



第1章 数据库系统概述

1.1 数据管理技术的发展

1.2 数据库系统

1.3 数据模型与数据模式

1.4 数据库管理系统 (DBMS)



1.1 数据管理技术的发展

- 研究数据库离不开**数据管理**的概念
- 数据库是数据管理的产物，而数据管理则是数据库的核心任务
- 伴随计算机硬件和软件的发展，数据管理经历了人工管理、文件系统和数据库系统三个发展阶段



人工管理阶段

■ 背景

- 没有操作系统，没有数据管理软件，用户用机器指令编码，通过纸带机输入程序和数据，程序运行完毕后，由用户取走纸带和运算结果

■ 人工管理阶段的数据管理有下列特点：

- ① 数据不保存在计算机内
- ② 没有专用的软件对数据进行管理
- ③ 只有程序（**program**）的概念，没有文件（**file**）的概念。数据的组织方式必须由程序员自行设计与安排
- ④ 数据面向程序。即一组数据对应一个程序

人工管理阶段

- 在人工管理阶段，应用程序与数据之间的关系可用图1-1表示

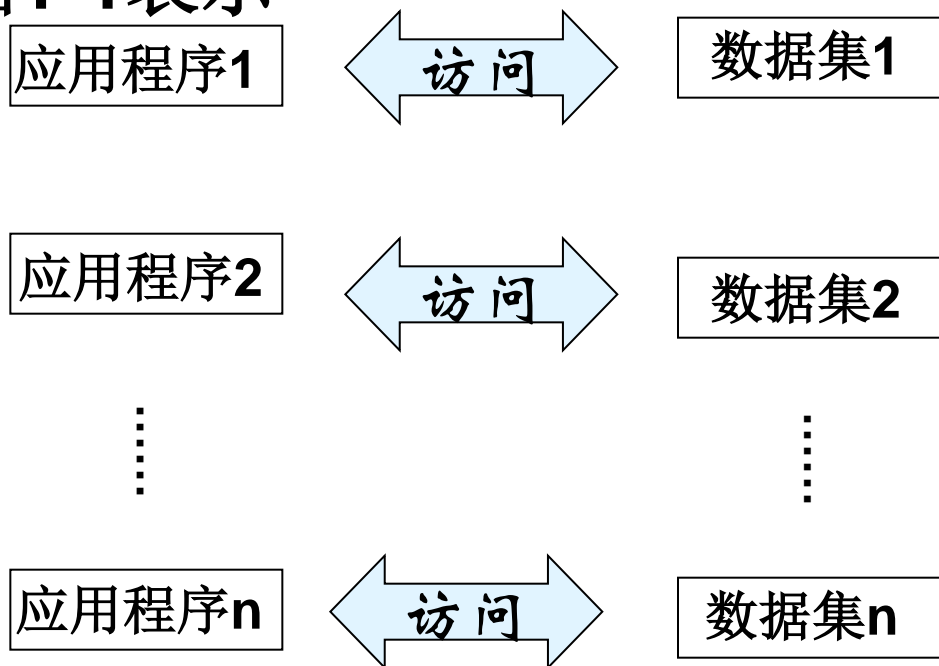


图1-1 程序与数据之间的关系(人工管理阶段)



文件系统阶段

■ 背景

- 外部存储器已有磁盘、磁鼓等直接存取存储设备
- 软件领域出现了高级语言和操作系统

■ 特点

- 系统提供一定的数据管理功能
 - 存取方法
 - 支持对文件的基本操作
 - 数据的存取基本上以记录为单位
- 数据仍是面向应用的
 - 一个数据文件对应一个或几个用户程序
- 数据与程序有一定的独立性
 - 文件的逻辑结构与存储结构由系统进行转换

文件系统阶段

- 在文件系统阶段，程序与数据之间的关系如下图所示

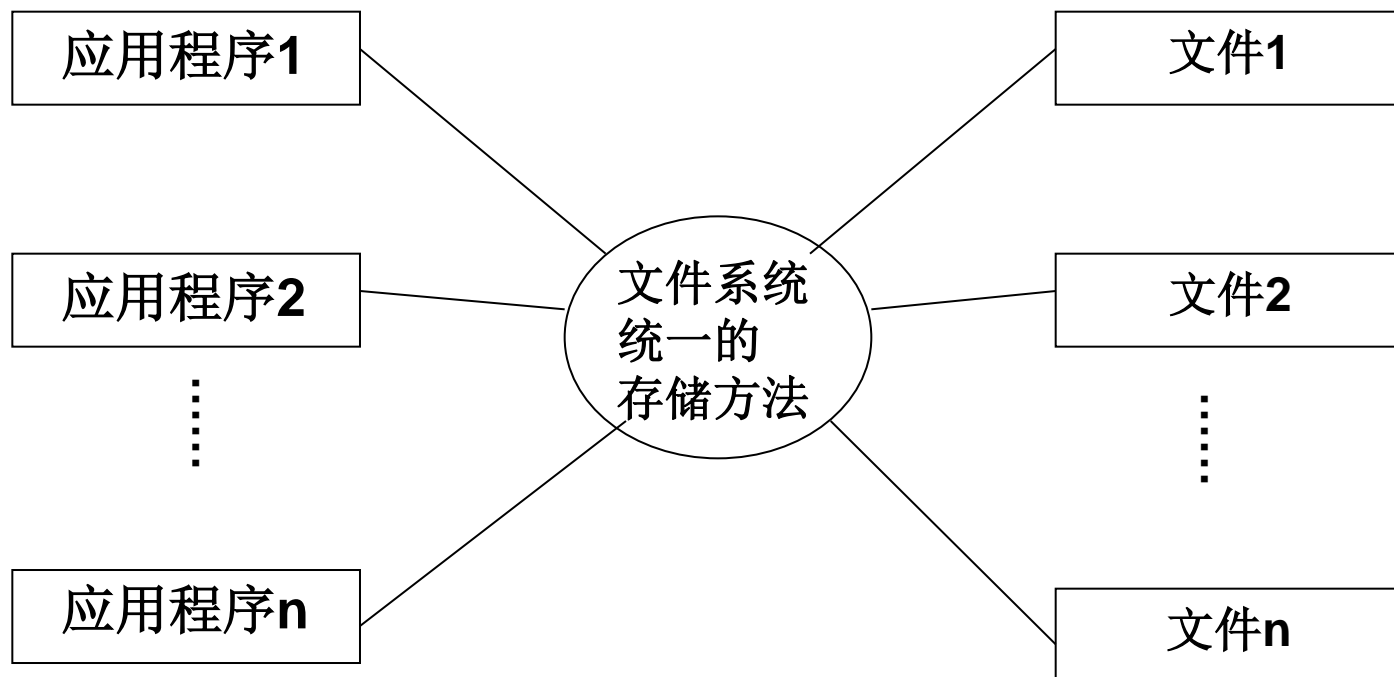


图1-2 程序与数据之间的关系(文件系统阶段)



文件系统阶段

■ 存在的问题

- 数据与程序的独立性差
- 数据的共享性差，冗余度大
 - 数据面向应用
 - 数据孤立
- 数据的不一致性
- 数据查询困难
 - 记录之间无联系
 - 应用自己编程实现
 - 对每个查询都重新编码
- 数据完整性难于保证



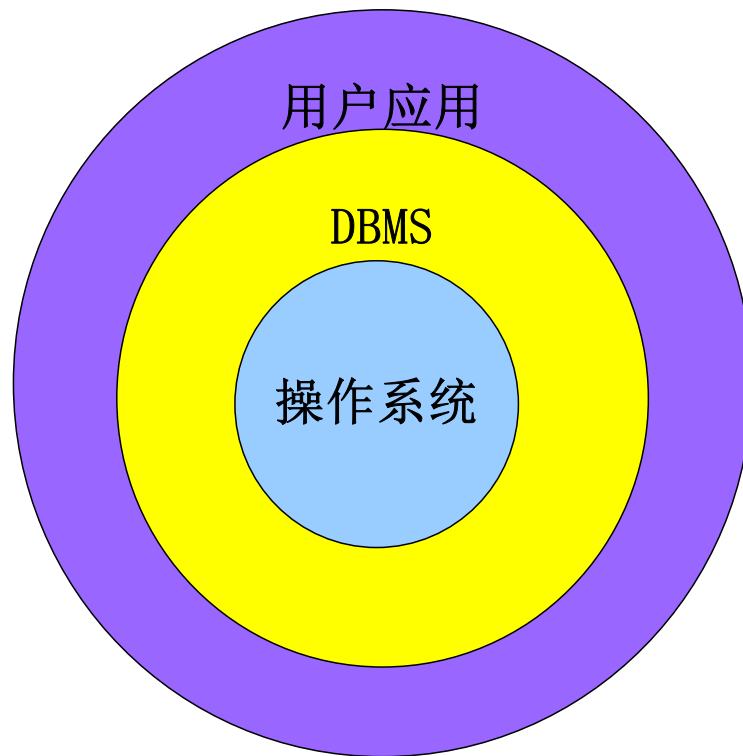
数据库系统阶段

- 从**20世纪60年代末**开始，数据管理全面进入数据库系统阶段，其标志是三件大事：
 - (1) **1968**年，美国**IBM**公司推出层次模型的**IMS**系统
 - (2) **1969**年，美国**CODASYL**(Congference on Data Systems Languages)组织发布了**DBTG**报告，对当时各式各样的数据库进行了总结，提出网状模型
 - (3) **1970**年，美国**IBM**公司的**E.F.Codd** 连续发表论文，提出关系模型，奠定了关系数据库的理论基础

数据库系统阶段

■ 背景

- 计算机管理的数据量大，关系复杂，共享性要求强
- 软、硬件技术的发展
- 文件系统的数据管理方法已很难适应开发应用系统的需要。
- 出现了统一管理数据的专门软件系统，即**数据库管理系统** (database management system, DBMS)





数据库系统阶段

■ 特点

- 数据共享性好、冗余少
- 数据结构化
- 数据独立性高
- 有统一的数据控制功能，数据共享程度高



数据库系统阶段

- 数据共享性好、冗余少

- 数据不再面向某个应用程序，而是面向整个系统。当前所有用户可同时存取数据库中的数据。数据库系统也不仅要描述数据本身，还要描述数据之间的联系

- 数据结构化

- 按照一定的数据模型，将各种数据组织到一个结构化的数据库中，数据库中的数据不再是一盘散沙



数据库系统阶段

■ 数据独立性高

- 数据的独立性得到显著提高，包括逻辑独立性和物理独立性
- **数据的逻辑独立性(logical data independence)**是指当数据的总体逻辑结构改变时，数据的局部逻辑结构不变，由于应用程序是依据数据的局部逻辑结构编写的，所以不必修改应用程序，这样保证了数据与程序间的逻辑独立性
- **数据的物理独立性(physical data independence)**是指当数据的存储结构改变时，数据的逻辑结构不变，当然更不必修改应用程序



数据库系统阶段

- 统一的数据控制功能，数据共享程度高
 - 数据的安全性控制（Security）
 - 保护数据以防止不合法的使用所造成的数据泄露和破坏
 - 措施：用户标识与鉴定，存取控制
 - 数据的完整性控制（Integrity）
 - 数据的正确性、有效性、相容性
 - 措施：完整性约束条件定义和检查
 - 并发控制（Concurrency）
 - 对多用户的并发操作加以控制、协调，防止其互相干扰而得到错误的结果并使数据库完整性遭到破坏
 - 措施：封锁

数据库系统阶段

在数据库系统阶段，程序与数据之间的关系可用图1-4表示

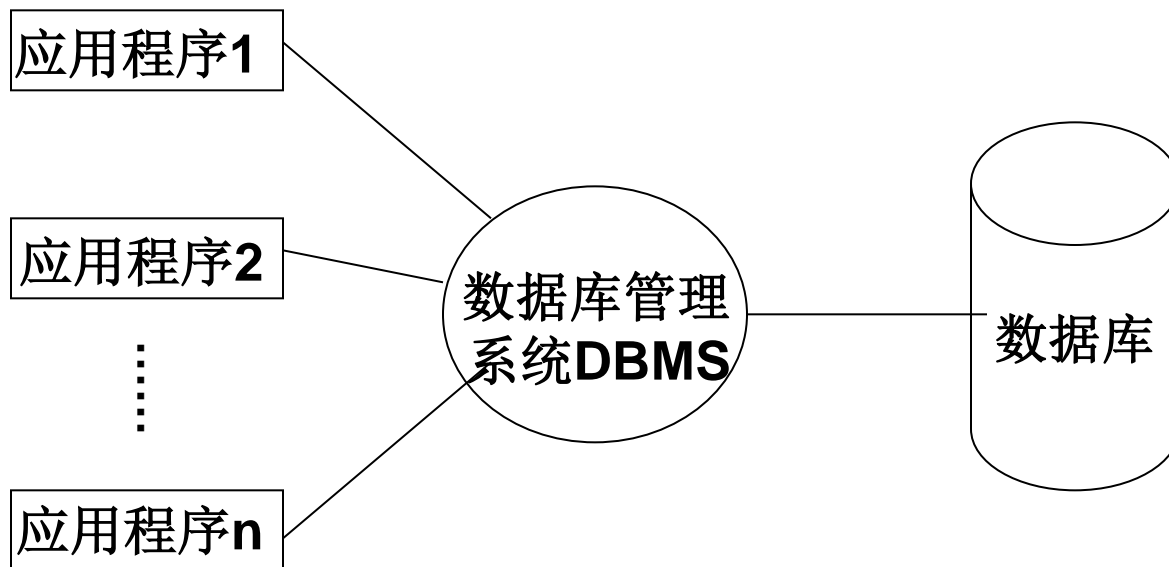


图1-4 程序与数据之间的关系(数据库系统阶段)

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/信息管理	大规模管理
	硬件水平	无直接存储设备	磁盘/磁鼓	大容量磁盘
	软件水平	没有操作系统和管理数据软件	高级语言和操作系统（有文件系统）	有数据库管理系统
	处理方式	批处理	联机实时处理、批处理	联机实时处理，分批处理，批处理
特点	数据的管理者	应用程序，数据不保存	文件系统，数据可长期保存	数据库管理系统
	数据面向的对象	某一应用程序	某一应用程序	现实世界

数据管理各阶段的比较

特点	数据的共享程度	无共享、冗余度极大	共享性差，冗余度大	共享性好
	数据的独立性	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
	数据的结构化	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
	数据控制能力	应用程序自己控制	应用程序自己控制	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
	用户接口	无	无	数据库系统为用户提供方便的用户接口



第1章 数据库系统概述

1.1 数据管理技术的发展

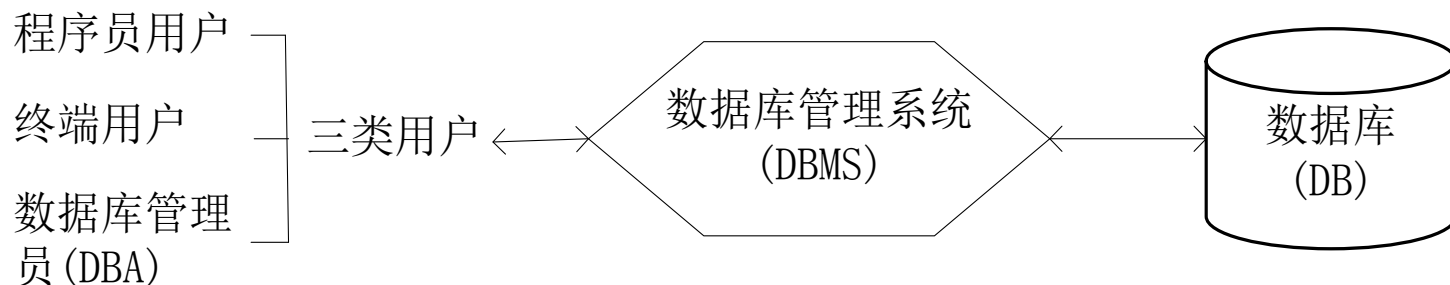
1.2 数据库系统

1.3 数据模型与数据模式

1.4 数据库管理系统 (DBMS)

1.2 数据库系统

数据库系统通常是指数据库和相应的软硬件系统。主要有数据库、数据库管理系统以及各类数据库用户三大部分，系统的组成结构如图所示





数据库系统的组成

- **数据库**是一个**统一管理**的**持久存储**的数据集合，这些数据应用于某个应用系统中
 - 描述了现实世界的某些方面
 - 数据间联系密切，具有最小的冗余度和较高的独立性
 - 服务于某个特定的应用
 - 可被多用户共享使用



数据库系统的组成

■ 数据库系统用户

- 数据库管理员（**DataBase Administrator, DBA**）：全面负责数据库系统的“管理、维护和正常使用”的人员
- 终端用户：指最终用户。通过应用系统的用户接口使用数据库。常用的接口方式有浏览器、菜单驱动、表格操作、图形显示、报表书写等，给用户提供了简明直观的数据表示
- 应用程序员：负责设计和编写应用系统的程序模块，并进行调试和安装



第1章 数据库系统概述

1.1 数据管理技术的发展

1.2 数据库系统

1.3 数据模型与数据模式

1.4 数据库管理系统 (DBMS)



1.3 数据模型与数据模式

- 数据如何描述、如何组织、如何存放在数据库中，这些都涉及到数据库系统的核心概念：**数据模型（Data Model）与数据模式（Data Schema）**



数据和信息

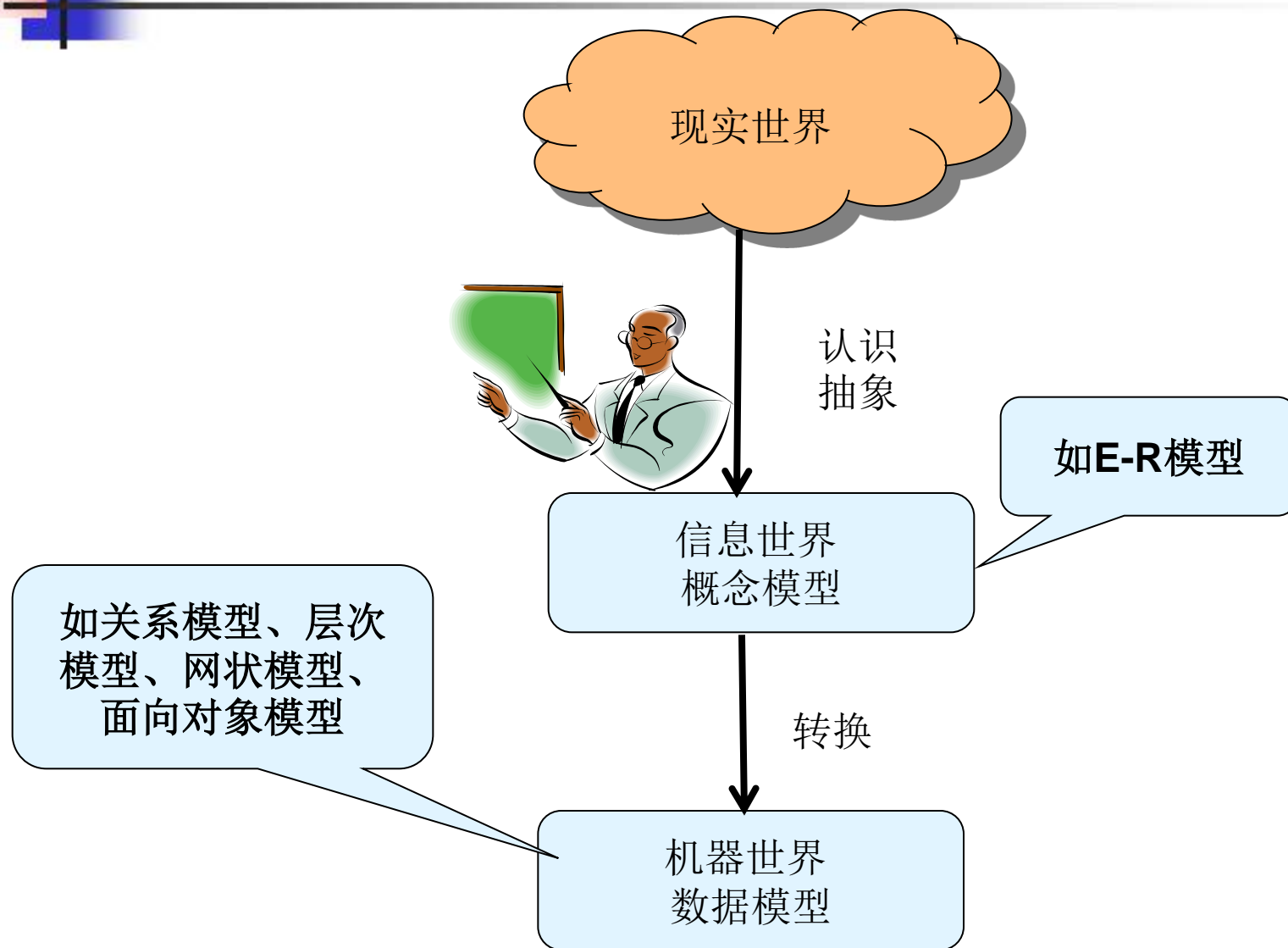
- 信息与数据之间存在着固有的联系，**数据**是信息的符号表示或称为载体，信息则是数据的内涵，是对数据语义的解释
- 对于任何数据而言，都有语法、语义两个方面要引起注意
- 语法（**Syntax**）指的是数据的格式规定，如规定数据是数值型、字符型、日期型
- 语义（**Semantic**）指的是数据本身的含义
- 例：
 - <张明， 2216111111 ， 男， 2021 级， 电信学部， 陕西， 2003.02 >



数据与数据模型

- 使用数据库技术，首先必须把现实世界中的事物表示为计算机能够处理的数据
- 模型是对现实世界特征的抽象
- 数据模型是对现实世界数据特征的抽象
- 从事物的特性到计算机中的具体表示，实际上经历了三个阶段——概念设计中的数据描述、逻辑设计中的数据描述和物理设计中的数据描述

数据抽象的层次





数据抽象的层次

- 现实世界：存在于人们头脑之外的客观世界
- 信息世界：是现实世界在人们头脑中的反映，人们可以通过文字符号形式将其记载下来。
- 机器世界：信息世界的信息在机器世界以二进制数据形式存储。



数据模型三要素

- **数据模型**（**Data Model**）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
- 数据模型应包含**数据结构**、**数据操作**和**数据完整性约束**三个要素：
 - **数据结构**：是指对实体类型和实体间联系的表达和实现
 - **数据操作**：是指对数据库的检索和更新（包括插入、删除和修改）两类操作
 - **数据完整性约束**：给出数据及其联系应具有制约和依赖规则

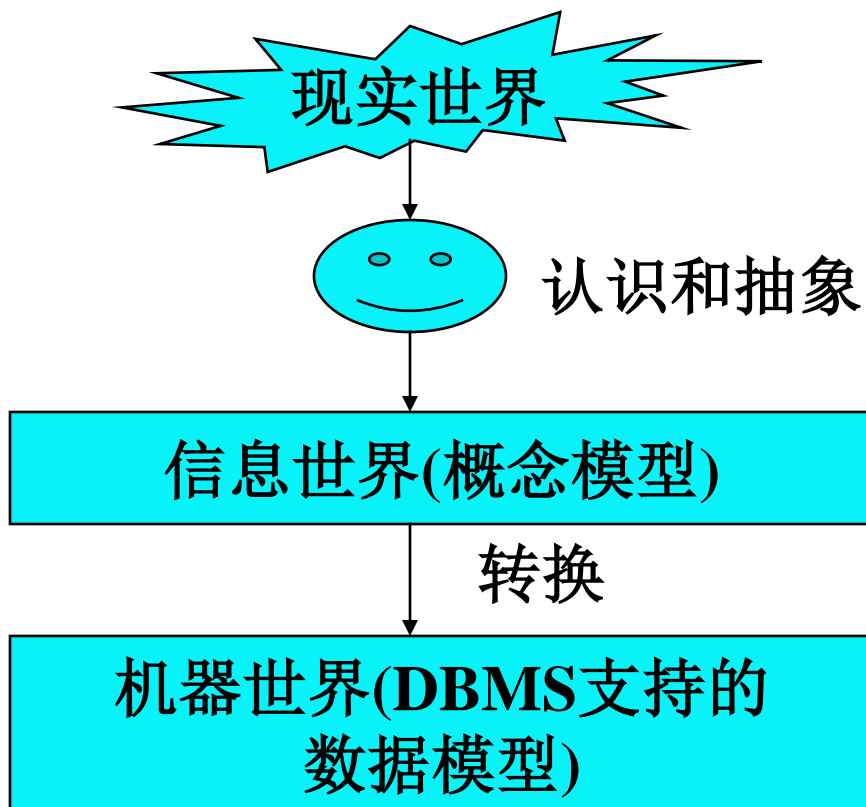


数据模型的级别

- 针对不同的使用对象和应用目的，采用不同层次的数据模型，一般可分为下面三个级别：
 - 概念数据模型(**conceptual data model**): 面向用户、面向现实世界的数据库模型，与特定的**DBMS**无关
 - 逻辑数据模型(**logical data model**): 用户从数据库系统所看到的数据库模型，与**DBMS**相关
 - 物理数据模型(**physical data model**): 反映数据存储结构的数据库模型

概念模型

概念模型是将数据从现实世界向抽象世界转换的第一步模型





数据模型的例子

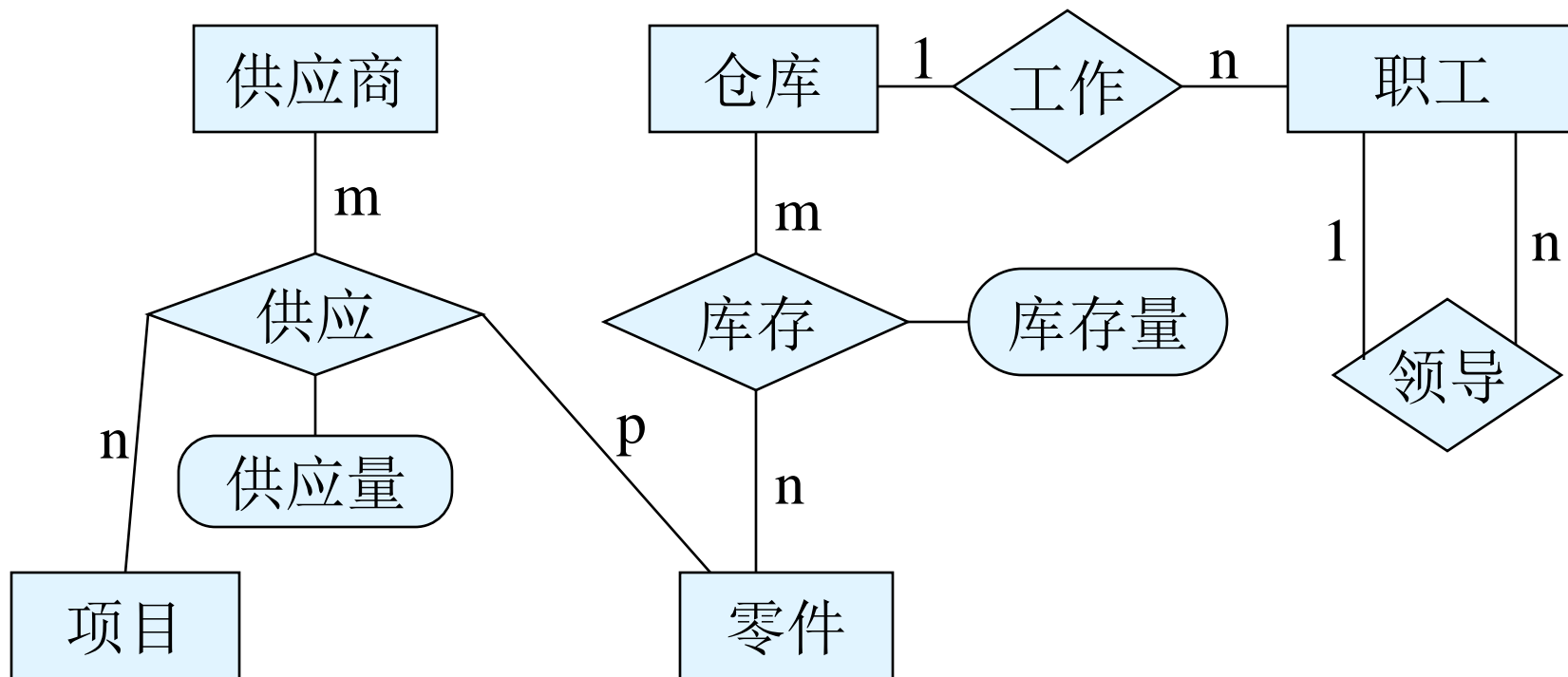
- 概念数据模型的例子：
 - 实体—联系(ER)模型
 - 扩展的实体—联系(EER)模型
 - 语义数据模型
- 逻辑数据模型的例子：
 - 层次模型
 - 网状模型
 - 关系模型
 - 面向对象的数据模型
 - 对象关系的数据模型



实体—联系 (ER) 模型

- 实体—联系模型(entity-relationship model)用以下三个概念描述现实世界：
 - 实体：实体是客观世界中可区别的客观对象或抽象概念
 - 属性：属性是实体特征的抽象，实体全部属性的集合称为实体型。当实体型全部属性的值确定之后，就得到实体型的一个实例
 - 实体间联系：实体间联系对应于客观世界中各种对象或抽象，概念之间的联系

实体—联系 (ER) 模型: 例



实体及其联系图



逻辑数据模型

- 经典逻辑数据模型有三类：
 - 层次模型(Hierarchical Model) (树)
 - 网状模型(Network Model) (图)
 - 关系模型(Relational Model) (表)
- 各种经典逻辑数据模型之间的根本区别在于数据之间的联系的表示方式不同

} 格式化模型

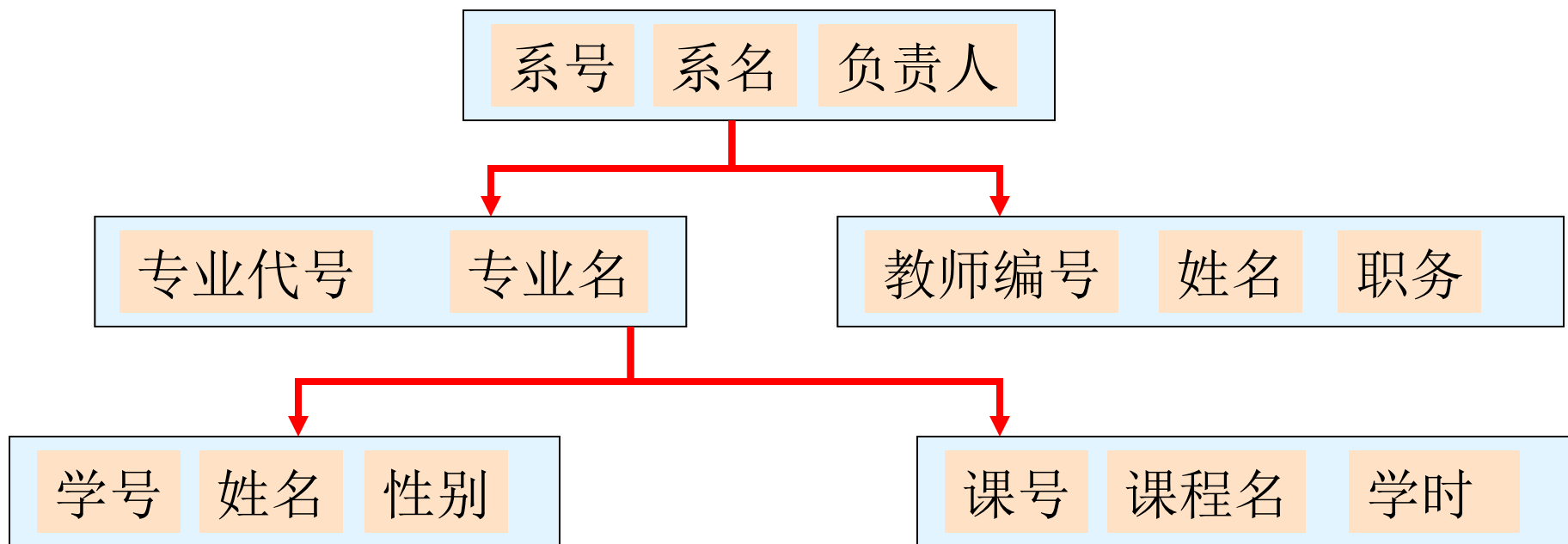


层次模型

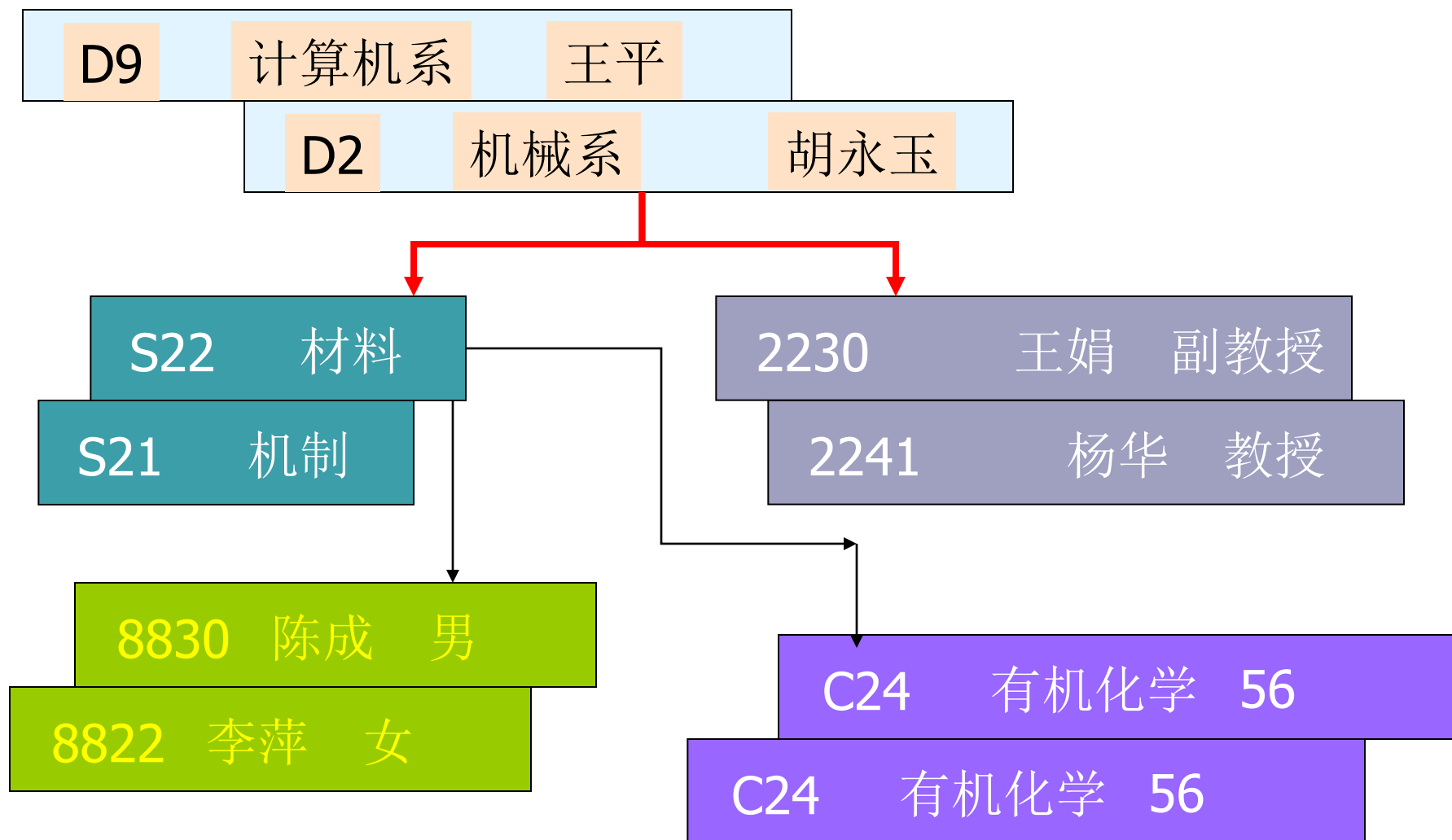
- 用树型（层次）结构表示实体类型及实体间联系的数据模型称为层次模型（**Hierarchical Model**）
- 与文件系统的数据管理方式相比，层次模型是一个飞跃，用户和设计者面对的是逻辑数据而不是物理数据，用户不必花费大量的精力考虑数据的物理细节
- 逻辑数据与物理数据之间的转换由**DBMS**完成
- 层次模型有两个缺点：
 - 一是只能表示**1:N**联系，虽然系统有多种辅助手段实现**M:N**联系但较复杂，用户不易掌握
 - 二是由于层次顺序的严格和复杂，引起数据的查询和更新操作很复杂，因此应用程序的编写也比较复杂

层次模型举例

- 在一个学校中，每个系分为若干个专业，而每个专业只属于一个系。系与教师，专业与学生，专业与课程之间也是一对多的联系



层次模型举例



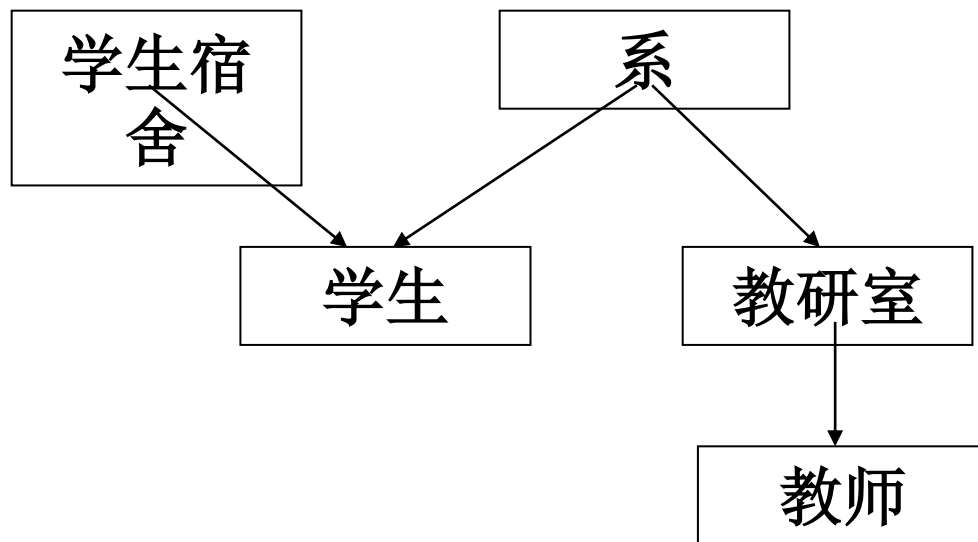


网状模型

- 用有向图结构表示实体类型及实体间联系的数据模型称为网状模型 (**network model**)
- 网状模型的特点是记录之间联系通过指针实现, **M:N**联系也容易实现 (一个**M:N**联系可拆成两个**1:N**联系), 查询效率较高
- 网状模型的缺点是数据结构和编程都比较复杂

网状模型举例

- **基本特点：**一个以上的结点无父结点；至少有一结点有多于一个的父结点。

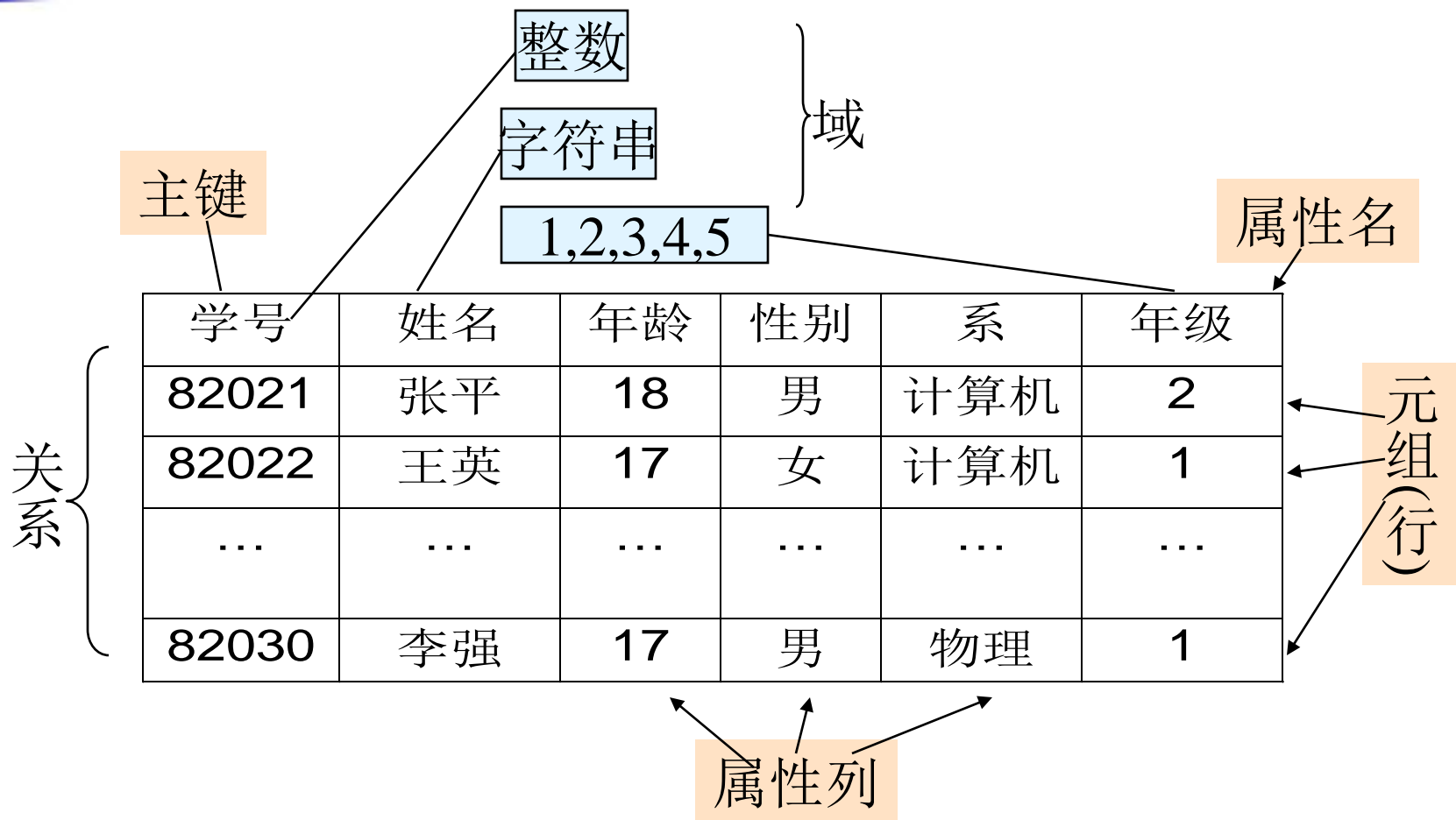




关系模型

- 关系模型（**Relational Model**）的主要特征是数据的逻辑结构是“二维表”
- 与前两种模型相比，数据结构简单，容易为初学者理解
- 关系模式相当于前面提到的记录类型，它的实例称为关系，每个关系实际上是一张二维表格

关系模型的直观表示



关系名：学生登记表

关系模式：学生（学号，姓名，年龄...）



面向对象模型

- 面向对象数据模型(**object-based data model**)的基本元素是对象(**Object**), 主要概念是对象类(**Class**)
- 对象类由一组变量和一组程序代码组成, 变量和程序代码分别定义对象类的特征(数据)和操作(方法)
- 当一个对象类的所有变量都赋予确定的值后, 就得到该对象类的一个对象实例(**Instance**)
- 允许一个对象的变量是另一个对象形成嵌套结构, 并且不限制对象的嵌套层次



四种逻辑数据模型的比较

	层次模型	网状模型	关系模型	面向对象模型
创始	1968年IBM公司的IMS系统	1969年CODASYL的DBTG报告(71年通过)	1970年F. Codd提出关系模型	20世纪80年代
数据结构	复杂 (树结构)	复杂 (有向图结构)	简单 (二维表)	复杂 (嵌套递归)
数据联系	通过指针	通过指针	通过表间的公共属性	通过对象标识
查询语言	过程性语言	过程性语言	非过程性语言	面向对象语言
典型产品	IMS	IDS/ II IMAGE/3000 IDMS TOTAL	Oracle Sybase DB2 SQL Server Informix	ONTOS DB
盛行期	20世纪70年代	70年代至80年代中期	80年代至现在	90年代至现在



数据库模式和实例

- **模式(Schema)**是数据库中**全体数据的逻辑结构和特征的描述**，它仅仅涉及类型的描述，不涉及具体的值
- 模式的一个具体值称为模式的一个**实例(Instance)**

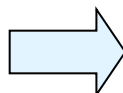


型和值

- 型 (**Type**) 是指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:
 - 类型—学生 (学号, 姓名, 年龄)
 - 值—— (2216111166, 张三, 22)

模式和实例举例

2个实例



1个模式



- 学生表 (学号, 姓名, 年龄)
- 课程表 (课程号, 课程名, 学分)
- 选课表 (学号, 课程号, 成绩)

实际中的模式描述
比本例要详细得多

S001	张三	21
S002	李四	20

C001	数据库	4
C002	英语	6
C003	数学	6

S001	C001	90
S002	C001	80

S001	张三	21
S002	李四	20
S003	王五	22

C001	数据库	4
C002	英语	6
C003	数学	6

S001	C001	90
S002	C001	80
S003	C001	90
S003	C002	96
S003	C003	98



模式的分级

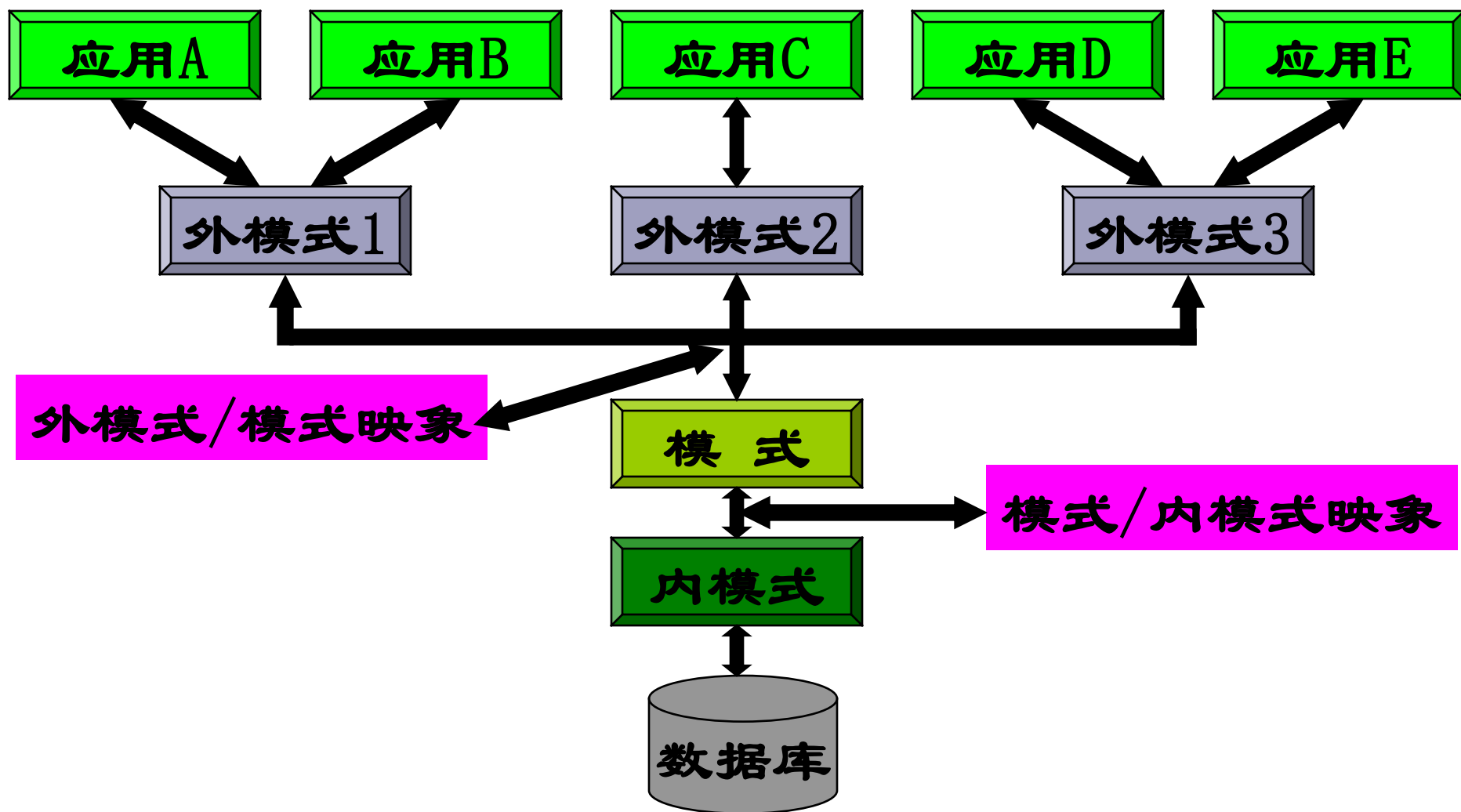
- 为了提高数据的物理独立性(physical data independence)和逻辑独立性(logical data independence)，使数据库的用户观点，即用户看到的数据库，与数据库的物理方面，即实际存储的数据库区分开来，数据库系统的模式是分级的



三级模式结构

- CODASYL（Conference On Data System Language,美国数据系统语言协商会）提出模式、外模式、内模式三级模式的概念。
- 三级模式之间有两级映象

三级模式结构





三级模式结构

■ 模式(Schema)

- 模式也称逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图，属于逻辑层抽象 (*Logical Level*)
- 模式是数据库系统模式结构的中间层，既不涉及数据的物理存储细节和硬件环境，也与具体的应用程序以及所使用的应用开发工具及高级程序设计语言无关
- 一个数据库只有一个模式。数据库模式以某一种数据模型为基础，统一综合地考虑了所有用户的需求，并将这些需求有机地结合成一个逻辑整体
- 定义模式时不仅要定义数据的逻辑结构，而且要定义数据之间的联系，定义与数据有关的安全性、完整性要求



三级模式结构

■ 外模式(Sub-Schema)

- **外模式**也称子模式（**Subschema**）或用户模式，属于视图层抽象（**View Level**），它是数据库用户（包括应用程序员和最终用户）能够看见和使用的**局部数据的逻辑结构和特征的描述**，是数据库用户的数据视图，是与某一应用有关的数据逻辑表示
- 外模式通常是**模式的子集**，一个数据库可以有多个外模式
- 由于外模式是各个用户的数据视图，如果不同的用户在不同应用需求、看待数据的方式、对数据保密的要求等方面存在差异，则其外模式描述就是不同的，即使对模式中同一数据，在外模式中的结构、类型、长度、保密级别等都可以不同。



三级模式结构

■ 内模式(**Storage Schema**)

- **内模式**也称存储模式（**Storage Schema**），属于物理层抽象（**Physical Level**）。
- 一个数据库只有一个内模式。它是**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
- 例如，记录的存储方式是顺序存储、按照**B**树结构存储还是按**hash**方法存储；索引按照什么方式组织；数据是否压缩存储，是否加密；数据的存储记录结构有何规定等

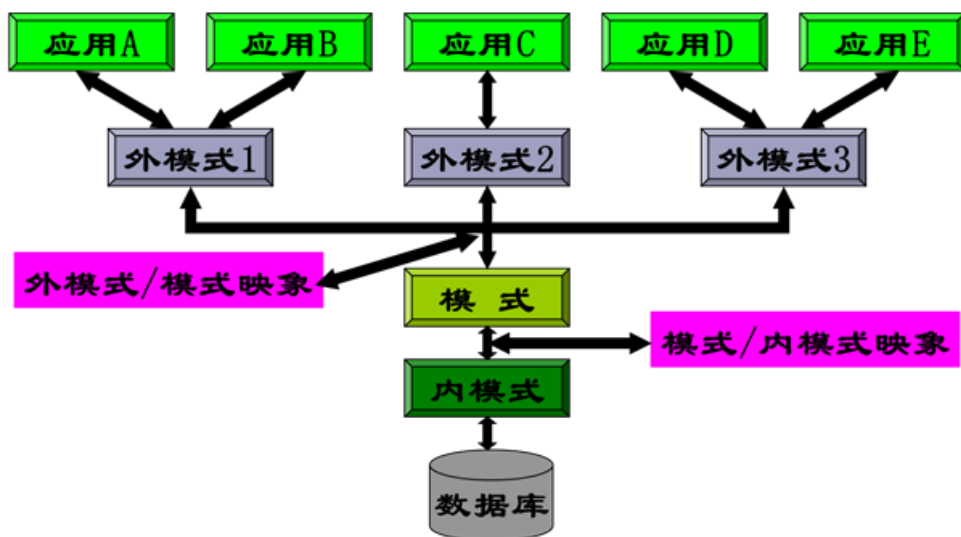
三级模式结构

■ 数据库系统在三级模式之间提供了两层映象：

➤ 外模式 / 模式映象

➤ 模式 / 内模式映象

■ 正是这两层映象保证了数据库系统中的数据能够具有较高的逻辑独立性和物理独立性





三级模式结构

■ 外模式/模式映象

- 对于每一个外模式，数据库系统都有一个外模式 / 模式映象，它定义了该外模式与模式之间的对应关系。
- 当模式改变时，对各个外模式 / 模式的映象作相应改变，可以使外模式保持不变。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

- 定义数据逻辑结构与存储结构之间的对应关系
- 存储结构改变时，修改模式/内模式映象，使模式保持不变，从而应用程序可以保持不变，称为数据的物理独立性



第1章 数据库系统概述

1.1 数据管理技术的发展

1.2 数据库系统

1.3 数据模型与数据模式

1.4 数据库管理系统 (DBMS)



1.4 数据库管理系统

- 数据库管理系统（**database management system, DBMS**）是指数据库系统中对数据进行管理的软件系统，是数据库系统的核心组成部分
- DBMS介于用户和操作系统之间，是一种通用的系统软件



DBMS的主要功能

■ 提供各类用户接口

- 程序员用户
- 终端用户
- 数据库管理员

■ 数据库的运行管理

- 安全性
- 完整性
- 并发控制
- 故障恢复
- 查询优化

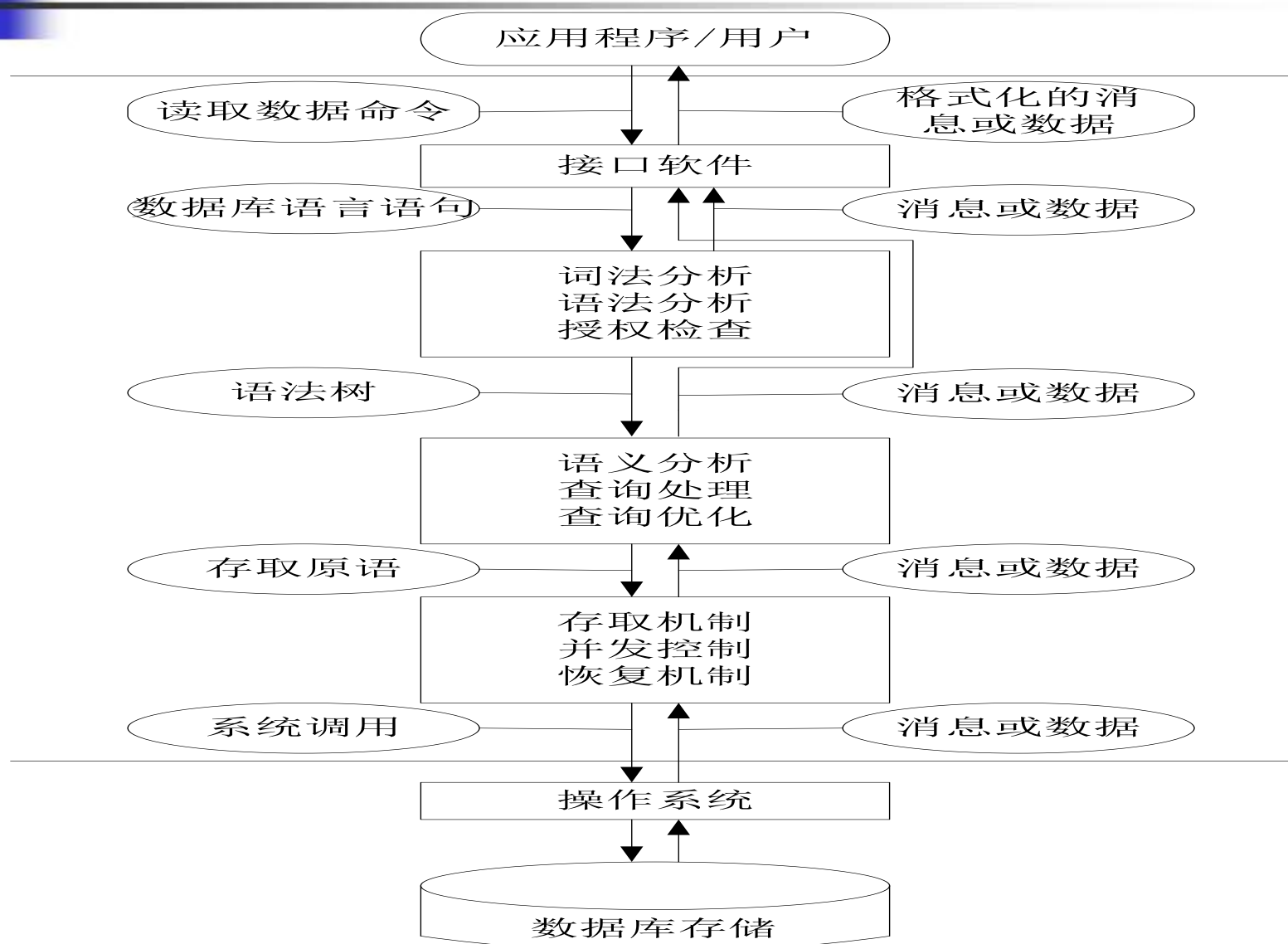
■ 数据目录管理



DBMS体系结构

- DBMS作为数据库系统的核心，所有访问数据库的请求都由DBMS来处理
- 考察用户访问数据的过程，可以简单了解DBMS的主要组成部分
- 与高级程序设计语言相似，作为数据库语言的一个实现的DBMS也有编译和解释两种实现方法
- 下图表示一个解释执行的DBMS的结构

解释执行的DBMS结构





DBMS运行过程

- 以上说明的是DBMS读取数据的过程，其它一些操作，如插入、删除、修改，其过程是类似的
- 对于插入、删除操作，先找到相应记录所在位置，然后执行插入、删除操作
- 对于修改操作，先要找到要修改的记录，进行完修改后，再把修改后的记录存储起来



DBMS运行过程

- 从DBMS执行的过程可以看出，解释方法具有灵活、应变性强的特点
- 能保持较高的数据独立性，对在解释过程中发生的数据结构、存储结构的变化能够适应
- 由于每次执行都要经过以上所有步骤，开销会很大，效率比较低。随着数据库技术的发展，这种方法已逐步被预编译方法所取代



DBMS的预编译方法

- 预编译方法的基本思想，在运行以前，对将要执行的数据库语句进行编译处理，保存生成的可执行代码，需要运行时，取出保存的可执行代码加以执行
- 当数据库中某些成分发生改变，使部分编译结果无效时，再对它们执行一次编译
- 实践证明，预编译方法的主要优点在于执行效率高，系统开销小



DBMS的进程结构

- **DBMS**是管理数据库的软件系统，必须要在操作系统的支持下才能工作
- 早期的计算机系统内存容量有限，**DBMS**常按功能划分为多个进程，但进程管理的开销较大
- 随着计算机内存越来越大，**DBMS**进程的划分不再受内存限制，更多考虑的是结构是否合理以及性能能否提高



线程的概念

- 现代操作系统中引入线程的概念，作为程序并发执行的单位，分享**CPU**时间
- 一个进程中可以创建多个可以切换的线程，为进程本身的并发执行提供了基础
- 线程是比进程更小的运行单位，称为轻量进程（**lightweight process**），而进程则称为重量进程（**heavyweight process**）



线程的优点

- ① 一个进程可划分为多个线程
- ② 线程的创建、切换、通信等开销远比进程少
- ③ 线程为进程本身的并发执行提供了手段



多线程DBMS

- 在单进程多线程结构的**DBMS**中，系统只创建一个**DBMS**进程。当有用户申请数据库服务时，**DBMS**分配至少一个线程为之服务，多个线程并行工作，共享资源
- 在支持多线程的系统中，线程管理功能可以由操作系统提供，也可以由**DBMS**进程扩充，最好由**DBMS**进程管理线程
- 因为线程作为**DBMS**的调度单位，可以按照**DBMS**的需要按一定的调度算法调度用户请求，由于调度算法是由**DBMS**设计与执行，因此会比操作系统直接对这些请求进行调度高效得多，提高了**DBMS**的性能和可移植性



多线程DBMS

- 由**DBMS**进程管理线程增加了**DBMS**的复杂性，还对操作系统提出以下两方面的新要求：
 - ① 提供非阻塞 I/O(nonblocking I/O)和异步 I/O(asynchronous I/O)功能
 - ② 支持“公平”调度(fair schedule)



数据目录

- **DBMS**的功能是管理大量的、共享的、持久的数据，有关这些数据的定义和描述（即元数据）也需要长期保存在系统中
- **数据目录(Catalog或Directory)**是一组关于数据的数据，也称为元数据(meta-data)
- 关系数据库中数据目录的组织通常与数据本身的组织相同
- 数据目录按不同的内容在逻辑上组织为若干张表
- 数据目录的特点是数据量比较小（与数据本身比）、使用频繁，因为任何数据库操作都要参照数据目录的内容



第1章 小结

■ 数据管理技术的发展

- 数据管理三个发展阶段

■ 数据库系统

- 数据库系统组成

■ 数据模型与数据模式

- 数据库基本概念
- 数据模型三要素
- 三种主要数据库模型
- 四种逻辑数据模型
- 数据库系统三级模式结构
- 数据库系统两层映像系统结构



第1章 小结（续）

■ 数据库管理系统

- 数据库管理系统体系结构
- 数据库管理系统数据目录