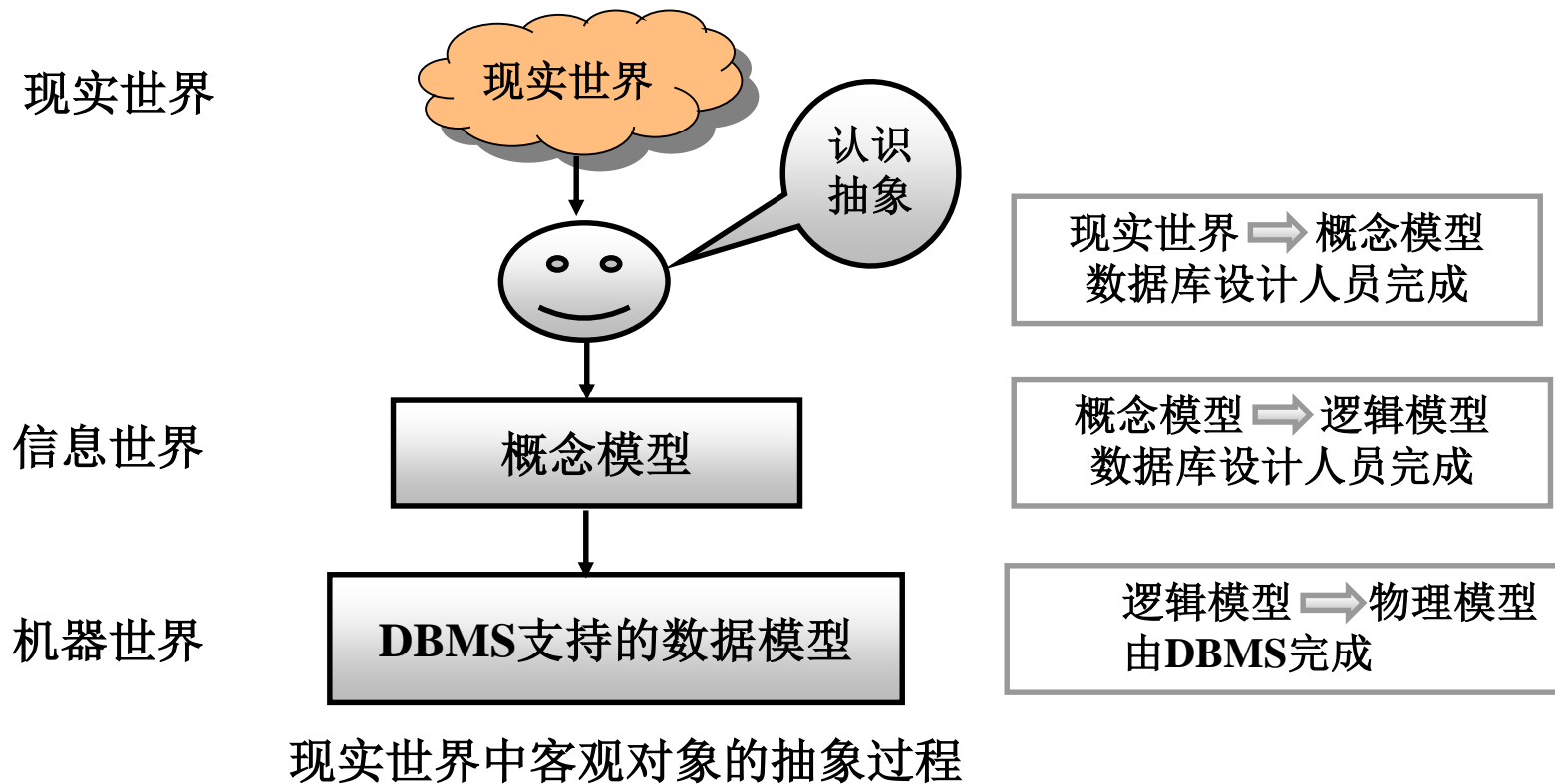
❖ 数据模型分为两类（分属两个不同的层次）

- ◆ **概念模型** 也称信息模型，它是按用户的观点来对数据和信息建模，用于数据库设计
- ◆ **逻辑模型和物理模型**
 - **逻辑模型**主要包括网状模型、层次模型、关系模型、面向对象模型等，按计算机系统的观点对数据建模，用于DBMS实现
 - **物理模型**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

❖ 客观对象的抽象过程

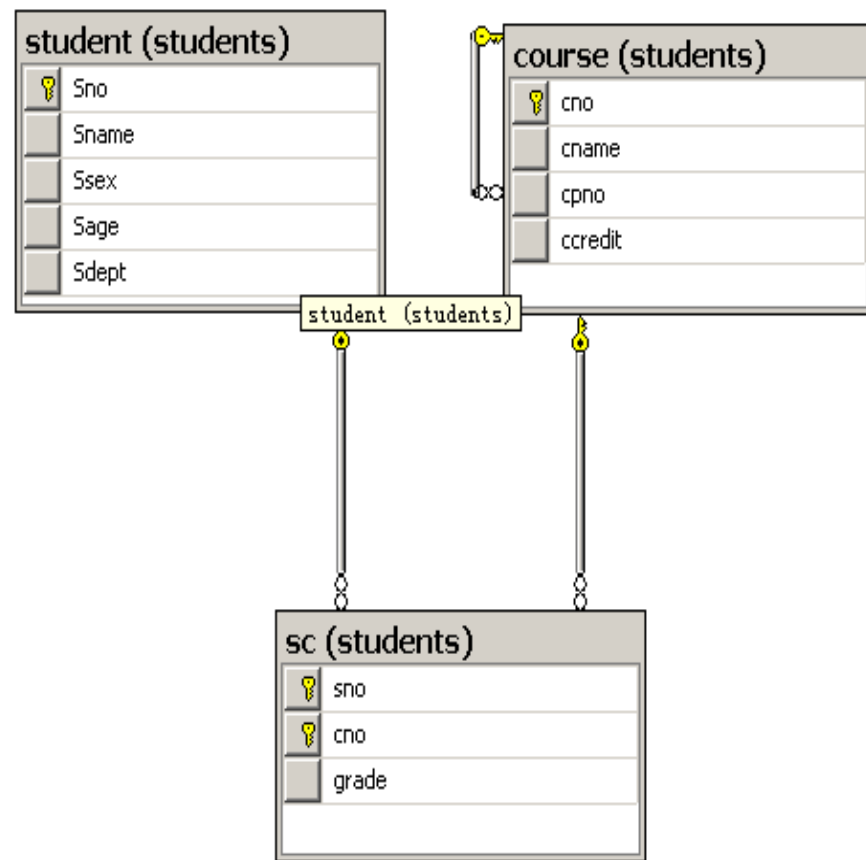
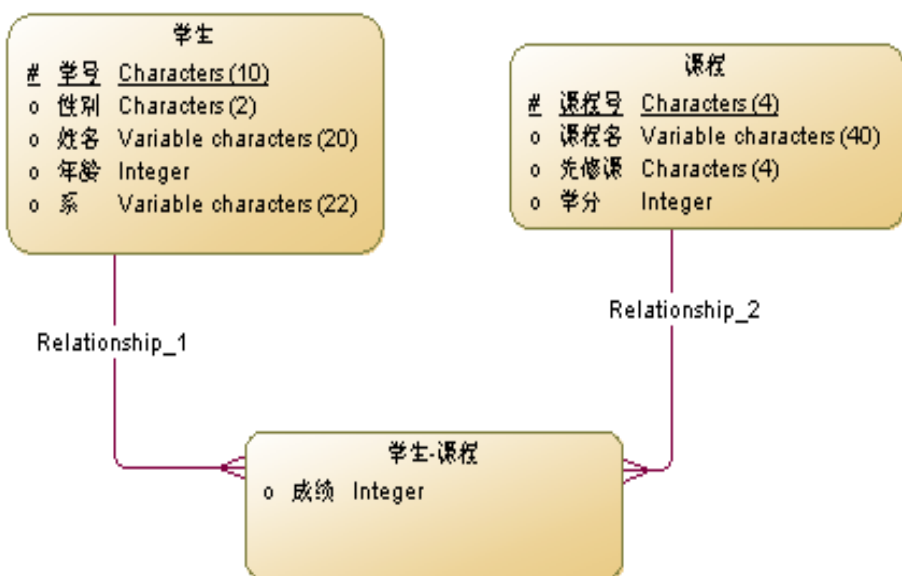
程---两步抽象

- ◆ 现实世界中的客观对象抽象为概念模型;
- ◆ 把概念模型转换为某一DBMS支持的数据模型。



信息世界

学生-课程概念模型



数据库逻辑模型

机器世界

数据结构

数据操作

数据的约束条件

❖ 什么是数据结构

- ◆ 描述数据库的组成对象，以及对象之间的联系

❖ 描述的内容

- ◆ 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



❖ 根据数据结构的类型来命名数据模型

- ◆ 非关系模型
 - 层次模型 (Hierarchical Model)
 - 网状模型(Network Model)
- ◆ 关系模型(Relational Model)
 - 数据结构：关系
- ◆ 面向对象模型(Object Oriented Model)
 - 数据结构：对象



❖ 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)

❖ 数据操作是对系统动态特性的描述

❖ 数据的完整性约束条件

- ◆ 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



❖ 数据模型对完整性约束条件的定义

- ◆ 反映和规定本数据模型必须遵守的**基本的通用**的完整性约束条件。例如在关系模型中，任何关系必须满足实体完整性和参照完整性两个条件
- ◆ 提供定义完整性约束条件的机制，以反映**具体应用**所涉及的数据必须遵守的特定的语义约束条件

- ❖ 关系数据模型的数据结构
- ❖ 关系数据模型的操纵和完整性约束
- ❖ 关系数据模型的存储结构
- ❖ 关系数据模型的优缺点
- ❖ 典型的关系数据库系统



- ❖ 关系数据库系统采用关系模型作为数据的组织方式
- ❖ 1970年美国IBM公司San Jose研究室的研究员E.F.Codd首次提出了数据库系统的关系模型
- ❖ 计算机厂商新推出的数据库管理系统几乎都支持关系模型
- ❖ <http://baike.baidu.com/view/68348.htm>

- ❖ 在用户观点下，关系模型中数据的**逻辑结构**是一张**二维表**，它由行和列组成。



学 号	姓 名	年 龄	性 别	系 名	年 级
2005004	王小明	19	女	社会学	2005
2005006	黄大鹏	20	男	商品学	2005
2005008	张文斌	18	女	法律	2005
...

学生登记表

❖ 关系 (Relation)

- ◆ 一个关系对应通常说的一张表

❖ 元组 (Tuple)

- ◆ 表中的一行即为一个元组

❖ 属性 (Attribute)

- ◆ 表中的一列即为一个属性，给每一个属性起一个名称即属性名

❖ 码 (Key)

- ◆ 唯一确定一个元组的属性或属性组

❖ 域 (Domain)

- ◆ 是一组具有相同数据类型的值的集合

❖ 分量

- ◆ 元组中的一个属性值

❖ 关系模式

- ◆ 对关系的描述，一般表示为

关系名 (属性1, 属性2,, 属性n)

学生 (学号, 姓名, 年龄, 性别, 系, 年级)



❖ 关系必须是规范化的，满足一定的规范条件

- ◆ 最基本的规范条件：关系的每一个分量必须是一个不可分的数据项。
- ◆ **不允许表中还有表**

职工号	姓名	职 称	工 资			扣 除		实 发
			基 本	津 贴	职 务	房 租	水 电	
86051	陈 平	讲 师	1305	1200	50	160	112	2283

术语对比

关系术语	一般表格的术语
关系名	表名
关系模式	表头（表格的描述）
关系	（一张）二维表
元组	记录或行
属性	列
属性名	列名
属性值	列值
分量	一条记录中的一个列值
非规范关系	表中有表（大表中嵌有小表）

关系数据模型的操纵和完整性约束

- ❖ 数据操作是集合操作，操作对象和操作结果都是关系
 - ◆ 查询
 - ◆ 插入
 - ◆ 删除
 - ◆ 更新
- ❖ 数据操作是集合操作，操作对象和操作结果都是关系，即若干元组的集合
- ❖ 存取路径对用户隐蔽，用户只要指出“干什么”，不必详细说明“怎么干”

关系数据模型的操纵和完整性约束

❖ 关系的完整性约束条件

实体完整性

参照完整性

用户定义的完整性

- ❖ 第一节 数据库系统概述
- ❖ 第二节 数据模型
- 第三节 数据库系统结构
- ❖ 第四节 数据库系统的组成



- ❖ 从数据库管理系统角度看，数据库系统通常采用三级模式结构，是数据库系统内部的系统结构
- ❖ 从数据库最终用户角度看（数据库系统外部的体系结构），数据库系统的结构分为：
 - ◆ 单用户结构
 - ◆ 主从式结构
 - ◆ 分布式结构
 - ◆ 客户 / 服务器
 - ◆ 浏览器 / 应用服务器 / 数据库服务器多层结构等

- ❖ **数据库系统模式的概念**
- ❖ 数据库系统的三级模式结构
- ❖ 数据库的二级映像功能与数据独立性



❖ “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

例如

学生记录型:

(学号, 姓名, 性别, 系别, 年龄, 籍贯)

一个记录值:

(201315130, 李明, 男, 计算机, 19, 江苏)

❖ 模式 (Schema)

- ◆ 数据库逻辑结构和特征的描述
- ◆ 是型的描述
- ◆ 反映的是数据的结构及其联系
- ◆ 模式是相对稳定的

❖ 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更新而变动

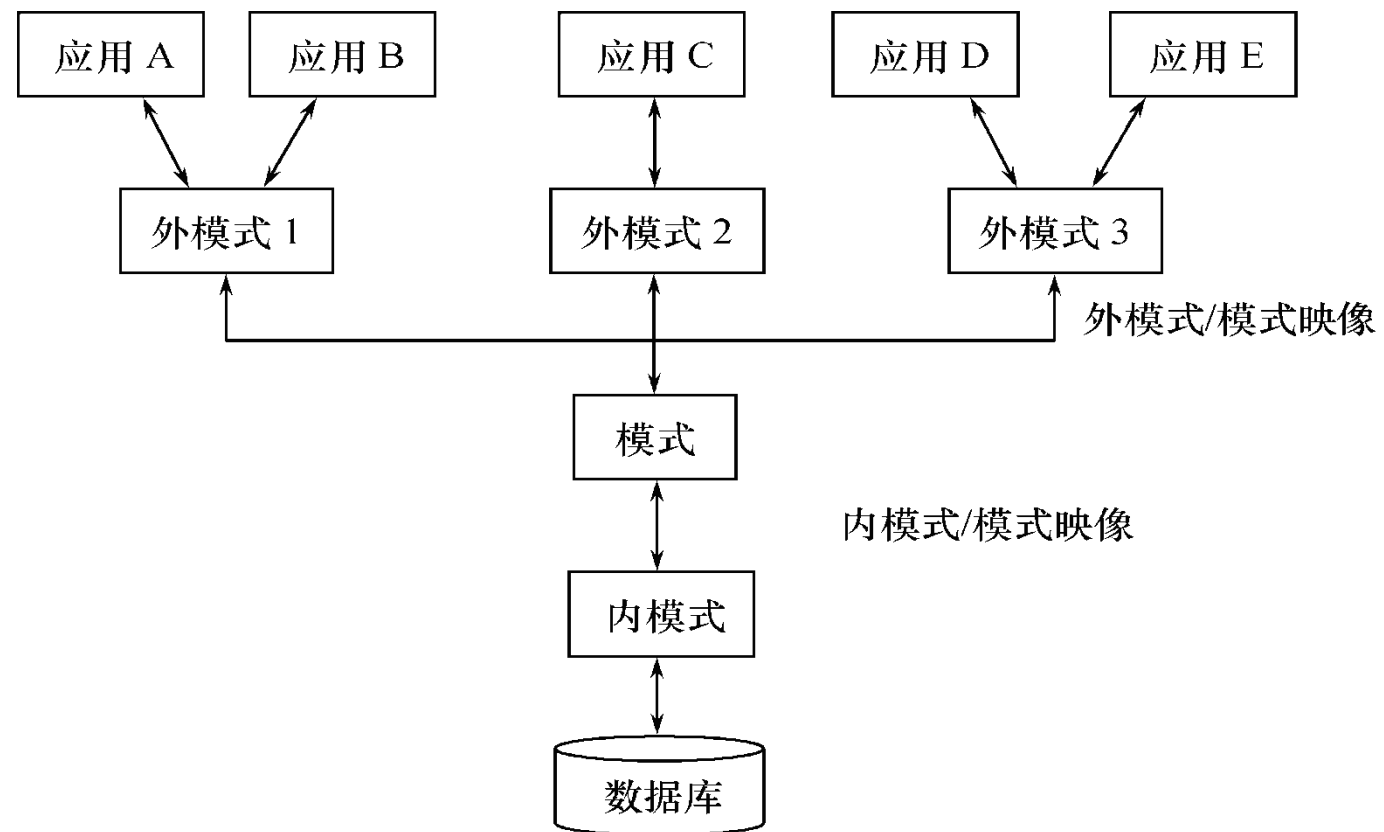
例如：在学生选课数据库模式中，包含学生记录、课程记录和学生选课记录

- ◆ 2013年的一个学生数据库实例，包含：
 - 2013年学校中所有学生的记录
 - 学校开设的所有课程的记录
 - 所有学生选课的记录
- ◆ 2012年度学生数据库模式对应的实例与 2013年度学生数据库模式对应的实例是不同的

- ❖ 数据库系统模式的概念
- ❖ **数据库系统的三级模式结构**
 - ◆ **模式 (Schema)**
 - ◆ **外模式 (External Schema)**
 - ◆ **内模式 (Internal Schema)**
- ❖ 数据库的二级映像功能与数据独立性



数据库系统的三级模式结构



❖ 模式（也称逻辑模式）

- ◆ 数据库中全体数据的逻辑结构和特征的描述
- ◆ 所有用户的公共数据视图，综合了所有用户的需求

❖ 一个数据库只有一个模式

❖ 模式的地位：是数据库系统模式结构的中间层

- ◆ 与数据的物理存储细节和硬件环境无关
- ◆ 与具体的应用程序、开发工具及高级程序设计语言无关

❖ 模式的定义

- ◆ 数据的逻辑结构（数据项的名字、类型、取值范围等）
- ◆ 数据之间的联系
- ◆ 数据有关的安全性、完整性要求



❖ 外模式（也称子模式或用户模式）

- ◆ 数据库用户（包括应用程序员和最终用户）使用的**局部**数据的逻辑结构和特征的描述
- ◆ 数据库用户的数据视图，是与某一应用有关的数据的逻辑表示



❖ 外模式的地位：介于模式与应用之间

- ◆ 模式与外模式的关系：一对多
 - 外模式通常是模式的子集
 - 一个数据库可以有多个外模式。反映了不同的用户的应用需求、看待数据的方式、对数据保密的要求
 - 对模式中同一数据，在外模式中的结构、类型、长度、保密级别等都可以不同
- ◆ 外模式与应用的关系：一对多
 - 同一外模式也可以为某一用户的多个应用系统所使用
 - 但一个应用程序只能使用一个外模式



❖ 内模式（也称存储模式）

- ◆ 是数据物理结构和存储方式的描述
- ◆ 是数据在数据库内部的表示方式
 - 记录的存储方式（顺序存储，按照B树结构存储，按hash方法存储）
 - 索引的组织方式
 - 数据是否压缩存储
 - 数据是否加密
 - 数据存储记录结构的规定

❖ 一个数据库只有一个内模式



- ❖ 数据库系统模式的概念
- ❖ 数据库系统的三级模式结构
- ❖ 数据库的二级映像功能与数据独立性
 - ◆ 外模式 / 模式映像
 - ◆ 模式 / 内模式映像



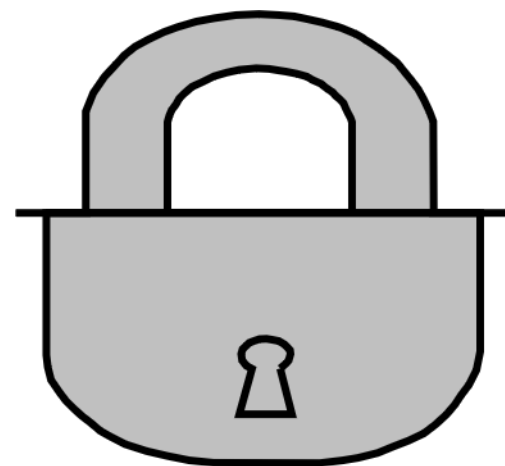
- ❖ 模式：描述的是数据的全局逻辑结构
- ❖ 外模式：描述的是数据的局部逻辑结构
- ❖ 同一个模式可以有任意多个外模式
- ❖ 每一个外模式，数据库系统都有一个外模式 / 模式映象，定义外模式与模式之间的对应关系
- ❖ 映象定义通常包含在各自外模式的描述中

❖ 保证数据的逻辑独立性

- ◆ 当模式改变时，数据库管理员修改有关的外模式 / 模式映象，使外模式保持不变
- ◆ 应用程序是依据数据的外模式编写的，从而应用程序不必修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性。



- ❖ 模式 / 内模式映象定义了数据全局逻辑结构与存储结构之间的对应关系。
 - ◆ 例如:说明逻辑记录和字段在内部是如何表示的
- ❖ 数据库中模式 / 内模式映象是唯一的
- ❖ 该映象定义通常包含在模式描述中



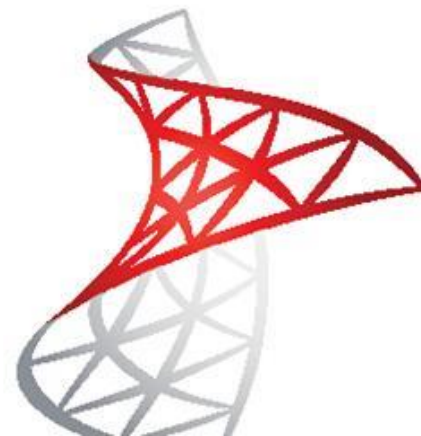
❖ 保证数据的物理独立性

- ◆ 当数据库的存储结构改变了（例如选用了另一种存储结构），数据库管理员修改模式 / 内模式映象，使模式保持不变
- ◆ 应用程序不受影响。保证了数据与程序的物理独立性，简称数据的物理独立性



❖ 数据库模式

- ◆ 即全局逻辑结构是数据库的中心与关键
- ◆ 独立于数据库的其他层次
- ◆ 设计数据库模式结构时应首先确定数据库的逻辑模式



❖ 数据库的内模式

- ◆ 依赖于它的全局逻辑结构
- ◆ 独立于数据库的用户视图，即外模式
- ◆ 独立于具体的存储设备
- ◆ 将全局逻辑结构中所定义的数据结构及其联系按照一定的物理存储策略进行组织，以达到较好的时间与空间效率

❖ 数据库的外模式

- ◆ 面向具体的应用程序
- ◆ 定义在逻辑模式之上
- ◆ 独立于存储模式和存储设备
- ◆ 当应用需求发生较大变化，相应外模式不能满足其视图要求时，该外模式就得做相应改动
- ◆ 设计外模式时应充分考虑到应用的扩充性

❖ 特定的应用程序

- ◆ 在外模式描述的数据结构上编制的
- ◆ 依赖于特定的外模式
- ◆ 与数据库的模式和存储结构独立
- ◆ 不同的应用程序有时可以共用同一个外模式

❖ 数据库的二级映像

- ◆ 保证了数据库外模式的稳定性
- ◆ 从底层保证了应用程序的稳定性，除非应用需求本身发生变化，否则应用程序一般不需要修改

- ❖ 数据与程序之间的独立性，使得数据的定义和描述可以从应用程序中分离出去
- ❖ 数据的存取由DBMS管理
 - ◆ 用户不必考虑存取路径等细节
 - ◆ 简化了应用程序的编制
 - ◆ 大大减少了应用程序的维护和修改



- ❖ 第一节 数据库系统概述
- ❖ 第二节 数据模型
- ❖ 第三节 数据库系统结构
- 第四节 数据库系统的组成



❖ 数据库系统对硬件资源的要求

◆ 足够大的内存

- 操作系统
- DBMS的核心模块
- 数据缓冲区
- 应用程序

◆ 足够大的外存

- 磁盘或磁盘阵列
 - 数据库
 - 光盘、磁带
 - 数据备份
- ### ◆ 较高的通道能力，提高数据传送率

- ❖ DBMS
- ❖ 操作系统
- ❖ 与数据库接口的高级语言及其编译系统
- ❖ 以DBMS为核心的应用开发工具
- ❖ 为特定应用环境开发的数据库应用系统



- ❖ 数据库管理员
- ❖ 系统分析员
- ❖ 数据库设计人员
- ❖ 应用程序员
- ❖ (最终用户)



❖ 不同的人员涉及不同的数据抽象级别，具有不同的数据视图，如下图所示

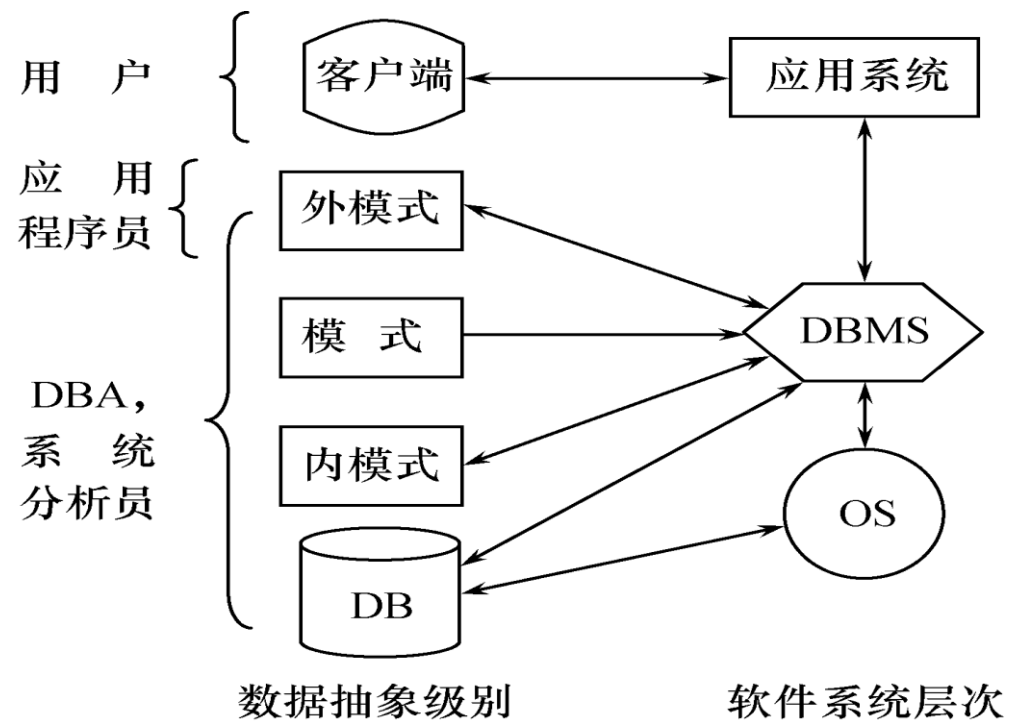


图1.30 各种人员的数据视图

❖ 具体职责:

- ◆ 决定数据库中的信息内容和结构
- ◆ 决定数据库的存储结构和存取策略
- ◆ 定义数据的安全性要求和完整性约束条件
- ◆ 监控数据库的使用和运行
- ◆ 数据库的改进和重组



❖ 系统分析员

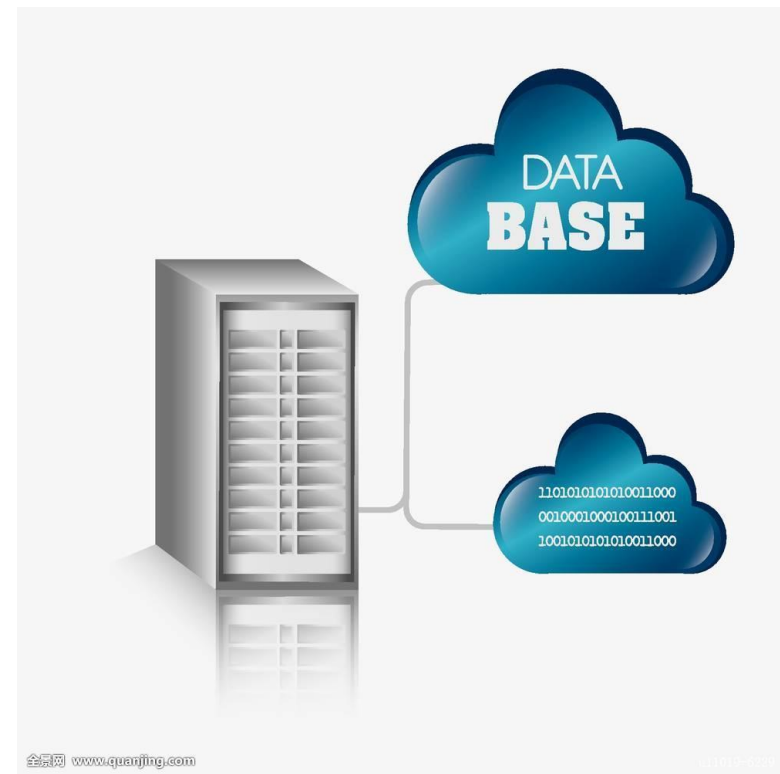
- ◆ 负责应用系统的需求分析和规范说明
- ◆ 与用户及DBA协商，确定系统的软硬件配置
- ◆ 参与数据库系统的概要设计

❖ 数据库设计人员

- ◆ 参加用户需求调查和系统分析
- ◆ 确定数据库中的数据
- ◆ 设计数据库各级模式



- ❖ 设计和编写应用系统的程序模块
- ❖ 进行调试和安装



- ❖ 用户是指最终用户（End User）。最终用户通过应用系统的用户接口使用数据库
- ❖ 偶然用户
 - ◆ 企业或组织机构的高中级管理人员
- ❖ 简单用户
 - ◆ 银行的职员、机票预定人员、旅馆总台服务员
- ❖ 复杂用户
 - ◆ 工程师、科学家、经济学家、科技工作者等
 - ◆ 直接使用数据库语言访问数据库，甚至能够基于数据库管理系统的API编制自己的应用程序

诲女知之乎！知之为知之，
不知为不知，是知也。

