

# 第1章 绪论

# 本章目标

理解并掌握四个基本名称的概念

数据

数据库

数据库管理系统

数据库系统

理解并掌握数据库的三级模式结构

理解并掌握数据库独立性与三级模式结构的关系

逻辑独立性

物理独立性

理解数据库系统的人员角色

理解并掌握关系数据库建模工具E-R图

理解并掌握数据模型的定义及其三个组成要素

数据结构

数据操纵

完整性约束

# 大纲

- **数据库系统概述**
- 数据模型
- 数据库系统结构
- 数据库系统的组成
- 本章小结

# 数据库系统概述

- **四个基本名词**
  - 数据 (Data)
  - 数据库 (Database)
  - 数据库管理系统 (DBMS)
  - 数据库系统 (DBS)
- **数据管理技术的产生与发展**
  - 人工管理阶段、文件系统阶段和数据库阶段
- **数据库系统的特点**

# 数据

## ■ 数据(Data)

- 描述事物的符号记录，是数据库中存储的基本对象
- 种类有数字、文字、图形、图像、音频、视频、学生的档案记录等
- 数据格式在不同的模型中表现不同：概念模型、物理模型
- 特点：与其语义不可分
  - 如93可代表的含义：某门课程的成绩、体重、班级人数...
  - (张强, 男, 20, 山东, 信息学院, 软件工程专业, 2020级)
- 语义：数据的含义

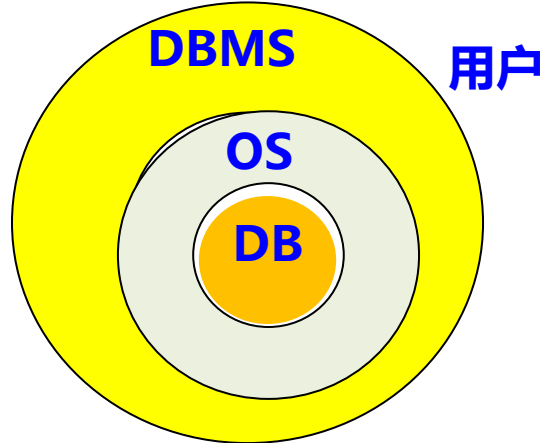
# 数据库

## ■ 数据库(Database, DB)

- 是指长期储存在计算机内的、有组织的、可共享的大量数据的集合
- 数据按一定的数据模型(Data model)组织、描述和储存
- 为各种用户共享
- 冗余度较小
- 数据独立性较高
- 易扩展性

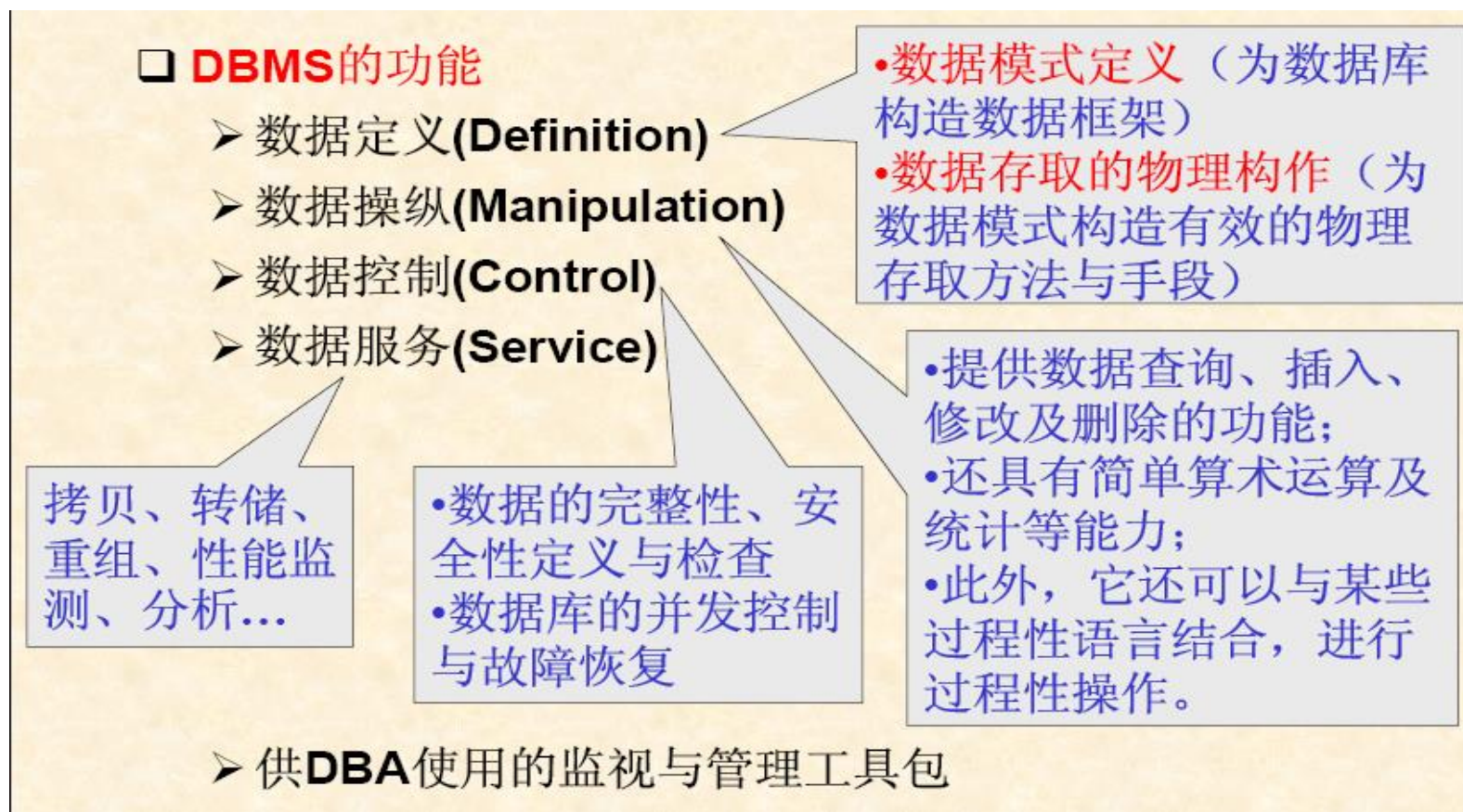
# 数据库管理系统

- 数据库管理系统(Database Management System, 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
  - 为用户或应用程序提供访问DB的方法，包括DB的建立、查询、更新及各种数据控制
  - 总是基于某种数据模型，如层次型、网状型、关系型和面向对象型等



# 数据库管理系统(cont'd)

## ■ DBMS是数据库的应用程序与数据库的接口





## ❑ 数据子语言 (data sub\_language)

### ➤ 数据定义语言

– **Data Definition Language**, 简称**DDL**, 负责数据的模式定义与数据的物理存取构造

### ➤ 数据操纵语言

– **Data Manipulation Language**, 简称**DML**, 负责数据的操纵, 包括查询及增、删、改等操作

### ➤ 数据控制语言

– **Data Control Language**, 简称**DCL**, 负责数据的完整性、安全性的定义与检查以及并发控制、故障恢复等功能

SQL 语言

## □ 数据子语言的结构形式

### ➤ 交互式命令语言

- 能在终端上即席操作，又称为自含型或自主型语言

### ➤ 宿主型语言

- 需要嵌入到某种宿主语言中

一种高级过程性程序设计语言，如FORTRAN, C, C<sup>++</sup>, COBOL等

# • 工业界常见DBMS产品

410 systems in ranking, February 2023

Rank			DBMS	Database Model	Score		
Feb 2023	Jan 2023	Feb 2022			Feb 2023	Jan 2023	Feb 2022
1.	1.	1.	Oracle +	Relational, Multi-model i	1247.52	+2.35	-9.31
2.	2.	2.	MySQL +	Relational, Multi-model i	1195.45	-16.51	-19.23
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model i	929.09	+9.70	-19.96
4.	4.	4.	PostgreSQL +	Relational, Multi-model i	616.50	+1.65	+7.12
5.	5.	5.	MongoDB +	Document, Multi-model i	452.77	-2.42	-35.88
6.	6.	6.	Redis +	Key-value, Multi-model i	173.83	-3.72	-1.96
7.	7.	7.	IBM Db2	Relational, Multi-model i	142.97	-0.60	-19.91
8.	8.	8.	Elasticsearch	Search engine, Multi-model i	138.60	-2.56	-23.70
9.	↑ 10.	↑ 10.	SQLite +	Relational	132.67	+1.17	+4.30
10.	↓ 9.	↓ 9.	Microsoft Access	Relational	131.03	-2.33	-0.23

来源: <https://db-engines.com/en/ranking>

- 该排名表明数据库产品的流程度。排名前10的DBMS中关系型模型占比70%，且前4位全部为关系型DBMS，表明关系型DBMS仍然是当前数据库管理技术的主流。

# • 墨天轮中国数据库流行度排行

最新解读报告：2023年2月中国数据库排行榜：OTO新格局持续三月，人大金仓、AnalyticDB排名创新高

2023年2月共259个数据库参与，点击查看排名规则更新

排行	上月	半年前	名称	模型 ∨	属性 ∨	三方评测 ∨	生态 ∨	专利	论文	得分
	1	↑ ↑ ↑ 4	OceanBase +	关系型	HP X Q			151	18	645.03
	2	↓ 1	TiDB +	关系型	HP X Q			26	44	634.73
	3	↓ 2	openGauss +	关系型	TP X Q			562	65	557.11
4	4	↓ 3	达梦 +	关系型	TP X Q			381	0	530.65
5	↑ ↑ 7	↑ ↑ 7	人大金仓 +	关系型	TP X Q			232	0	410.05
6	↓ 5	6	PolarDB +	关系型	X Q HP			512	26	374.34
7	↓ 6	↓ ↓ 5	GaussDB +	关系型	X Q HP			562	65	356.47
8	8	↑ 9	TDSQL +	关系型	X Q HP			39	10	281.99
9	↑ 10	↑ 10	AnalyticDB +	关系型	AP Q X			480	28	208.71
10	↓ 9	↓ ↓ 8	GBase +	关系型	TP AP Q X			152	0	206.65

来源：<https://www.modb.pro/dbRank>



# 经典DBMS结构

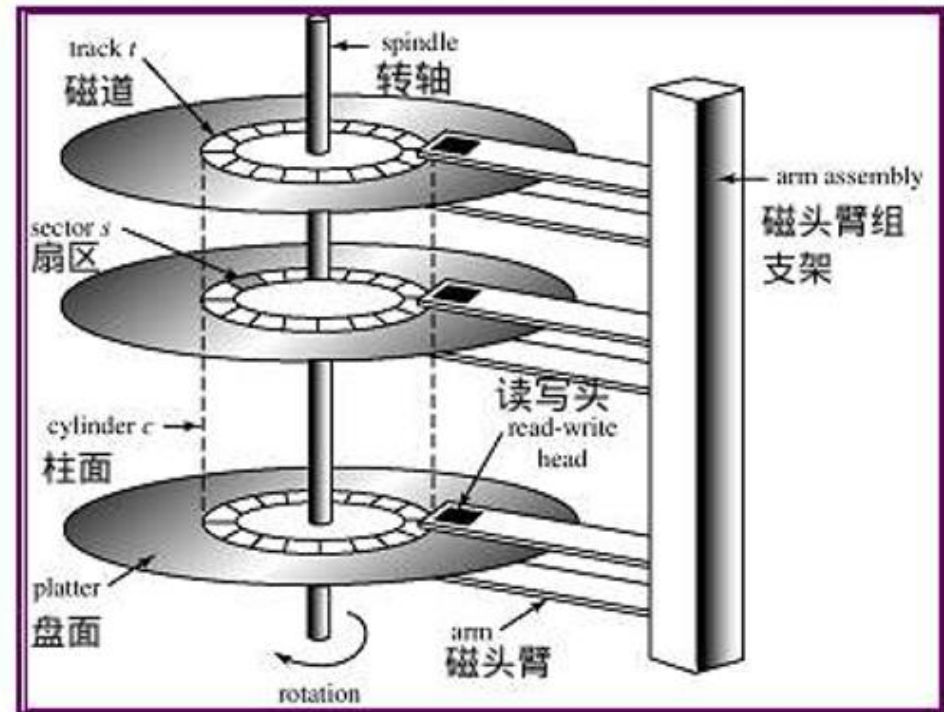
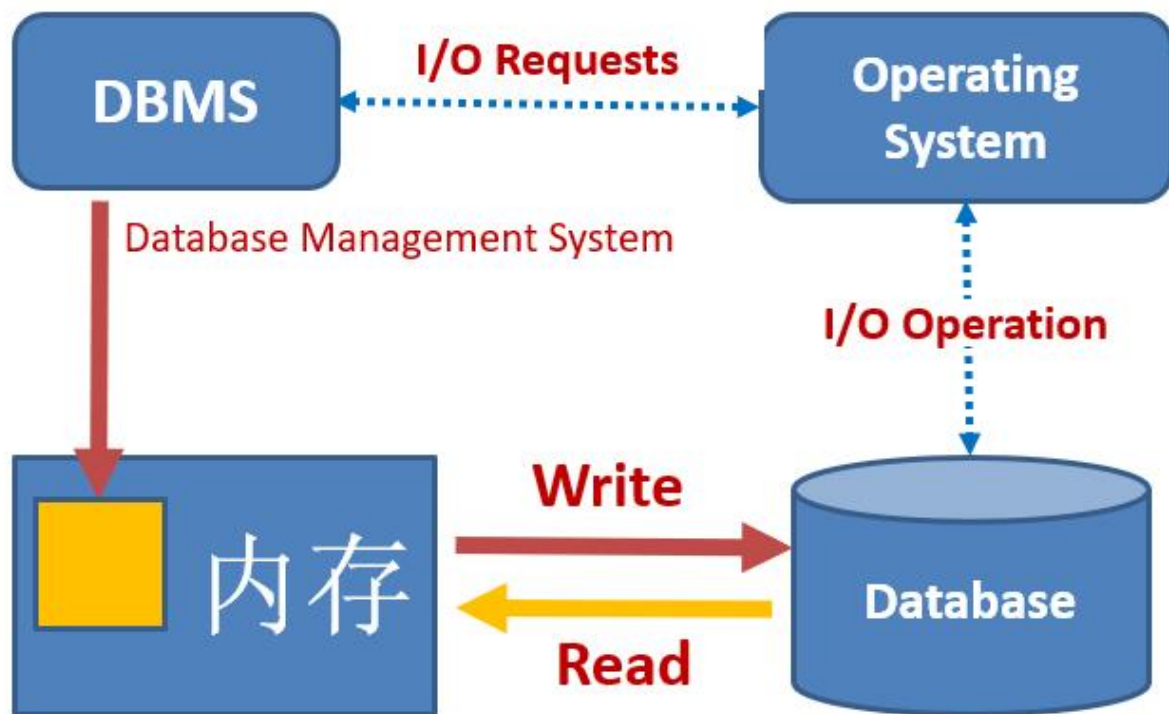
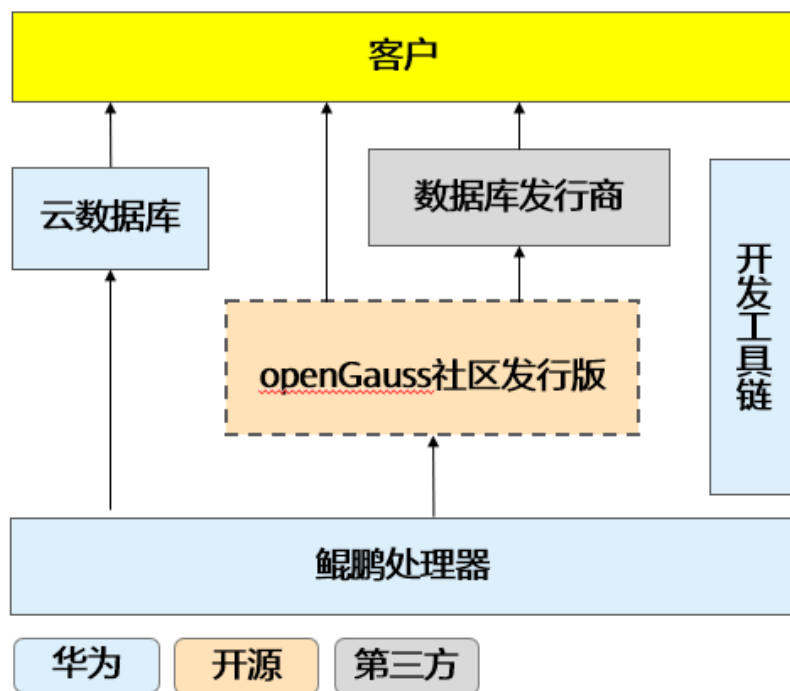


Fig. Conventional Architecture of DBMS

# 华为openGauss简介

- 华为openGauss是一款**高性能、高安全、高可靠**的企业级开源关系型数据库。
- openGauss集中式版本内核全开源
  - PG开源数据库 -> 内部自用孵化阶段 -> 联创产品化阶段 -> 共建生态



- openGauss的更详细内容请参见  
华为公司提供的材料：  
01.openGauss概述.pptx

# openGauss运行环境

- 支持的硬件平台
  - openGauss支持运行在鲲鹏服务器和通用的x86服务器上：
    - 支持鲲鹏服务器和基于x86\_64的通用PC服务器。
    - 支持本地存储（SATA、SAS、SSD）。
    - 支持千兆、万兆Ethernet网络。
- 支持的操作系统
  - openEuler release 20.03 (LTS) on ARM。推荐采用此操作系统。
  - openEuler 20.03 on X86-64。
  - CentOS 7.6 on X86-64。

# openGauss技术指标

- openGauss的技术指标如表所示。

技术指标	最大值	技术指标	最大值
数据库容量	受限于操作系统与硬件	复合索引包含列数	32
单表大小	32 TB	单表约束个数	无限制
单行数据大小	1 GB	并发连接数	10000
每条记录单个字段的大小	1 GB	分区表的分区个数	32768
单表记录数	$2^{48}$	分区表的单个分区大小	32 TB
单表列数	250~1600（随字段类型不同会有变化）	分区表的单个分区记录数	$2^{55}$
单表中的索引个数	无限制		



# openGauss基本功能

- 标准SQL支持

- 支持标准的SQL92/SQL99/SQL2003/SQL2011规范，支持GBK和UTF-8字符集，支持SQL标准函数与分析函数，支持存储过程

- 数据库存储管理功能

- 支持表空间，可以把不同表规划到不同的存储位置

- 提供主备双机

- 事务支持ACID特性、单节点故障恢复、双机数据同步，双机故障切换等

- 应用程序接口

- 支持标准JDBC 4.0的特性、ODBC 3.5特性

- 管理工具

- 提供安装部署工具、实例启停工具、备份恢复工具

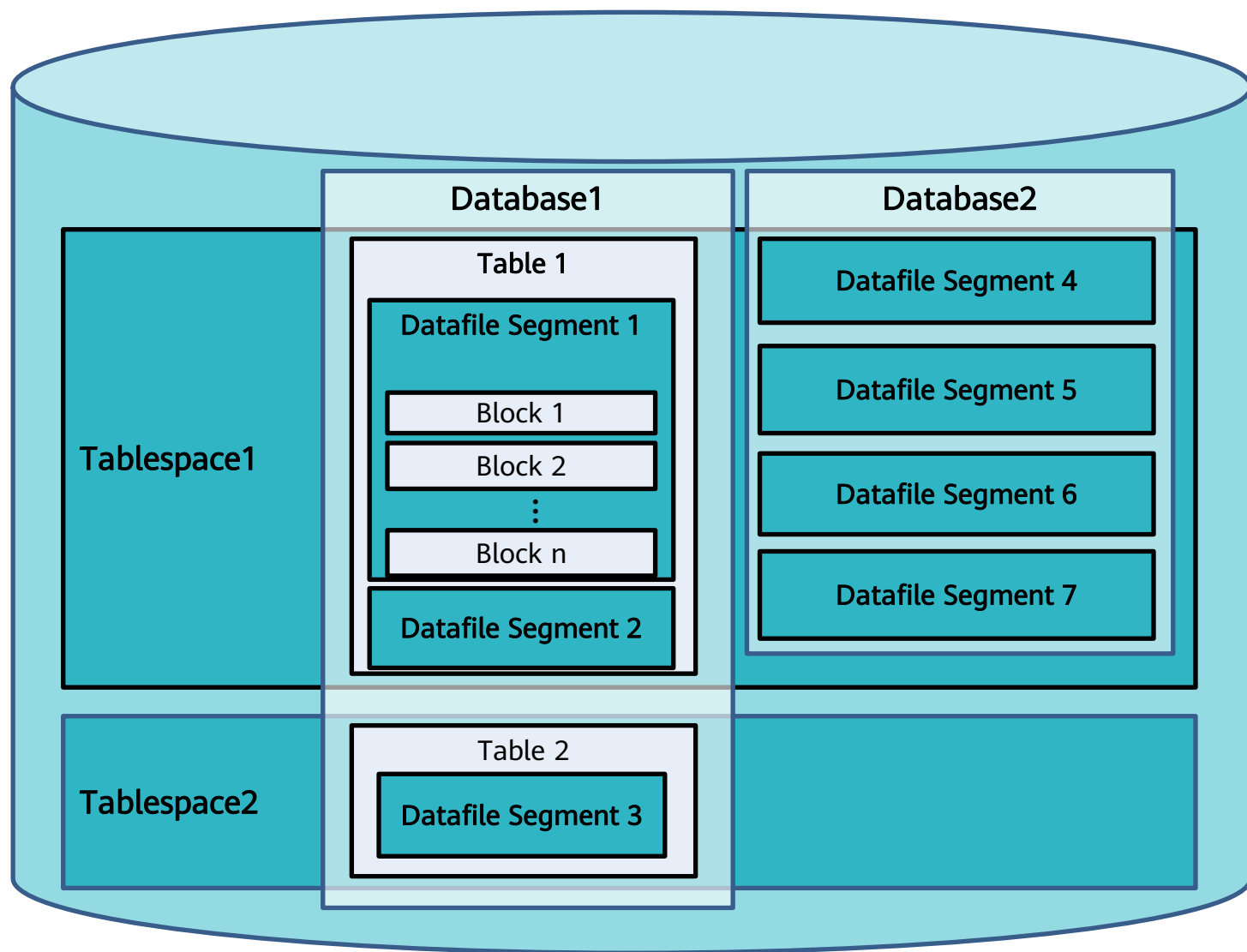
- 安全管理

- 支持SSL安全网络连接、用户权限管理、密码管理、安全审计等功能，保证数据库在管理层、应用层、系统层和网络层的安全性

# openGauss体系架构



# openGauss逻辑架构

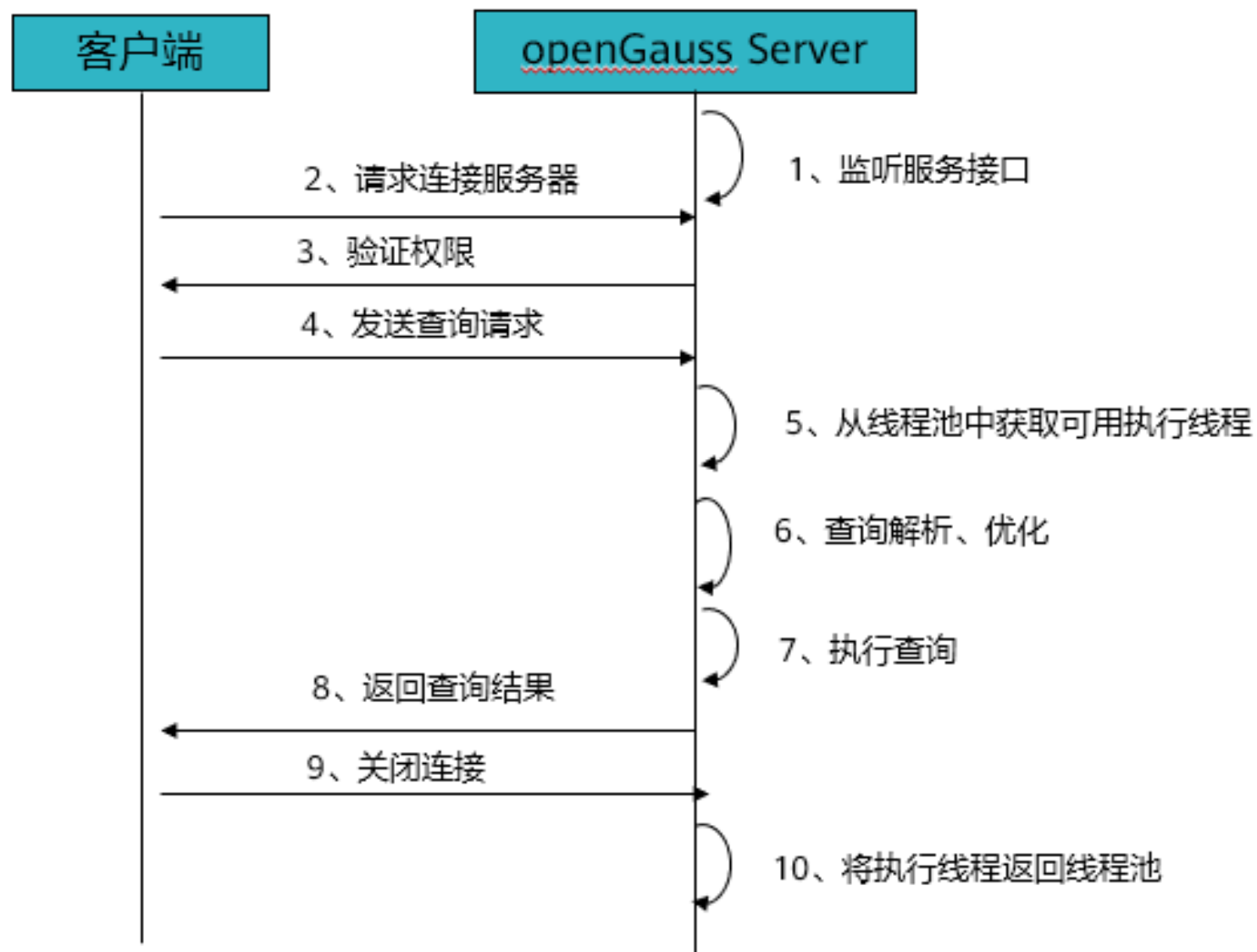


- **Tablespace:** 表空间，是一个目录，可以存在多个，里面存储的是它所包含的数据库的各种物理文件。每个表空间可以对应多个 Database。
- **Database:** 数据库，用于管理各类数据对象，各数据库间相互隔离。数据库管理的对象可分布在多个 Tablespace 上。
- **Datafile Segment:** 数据文件，通常每张表只对应一个数据文件。如果某张表的数据大于 1GB，则会分为多个数据文件存储。
- **Table:** 表，每张表只能属于一个数据库，也只能对应到一个 Tablespace。每张表对应的数据文件必须在同一个 Tablespace 中。
- **Block:** 数据块，是数据库管理的基本单位，默认大小为 8KB。

# openGauss物理架构

- 数据库的文件默认保存在initdb时创建的数据目录中
- 在数据目录中有很多类型、功能不同的目录和文件
  - 除了数据文件之外，还有参数文件、控制文件、数据库运行日志及预写日志等。

# openGauss服务响应流程



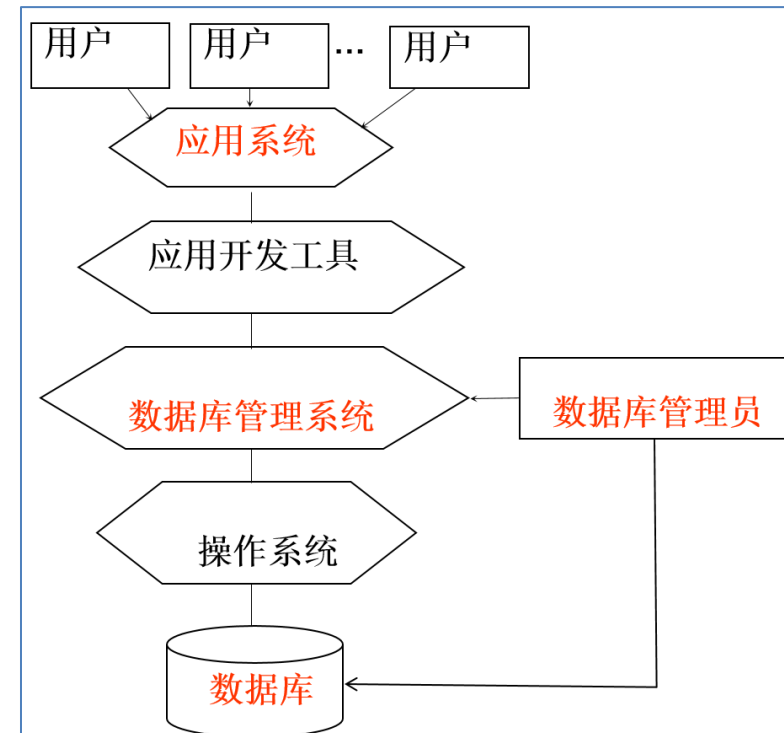
# 数据库系统

## ■ 数据库系统 (Database System, DBS)

– DBS是实现有组织地、动态地存储大量关联数据、方便多用户访问的计算机**硬件、软件和数据资源**组成的系统，即它是采用数据库技术的计算机系统。

– 构成：

- 数据库
- 数据库管理系统（及其开发工具）
- 应用系统
- 数据库管理员(Database Administrator, DBA)
- 用户



# 数据库系统概述

- 四个基本名词
  - 数据 (Data)
  - 数据库 (Database)
  - 数据库管理系统 (DBMS)
  - 数据库系统 (DBS)
- 数据管理技术的产生与发展
  - 人工管理阶段、文件系统阶段和数据库阶段
- 数据库系统的特点

# 数据管理技术的产生与发展

- 数据处理

- 指对各种数据进行收集、存储、加工和传播的一系列活动的总和

- 数据管理

- 指对数据进行分类、组织、编码、存储、检索和维护等操作
  - 是数据处理的中心问题

- 数据处理与数据管理的关系

- 数据管理技术的优劣，将直接影响数据处理的效率



# 数据管理技术的三个发展阶段

		人工管理阶段	文件系统阶段	数据库系统阶段
背景	应用背景	科学计算	科学计算、管理	大规模管理
	硬件背景	无直接存取存储设备	磁盘、磁鼓	大容量磁备盘
	软件背景	没有操作系统	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、分布处理、批处理
特点	数据的管理者	用户（程序员）	文件系统	数据库管理系统
	数据面向的对象	某一应用程序	某一应用	现实世界
	数据的共享程度	无共享，冗余度极大	共享性差，冗余度大	共享性高，冗余度小
	数据的独立性	不独立，完全依赖于程序	独立性差	具有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	记录内有结构，整体无结构	整体结构化，用数据模型描述
	数据控制能力	应用程序自己控制	应用程序自己控制	由数据库管理系统提供数据安全性、完整性、并发控制和恢复能力

• **人工管理**(20世纪40年代中--50年代中)

• **文件系统**(20世纪50年代末--60年代中)

• **数据库系统**(20世纪60年代末--现在)

# 数据库系统概述

- 四个基本名词
  - 数据 (Data)
  - 数据库 (Database)
  - 数据库管理系统 (DBMS)
  - 数据库系统 (DBS)
- 数据管理技术的产生与发展
  - 人工管理阶段、文件系统阶段和数据库阶段
- 数据库系统的特点

# 数据库系统的特点

- 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

# 数据结构化

- 整体数据的结构化是数据库的主要特征之一，是与文件系统的本质区别
- 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

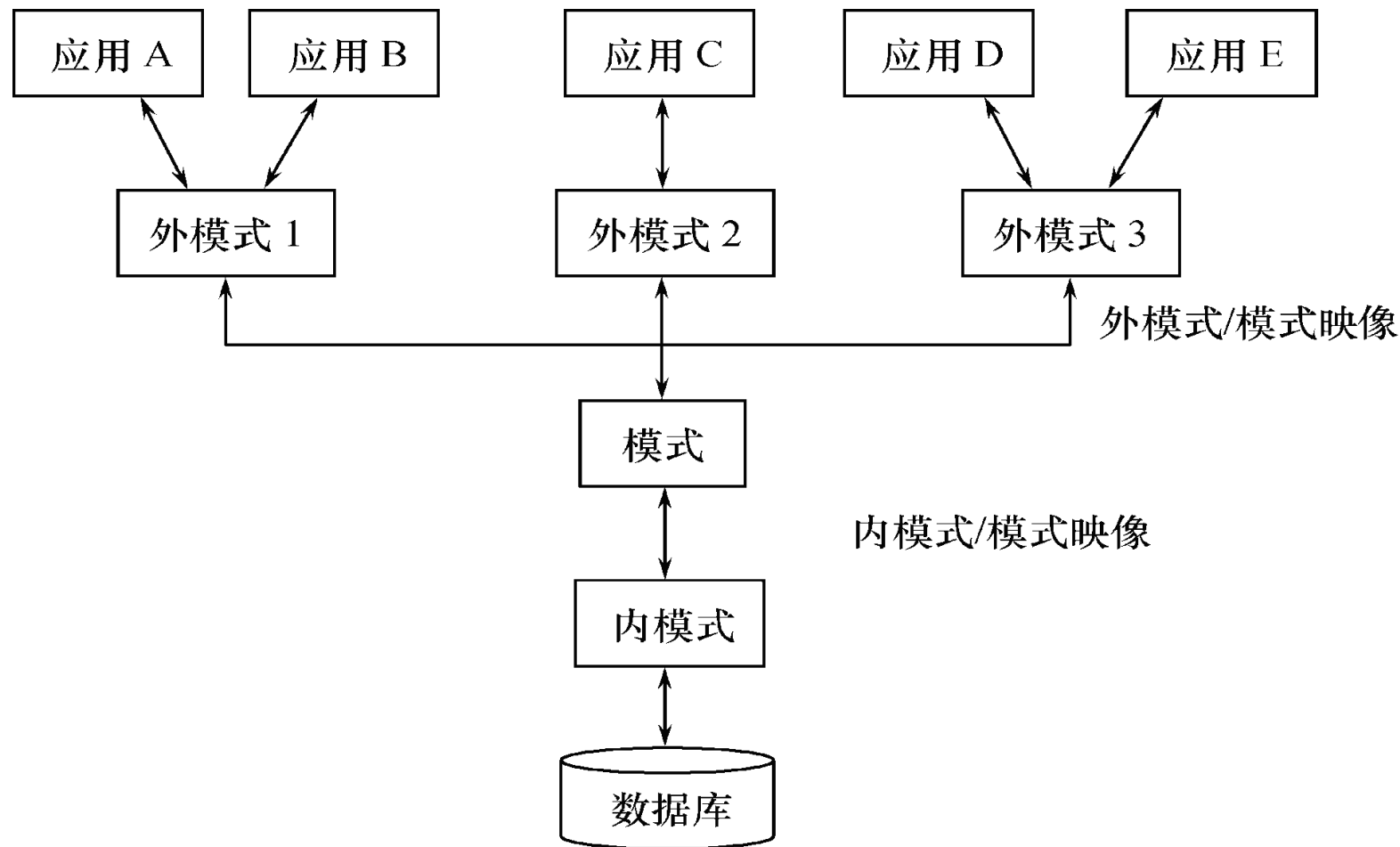
# 数据的共享性高，冗余度低，易扩充

- 数据面向整个系统，因此数据可被多个用户、多个应用共享使用
- 减少了数据冗余，节约存储空间
- 避免数据间的不相容性与不一致性



指同一数据不同拷贝的值不一样

# 数据独立性高



- **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

**三级模式+二级映象**

# 数据由DBMS统一管理 and 控制

- 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

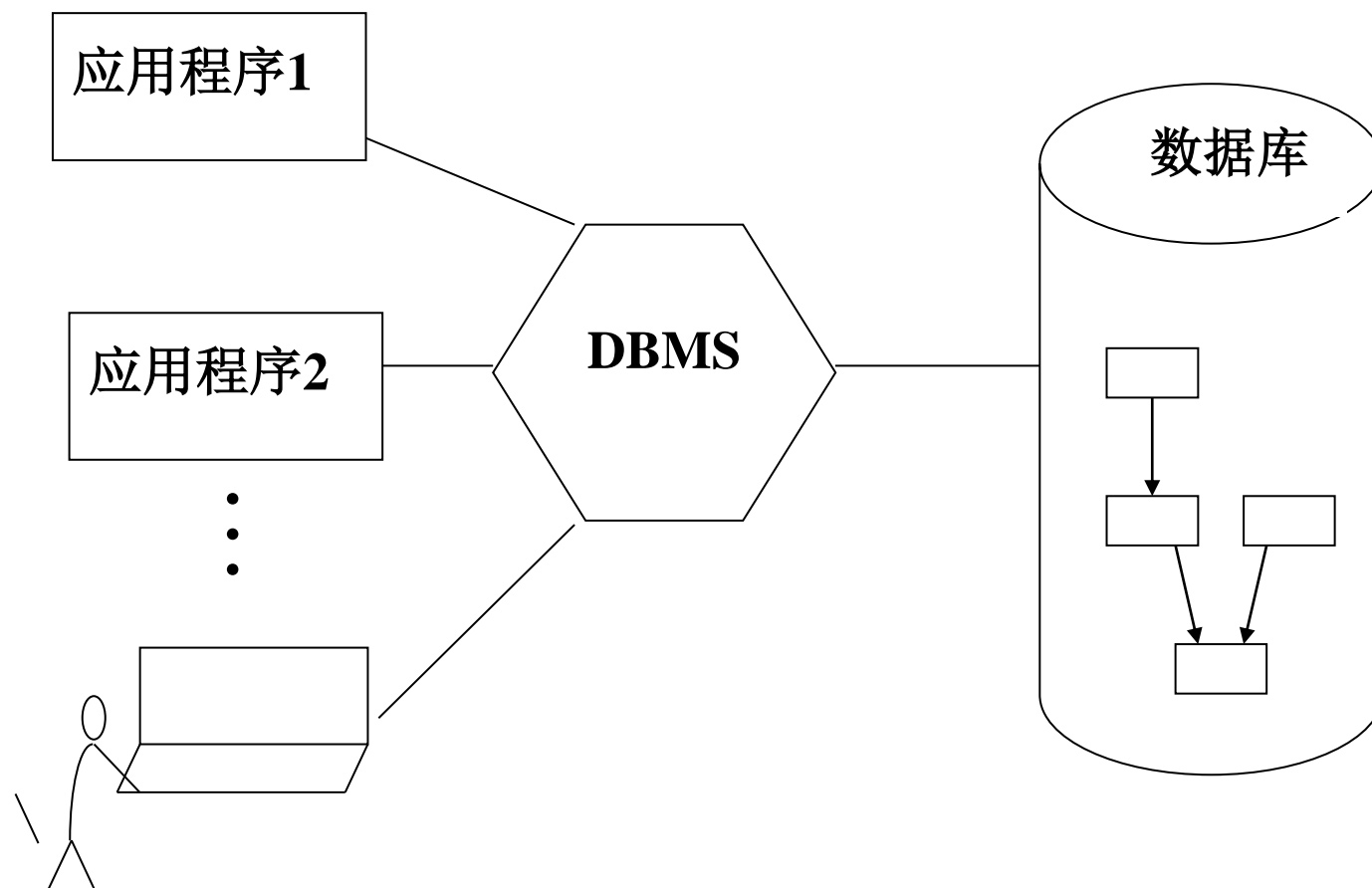
- 数据的完整性是指数据的正确性、有效性和相容性
    - 将数据控制在有效的范围内，或保证数据之间满足一定的关系

- **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



图：应用程序与数据的对应关系(数据库系统)

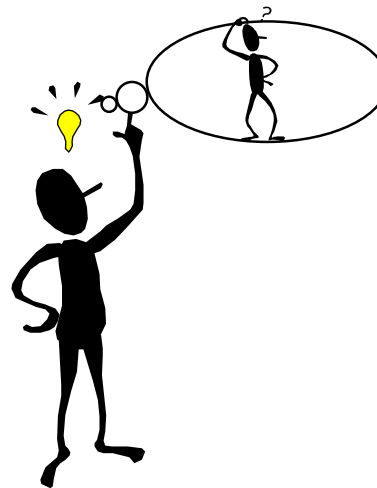


# 数据库概念小结

- 数据库是长期存储在计算机内有组织的大量的共享的数据集合。它可以供各种用户共享，具有最小冗余度和较高的数据独立性。
- DBMS在数据库建立、运用和维护时对数据库进行统一控制，来保证数据的完整性、安全性，并在多用户同时使用数据库时进行并发控制，在发生故障后对数据库进行恢复。

# 大纲

- 数据库系统概述
- **数据模型**
- 数据库系统结构
- 数据库系统的组成
- 本章小结



**Student** (*sid: string, name: string, login: string, age: integer, gpa: real*)



# 数据模型(Data Model)

- 什么是数据模型?
- 两大类数据模型
  - 概念模型与结构模型(逻辑与物理)
- 概念模型
  - 几个基本概念, E-R方法
- 数据模型的组成要素
  - 数据结构、数据操作与完整性约束
- 最常用的数据模型
  - 层次、网状、关系与面向对象模型

# 什么是数据模型

- A **data model** is an abstract model that describes how data is **represented** and **accessed**.
- 数据模型是数据库系统的**核心和基础**
- 数据模型应满足三方面的要求：
  - 较真实地模拟现实世界
  - 易为人理解
  - 便于计算机实现
- 建立数据模型的过程称为**数据建模(Data modeling)**

# 两类数据模型

## ■ 两类分属两个不同层次的数据模型

### — 概念模型(或信息模型)

- 按用户的观点对数据和信息进行建模，主要用于数据库设计，与DBMS无关，如E-R模型

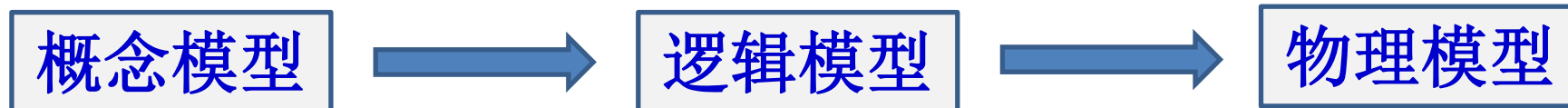
### — 逻辑模型和物理模型

#### • 逻辑模型

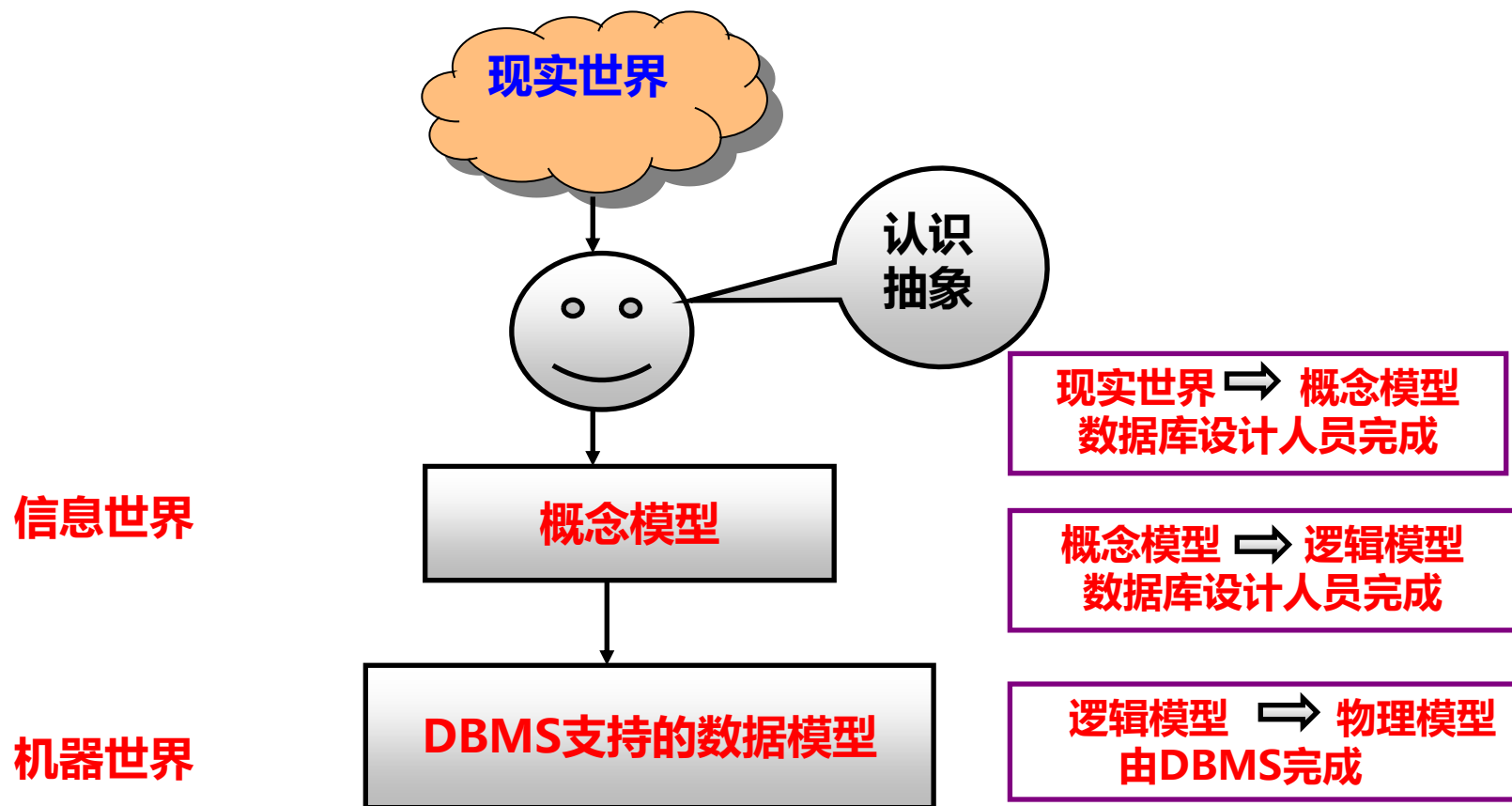
- 按计算机系统的观点对数据建模，用于DBMS实现。常见的模型有：网状、层次、关系、面向对象、半结构化等

#### • 物理模型

- 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—两步抽象



- 现实世界中的客观对象抽象为概念模型：将现实世界抽象为信息世界。
- 把概念模型转换为某一数据库管理系统支持的数据模型：将信息世界转换为机器世界。

# 概念模型

- 作用

- 是现实世界到机器世界的一个中间层次。用于信息世界的建模，是数据库设计人员进行数据库设计的有力工具，也是数据库设计人员和用户之间进行交流的语言

- 主要基本概念

- 实体、属性、码、域、实体型、实体集和联系

- 表示方法

- E-R方法

## ■ 实体(Entity)

- 客观存在并可相互区别的事物，如一个学生、一门课、部门的一次订货、老师与系的工作关系。

## ■ 属性(Attribute)

- 实体所具有的某一特性，一个实体可由若干个属性来刻画。

## ■ 码(Key)

- 唯一标识实体的属性或属性集，如学号是学生实体的码。

## ■ 域(Domain)

- 属性的取值范围，如性别的域为（男，女）

## ■ 实体型(Entity Type)

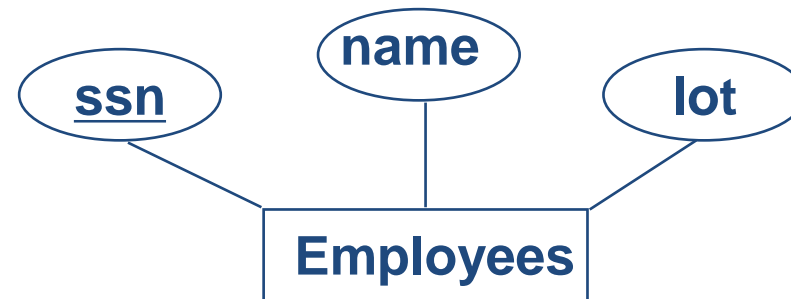
- 用实体名及其属性名集合来抽象和刻画同类实体，如学生（学号，姓名，性别，出生年份，系别，入学时间）就是一个实体型。

## ■ 实体集(Entity Set)

- 同型实体的集合，如全体学生就是一个实体集

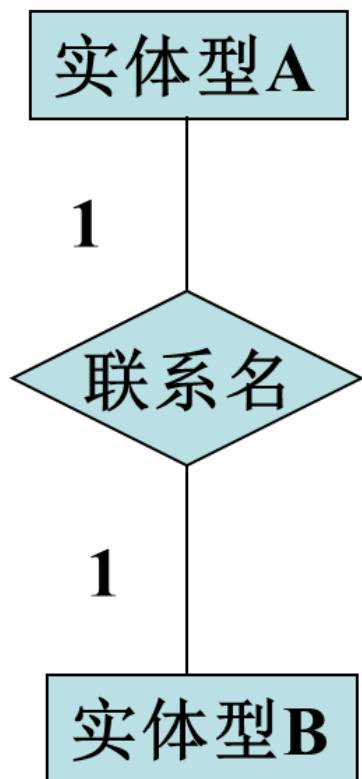
## ■ 联系(Relationship)

- 实体（型）内部的联系：同一实体的各属性间的联系
- 实体(型)之间的联系：不同实体集间的联系

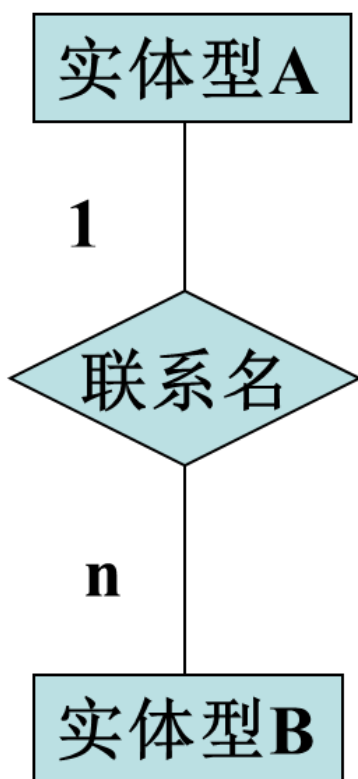




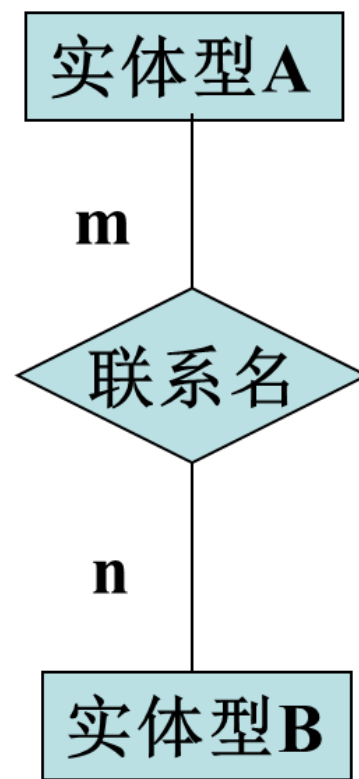
# 两个不同实体型之间的三类联系



1:1联系



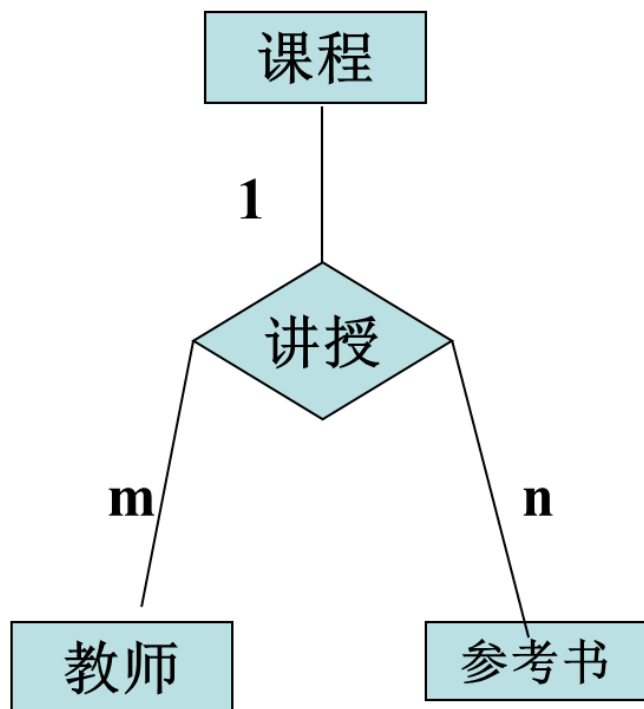
1:n联系



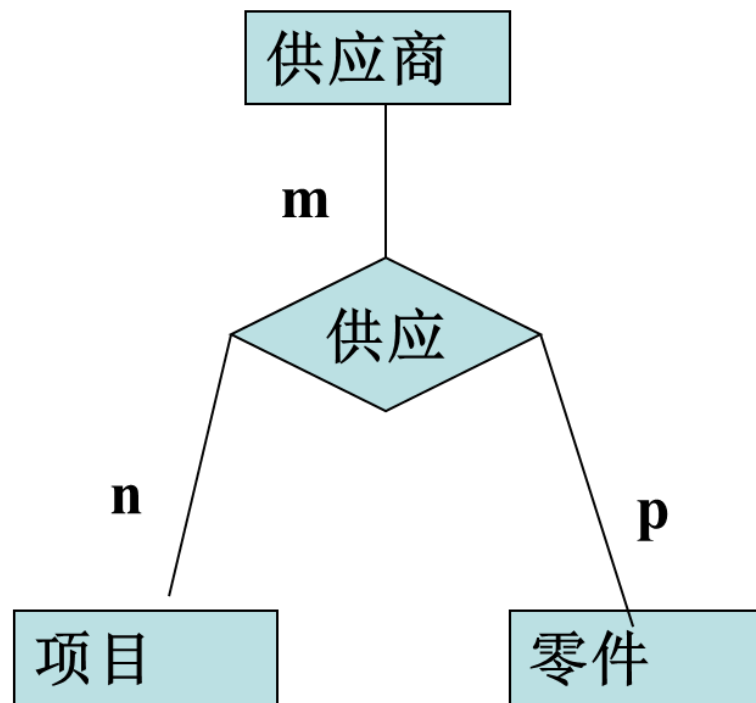
m:n联系

# 两个以上实体型间的三类联系

- 同样存在**一对一**、**一对多**、**多对多**联系。



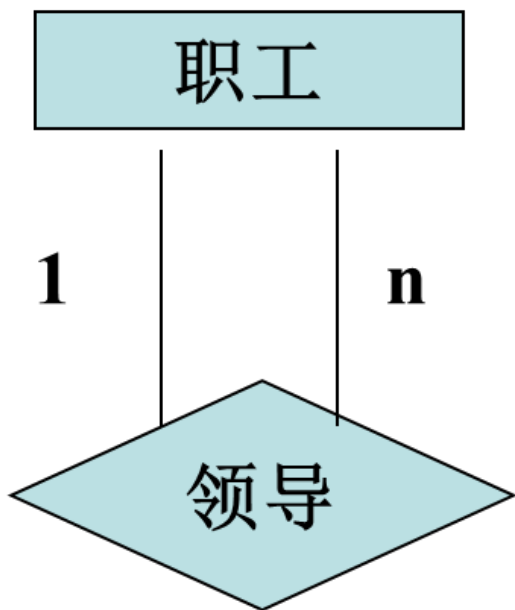
两个以上实体型间1:n联系



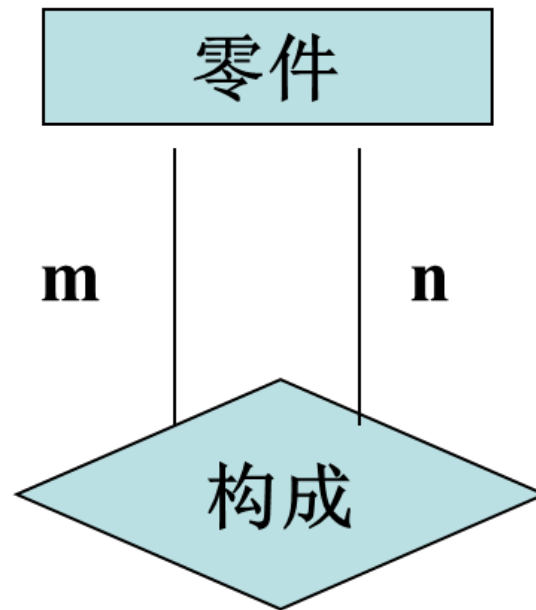
两个以上实体型间m:n联系

# 单个实体型内部的三类联系

- 同样存在**一对一**、**一对多**、**多对多**联系。



单个实体型内部1:n联系



单个实体型内的m:n联系

# 问题：怎么确定实体间的联系？

- 通过分析上下文的语义
- 小例：
  - 一个顾客可以购买多种商品，一种商品可以被多个顾客购买
  - 分析结果：\_\_\_\_\_。

# E-R图

- E-R图(Entity-Relationship Diagram)

- E-R方法(E-R Approach)
- E-R建模(E-R Modeling)
- P.P.S.Chen(陈品山)1976年提出



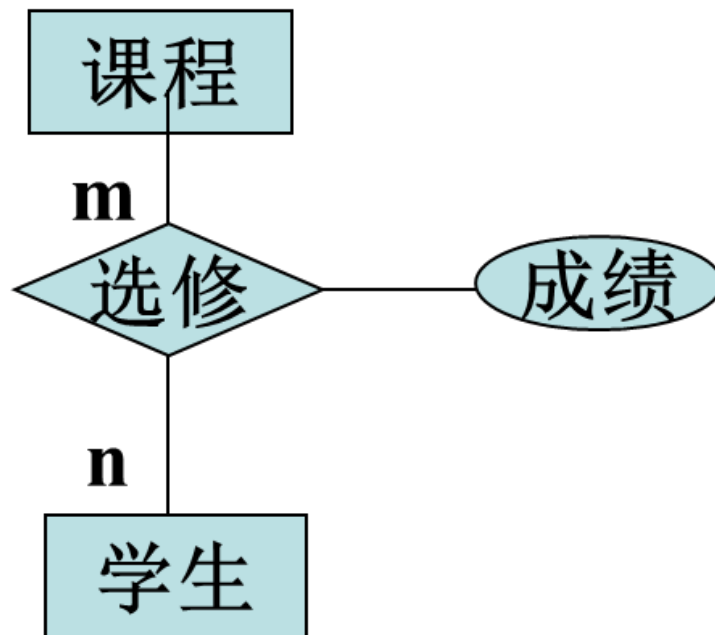
- Peter Chen. The Entity-Relationship Model--Toward a Unified View of Data. ACM Transactions on Database Systems, Vol. 1(1), p.9-36,1976

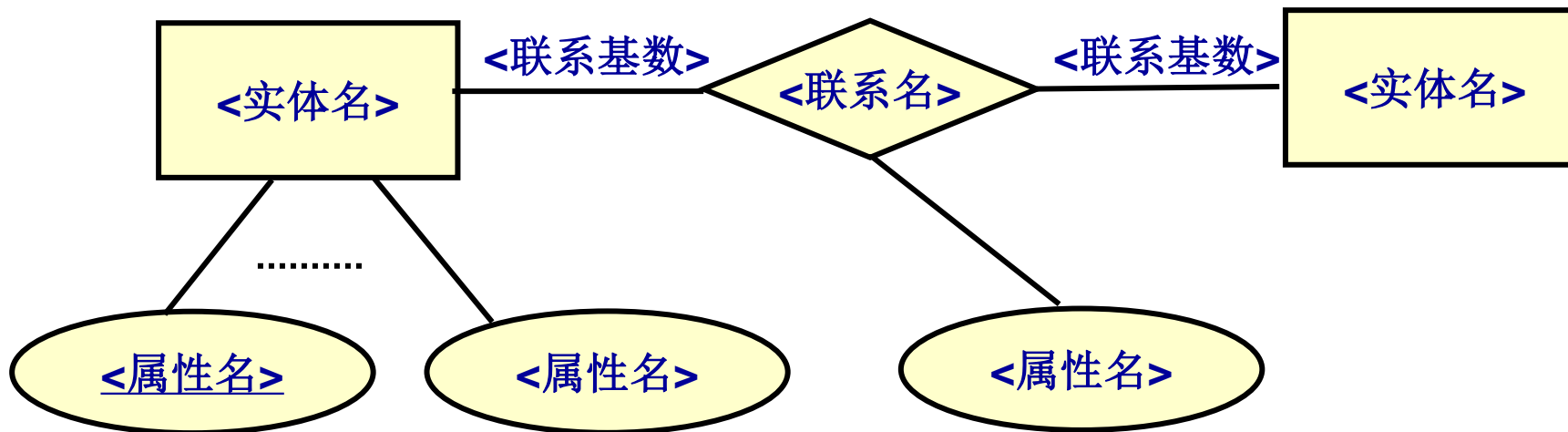
## ■ E-R图表示要点

- 实体：矩形
- 属性：椭圆
- 联系：菱形
  - 1:1、1:n或m:n

## ■ 命名规则：

- 尽可能地传递更多的含义
- 实体用名词；联系用动词
- 英文命名
  - 应用单数不用复数，如，customer而不是customers





**矩形：表示实体集。**

**菱形：表示联系集。两端写上联系的基数 (1:N, M:N, 1:1)**

**椭圆形：表示属性。实体的码加下划线，联系也可有属性**

**线段：将属性连接到实体集或将实体集连接到实体集。**

**双椭圆：表示多值属性。**

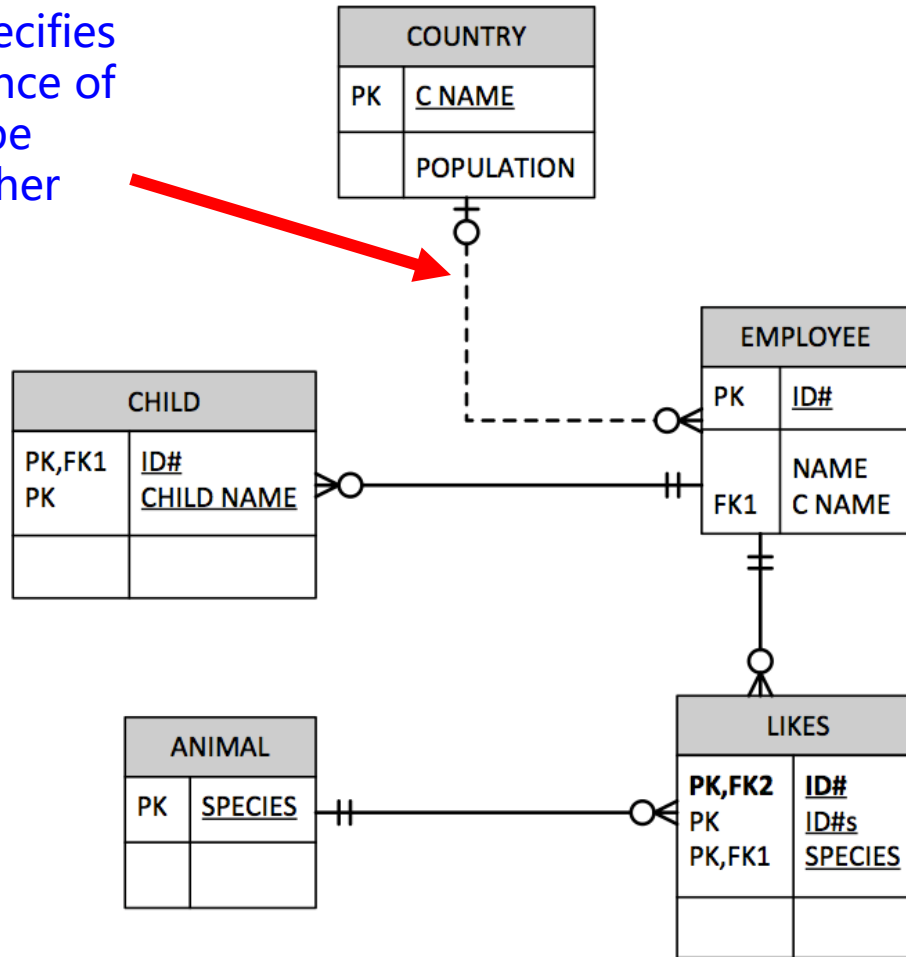
**虚椭圆：表示派生属性。**

**双线：表示一个实体集全部参与到联系集中。**

**双矩形：表示弱实体集。**

# Crow's Foot(鱼尾纹)表示法

- Dashed line specifies that each instance of an entity may be related to another instance.



Symbol	Meaning
	One—Mandatory
	Many—Mandatory
	One—Optional
	Many—Optional

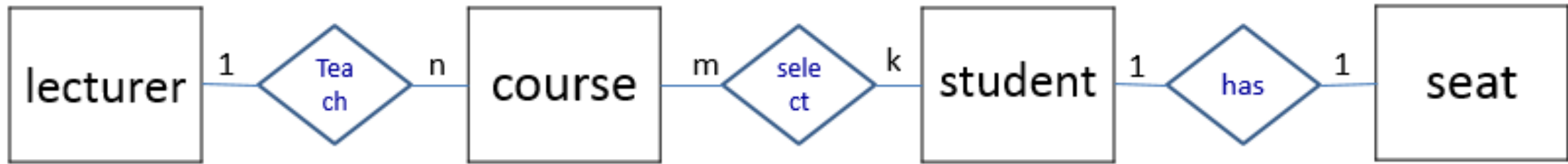
<https://www.conceptdraw.com/examples/crowfoot-notation>



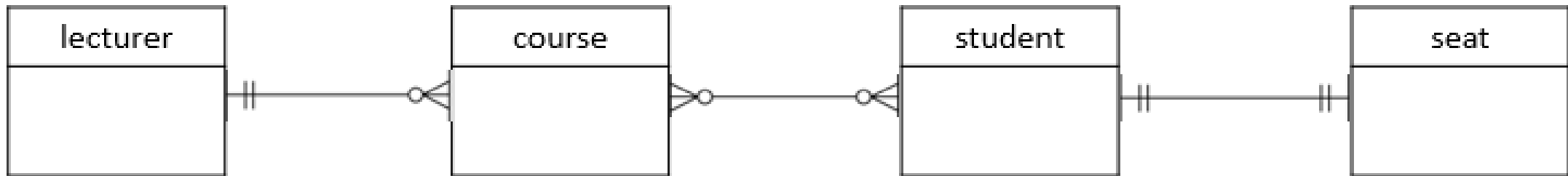
# 课堂练习

- 在教学系统中有业务规则(business rule)如下：
  - 一个学生(student)有一个座位(seat)，每个座位只能有一个学生坐；
  - 每个老师(lecturer)可以教多门课(course)，也可以不教课；每门课只能由一个老师教；
  - 一个学生可以选修多门课，也可以不选任何课；一门课可以被多个学生选，也可以允许没有学生选

请根据以上语义分别使用陈氏记法和鱼尾纹记法画出该E-R图。



**Fig 1. Chen's Notation**

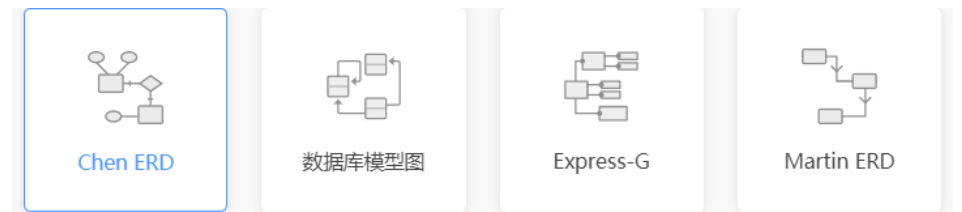
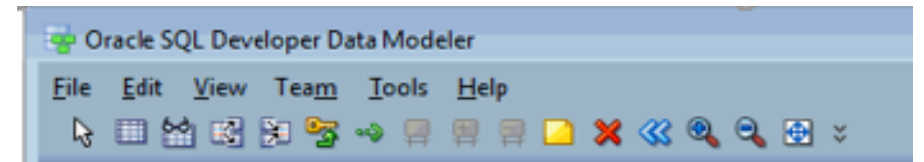


**Fig 2. Crow's Foot Notation**

## ■ E-R图建模工具

- 手工
- 数据建模工具
  - SAP Sybase PowerDesigner
  - CA Erwin
  - Oracle SQL Developer Data Modeler
  - Microsoft Visio
  - 亿图在线版

<https://www.edrawmax.cn/online/>



# 数据模型的三个组成要素

■ 数据模型 = 数据结构 + 数据操作 + 完整性约束

## 数据结构

- 是所研究的对象类型的集合
- 描述了系统的**静态特性**
- 例如，关系模型中的域、属性、关系
- 通常按数据结构的类型命名数据类型

## 数据操作

- 指对数据库中各种对象（型）的实例（值）允许执行的操作的集合，包括操作及有关的操作规则
- 描述了系统的**动态特性**
- 如，规定数据库的检索和更新（增、删、改）操作规则

## 完整性约束

- 一组**完整性规则**的集合
- 完整性规则是给定的数据模型中数据及其联系所具有的制约和依存规则，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

实体完整性  
参照完整性

## 四种数据结构模型比较

	层次模型	网状模型	关系模型	面向对象模型
创始	1968 IBM IMS系统	1969 CODASYL DBTG报告	1970 E.F.Codd 提出	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	面向对象标识
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	Oracle, Sybase, DB2, SQL Server	ONTOS DB
盛行期	20世纪70年代	20世纪70年代 到80年代中期	20世纪80年代 到现在	20世纪101年代到 现在

# 最常用的数据模型

- 层次模型 (Hierarchical Model)
- 网状模型 (Network Model)
- 面向对象模型 (Object Oriented Model)
- 关系模型 (Relational Model)

非关系模型

**非关系数据模型请自行看书，课堂不讲**

# 关系模型

- 关系数据模型简称
- 目前最重要的一种数据模型
- 关系数据库系统采用关系模型作为数据的组织方式
- 由IBM San Jose的E.F.Codd于1970年首次提出
- 关系数据库管理系统在当前市场上仍然占据主要市场份额

学生登记表

属性

元组

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
...	...	...	...	...	...

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



职工号	姓名	职 称	工 资			扣 除		实 发
			基 本	津 贴	职 务	房 租	水 电	
88051	陈 平	讲 师	1305	1200	50	160	112	2283
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

**!工资和扣除是可分的数据项,不符合关系模型要求**

- 关系模型要求关系必须是规范化的, 即要求关系必须满足一定的规范条件
- 最基本的一条: **关系的每一个分量必须是不可分的数据项**

# 术语对比

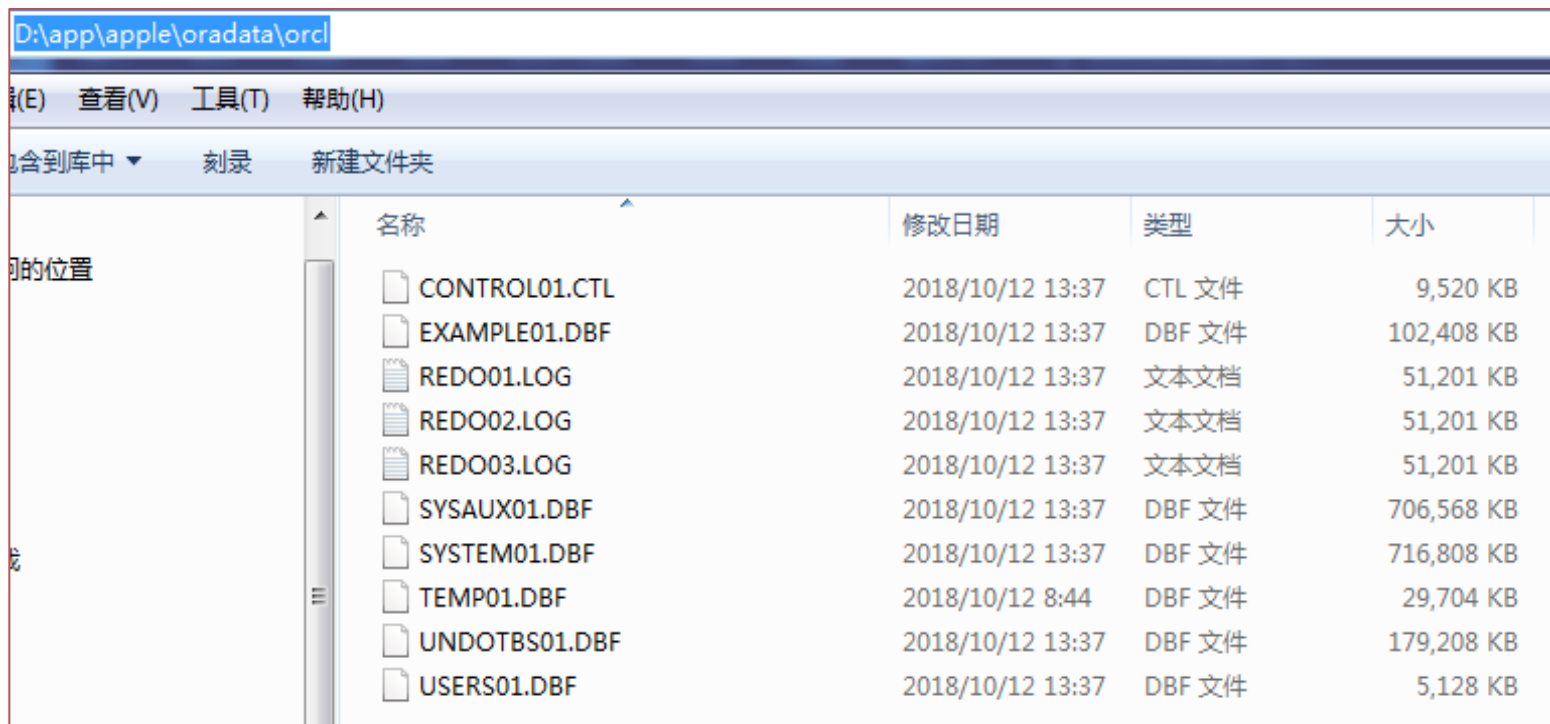
关系术语	一般表格的术语
关系名	表名
关系模式	表头（表格的描述）
关系	（一张）二维表
元组	记录或行
属性	列
属性名	列名
属性值	列值
分量	一条记录中的一个列值
非规范关系	表中有表（大表中嵌有小表）

# 关系数据模型的操纵与完整性约束

- 主要操作
  - 查询、插入、删除和修改数据
- 完整性约束条件
  - 实体完整性
  - 参照完整性
  - 用户自定义的完整性
- 数据操作是集合操作
  - 操作对象和操作结果都是关系
  - 区别于非关系模型中的单记录的操作方式
  - 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”

# 关系模型的存储结构

- 实体及实体间的**联系**都用**表**来表示
- 表以**文件形式**存储
  - 有的DBMS一个表对应一个操作系统文件；有的DBMS自己设计文件结构



名称	修改日期	类型	大小
CONTROL01.CTL	2018/10/12 13:37	CTL 文件	9,520 KB
EXAMPLE01.DBF	2018/10/12 13:37	DBF 文件	102,408 KB
REDO01.LOG	2018/10/12 13:37	文本文档	51,201 KB
REDO02.LOG	2018/10/12 13:37	文本文档	51,201 KB
REDO03.LOG	2018/10/12 13:37	文本文档	51,201 KB
SYSAUX01.DBF	2018/10/12 13:37	DBF 文件	706,568 KB
SYSTEM01.DBF	2018/10/12 13:37	DBF 文件	716,808 KB
TEMP01.DBF	2018/10/12 8:44	DBF 文件	29,704 KB
UNDOTBS01.DBF	2018/10/12 13:37	DBF 文件	179,208 KB
USERS01.DBF	2018/10/12 13:37	DBF 文件	5,128 KB

# 关系模型的优缺点

## ■ 优点

- 建立在严格的数学概念的基础上
- 概念单一
  - 实体和各类联系都用关系来表示
  - 对数据的检索结果也是关系
- 存取路径对用户透明
  - 具有更高的数据独立性，更好的安全保密性
  - 简化了程序员的工作和数据库开发建立的工作

## ■ 缺点

- 查询效率往往不如非关系数据模型
- 对用户的查询请求的优化增加了开发DBMS的难度

# 大纲

- 数据库系统概述
- 数据模型
- **数据库系统结构**
- 数据库系统的组成
- 本章小结

# 数据库系统结构

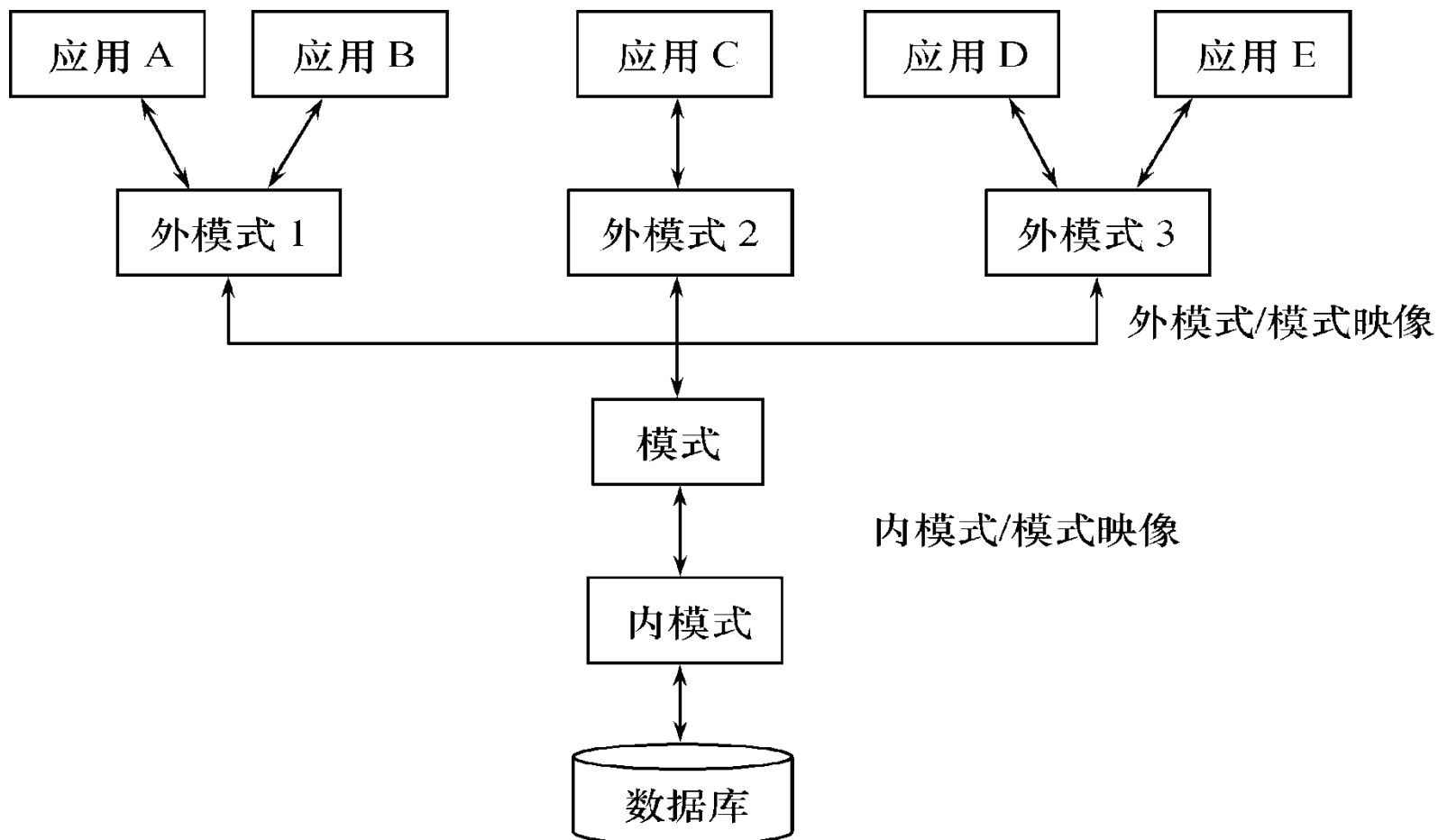


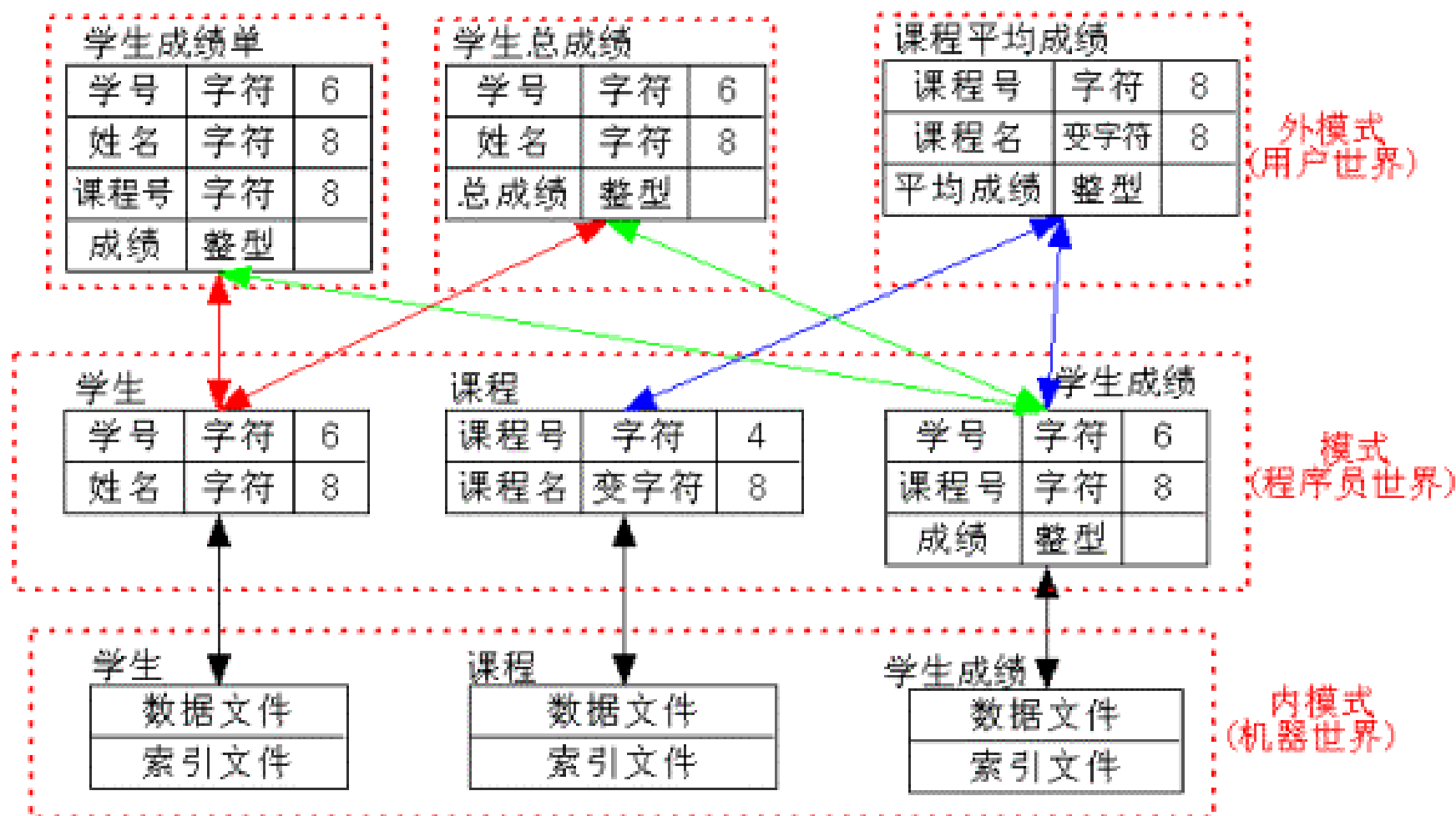
Fig. ANSI-SPARC Architecture

(American National Standards Institute, Standards Planning and Requirements Committee)

	外模式 (External Schema)	模式 (Schema)	内模式 (Internal Schema)
定义	也称子模式或用户模式 还称用户级模式	也称逻辑模式 还称概念级模式	也称存储模式 还称物理级模式
	是数据库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	是数据库中全体数据的逻辑结构和特征的描述, 它包括: 数据的逻辑结构、数据之间的联系和与数据有关的安全性、完整性要求。	它是数据物理结构和存储方式的描述
特点1	是各个具体用户所看到的数据视图, 是用户与DB的接口。	是所有用户的公共数据视图	数据在数据库内部的表示方式
特点2	可以有多个外模式	只有一个模式	只有一个内模式
特点3	针对不同用户, 有不同的外模式描述。每个用户只能看见和访问所对应的外模式中的数据, 数据库中其余数据是不可见的。所以外模式是保证数据库安全性的一个有力措施。	数据库模式以某一种数据模型(层状、网状、关系)为基础, 统一综合地考虑所有用户的需求, 并将这些需求有机地结合成一个逻辑整体。	
特点4	面向应用程序或最终用户	由DBA定义	以前由DBA定义, 现基本由DBMS定义



## ■ 示例:



# 数据库的二级映象功能与数据独立性

## ■ 外模式/模式映象

- 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，外模式不变，应用程序就没必要修改。所以外模式/模式映像功能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

- 当数据库的存储结构改变了，模式/内模式映像作相应的改变，以使模式保持不变。模式不变，与模式没有直接联系的应用程序也不会改变，所以模式/内模式映像功能保证了数据与程序的物理独立性，简称数据的物理独立性。

## ■ 示例:

□ 外模式: EMP (EMP, DEPT, NAME)

```
Create View EMP(EMP,DEPT,NAME)
As
Select E# as EMP,D# as DEPT,name
From Employee
```

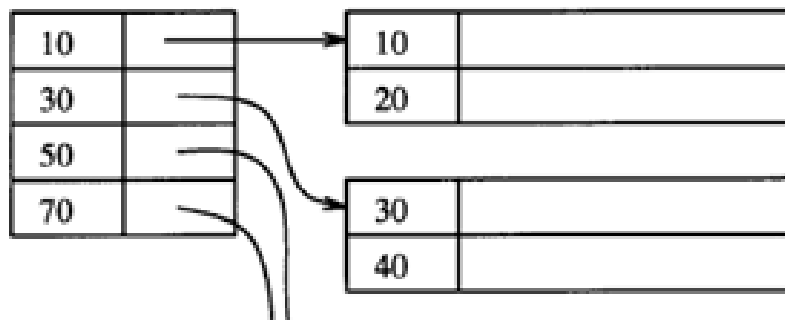
此语句定义了外模式，同时也定义了外/模映象关系

E# → EMP#

E# → EMP#

□ 模式: Employee(E# D#,Name,Salary)

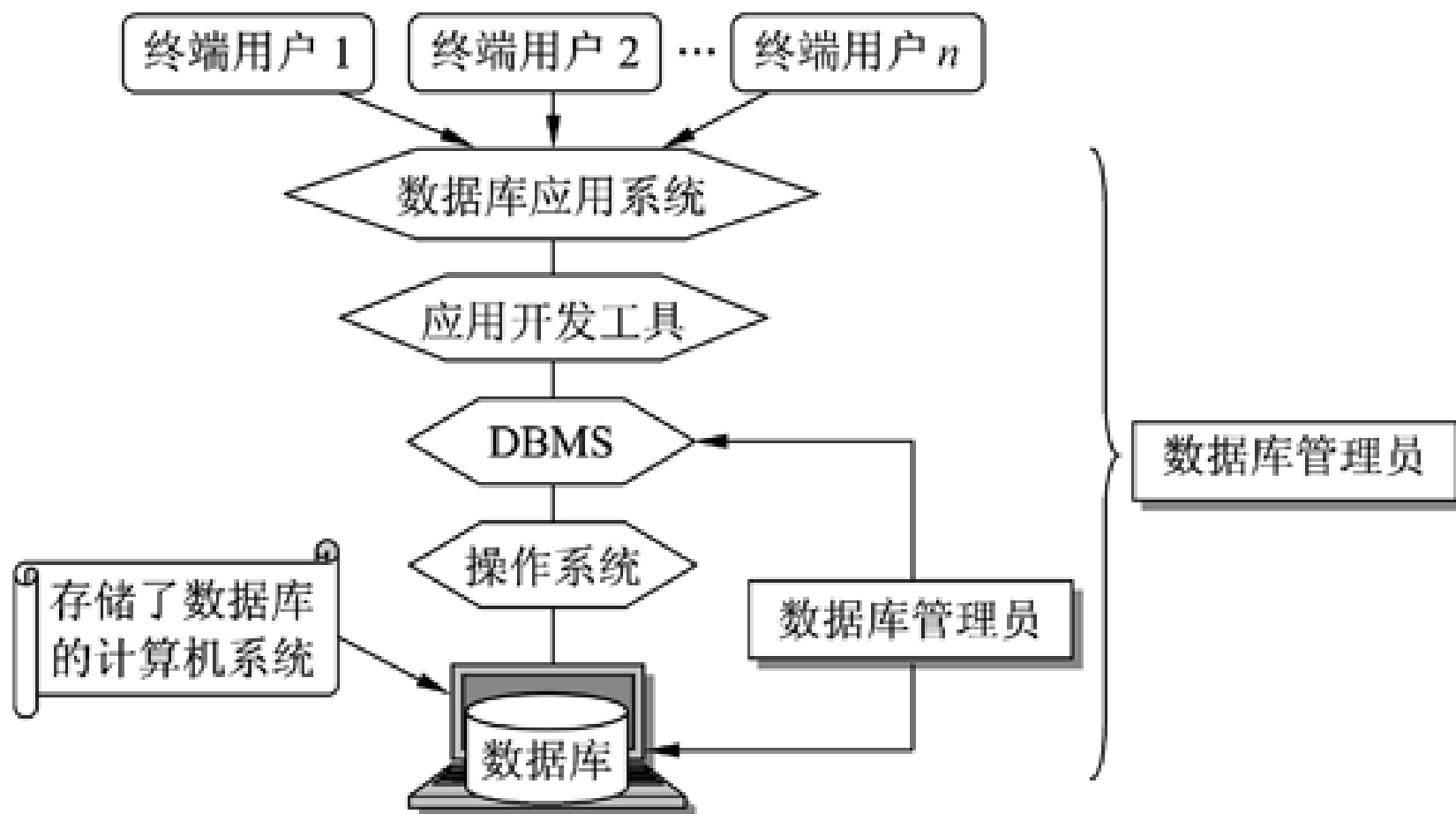
□ 内模式: 顺序文件, 索引文件, .....



# 大纲

- 数据库系统概述
- 数据模型
- 数据库系统结构
- **数据库系统的组成**
- 本章小结

# 数据库系统的组成

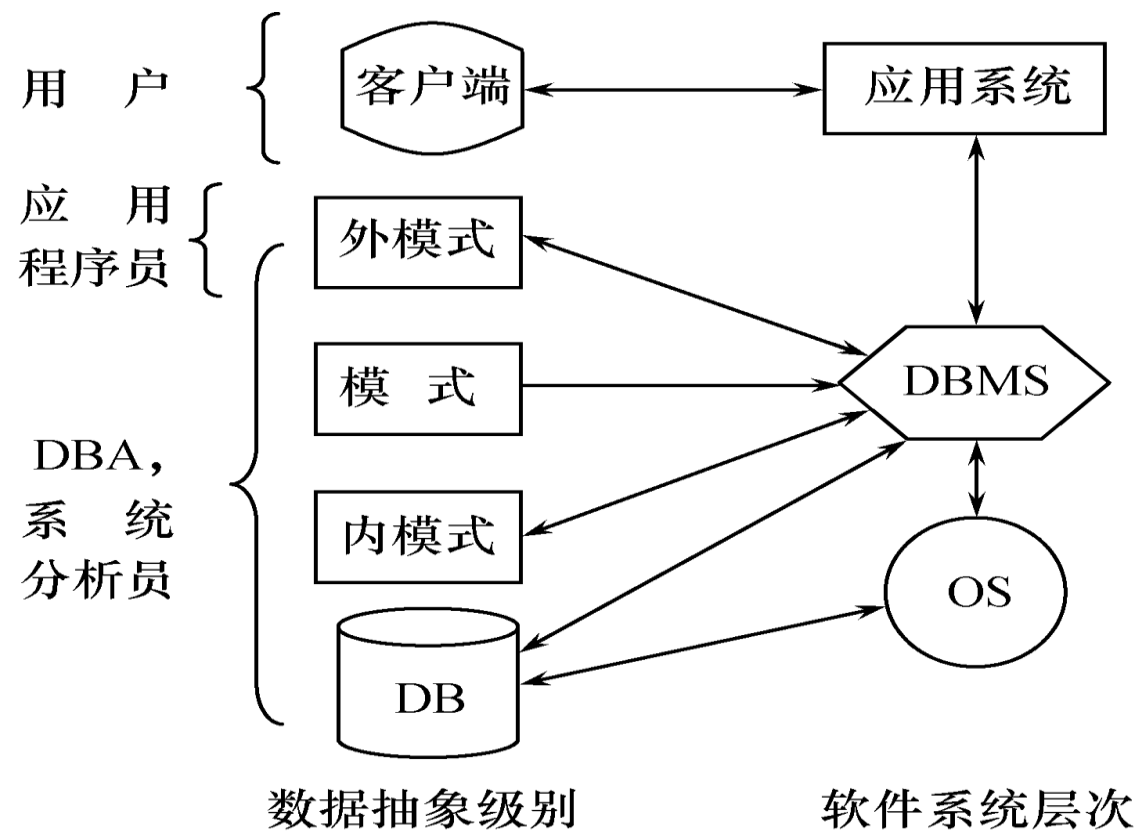


数据库系统=硬件+数据库+OS+DBMS+应用系统+应用开发工具+DBA+用户

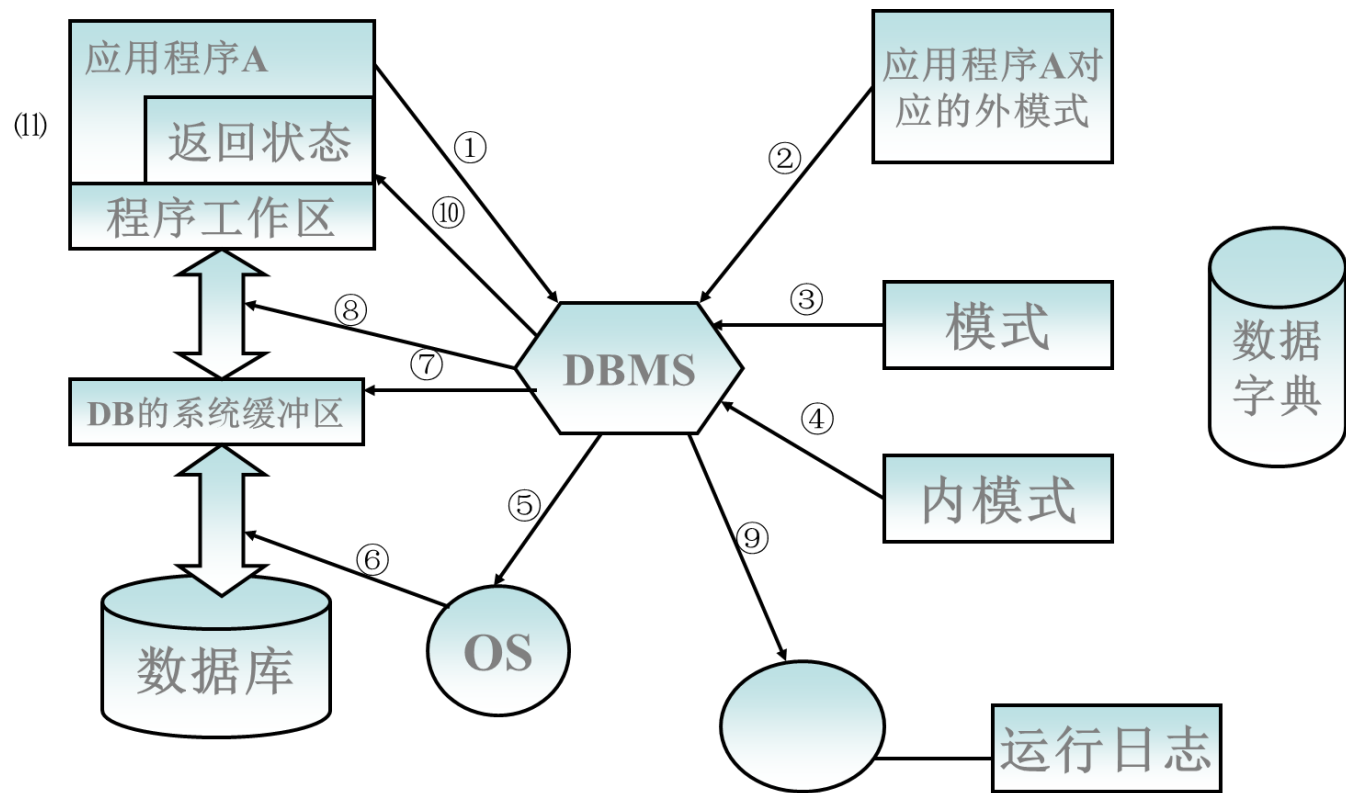
# 人员

- **数据库管理员(Database Administrator, DBA)**
  - 决定数据库中的信息内容和结构、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重构
- **系统分析员**
  - 负责应用系统的需求分析和规范说明，与用户和DBA一起确定系统的软硬件配置，并参与系统的概要设计
- **数据库设计人员**
  - 负责数据库中数据的确定和数据库各级模式的设计。常由DBA担任
- **应用程序员**
  - 负责应用系统的设计与编码，并进行调试和安装
- **最终用户(End User, EU)**
- **偶然用户、简单用户和复杂用户**

## ■ 各类人员的数据视图:



## ■ 用户访问数据的过程:





# 课堂练习

- 数据库系统的核心和基础是 ( )  
A.物理模型      B.概念模型      C.数据模型      D.逻辑模型
- 实现将现实世界抽象为信息世界的是 ( )  
A.物理模型      B.概念模型      C.关系模型      D.逻辑模型
- 能够保证数据库系统中的数据具有较高的逻辑独立性是 ( )  
A.外模式/模式映像      B.模式      C.模式/内模式映像      D.外模式
- DBMS是一类系统软件，它是建立在下列哪些系统之上的？ ( )  
A.应用系统      B.编译系统      C.操作系统      D.硬件系统

- 下列说法正确的是（ ）
  - A.数据库的概念模型与具体的DBMS有关
  - B.三级模式中描述全体数据的逻辑结构和特征的是外模式
  - C.数据库管理员负责设计和编写应用系统的程序模块
  - D.从逻辑模型到物理模型的转换一般是由DBMS完成的
- 长期存储在计算机内，有组织的、可共享的大量数据的集合是（ ）
  - A.数据      B.数据库      C.数据库管理系统      D.数据库系统
- 在数据库管理技术发展过程中，需要应用程序管理数据的是（ ）
  - A.人工管理阶段      B.人工管理阶段和文件系统阶段
  - C.文件系统阶段和数据库系统阶段      D.数据库系统阶段

- 在文件系统管理阶段，由文件系统提供数据存取方法，所以数据已经达到很强的独立性。（对or错）
- 数据库管理系统是指在计算机系统中引入数据库后的系统，一般由DB, DBS, 应用系统和DBA组成。（对or错）
- 在数据模型的组成要素中，数据结构是刻画一个数据模型性质最重要的方面，人们通常按照数据结构的类型来命名数据模型。（对or错）
- 数据库系统的三级模式是对数据进行抽象的三个级别，把数据的具体组织留给DBMS管理。（对or错）
- 层次模型是比网状模型更具普遍性的结构，网状模型是层次模型的一个特例。（对or错）

- 数据库系统的逻辑模型按照计算机的观点对数据建模，主要包括\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、面向对象模型、对象关系模式和半结构化数据模型等。
- 最经常使用的概念模型是\_\_\_\_\_。
- 数据独立性是数据库领域的重要概念，包括数据的\_\_\_\_\_独立性和数据的\_\_\_\_\_独立性。
- 数据库的三级模式结构中，描述局部数据的逻辑结构和特征的是\_\_\_\_\_。
- 数据模型的组成要素中描述系统的静态特性和动态特性的分别是\_\_\_\_\_和\_\_\_\_\_。

# 本章小结

- 数据库系统概述
  - 数据库的基本概念
  - 数据管理的发展过程
  - 数据库系统的特点
- 数据模型
  - 数据模型三个组成要素
  - 概念模型, E-R 模型
  - 三种主要数据库模型
- 数据库系统内部的系统结构
  - 三级模式+两级映像的体系结构
- 数据库系统的组成

# 本章作业

- 教材第一章全部习题.