

# 第一章 绪论

---

## 1.1 数据库系统概述

## 1.2 数据模型

## 1.3 数据库系统结构

## 1.4 数据库系统的组成（自学）

# 1.1 数据库系统概述

- 发展了一门计算机基础学科
- 带动了一个巨大的软件产业
- 造就了四位图灵奖得主

13年甲骨文公司超越IBM,  
成为全球第二大软件公司

- Charles W.Bachman (网状数据库之父, 1973年数据库界第一个图灵奖), 开发了世界上第一个网状数据库IDS;
- E.F.Codd (关系数据库之父, 1981年数据库界第二个图灵奖), 提出关系模型概念;
- James Grayz 1998年获得了数据库界第三个图灵奖, 在数据库事务处理研究及系统实现方面做出巨大贡献。
- Michael Stonebraker 获得2014年图灵奖, 现代主流数据库系统架构的奠基人。

# 1.1.1 四个基本概念

---

- **数据(Data)**
- **数据库(DataBase)**
- **数据库管理系统(DBMS)**
- **数据库系统(DBS)**

# 1.1.1 四个基本概念

## □ 数据(Data)：描述事物的符号记录

■ 种类：数字、文字、图形、图象、音频、视频等

■ 特点：数据与其语义是不可分的

90

数据的形式不能  
完全表达其内容

✓ 现实世界常用自然语言描述事物：

如李明同学，男，1997年出生，2016年考入电子信息系

✓ 计算机中常用记录描述事物：

如学生记录 (李明，男，1997，电子信息系，2016)

■ 数据含义：学生姓名、性别、出生年、系、入学时间。

■ 数据是有结构的：记录是计算机存储数据的一种格式

有结构的数据如何保存



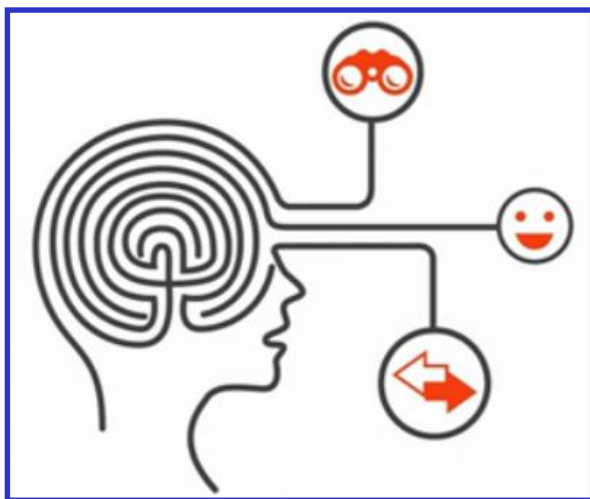
Excel



共享性



# 1.1.1 四个基本概念



你认识的人中有多少戴眼镜



人脑的记忆 → 非结构化



A的名字	XXX
B的年龄	19
C的身高	170
C的性别	男
...	...

结构化的数据



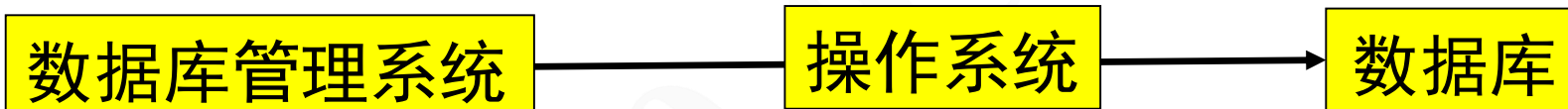
序号	姓名	年龄	戴眼镜吗?
1	张三	19	是
2	李四	20	不是
3	王五	18	不清楚
...	...	...	...

# 1.1.1 四个基本概念

□ 数据库(Database, DB): 存放数据的仓库

■ 长期储存在计算机内、有组织的、可共享的  
大量数据集。  
*数据之间有联系*

如何管理数据库 ?



□ 数据库管理系统 (Database Management System, DBMS) :  
用户与操作系统之间的数据管理软件。

■ 主要功能: 数据定义、组织、存储、管理、操纵功能,  
数据库的事务和运行管理, 数据库的建立和维护等功能。

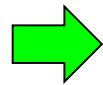
# 1.1.1 四个基本概念

谁使用DBMS?



数据库管理员 (DataBase Administrator, DBA)

DBA如何使用DBMS管理数据?



数据库语言

普通用户如何使用DBMS中的数据?



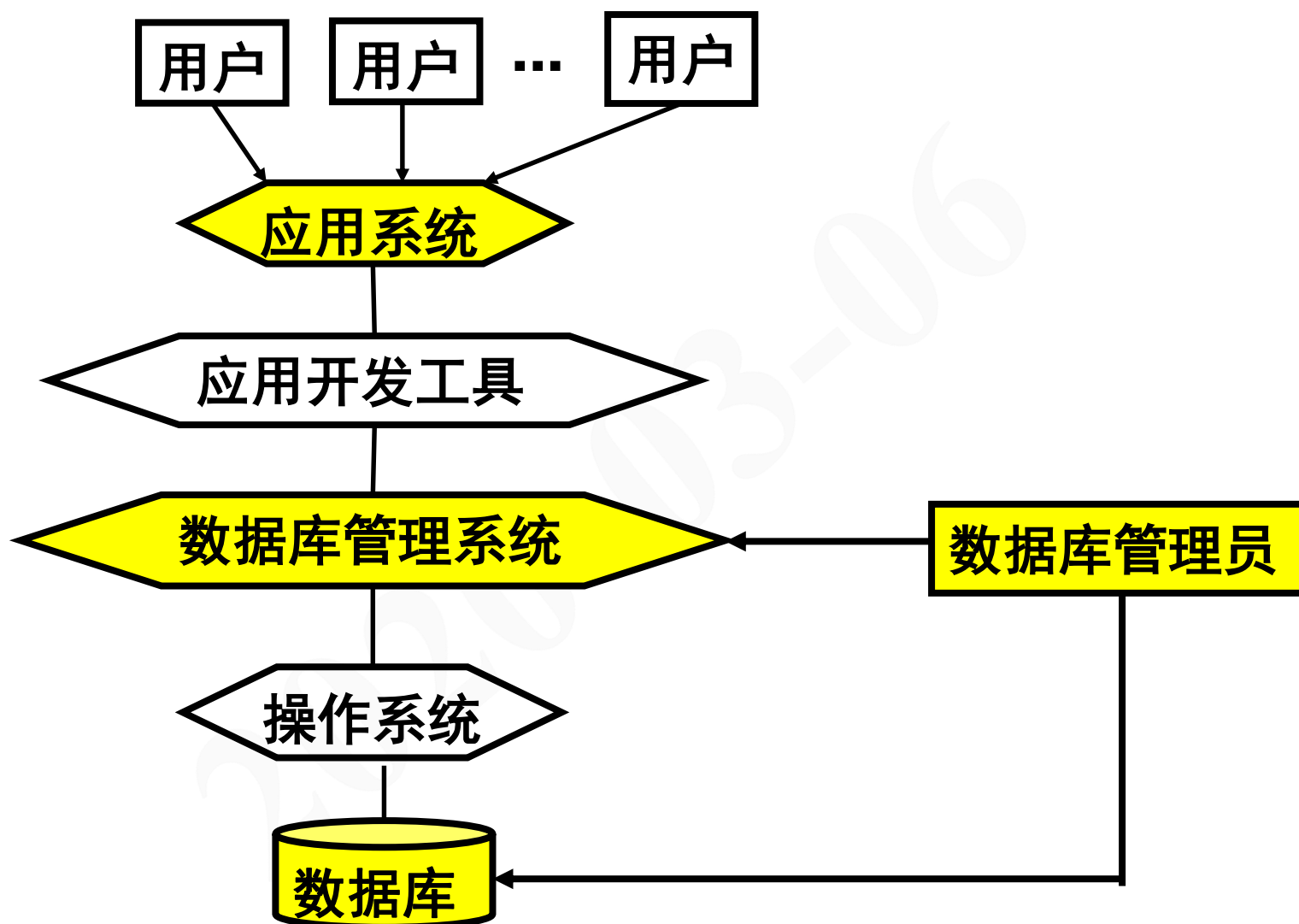
数据库应用程序 (DBAP)

□ 数据库系统 (Database System, DBS) : 是指在计算机系统中引入数据库后的系统。

- DB
- DBMS
- DBA
- DBAP

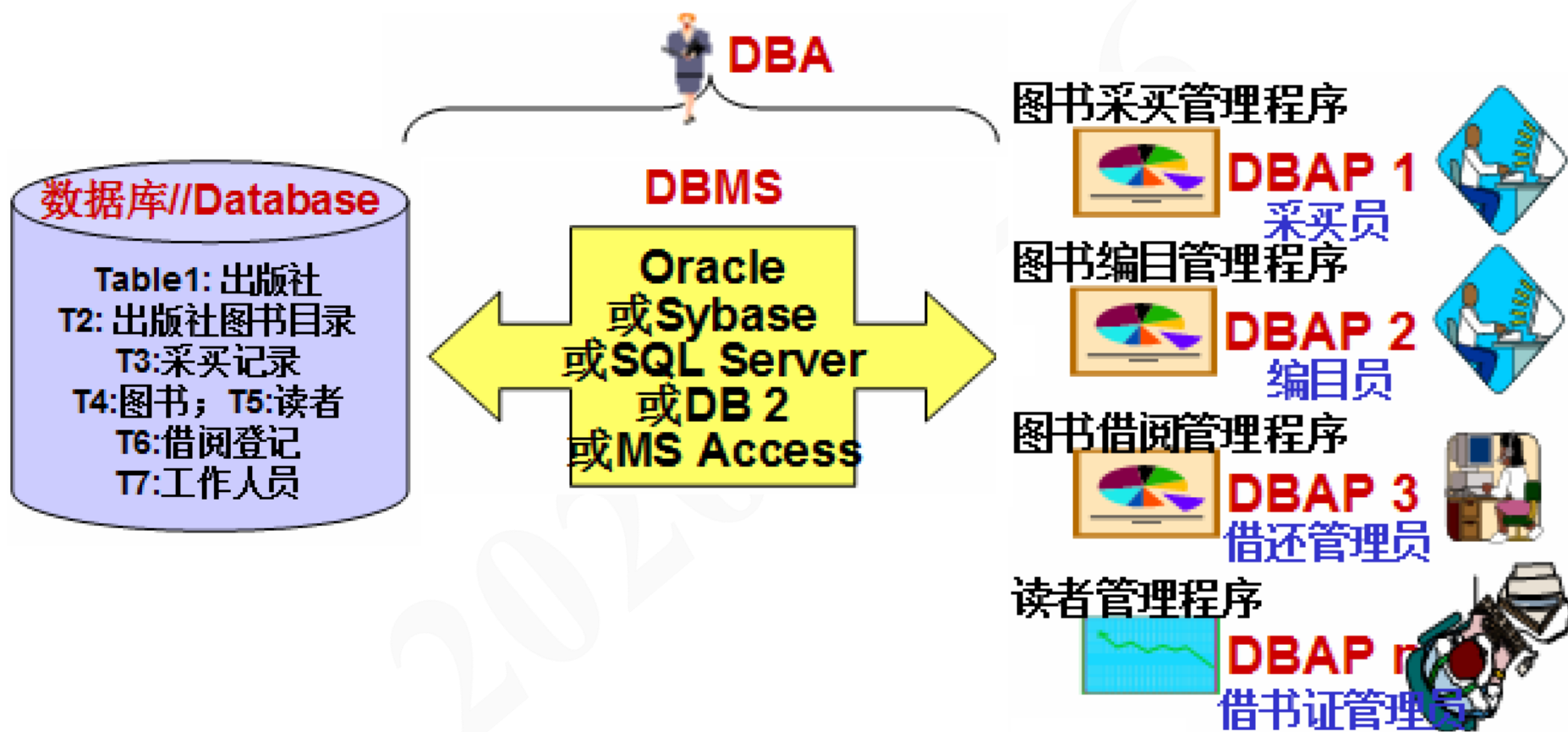
在不引起混淆的情况下常把  
数据库系统简称为数据库

# 数据库系统





# 数据库系统实例：图书管理系统



## 数据库管理系统是

- ☐ A 采用了数据库技术的计算机系统
- ☒ B 位于用户与OS之间的数据管理软件
- ☐ C 包括DBA、计算机硬件及DB的系统
- ☐ D 包括OS在内的数据管理软件

提交

# 1.1.2 数据管理技术的产生与发展

---

## 数据库技术是如何诞生的

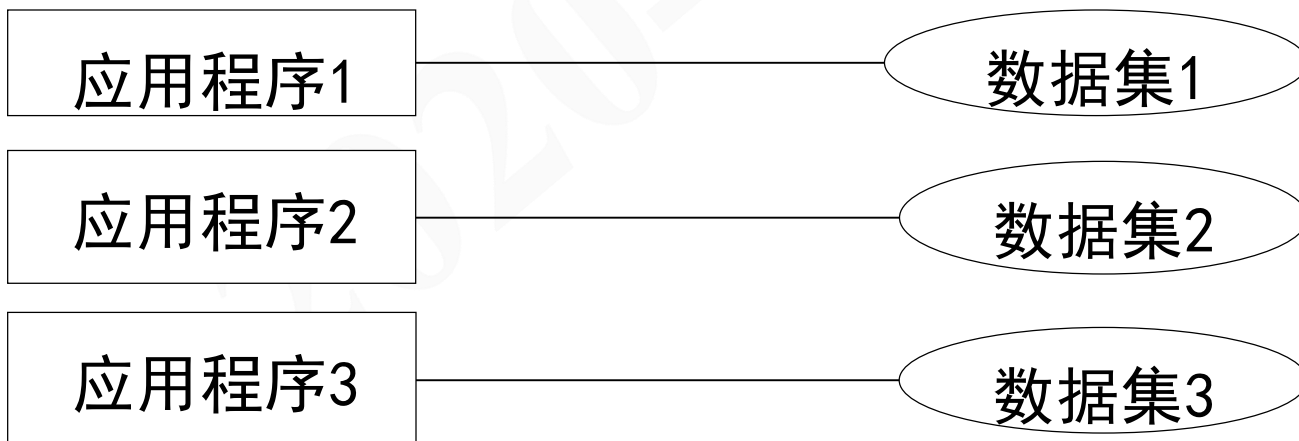
- **数据管理：**对数据进行分类、组织、编码、存储、检索和维护
  
- **数据管理技术的发展过程**
  - 人工管理阶段(20世纪50年代中之前)
  - 文件系统阶段(20世纪50年代末--60年代中)
  - 数据库系统阶段(20世纪60年代末--现在)

## 1.1.2 数据管理技术的产生与发展

### □ 人工管理阶段(20世纪50年代中之前)

- ★ 数据不保存
- ★ 程序员负责数据管理的一切工作
- ★ 数据和程序一一对应，没有独立性和共享性

### ■ 应用程序与数据的关系



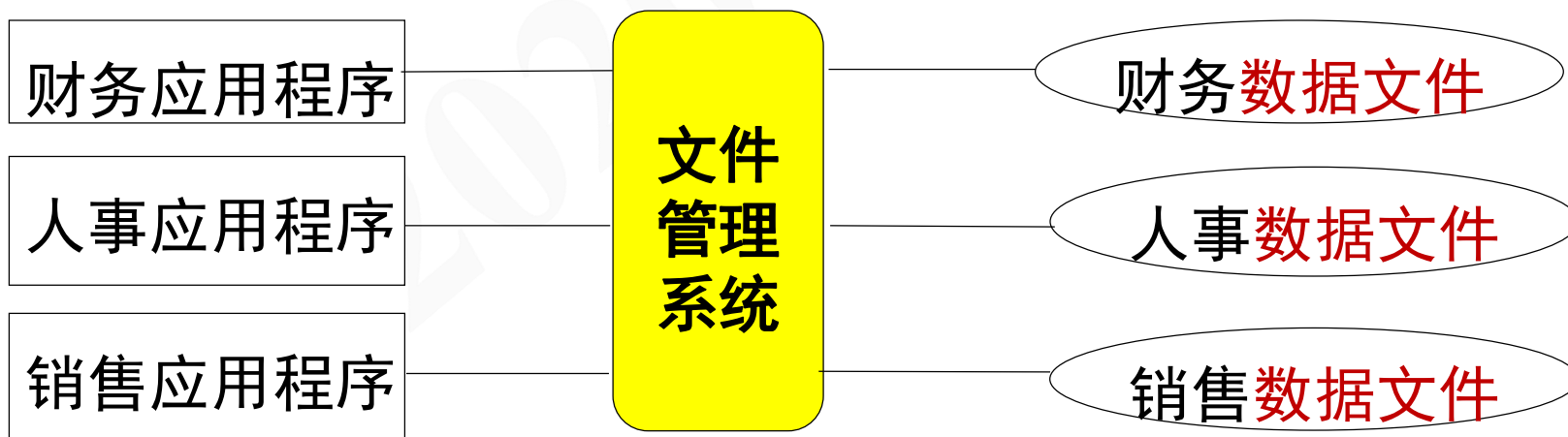
## 1.1.2 数据管理技术的产生与发展

### □ 文件系统阶段 (20世纪50年代末--60年代中)

- ★ 数据共享困难：文件私有，**记录内有结构**，记录之间无联系
- ★ 数据高度冗余：数据面向某一应用程序或特定用户。
- ★ 数据和程序缺乏独立性：具有一定的**物理独立性**，**无逻辑独立性**，**数据的最小存取单位是记录**。

0371041,数据库原理与应用(含实验),2,限,1,电气,周艳

#### ■ 应用程序与数据的关系



## 1.1.2 数据管理技术的产生与发展

### □ 数据库系统阶段 (20世纪60年代末以来)

- ★ **数据共享性高**，冗余度低
- ★ 数据高度结构化：**整体数据结构化**（**数据模型**描述数据及**联系**），面向整个系统，**数据库与文件系统的根本区别**。
- ★ 数据和程序的独立性：高度的**物理独立性**和一定的**逻辑独立性**，**数据的最小存取单位是数据项**。

### ■ 应用程序与数据的关系

0371041	数据库原理与应用（含实验）	2.0	限	1	电气	周艳
---------	---------------	-----	---	---	----	----



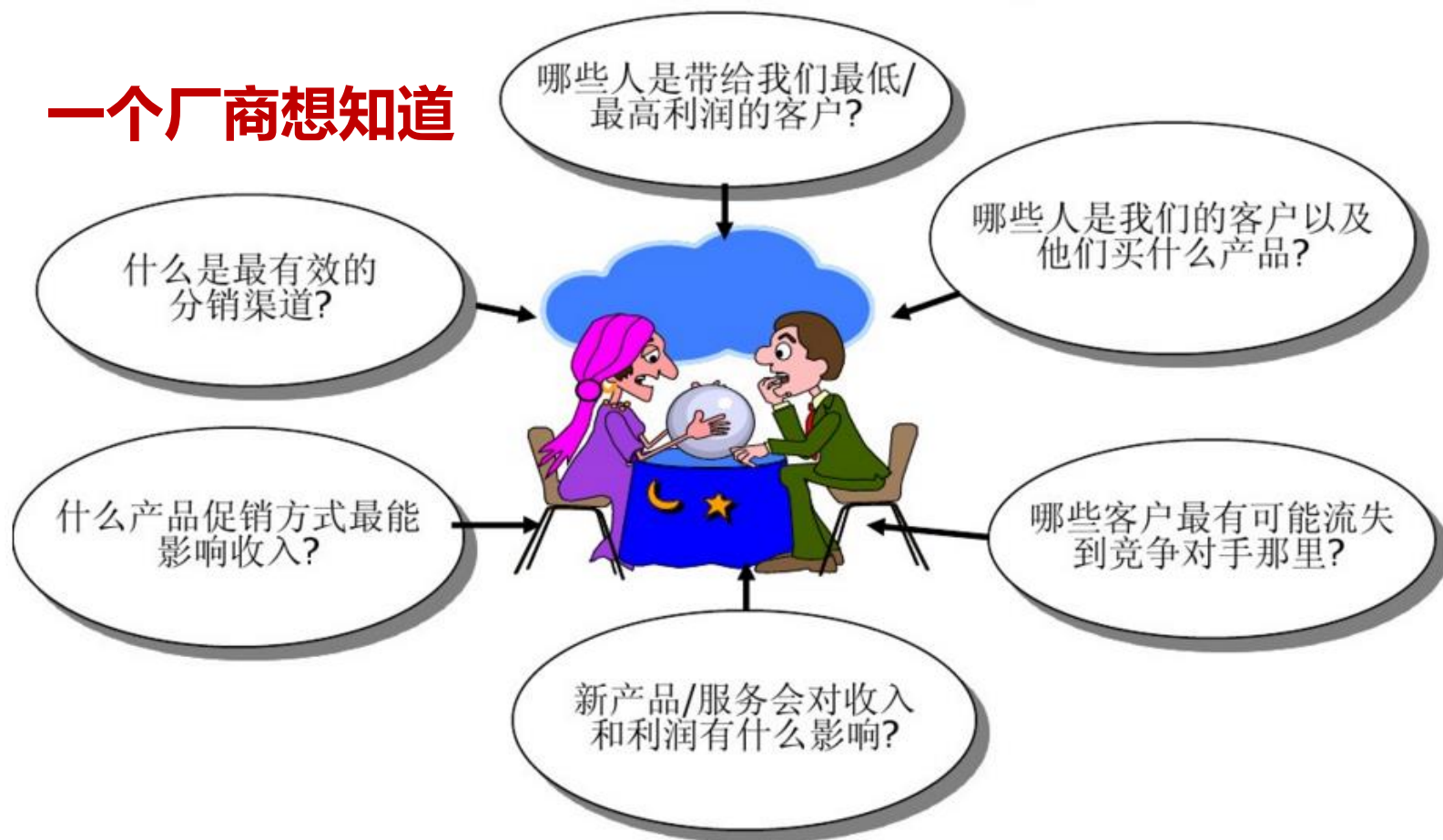
数据库的特点之一是数据的共享，严格的讲，  
这里的数据共享是指

- ☐ A 多个用户共享一个数据文件
- ☐ B 多个用户、同一种语言共享程序
- ☒ C 多种应用、多种语言、多个用户共享数据集合
- ☐ D 一个应用的多个程序共享一个数据集合

提交

# 数据库能满足用户的所有需求吗

## 一个厂商想知道



**需要数据的分析处理：通过历史的、综合的、集成的数据。**

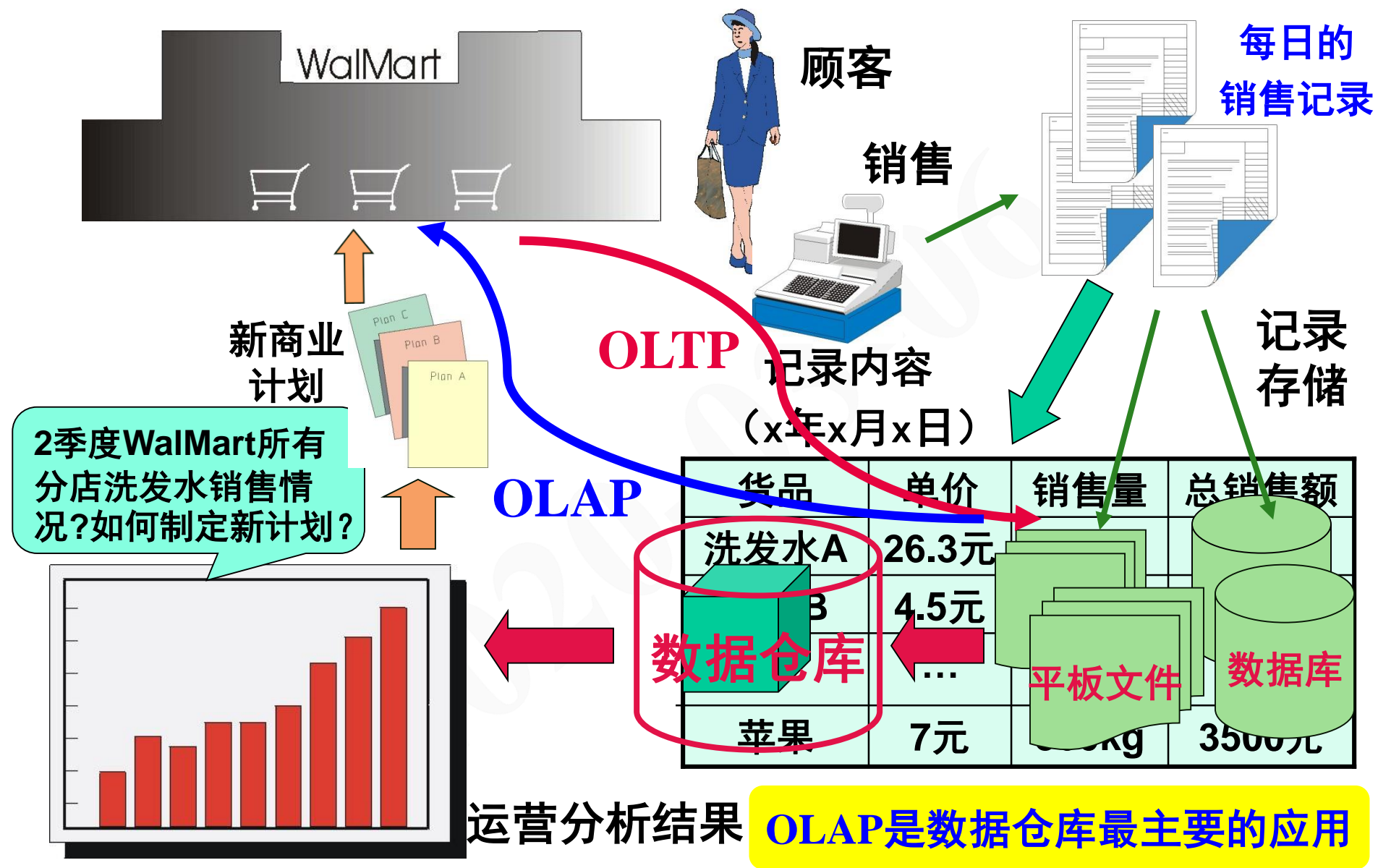


# 从数据库到数据仓库



- ❑ 传统的数据库技术主要用于联机事务处理( OLTP, On-Line Transaction Process ), 但对分析处理支持不够。
- ❑ 必须把分析型数据从事务处理环境中提取出来, 按照决策支持系统处理的需要进行重新组织, 建立单独的分析处理环境。
- ❑ 数据仓库技术正是为了构建这种新的分析处理环境而出现的一种数据存储和组织技术。

# 数据仓库、OLTP 和 OLAP(Online Analytical Processing)



# OLAP/Data Warehouse

---

- ❑ 1991, W.H. Inmon提出数据仓库(Data Warehouse)的概念
- ❑ 1993, Codd定义OLAP(Online Analytical Processing, 联机分析处理), 推动建立数据仓库。
- ❑ 数据仓库≠数据库
  - 数据库面向商业处理, 回答发生了什么(What happened)。
  - OLAP支持经营管理中的决策制定过程。回答下一步会怎么样(What next), 采取这样的措施又会怎么样(What if)

数据仓库就是用以更好地支持决策分析处理的、面向主题的、集成的、不可更新的、随时间不断变化的数据集合。

对学生按平均成绩排序，前10%的学生可获得免研资格

**What happened**



学校招生时要决定今年在安徽的招生指标，不能简单参照去年的计划，需要参考多维度的数据积累进行决策。

**What next**

**招生  
指标**

过去5年在安徽的招生指标变化

过去5年全国各地考生人数的变化

过去5年学校在各地招生指标的变化

过去5年各地考生毕业去向的情况等

# 大数据时代

大数据 (big data) 时代到来，引发数据管理/处理技术变革



# 从一个笑话体会大数据时代

---

某比萨店的电话铃响了，客服人员拿起电话。

客服：XXX比萨店。您好，请问有什么需要我为您服务？

顾客：你好，我想要一份……

客服：先生，烦请先把您的会员卡号告诉我。

顾客：16846\*\*\*。

客服：陈先生，您好！您是住在泉州路1号1205室，您公司电话是46\*\*\*\*，您手机是139\*\*\*\*。请问您用哪个电话付费？

顾客：你为什么知道我所有的电话号码？

客服：陈先生，因为我们联机到CRM系统。

顾客：我想要一个海鲜比萨……

客服：陈先生，海鲜比萨不适合您。

顾客：为什么？

客服：根据您的医疗记录，你的血压和胆固醇都偏高。

顾客：那你们有什么可以推荐的？

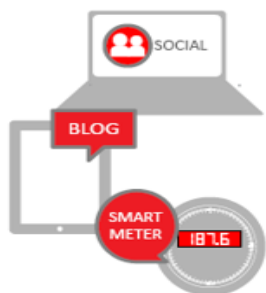
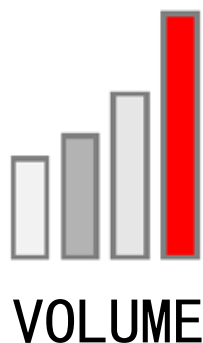
客服：您可以试试我们的低脂健康比萨。

顾客：你怎么知道我会喜欢吃这种的？

客服：您上星期一在中央图书馆借了一本《低脂健康食谱》。

。 。 。 。 。 。 。 。

# 大数据的4V特性



VARIETY



VELOCITY



VALUE

## □ 数据量巨大 Volume

- 集中存储/集中计算已无法处理

## □ 数据类型繁多 Variety

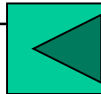
- 来源多、格式多

## □ 处理速度快 Velocity

- 1分钟时间
  - 新浪可以发送2万条微博
  - 淘宝可以卖出6万件商品
  - 百度可以产生90万次搜索查询

## □ 价值密度低，商业价值高 Value

- 真正可用的数据可能只有很小一部分

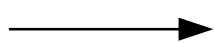




# NoSQL的诞生



概念演变



**Not only SQL**

最初表示“反SQL”运动

用新型的非关系数据库取代关系数据库

现在表示关系和非关系型数据库各有优缺点

彼此都无法互相取代

■ 关系数据库已经无法满足Web2.0的需求。主要表现在：

- (1) 无法满足海量数据的管理需求
- (2) 无法满足数据高并发的需求
- (3) 无法满足高可扩展性和高可用性的需求

■ NoSQL数据库数量众多，典型的NoSQL数据库通常包括键值数据库、列族数据库、文档数据库和图数据库。

如百度云数据库Redis使用键值数据库

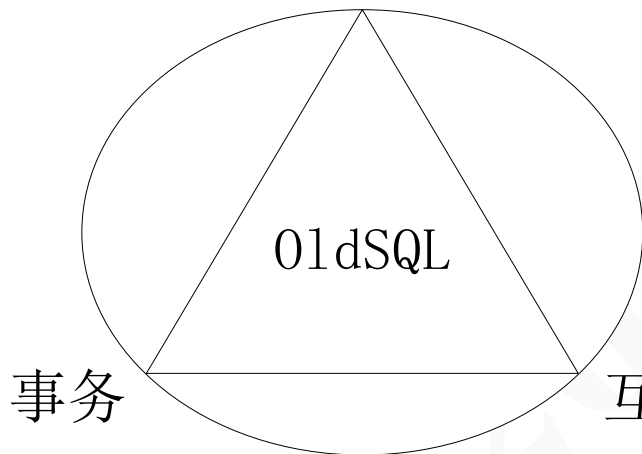


# 从NoSQL到NewSQL

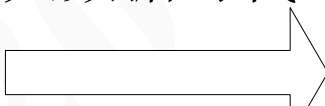
□ NoSQL数据库较好地满足了大数据时代的各种非结构化数据的存储需求，得到广泛的应用。

□ NewSQL数据库开始融合OldSQL和NoSQL的优点

一种架构支持多类应用  
(One Size Fits All)  
分析

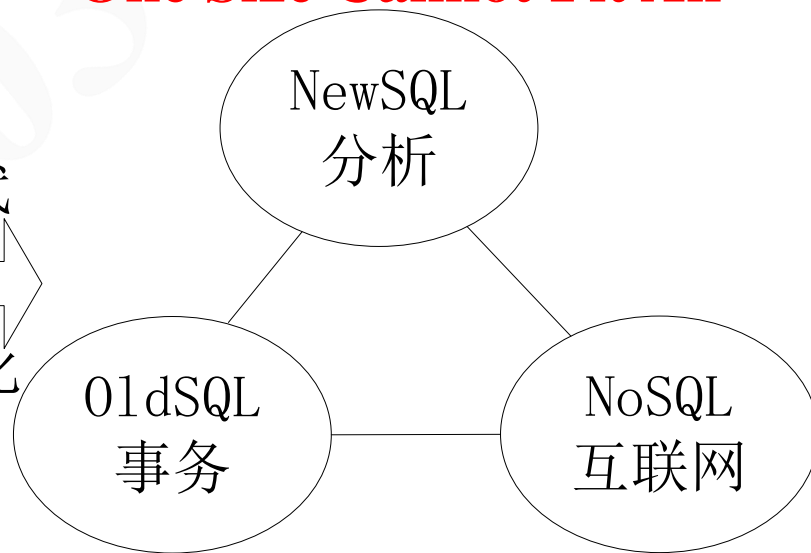


大数据时代



架构多元化

多架构支持多类应用  
**One Size Cannot Fit All**



## 大数据引发数据处理架构变革

## 1.1.2 数据管理技术的产生与发展

### DB-Engines 2019年12月份全球数据库排名

Rank Dec 2019	DBMS	Database Model	Score	
			Dec 2019	Nov 2019
1.	Oracle	Relational	1346.39	+10.33
2.	MySQL	Relational	1275.67	+9.38
3.	Microsoft SQL Server	Relational	1096.20	+14.29
4.	PostgreSQL	Relational	503.37	+12.30
5.	MongoDB	Document	421.12	+7.94

目前商品化的数据库管理系统以**关系数据库**为主导

## 1.1.2 数据管理技术的产生与发展

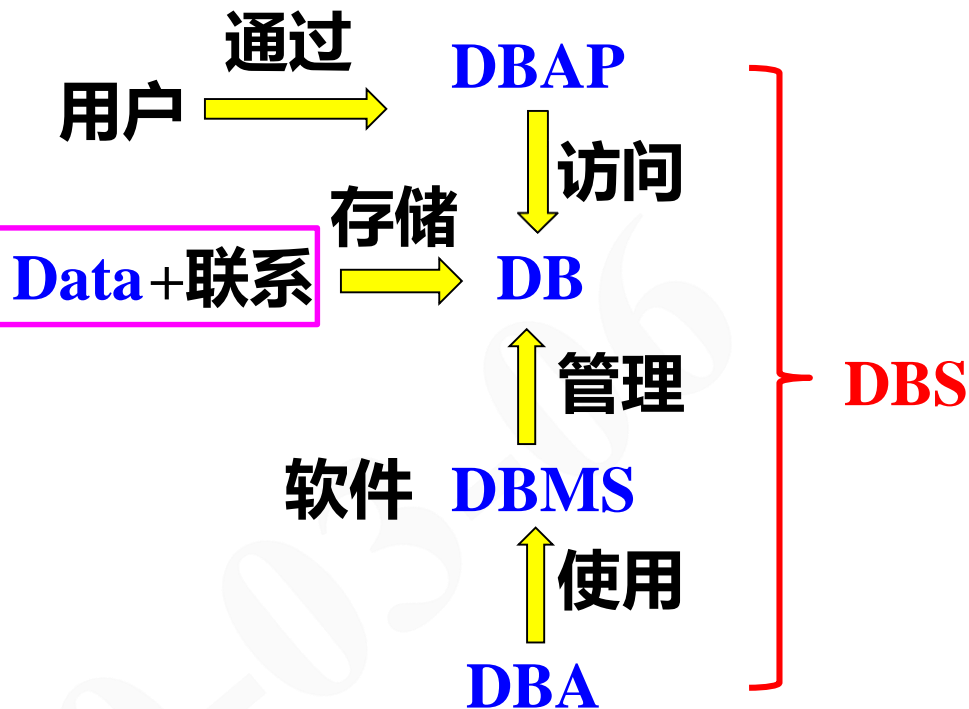
---

- **Oracle:** 世界第二大的计算机软件供应商，Oracle一直是数据库**市场占有率最高**的产品。
- **MySQL:** **小型**关系型数据库管理系统，也是Oracle公司的产品，是最流行的**开源**数据库。
- **Microsoft SQL Server:** 微软公司的产品，可以说是Windows上最好的数据库，已开始支持Linux。
- **PostgreSQL:** 功能最强大的**开源**数据库，国内应用逐步增多。
- **MongoDB:** 最好的**文档型数据库**，当前最成功的**NoSQL**数据库。弥补了关系型数据库的很多问题。

数据库是不是  
只存放数据

如何实现整体  
数据结构化

数据模型



数据管理技术的发展

- 人工管理
- 文件系统
- 数据库系统

无共享，冗余高，不独立，无结构

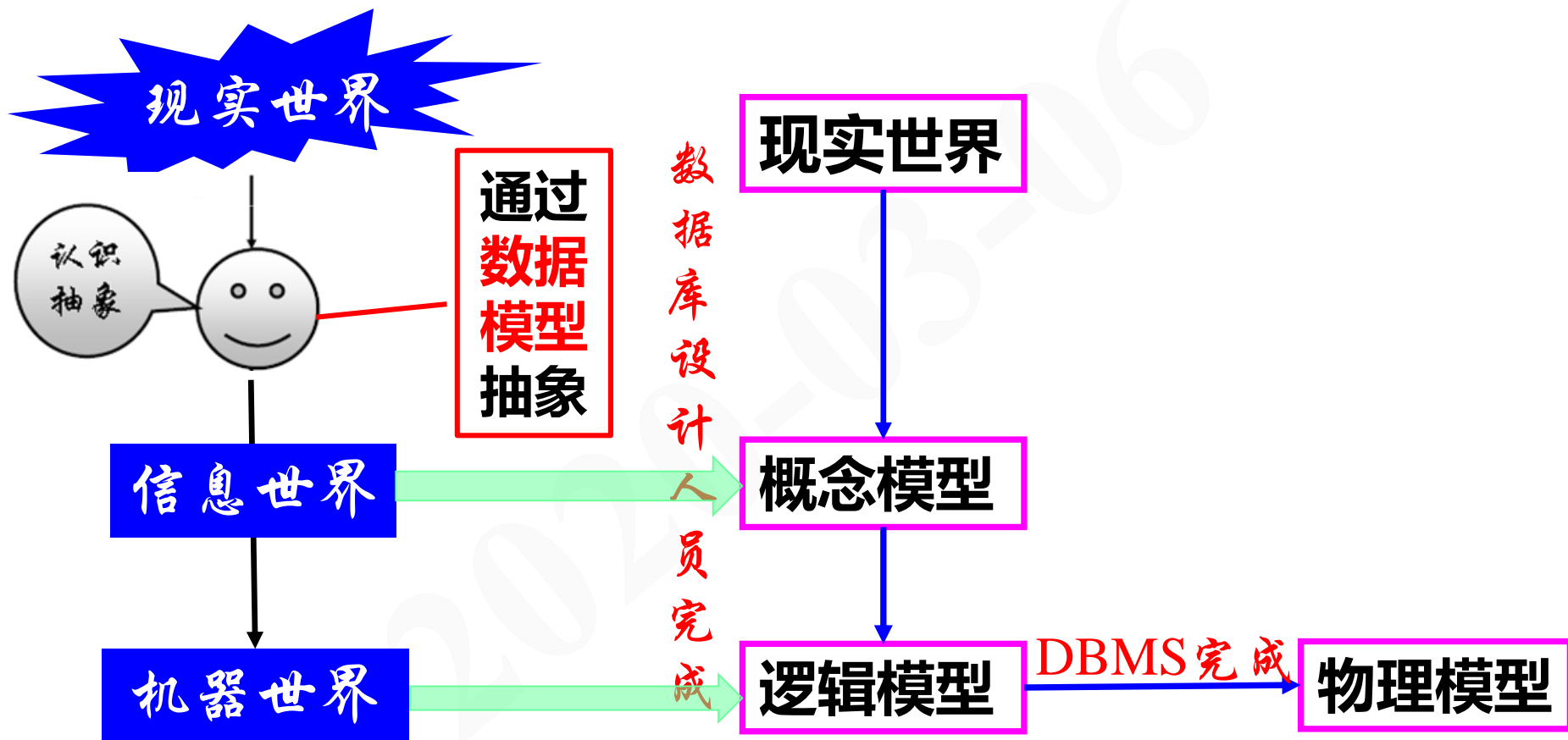
无共享，冗余高，物理独立性，记录内有结构，最小存取单位为记录。

本质区别

共享高，冗余低，物理和逻辑独立性，整体数据结构化，最小存取单位为数据项。

## 1.2 数据模型

□ 数据模型是对现实世界数据特征的抽象，数据库用**数据模型**来表示和处理现实世界中的数据。



现实世界中客观对象的抽象过程

## 1.2 数据模型

□ **数据模型是数据库系统的核心和基础**，各种DBMS软件都是基于某种数据模型或者支持某种数据模型的。

■ **数据模型分成两个不同的层次**

(1) 用于数据库设计的概念模型

- 较真实地模拟现实世界
- 易于理解
- 便于计算机实现

按**用户的观点**对数据和信息建模，与DBMS**无关**。

(2) 用于数据库实现的逻辑模型和物理模型

按**计算机系统的观点**对数据建模，与DBMS**有关**。

## 1.2.1 概念模型

---

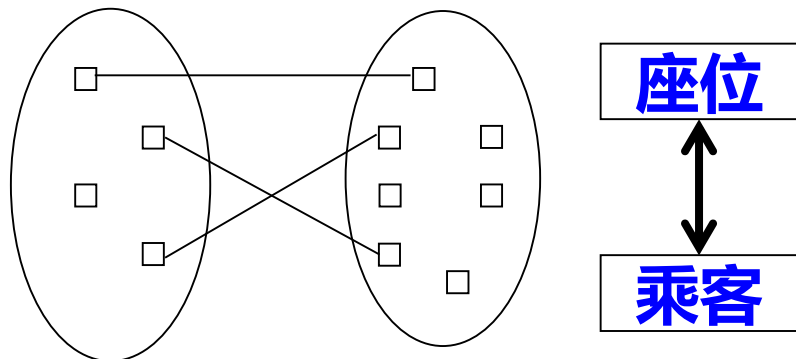
### 一、信息世界中的基本概念

- ★ **实体(Entity)**: 客观存在并可相互区别的事物。
- ★ **属性(Attribute)**: 实体所具有的某一特性。
- ★ **码、键(Key)**: 能唯一标识实体的属性集。
- ★ **实体集(Entity Set)**: 同型实体的集合。
- ★ **联系(Relationship)**: 现实世界中事物内部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。

**实体之间的联系有一对一(1:1)、一对多(1:n)和多对多(m:n)**

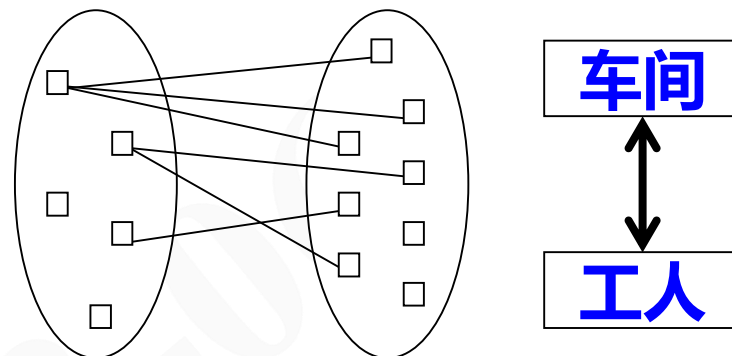
## 二、两个实体型之间的联系

实体集E1 实体集E2



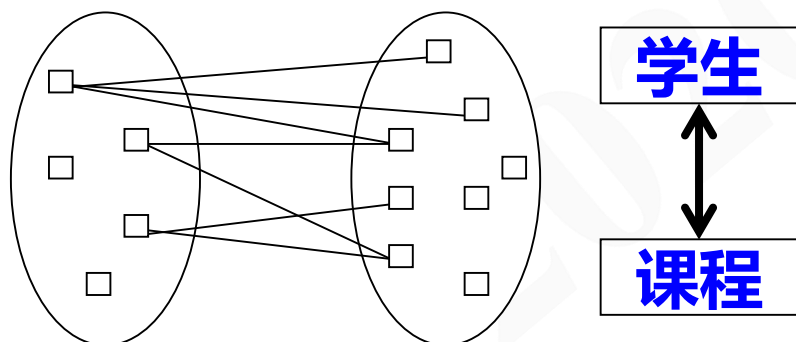
一对一联系

实体集E1 实体集E2



一对多联系

实体集E1 实体集E2



多对多联系

当实体较多时，  
如何表示





### 三、概念模型的一种表示方法

□ 实体 - 联系方法(ER方法)，也称ER模型，ER图。

■ 实体：用矩形表示。 

■ 属性：用椭圆形表示，以无向边将其与相应的实体连接起来。 

■ 联系：用菱形表示，用无向边与有关实体连接起来，在无向边旁标上联系类型。 

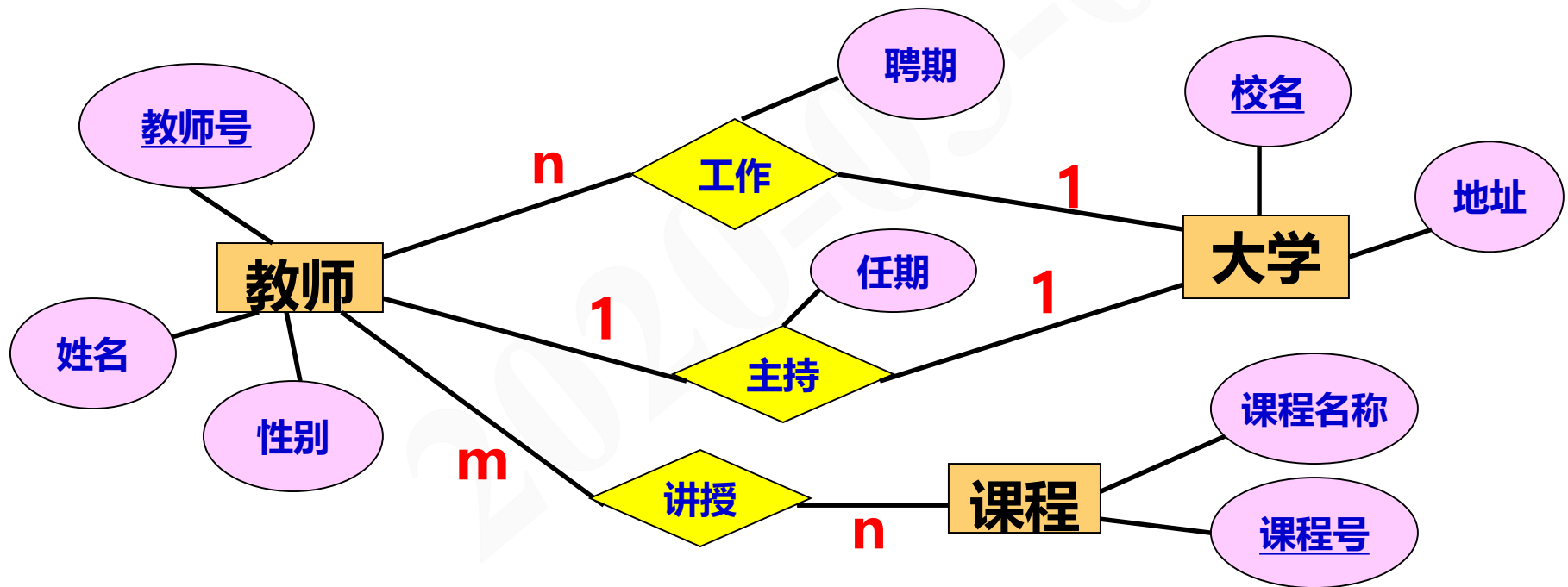
- 联系的属性：联系本身也是一种实体型，也可以有属性。联系的属性也要用无向边与该联系连接。

# ER图实例（大学、教师和课程三个实体）

**1: 1联系**：每所大学只有一个正校长，每个正校长只能主持一所大学的工作。

**1: n联系**：一个教师一段时间内只能在一所大学工作，一所大学可有许多教师。

**m: n联系**：一个教师可以上多门课程,每门课程可以有多个教师



## 四、实体联系模型实例

◆ 例1：为仓库管理设计一个ER模型。仓库主要管理零件的采购和供应等事项。仓库根据需要向外面供应商订购零件，而许多工程项目需要仓库提供零件。

1. 首先确定实体类型。

■ 本问题有四个实体类型：仓库，零件，供应商，工程项目。

2. 确定联系及联系的类型。

■ 仓库和零件是1:N联系

■ 零件和供应商之间是M:N联系

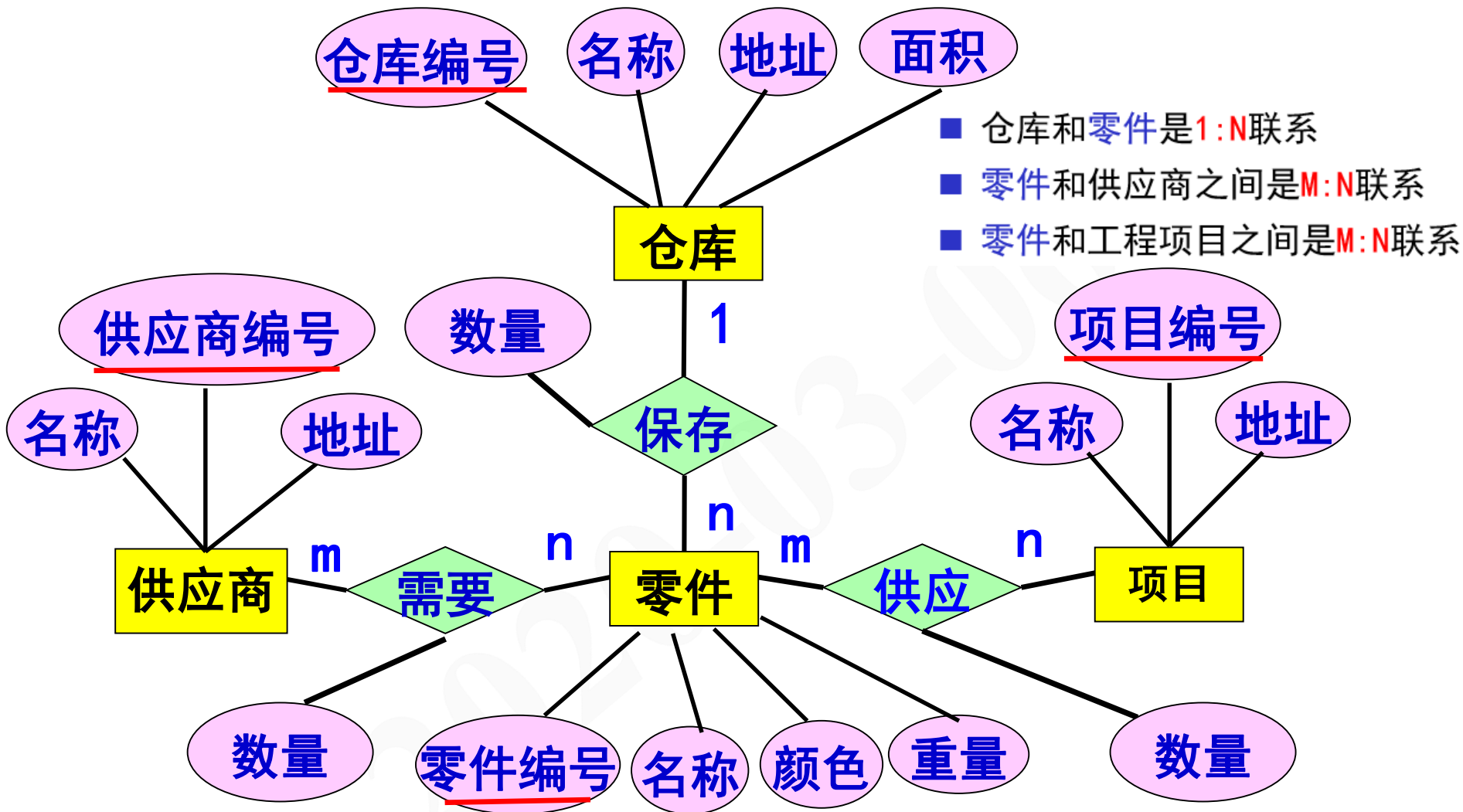
■ 零件和工程项目之间是M:N联系

3. 把实体类型和联系类型组合成ER图。

4. 确定实体类型和联系类型的属性。

5. 确定实体类型的码，在ER图中属于码的属性下画一条横线。

# 仓库管理的ER图



概念模型如何转换为逻辑模型，进入机器世界



## 1.2.2 (逻辑) 数据模型的组成要素

---

### ◆ 数据结构：

描述系统的**静态特性**。数据结构不仅要描述数据库的组成对象本身还要描述对象之间的联系。

### ◆ 数据操作：

描述系统的**动态特性**包括操作及有关的操作规则。

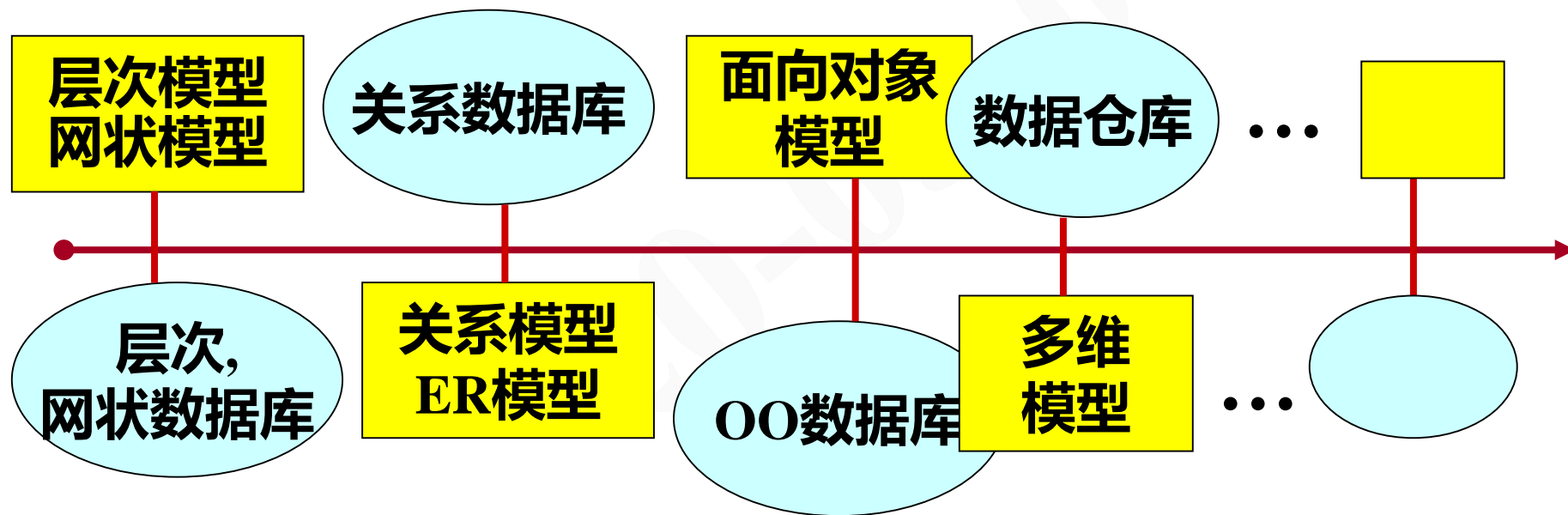
### ◆ 数据的完整性约束条件：

是一组完整性规则的集合。完整性规则是数据模型中数据及其联系所具有的约束规则，用来限定数据库状态以及状态的变化，以保证数据的正确。

## 1.2.3 最常用的逻辑数据模型

□ 数据模型就是对现实世界的模拟、描述或表示。

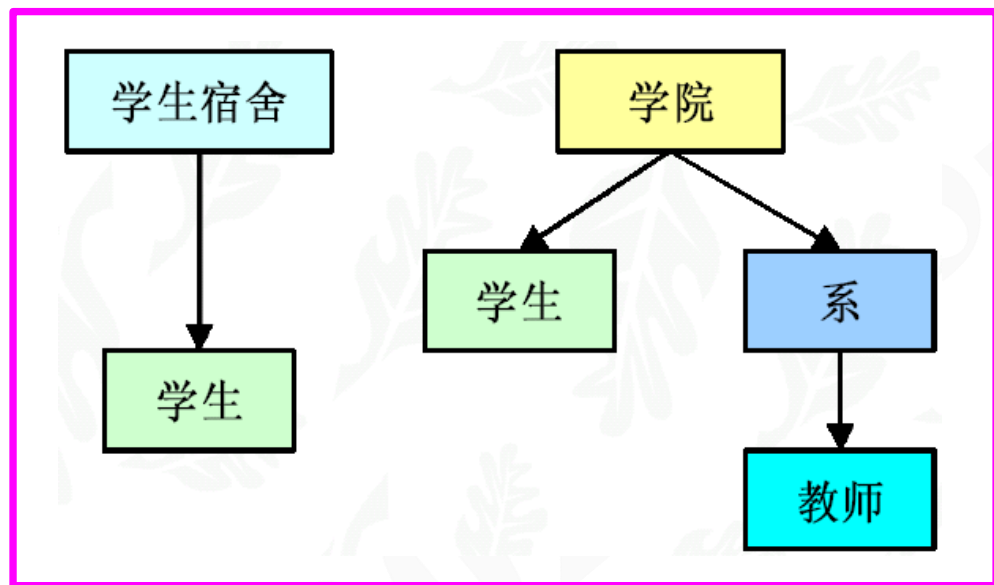
■ 数据库的发展是由数据模型的进步主导的



# 一、层次模型

□ 用**树形**结构表示各类实体以及实体之间的联系

- 只有一个结点无双亲——根结点
- 根以外的其它结点有且仅有一个双亲



优点：

- ★ 数据结构简单清晰
- ★ 查询效率高

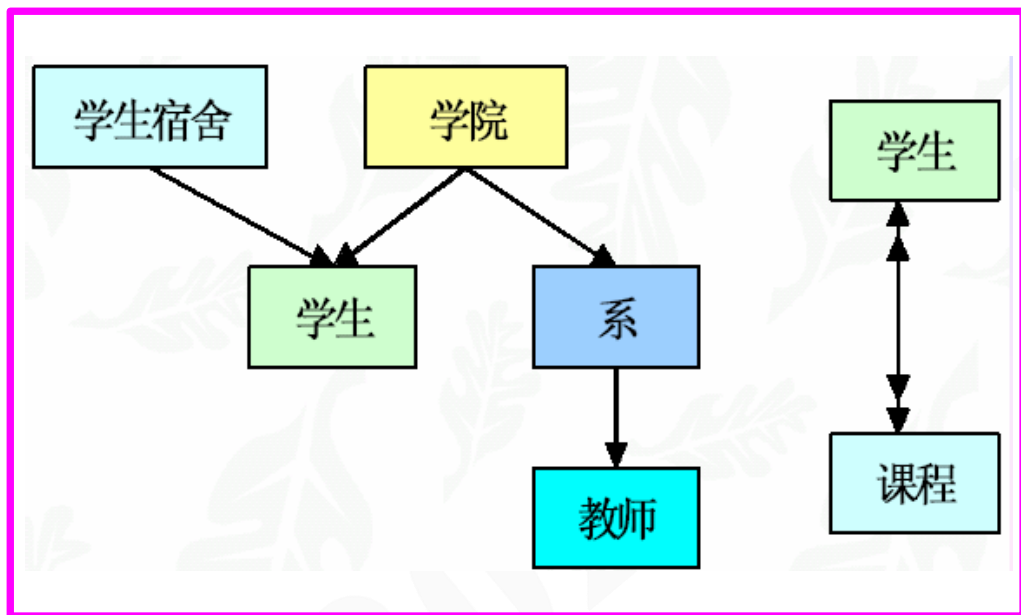
缺点：

- ★ 查询子女结点必须通过双亲结点，插入和删除操作限制多
- ★ 难以描述现实世界中的复杂联系。

## 二、 网状模型

### □ 去掉了层次模型的两个限制:

- 可以有一个以上的结点无双亲
- 至少有一个结点有多于一个的双亲



缺点:

- ★ 结构比较复杂
- ★ 语言复杂

优点:

- ★ 能更为直接地描述现实世界
- ★ 性能良好, 存取效率较高



### 三、关系模型

□ 关系模型由**一组关系**组成，每个关系的数据结构是一张规范化的**二维表**，由行和列组成。

**学生(学号, 姓名, 性别, 年龄, 系)**

学生关系

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	系 Sdept
200215121	李晨	男	20	CS
200215122	刘晨	女	19	CS
200215123	王敏	女	20	MA

关系  
模式

表 - dbo.Student 摘要					
Sno	Sname	Ssex	Sage	Sdept	
200215121	李勇	男	20	CS	
200215122	刘晨	女	19	CS	
200215123	王敏	女	20	MA	

Sql Server中存储的学生关系表

概念模型如何转换为关系模型



### 三、关系模型

实体以及实体间的联系在关系模型中都用关系来表示。

#### □ 实体及实体间的联系的表示方法

■ 实体型：直接用关系（表）表示。

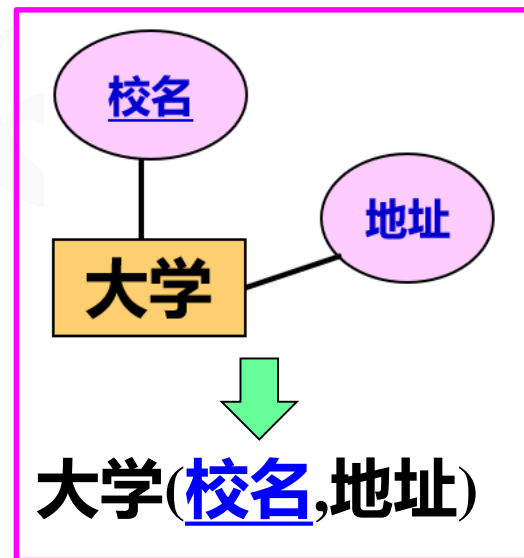
■ 属性：用属性名表示。

■ 一对一和一对多的联系：

○ 隐含在实体对应的关系中(一般选用此方式)

○ 转换为一个独立的关系模式

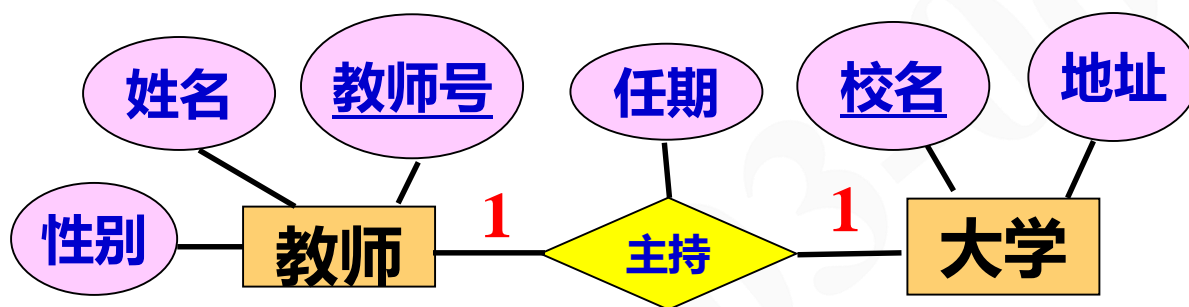
■ 多对多联系：转换为一个独立的关系模式



# 概念模型转换为关系模型

□ 1:1的联系：一般隐含在实体对应的关系中。

■ 在两个实体类型转换成的两个关系模式中任意一个关系模式属性中加入另一个关系模式的码和联系类型的属性。



大学(校名,地址,**教师号**,任期)

教师(教师号,姓名,性别)

或

大学(校名,地址)

教师(教师号,姓名,性别,**校名**,任期)

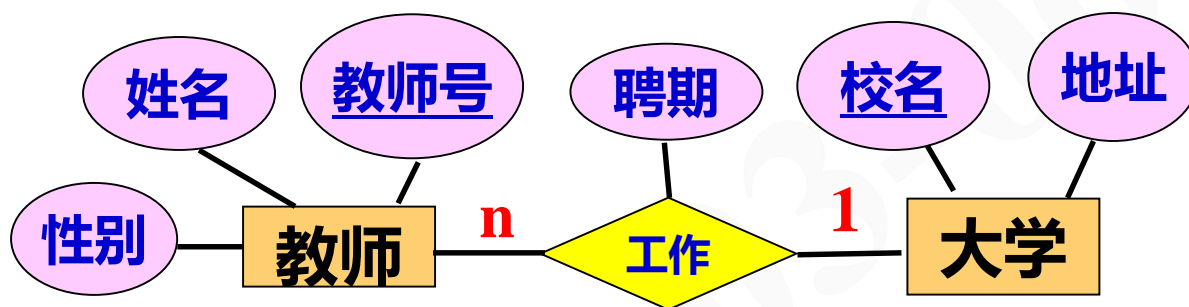
1:1的联系也可以转换为一个独立的关系模式：

主持(校名,教师号,任期) 或 主持(教师号,校名,任期)

# 概念模型转换为关系模型

□ 1:n的联系：一般隐含在实体对应的关系中。

■ 在N端实体类型转换成的关系模式中加入1端实体类型的码和联系类型的属性。



大学(校名,地址)

教师(教师号,姓名,性别,校名,聘期)

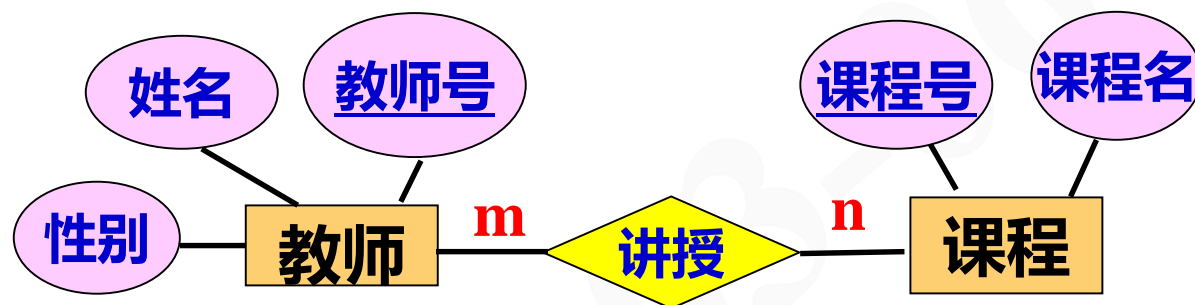
1:n的联系也可以转换为一个独立的关系模式

聘用 (教师号, 校名, 聘期)

# 概念模型转换为关系模型

□ m:n的联系：转换为一个独立的关系模式

■ 将联系转换成新的关系模式，其属性为两端实体类型的码加上联系类型的属性，而码为两端实体码的组合。



教师 (教师号, 姓名, 性别)

选课 (教师号, 课程号)

课程 (课程号, 课程名)

思考：关系模型中实体之间的联系如何实现



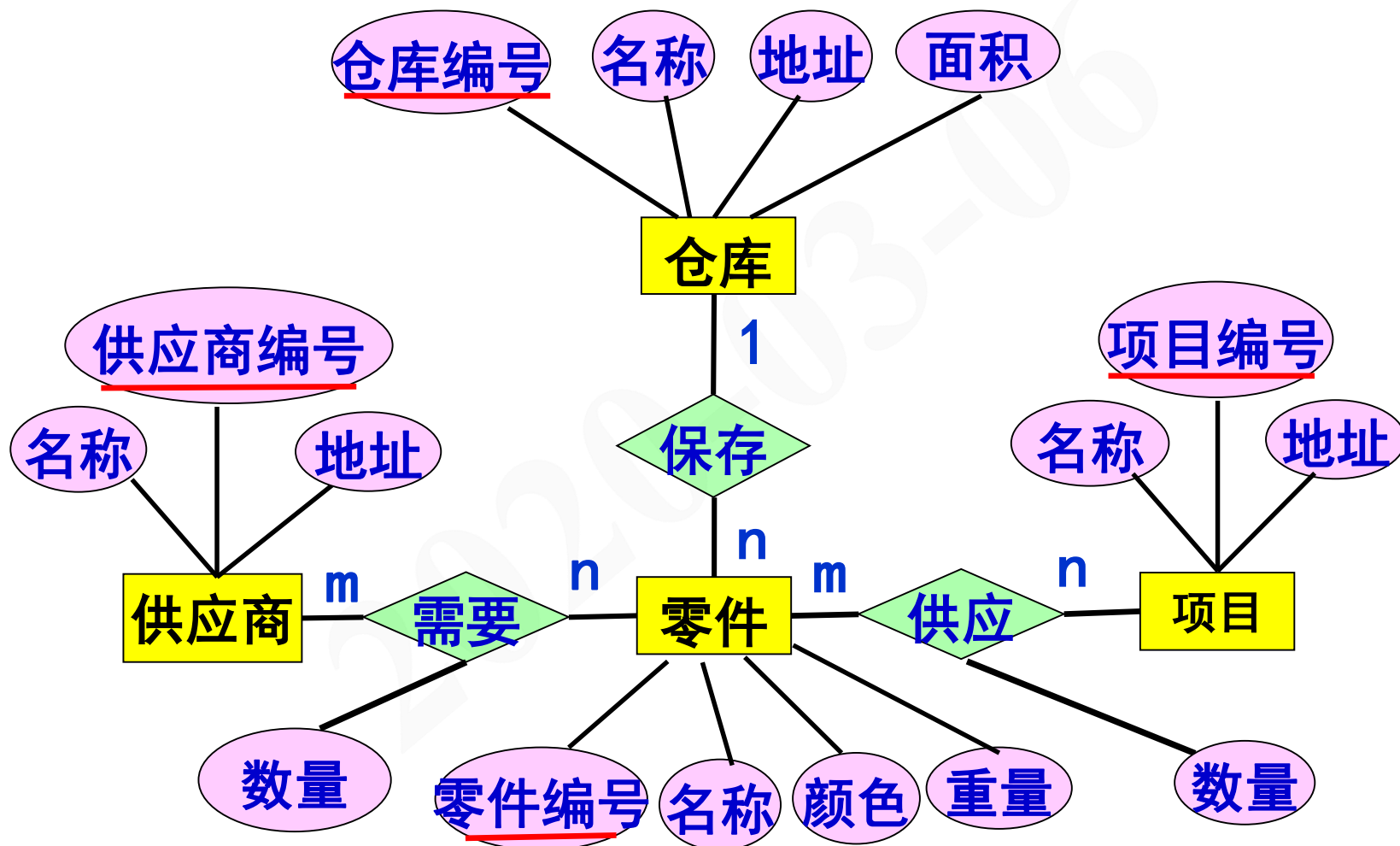
增加新的关系



实体之间的联系通过公共属性来实现

### 三、关系模型

◆ 例1：仓库管理问题。仓库管理零件的采购和供应等事项。仓库向供应商订购零件，工程项目需要仓库提供零件。



# 三、关系模型

## 仓库管理的ER图 → 关系模式 (几个关系模式?)

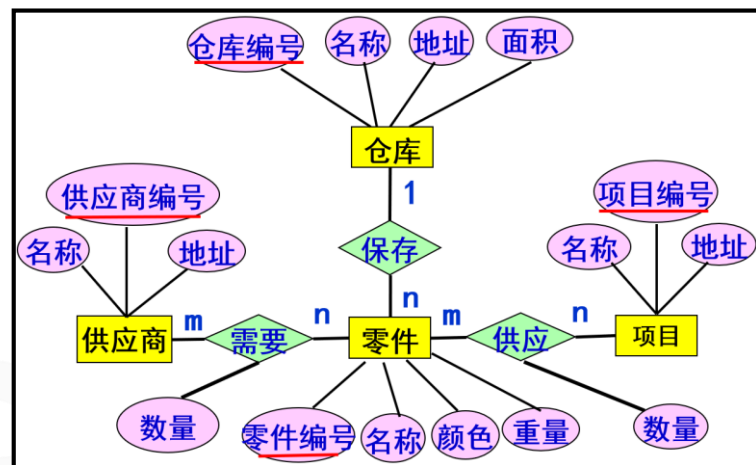
□ 4个实体转换为4个关系模式:

仓库 (仓库编号, 名称, 地址, 面积)

项目 (项目编号, 名称, 地址)

零件 (零件编号, 名称, 颜色, 重量)

供应商 (供应商编号, 名称, 地址)



□ 2个M:N的联系转换为2个新的关系模式:

项目零件 (项目编号, 零件编号, 数量)

供应商零件 (供应商编号, 零件编号, 数量)

转换结果:  
6个关系模式  
(红色关系模式)

□ 1个1:N的联系可隐含在N端实体关系模式中:

零件 (零件编号, 名称, 颜色, 重量, 仓库编号, 数量)

# 三、关系模型

---

## 优点:

- ★ 建立在严格的数学概念基础上;
- ★ 概念单一, 实体和联系都用关系表示, 数据操作的结果也为关系;
- ★ 数据的存取路径对用户是透明的。

## 缺点:

- ★ 存取路径透明使得数据存取的效率不如非关系模型。
- ★ 为提高性能, 必须对用户的查询请求进行优化, 增加了开发数据库管理系统的难度。



**组成关系数据模型的三大要素是（ D ）**

- A.关系数据语言、关系操作集合和关系数据控制**
- B.关系数据结构、关系数据定义和关系完整性约束**
- C.关系数据定义、关系数据操纵和关系数据控制**
- D.关系数据结构、关系操作集合和关系完整性约束**

数据库类型是根据( C )划分的。

A.文件形式

B.记录形式

C.数据模型

D.存取数据的方法

# 1.3 数据库系统结构

□ **数据库的数据抽象：对数据进行计算机化的管理**

- 如管理学生(20130311)的数据库课程成绩，以**关系**（二维表）形式组织数据，使其**结构化**——**数据模型**

确定数据模型后，如何设计具体的数据结构



- 利用 **“数据模型”** 工具完成对具体系统的数据抽取、组织、描述，得到的具体表达结果就是 **“数据模式”**

## 数据模式

学号	课程名称	成绩
20130311	数据库	80
20130322	程序设计	90

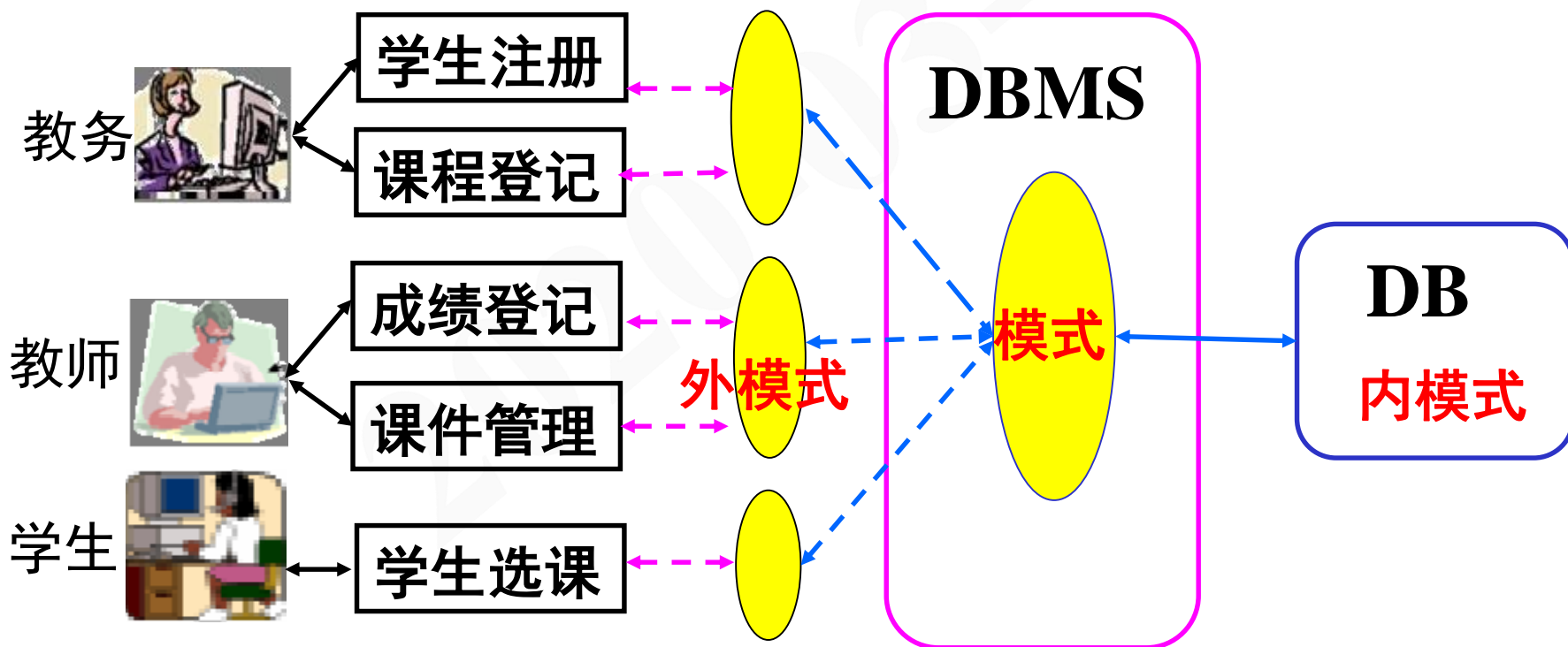
当应用比较复杂时，  
如何抽象数据模式



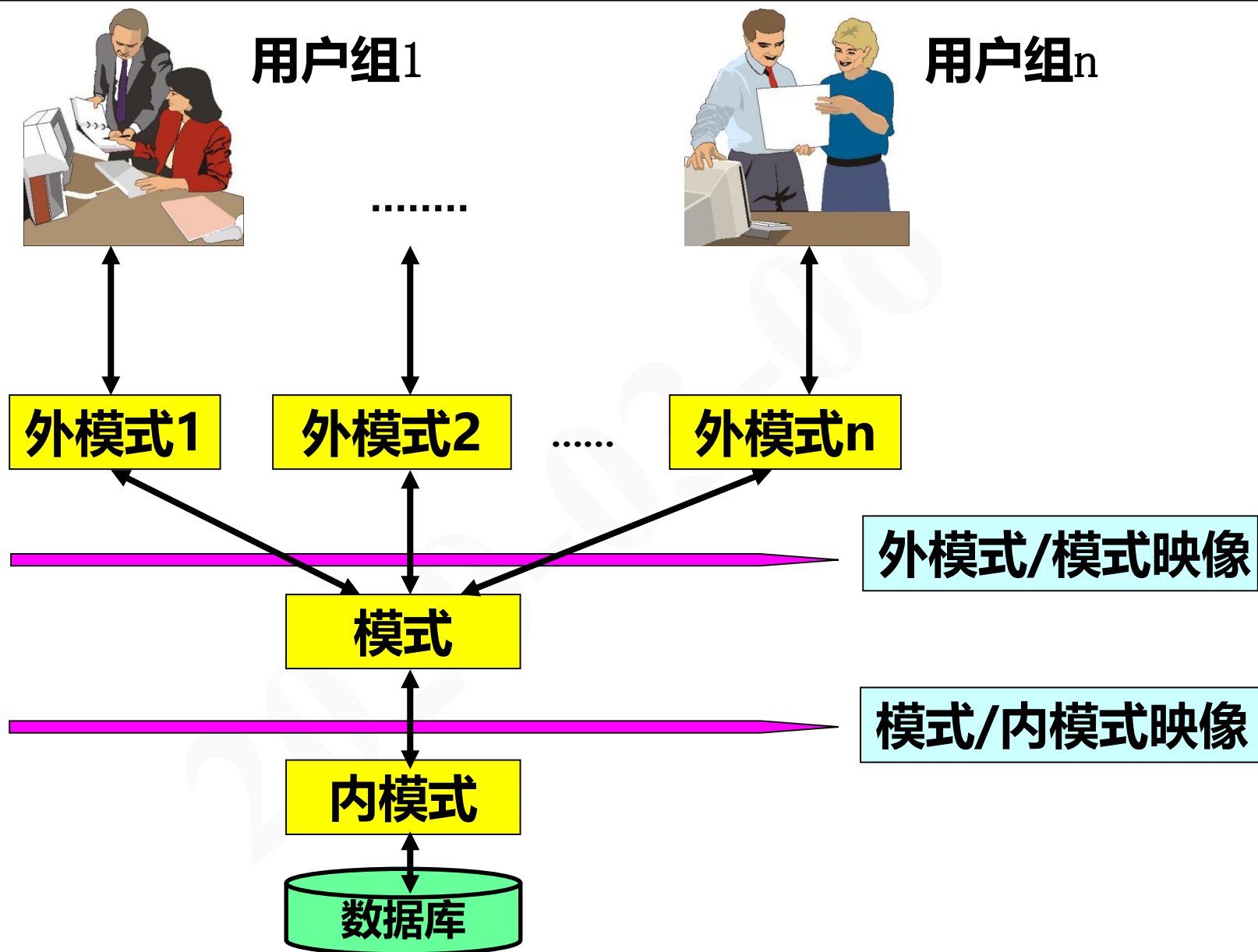
# 数据库系统的分层抽象

□ DBMS管理数据的**三个层次** **对数据的结构抽象** → **三级模式**

- 某一用户可看到与处理的数据，全局数据中的一部分 **外模式**
- 从全局角度理解/管理的数据 **模式**
- 存储在介质上的数据 **内模式**



# 一、数据库三级模式结构



# 1、模式

## □ 模式（逻辑模式）

是数据库中**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**。
- 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、开发工具及程序设计语言无关

## 模式

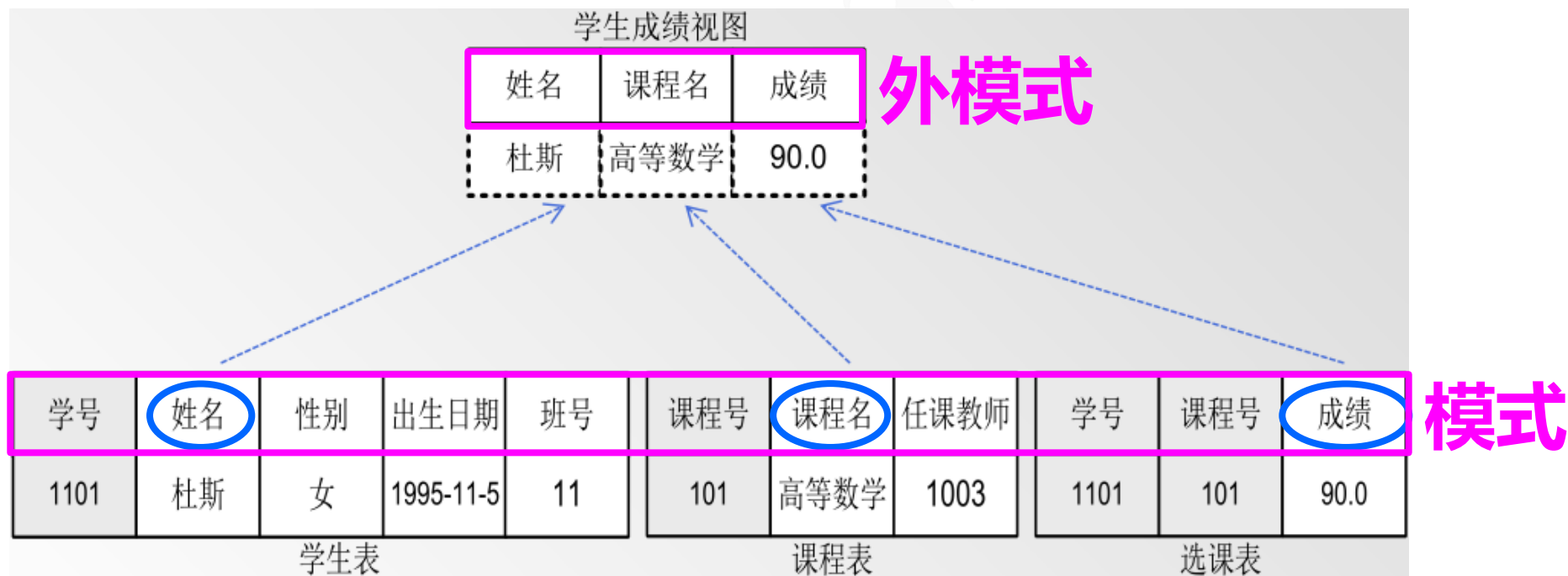
学号	姓名	性别	出生日期	班号	课程号	课程名	任课教师	学号	课程号	成绩
1101	杜斯	女	1995-11-5	11	101	高等数学	1003	1101	101	90.0
学生表					课程表			选课表		

## 2、外模式

### □ 外模式（子模式、用户模式）

#### 局部数据的逻辑结构和特征的描述

- 外模式通常是模式的子集
- 一个数据库可以有**多个外模式**



### 3、内模式

#### □ 内模式（存储模式）

- 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；
- 一个数据库只能有一个**内模式**。

#### □ 三级模式是对数据的三个抽象级别

三级模式之间如何联系和转换



#### □ 通过**二级映像**实现三个抽象层次的联系和转换

三级模式和二级映像的作用

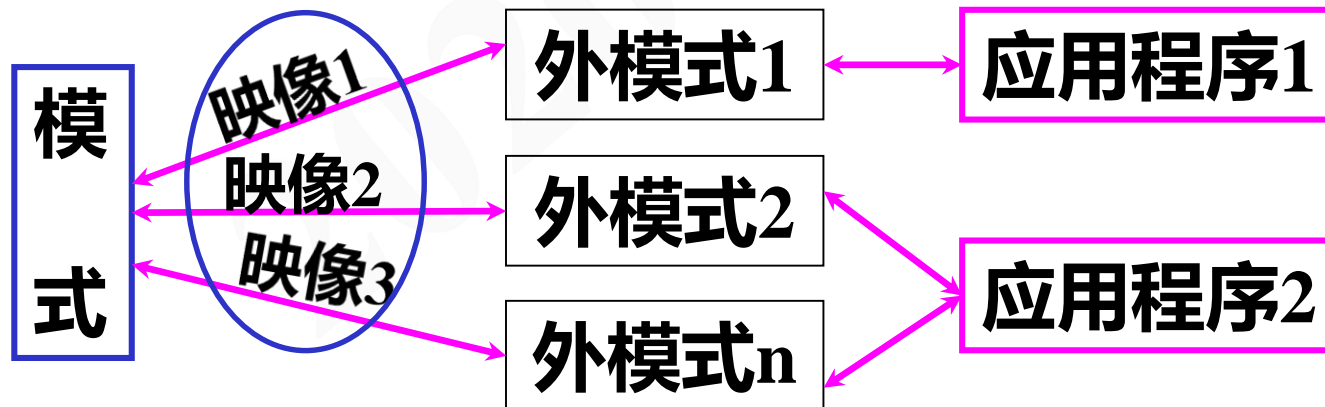


**保证数据独立性：数据独立于应用程序**



# 1、外模式/模式映像

- 定义外模式与模式之间的对应关系;
- 每一个外模式都对应一个外模式 / 模式映象;
- 保证了数据的**逻辑独立性**。
  - **模式改变**时, **修改**相关**外模式/模式映象**, 使**外模式不变**, 则应用程序不必修改, 保证数据与程序的逻辑独立性。



# 1、外模式/模式映像

学号	姓名	性别	出生日期	班号	课程号	课程名	任课教师	学号	课程号	成绩
1101	杜斯	女	1995-11-5	11	101	高等数学	1003	1101	101	90.0

学生表                      课程表                      选课表

原有  
模式

学号	姓名	性别	出生日期	班号	学号	课程号	课程名	任课教师	成绩
1101	杜斯	女	1995-11-5	11	1101	101	高等数学	1003	90.0

学生表                      选课表

模式改变

学生成绩视图		
姓名	课程名	成绩
杜斯	高等数学	90.0

外模式不变

模式改变，外模式不变，则应用程序不必修改。

## 2、模式/内模式映像

- 定义数据全局逻辑结构与存储结构之间的对应关系;
- 数据库中模式 / 内模式映象是**唯一**的。
- 保证了数据的**物理独立性**
  - **存储结构改变**时, **修改模式 / 内模式映象**, 使**模式不变**, 应用程序不受影响, 保证数据与程序的物理独立性。



数据库的二级映像功能使数据的定义和描述从应用程序中分离出去, **数据与程序之间具有了独立性。**

在数据库系统中，当数据库的模式改变时，用户程序可以不做改变，这是数据的（ C ）。

A.位置独立性

B.存储独立性

C.逻辑独立性

D.物理独立性

要保证数据库的数据独立性，需要修改的是（ C ）。

A.模式与外模式

B.模式与内模式

C.三级模式之间的两层映射

D.三层模式

**数据库系统的数据独立性是指（ D ）。**

**A.数据与程序独立存放**

**B.不同的数据被存放在不同的文件中**

**C.不同的数据只能被对应的应用程序所使用**

**D.三种说法都不对**

**数据库中，数据的物理独立性是指（ C ）**

**A.数据库与DBMS的相互独立**

**B.用户程序与DBMS的相互独立**

**C.用户的应用程序与存储在磁盘上数据库中的数据是相互独立的**

**D.应用程序与数据库中数据的逻辑结构相互独立**