



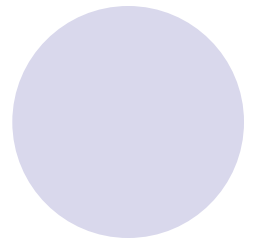
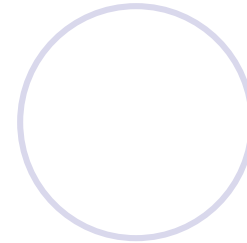
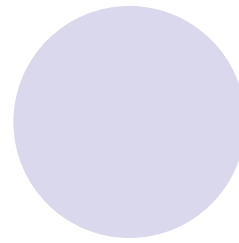
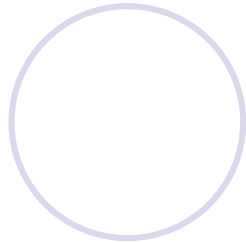
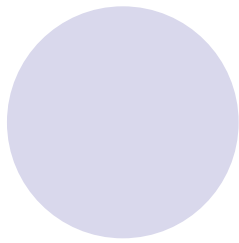
数据库系统

Database System

主讲：张仲楠 教授

Email: zhongnan_zhang@xmu.edu.cn

Office: 海韵A416



数据库系统

Database System

第一章 绪论

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

1.5 小结

数据库的地位



- 数据库技术产生于六十年代(63年, **C.W.Bachman** 设计开发了**IDS**), 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**。

大数据引发数据库架构大变革

一个数据库**操作序列**，
这些操作要么全做，
要么全不做

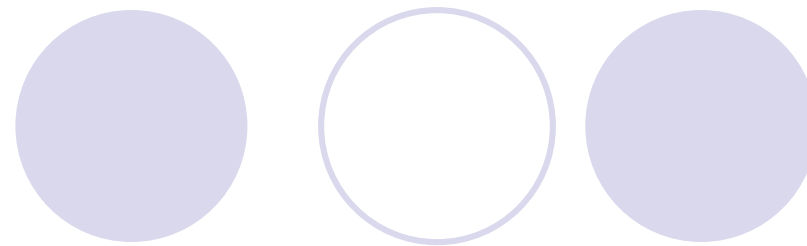
- 传统数据库的基本架构

- 以**事务(Transaction)处理**为主要应用设计的
- 一种架构支持所有应用

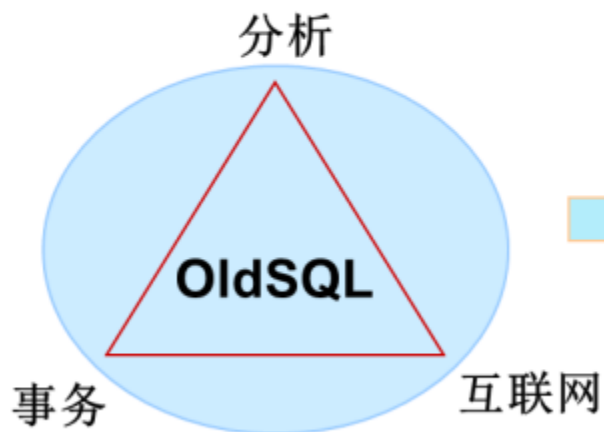
- 在大数据时代，对于**分析类应用**的需求不断增加，这种变化导致

- 一种架构难以完全满足大数据的需求
- 数据库行业出现三个互为补充的三大阵营，
OldSQL、NewSQL和NoSQL。

数据库架构变革



一种架构支持多类应用

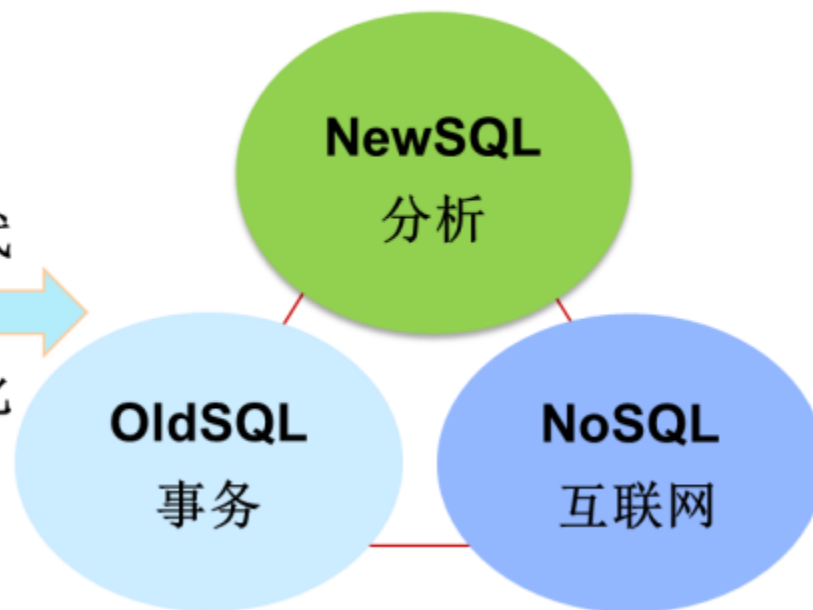


大数据时代

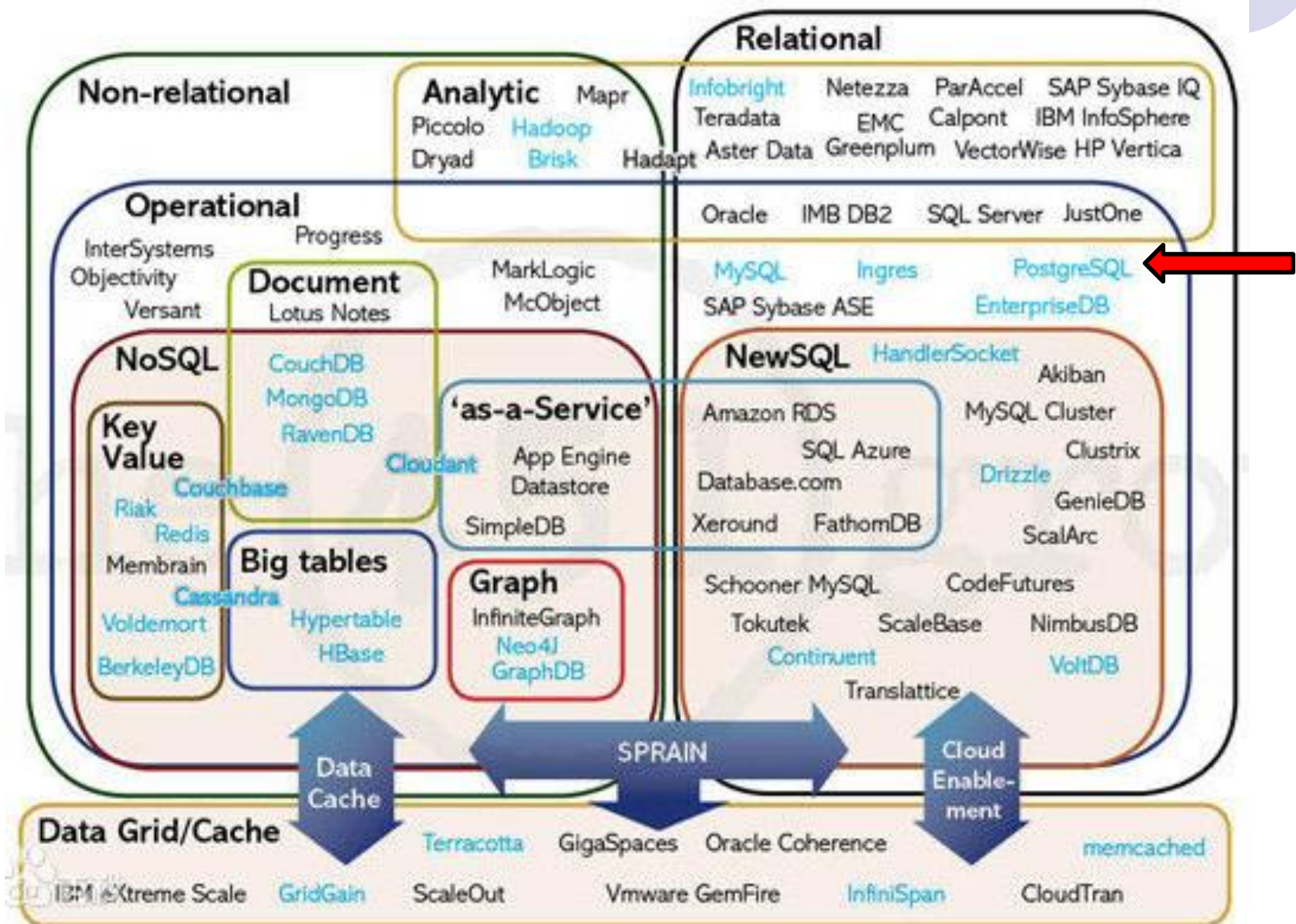


架构多元化

多种架构支持多类应用



数据库架构变革



数据库流行排行榜



INGRES
1973
Postgres
1986

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	1247.52	+2.35	-9.31
2.	2.	2.	MySQL +	Relational, Multi-model	1195.45	-16.51	-19.23
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model	929.09	+9.70	-19.96
4.	4.	4.	PostgreSQL +	Relational, Multi-model	616.50	+1.65	+7.12
5.	5.	5.	MongoDB +	Document, Multi-model	452.77	-2.42	-35.88
6.	6.	6.	Redis +	Key-value, Multi-model	173.83	-3.72	-1.96
7.	7.	7.	IBM Db2	Relational, Multi-model	142.97	-0.60	-19.91
8.	8.	8.	Elasticsearch	Search engine, Multi-model	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

根据数据库管理系统的流行程度进行排名（每月更新）

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

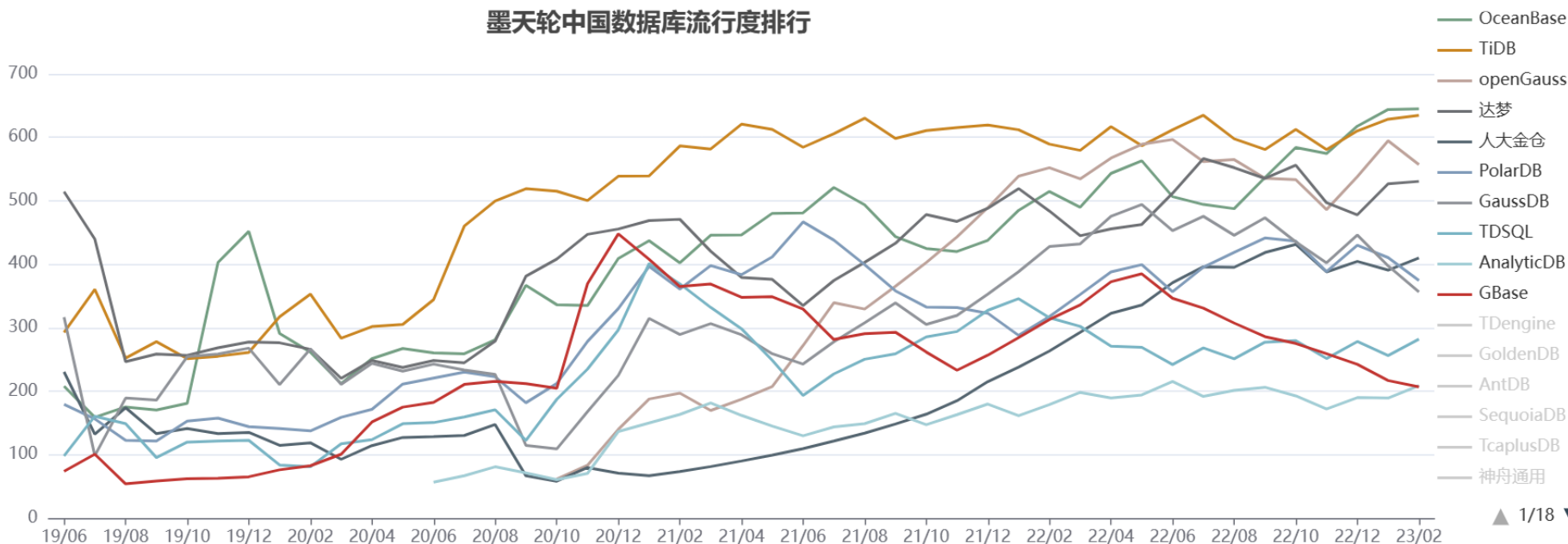
数据库流行排行榜

OCEANBASE



openGauss

墨天轮中国数据库流行度排行

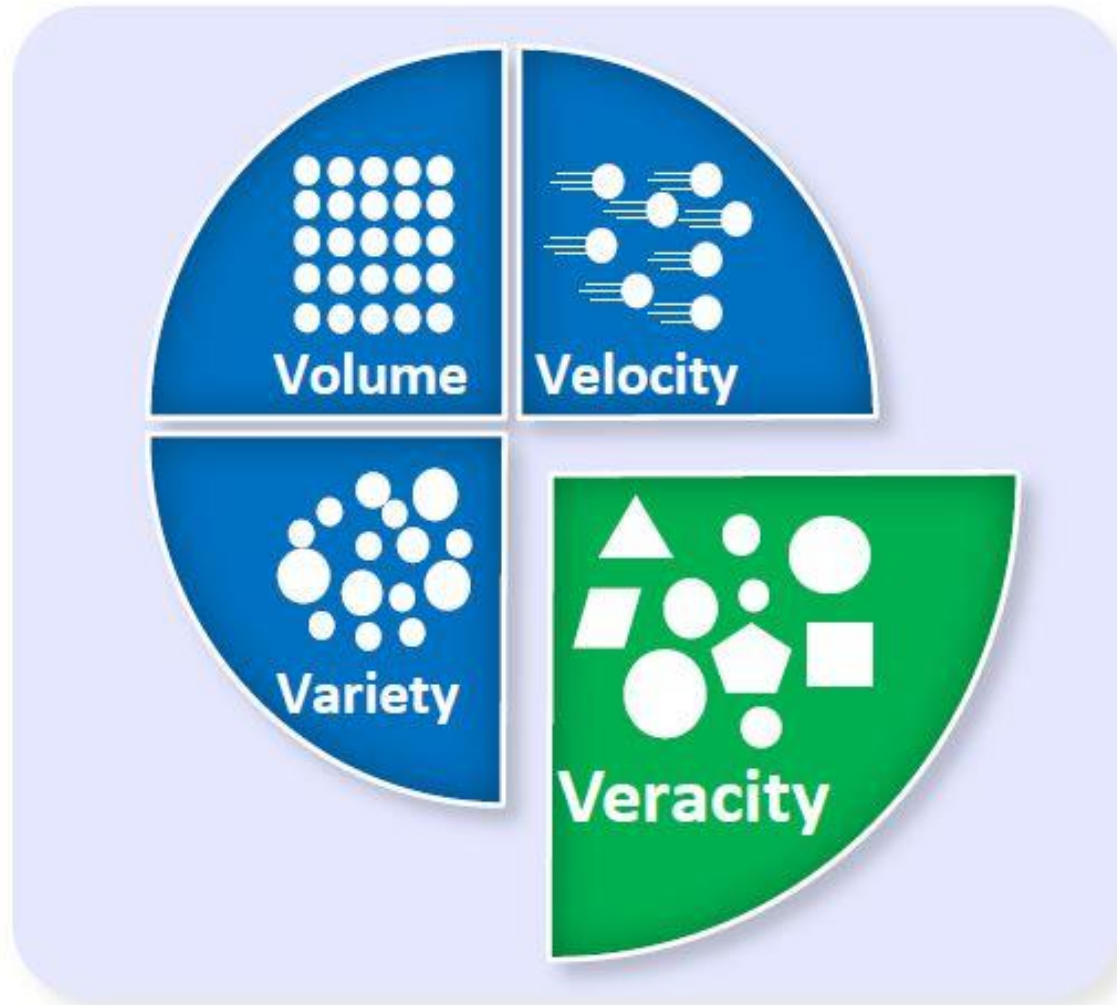


来源: <https://www.modb.pro/dbRank>

什么是大数据？

- 大数据是如此**庞大**和**复杂**的数据集，以至于**传统的**数据处理应用软件不足以处理它们。
- **2012年, Gartner 重新定义**
 - “大数据是**大容量**、**高速率**和**类型多样**的信息资产，需要新的处理形式来实现增强的决策制定、洞察发现和流程优化。”

IBM的四个“V”



IBM的四个“V”: Volume(巨量)

40 ZETTABYTES

[43 TRILLION GIGABYTES]

of data will be created by 2020, an increase of 300 times from 2005

2020

2005

It's estimated that

2.5 QUINTILLION BYTES

[2.3 BILLION GIGABYTES]

of data are created each day

Volume
SCALE OF DATA

GB TB PB EB ZB

6 BILLION PEOPLE

have cell phones

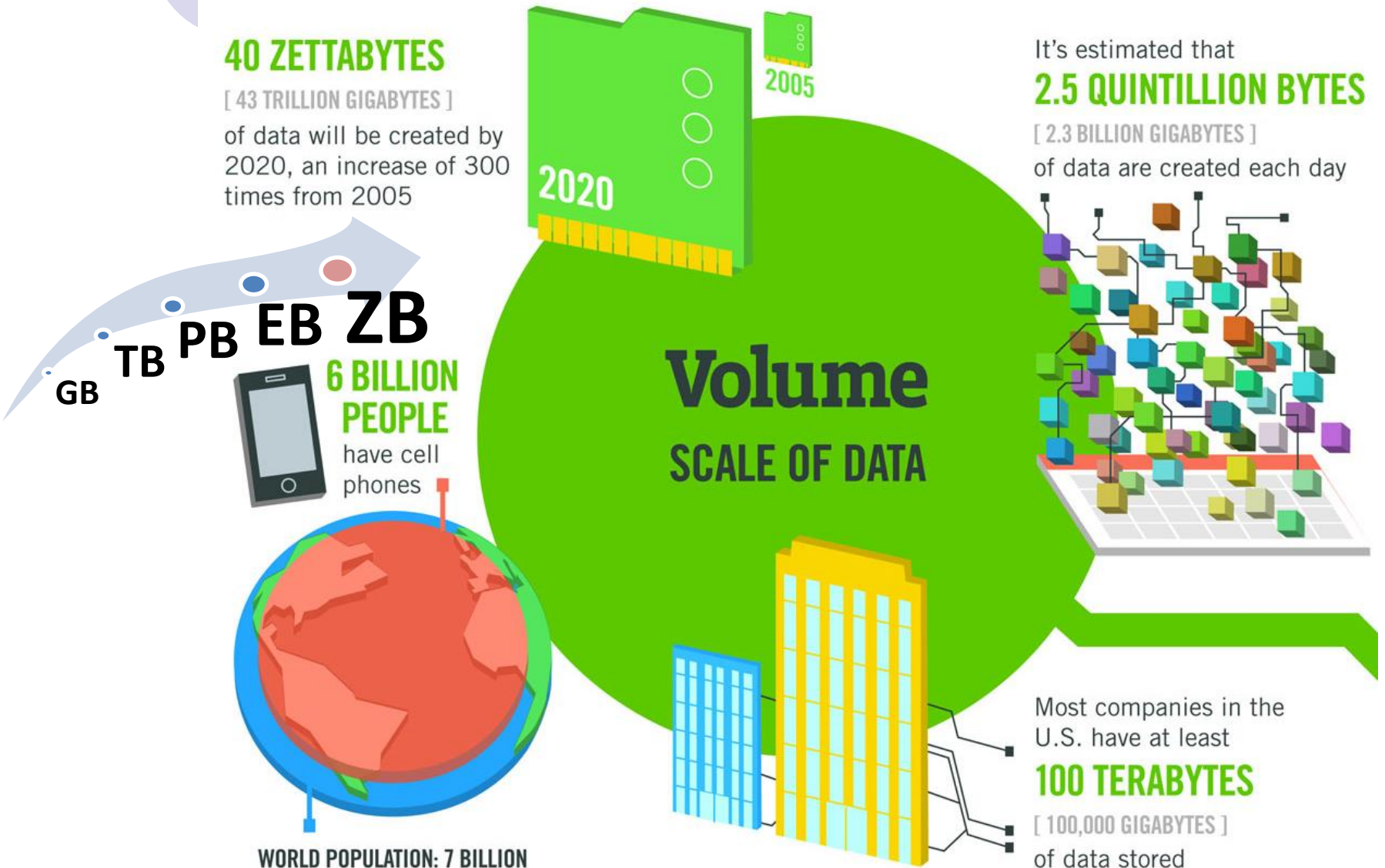
WORLD POPULATION: 7 BILLION

Most companies in the U.S. have at least

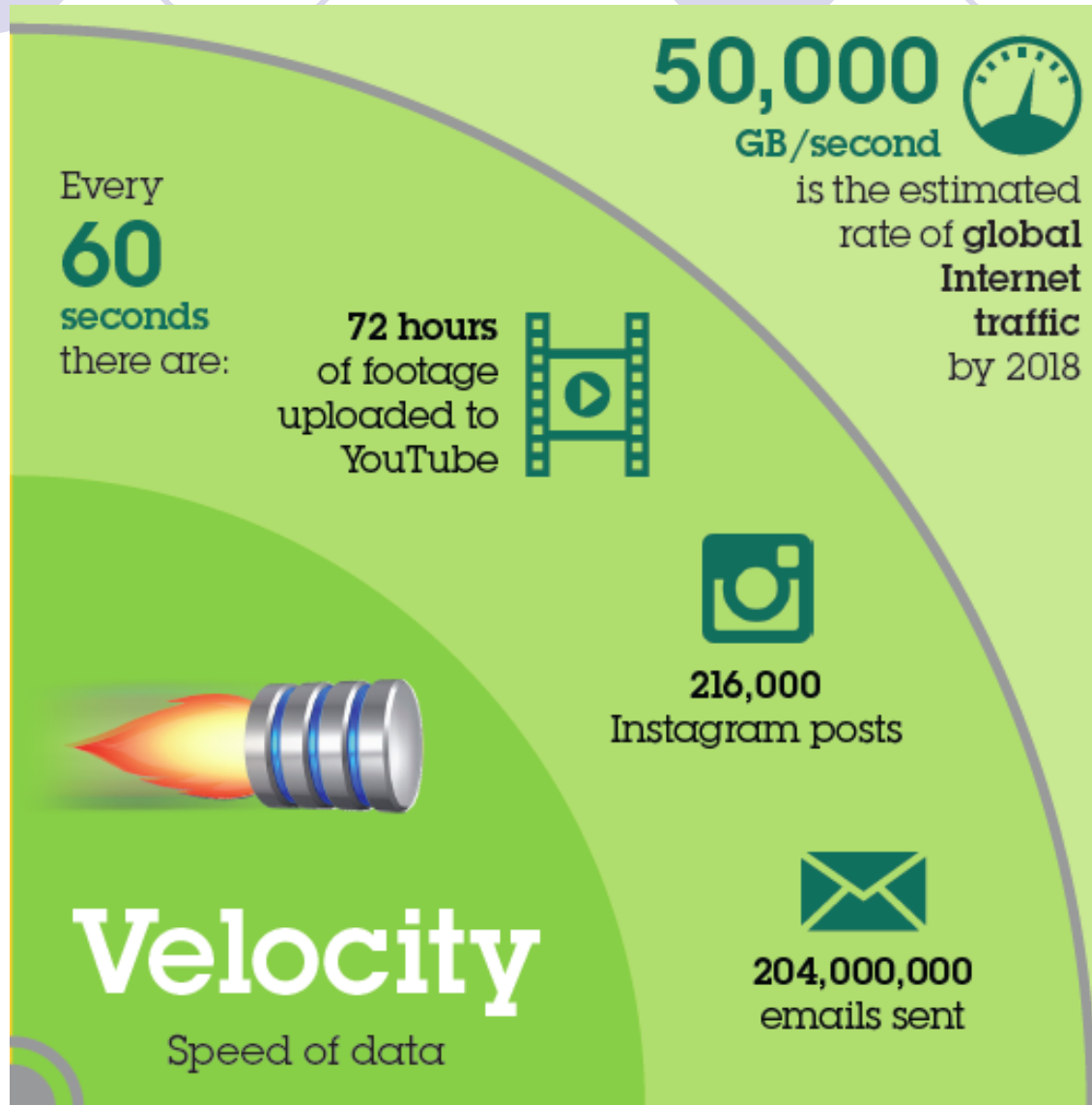
100 TERABYTES

[100,000 GIGABYTES]

of data stored



IBM的四个“V”: Velocity(快速)



IBM的四个“V”: Variety(多样)

As of 2011, the global size of data in healthcare was estimated to be

150 EXABYTES

[161 BILLION GIGABYTES]



By 2014, it's anticipated there will be

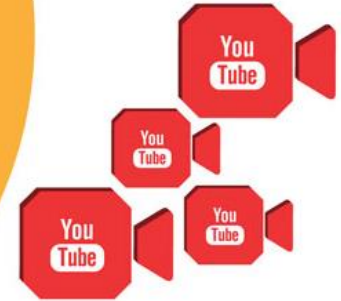
**420 MILLION
WEARABLE, WIRELESS
HEALTH MONITORS**



Variety
DIFFERENT
FORMS OF DATA

**4 BILLION+
HOURS OF VIDEO**

are watched on
YouTube each month



**30 BILLION
PIECES OF CONTENT**

are shared on Facebook
every month



400 MILLION TWEETS

are sent per day by about 200
million monthly active users



IBM的四个“V”: Veracity(真实性)

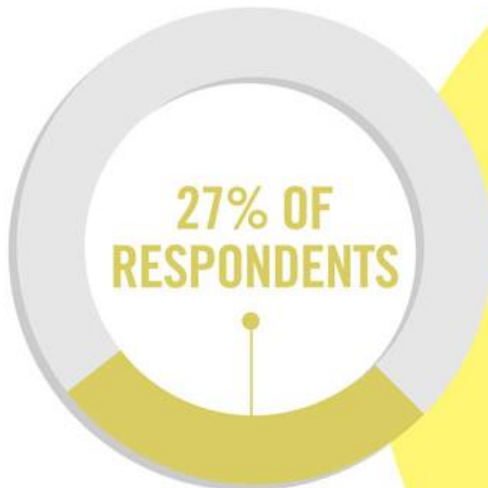
1 IN 3 BUSINESS LEADERS

don't trust the information they use to make decisions



Poor data quality costs the US economy around

\$3.1 TRILLION A YEAR



in one survey were unsure of how much of their data was inaccurate

Veracity
UNCERTAINTY
OF DATA

第五个“V”: Value(价值)

The fifth “V”?

Big data = the ability to achieve greater **Value** through insights from superior analytics



Case study: A US-based aircraft engine manufacturer now uses analytics to predict engine events that lead to costly airline disruptions, with 97% accuracy. If this prediction capability had been available in the previous year, it would have saved \$63 million.

第一章

绪论

1.1 数据库系统概述

1.1.1 四个基本概念

1.1.2 数据管理技术的产生和发展

1.1.3 数据库系统的特点

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)

以“数据”为基础
联系而又有区别

一、数据

- 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
 - 数字、文本、图形、图像、音频、视频
- 数据的特点
 - 数据的表现形式还不能完全表达其内容，需要解释

语义



- 数据的解释是指对数据含义的说明
- 数据的含义称为数据的语义(**semantic**)，数据与其语义是**不可分**的。
 - 例如 **93**是一个数据
 - 语义1：学生某门课的成绩
 - 语义2：某人的体重
 - 语义3：某个年级的学生人数

记录

- 记录是一种**结构化**的数据表示和存储方式
- 学生档案中的学生记录
 - （李明, 男, **199205**, 福建厦门市, 软件工程系, **2010**）
 - 语义：学生姓名、性别、出生年月、籍贯、所在系、
入学时间
 - 解释：李明是个大学生，**1992年5月**出生，福建厦门市人，
2010年考入软件工程系

文本、图形、图像、音频、视频**都是非结构化的**

二、数据库

- 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(redundancy)较小
- 数据独立性(data independency)较高
- 易扩展

通俗的讲就是数据的重复度。在一个数据集合中重复的数据称为冗余数据

当有新的业务逻辑，数据库结构变动无须太大

三、数据库管理系统

● 什么是DBMS

○ 一个系统软件

○ 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



DBMS的主要功能

○ 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的组成与结构

○ 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

提高存储空间利用率、存取效率

数据字典，用户数据，数据的存取路径等

○ 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)



DBMS的主要功能

○数据库的**事务管理**和**运行管理**

保证**事务**的正确运行

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**(实用程序)

数据库初始数据装载转换

数据库**转储、恢复**

数据库的重组

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

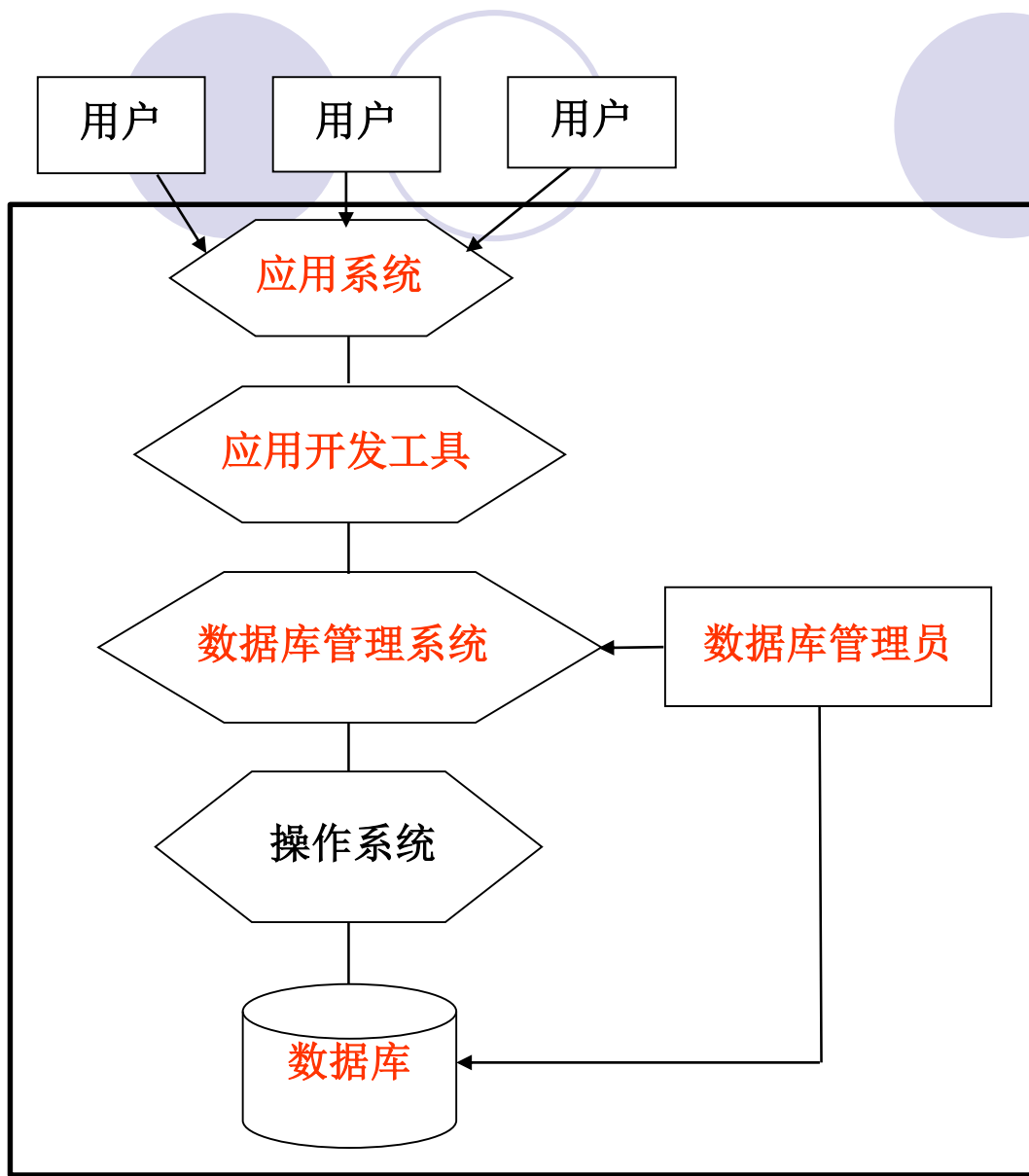
两个**DBMS**系统的数据转换

异构数据库之间的互访和互操作

Atomic
Consistent
Isolated
Durable

四、数据库系统

- 什么是数据库系统（**Database System**，简称**DBS**）
 - 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
 - 存储、管理、处理和维护数据的系统



数据库系统范畴

1.1 数据库系统概述

1.1.1 四个基本概念

1.1.2 数据管理技术的产生和发展

1.1.3 数据库系统的特点

数据管理技术的产生和发展

- 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(收集,存储,加工,传播)的**中心问题**

- 数据管理技术的发展过程

- 人工管理阶段(**20世纪40年代中--50年代中**)
- 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**)

数据管理技术的产生和发展(续)

- 数据管理技术的发展动力

- 应用需求的推动

- 计算机硬件的发展

- 计算机软件的发展

一、人工管理阶段

- 时期

- 20世纪40年代中--50年代中

- 产生的背景

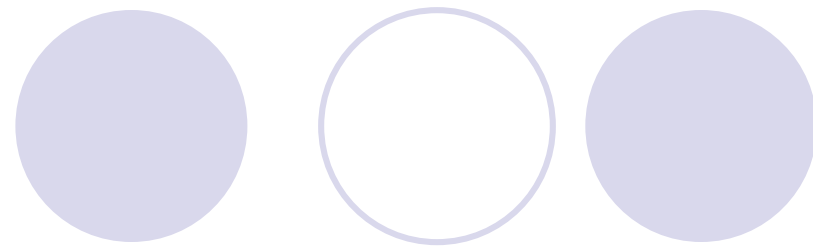
- 应用需求 科学计算

- 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

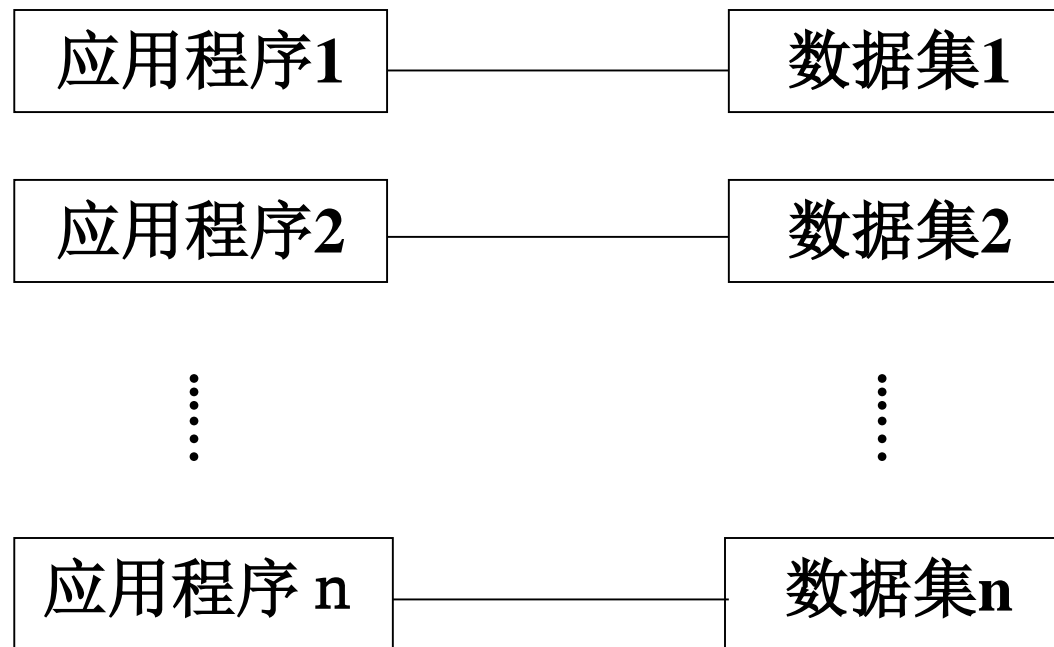
人工管理阶段(续)



● 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

应用程序与数据的对应关系(人工管理阶段)



人工管理阶段应用程序与数据之间的对应关系

二、文件系统阶段

- 时期

- 20世纪50年代末--60年代中

- 产生的背景

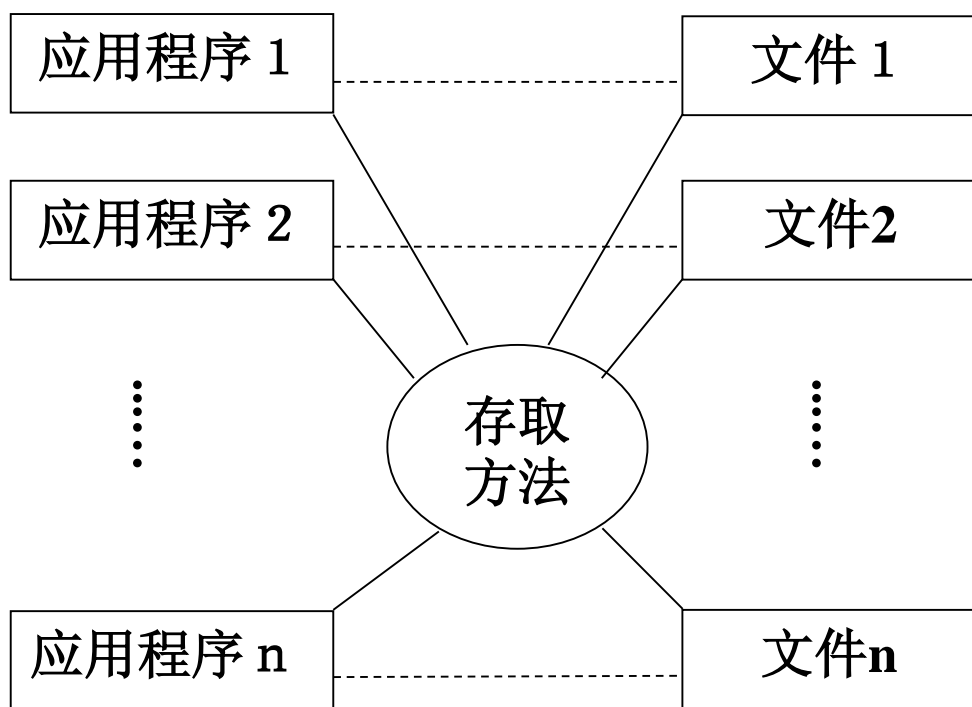
- 应用需求 科学计算、数据管理
 - 硬件水平 磁盘、磁鼓
 - 软件水平 有文件系统
 - 处理方式 联机实时处理、批处理

文件系统阶段(续)

● 特点

- 数据的管理者：文件系统，数据可长期保存
- 数据面向的对象：某一应用程序
- 数据的共享程度：共享性差、冗余度大
- 数据的结构化：记录内有结构,整体无结构
- 数据的独立性：独立性差
- 数据控制能力：应用程序自己控制

应用程序与数据的对应关系(文件系统阶段)



文件系统阶段应用程序与数据之间的对应关系

三、数据库系统阶段

- 时期

- 20世纪60年代末以来

- 产生的背景

- 应用背景

大规模数据管理

- 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

从文件系统到数据库系统，是数据管理技术的一个飞跃



1.1 数据库系统概述

1.1.1 四个基本概念

1.1.2 数据管理技术的产生和发展

1.1.3 数据库系统的特点

1.1.3 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

1. 数据结构化

- 整体数据的结构化是数据库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

文件系统的记录示例

学生文件的记录结构

学号	姓名	性别	系	年龄	住址	联系电话
----	----	----	---	----	----	------

课程文件的记录结构

课程号	课程名	学时
-----	-----	----

学生选课文件的记录结构

学号	课程号	成绩
----	-----	----

文件中记录内部有结构，但记录间无联系

2. 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。

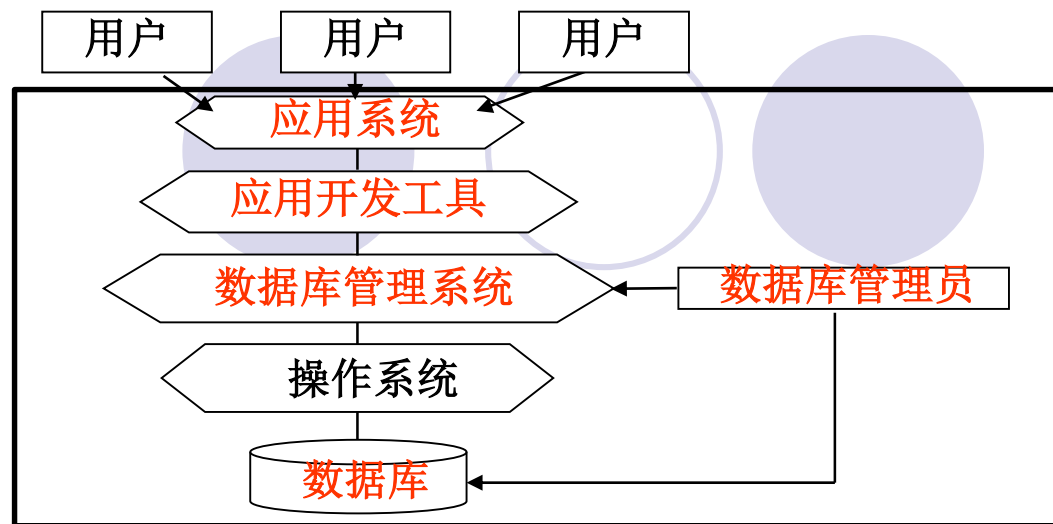
- 数据共享的好处

- 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

同一数据的不同副本的值不一样

表示同一事实的数据逻辑上有冲突

3. 数据独立性高



- **物理独立性(physical data independence)**

- 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(logical data independence)**

- 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

4. 数据由DBMS统一管理和控制

● DBMS提供的数据库控制功能

○ (1)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。

○ (2)数据的完整性（Integrity）检查

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。

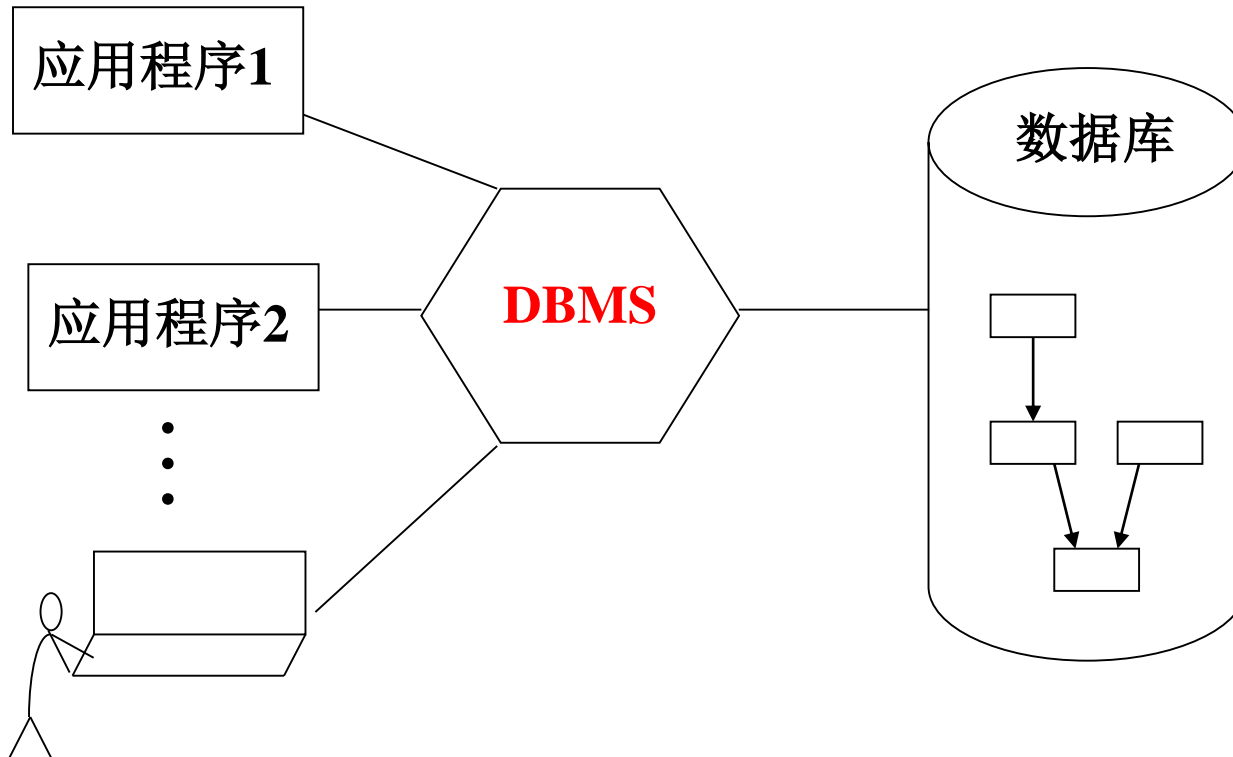
○ (3)并发（Concurrency）控制

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。

○ (4)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。

应用程序与数据的对应关系(数据库系统)



数据库系统阶段应用程序与数据之间的对应关系

本节小结

- 数据库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。
- 数据库管理系统是位于用户与操作系统之间的一层数据管理软件。
- 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
- 遗留的问题：DBMS的二级映像

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

1.5 小结

数据模型(data model)

- 对现实世界数据特征的抽象
- 用来描述、组织和操作现实世界中的数据
- 通俗地讲数据模型就是现实世界的模拟
- 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



1.2 数据模型

1.2.1 两大类数据模型

1.2.2 数据模型的组成要素

1.2.3 概念模型

1.2.4 常用的数据模型

1.2.5 层次模型

1.2.6 网状模型

1.2.7 关系模型

1.2.1 两大类数据模型



一种模型满足
三个要求很难

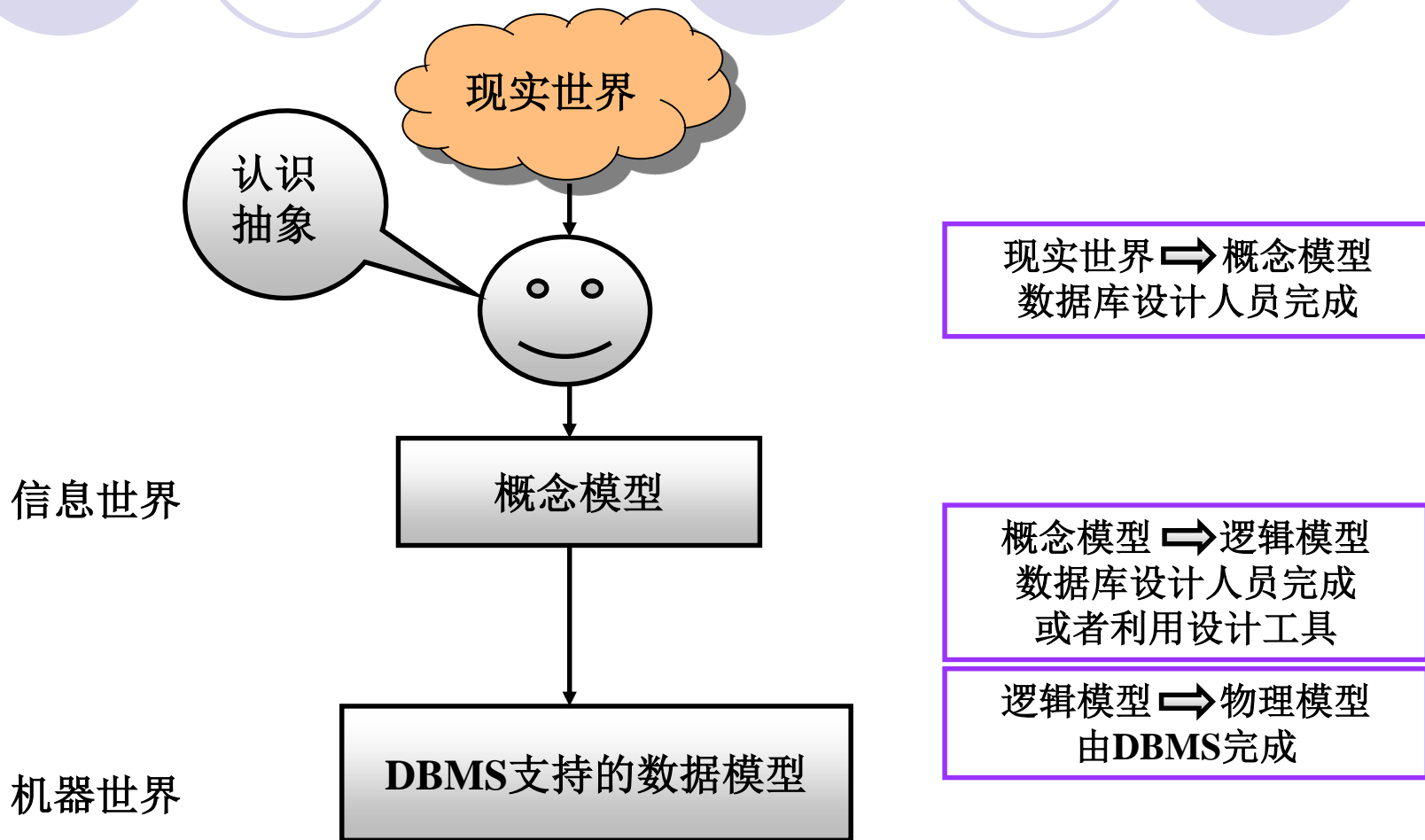
- 数据模型分为两类（分属两个不同的层次）

(1) 概念模型(**Conceptual Model**) 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**。

(2) 逻辑模型(**Logical Model**)和物理模型(**Physical Model**)

- 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。
- 物理模型具体实现是**DBMS**的任务，用户不必考虑物理级的细节

两大类数据模型 (续)



1.2 数据模型

1.2.1 两大类数据模型

1.2.2 数据模型的组成要素

1.2.3 概念模型

1.2.4 最常用的数据模型

1.2.5 层次模型

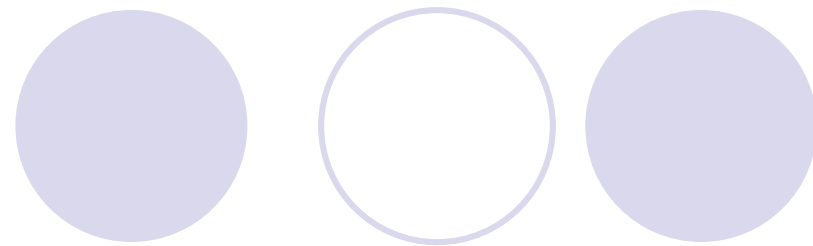
1.2.6 网状模型

1.2.7 关系模型

1.2.2 数据模型的组成要素

- 数据结构
- 数据操作
- 完整性约束条件(integrity constraints)
- 精确地描述了系统的静态特性、动态特性和完整性

一、数据结构



- 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

- 与对象的类型、内容、性质有关的

- 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

- 通常按照数据结构类型来命名数据模型

举例：学生选课系统的数据结构

学生关系模式

属性	类型	空	说明
学号	CHAR(8)		码
姓名	CHAR(10)	N	
性别	INT	N	0 男 1 女
专业	CHAR	Y	

课程关系模式

属性	类型	空	说明
课程号	CHAR(8)		码
课程名	CHAR(30)	N	
学时	INT	N	

学生选课关系模式

属性	类型	空	说明
学号	CHAR(8)		分别是外码 一起是主码
课程号	CHAR(8)		
成绩	INT	Y	0=< >=100

二、数据操作

- 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

- 数据操作的类型和规则

- 查询、更新(包括插入、删除、修改)
- 优先级

- 数据操作是对系统动态特性的描述

三、数据的完整性约束条件

- 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对象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。

举例：学生选课系统的数据操作及完整性约束

学生关系（值）

学号	姓名	性别	专业
05001	李明	男	计算机
05010	王丽	女	计算机
05130	张伟	男	数控
06008	刘力	男	数控
05130	李平	男	计算机

课程关系（值）

课程号	课程名	学时
J001	语言	64
J002	数据结构	48
J012	AutoCAD	48
J003	数据库	56

不符合实体完整性

选课关系（值）

学号	课程号	成绩
05001	J001	85
05010	J001	90
05130	J012	89
05001	J002	75
05001	J003	180

不符合参照完整性

不符合用户定义的完整性

1.2 数据模型

1.2.1 两大类数据模型

1.2.2 数据模型的组成要素

1.2.3 概念模型

1.2.4 常用的数据模型

1.2.5 层次模型

1.2.6 网状模型

1.2.7 关系模型

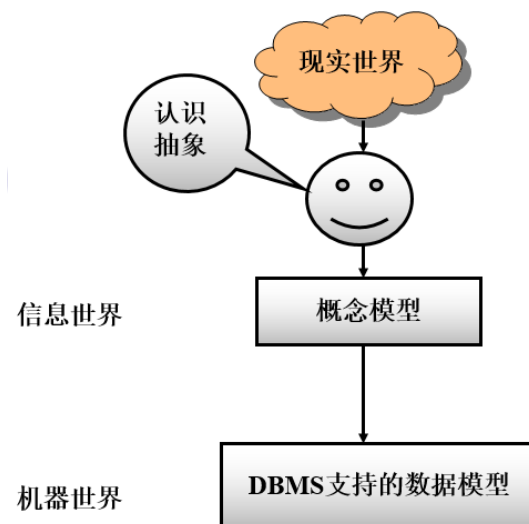
1.2.3 概念模型

- 概念模型的用途

- 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
- 是数据库设计的有力工具
- 数据库设计人员和用户之间进行交流的语言

- 对概念模型的基本要求

- 较强的语义表达能力
- 能够方便、直接地表达应用中的各种语义知识
- 简单、清晰、易于用户理解



一、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。

(2) 属性 (Attribute)

实体所具有的某一特性称为属性。

一个实体可以由若干个属性来刻画。

(3) 码 (Key)

唯一标识实体的属性集称为码。

信息世界中的基本概念(续)

(4) 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

比如：学生(学号,姓名,性别,生日,专业,....)

(5) 实体集 (Entity Set)

同一类型实体的集合称为实体集

比如：全体学生就是一个实体集

信息世界中的基本概念(续)

(6) 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。
- 实体内部的联系通常是指组成实体的各属性之间的联系
- 实体之间的联系通常是指不同实体集之间的联系或者同一实体集内不同实体间的联系

二、概念模型的一种表示方法

- 实体—联系模型(E-R模型)

- 用E-R图来描述现实世界的概念模型



Dr. Peter Pin-Shan Chen

陈品山 教授

路易斯安纳州立大学(LSU)计算机科学系

<http://www.csc.lsu.edu/~chen/>

E-R图

● 实体型

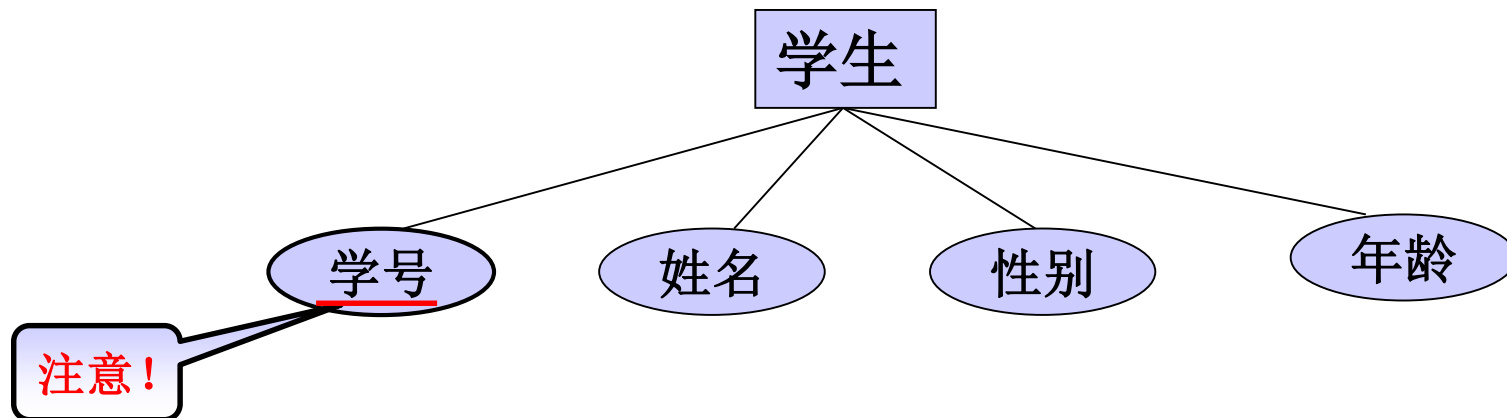
用**矩形**表示，**矩形框内写明实体名**。一般用**名词**。

学生

教师

● 属性

用**椭圆形**表示，并用**无向边**将其与相应的实体(或者联系)连接起来，选为**码**的属性要**加下划线**



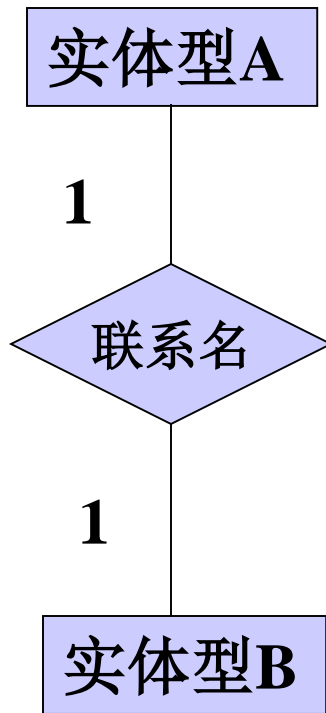
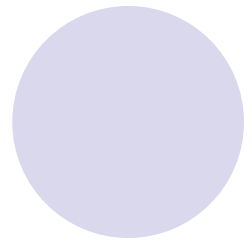
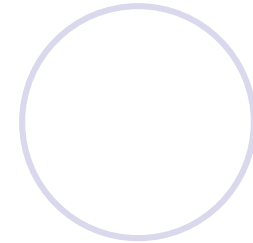
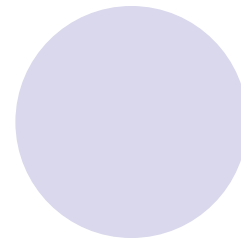
E-R图(续)

●联系

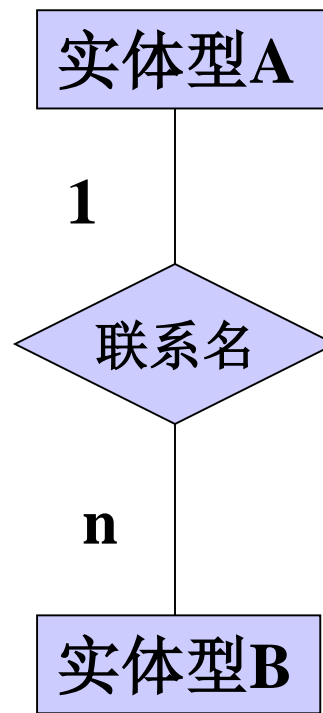
❖联系的本身：

- 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，同时无向边旁标上联系类型（1:1、1:n或m:n）
- 一般用动词

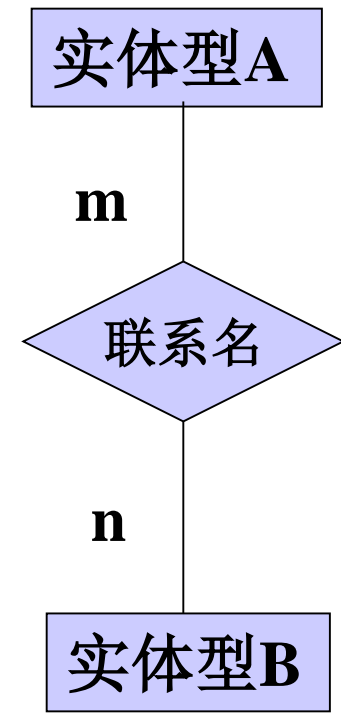
联系的表示方法



1:1联系



1:n联系



m:n联系

三、两个实体型之间的联系

- 一对一联系 (1:1)

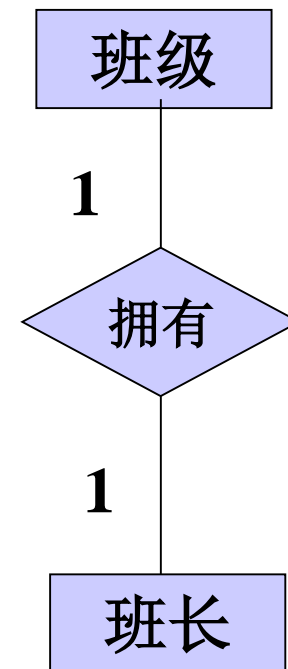
- 实例

一个班级只有一个正班长

一个班长只在一个班中任职

- 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



1:1联系

两个实体型之间的联系 (续)

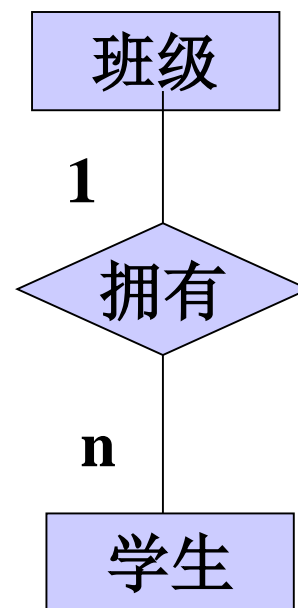
● 一对多联系 (1:n)

○ 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

○ 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** ($n \geq 0$) 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



1:n联系

两个实体型之间的联系 (续)

● 多对多联系 (m:n)

○ 实例

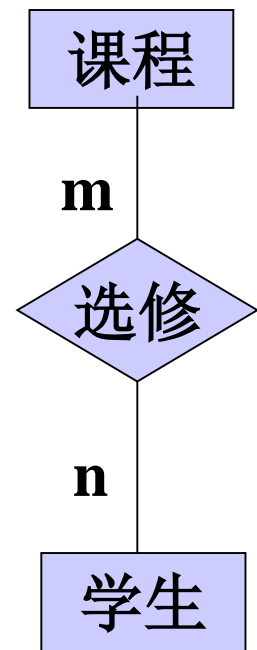
课程与学生之间的联系：

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

○ 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** ($n \geq 0$) 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** ($m \geq 0$) 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

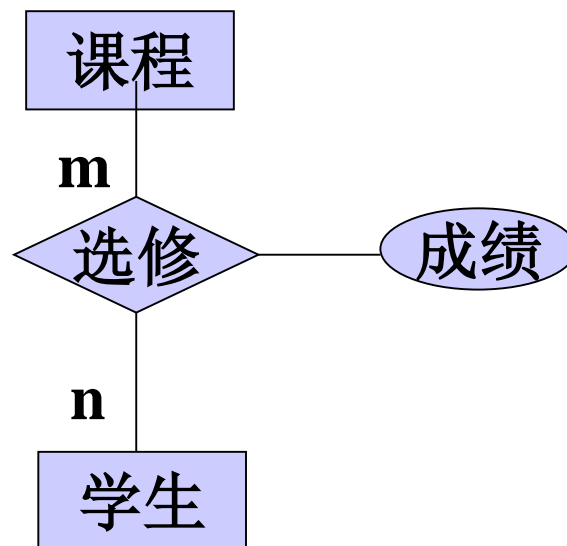


m:n联系

联系的属性

❖ 联系的属性:

联系也可以有属性。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



四、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, \dots, E_m 存在联系, 对于实体集 E_j ($j=1, 2, \dots, i-1, i+1, \dots, m$) 中的给定实体, 最多只和 E_i 中的一个实体相联系, 而 E_i 中的一个实体可以与 E_j ($j=1, 2, \dots, i-1, i+1, \dots, m$) 中的 n 个实体 ($n \geq 0$) 相联系, 则我们说 E_i 与 $E_1, E_2, \dots, E_{i-1}, E_{i+1}, \dots, E_m$ 之间的联系是一对多的

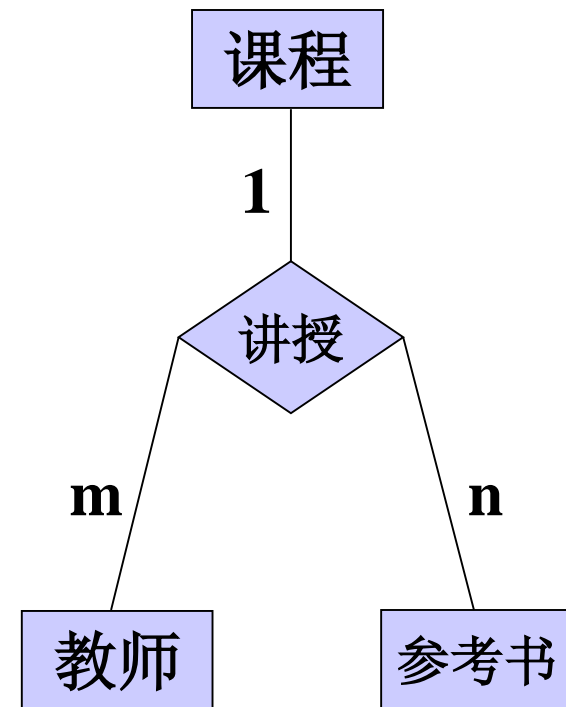
- 两个以上实体型之间其他类型联系呢?

两个以上实体型之间的联系(续)

● 实例：课程、教师与参考书

参考书	教师
A	Tom
A	John
A	Smith
B	Tom
B	John
B	Smith

课程
高等数学
数学分析
微积分



两个以上实体型间1:n联系

函数依赖：(参考书, 教师) → 课程

两个以上实体型之间的联系(续)

- 两个以上实体型间的多对多联系

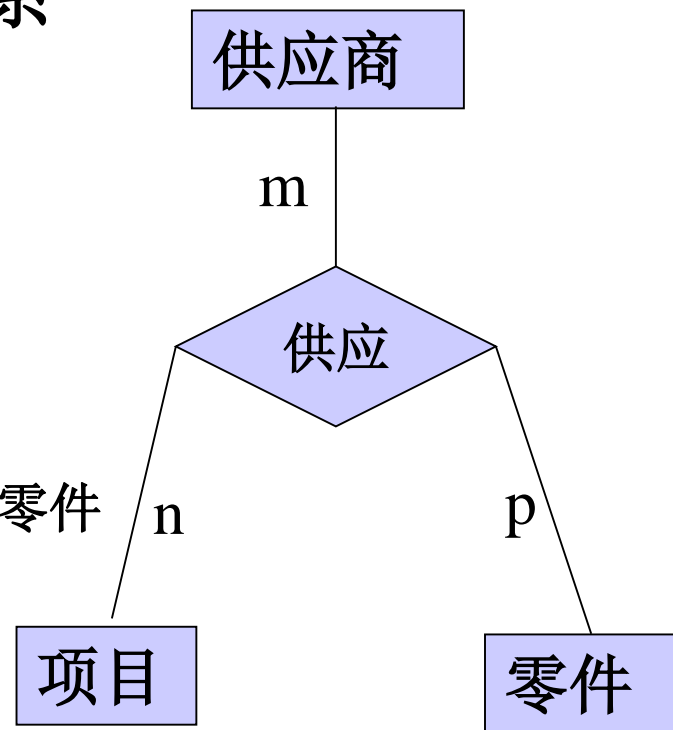
- 实例

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不同零件

每种零件可由不同供应商供给不同的项目



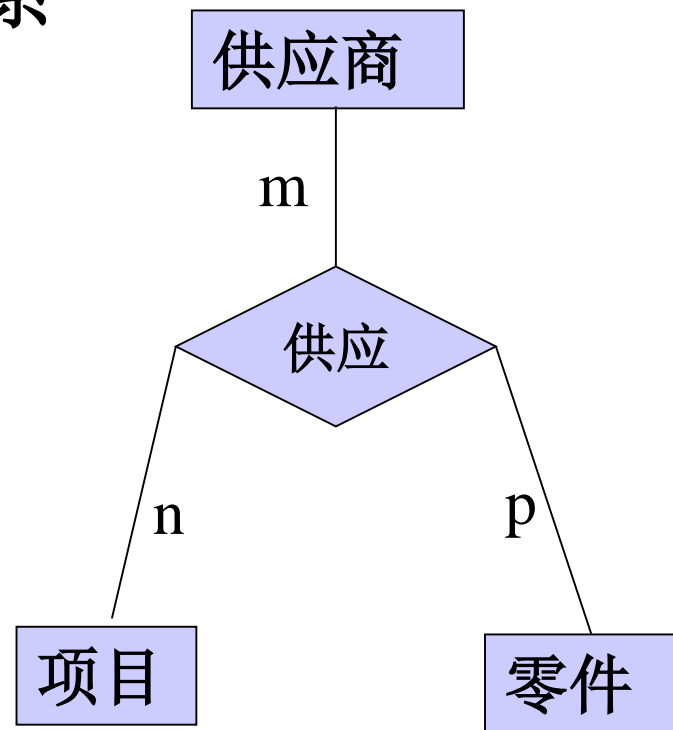
两个以上实体型间m:n联系

两个以上实体型之间的联系(续)

- 两个以上实体型间的多对多联系

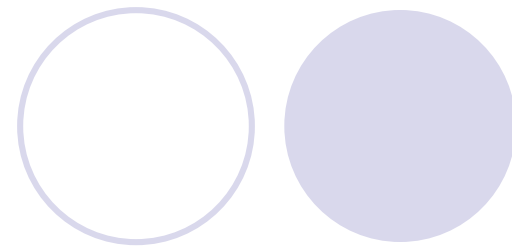
- 实例

供应商	项目	零件
1	A	X
1	A	Y
1	B	X
1	B	Y
2	A	X
2	A	Y
2	B	X
2	B	Y



两个以上实体型间m:n联系

五、单个实体型内的联系



● 一对多联系

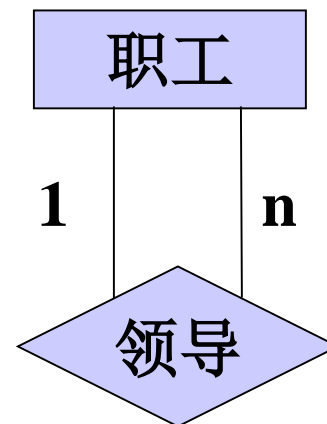
○ 实例

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



单个实体型内部
1:n联系

● 一对一联系

○ 请举例

六、一个实例



用E-R图表示某个工厂物资管理的概念模型

● 实体型

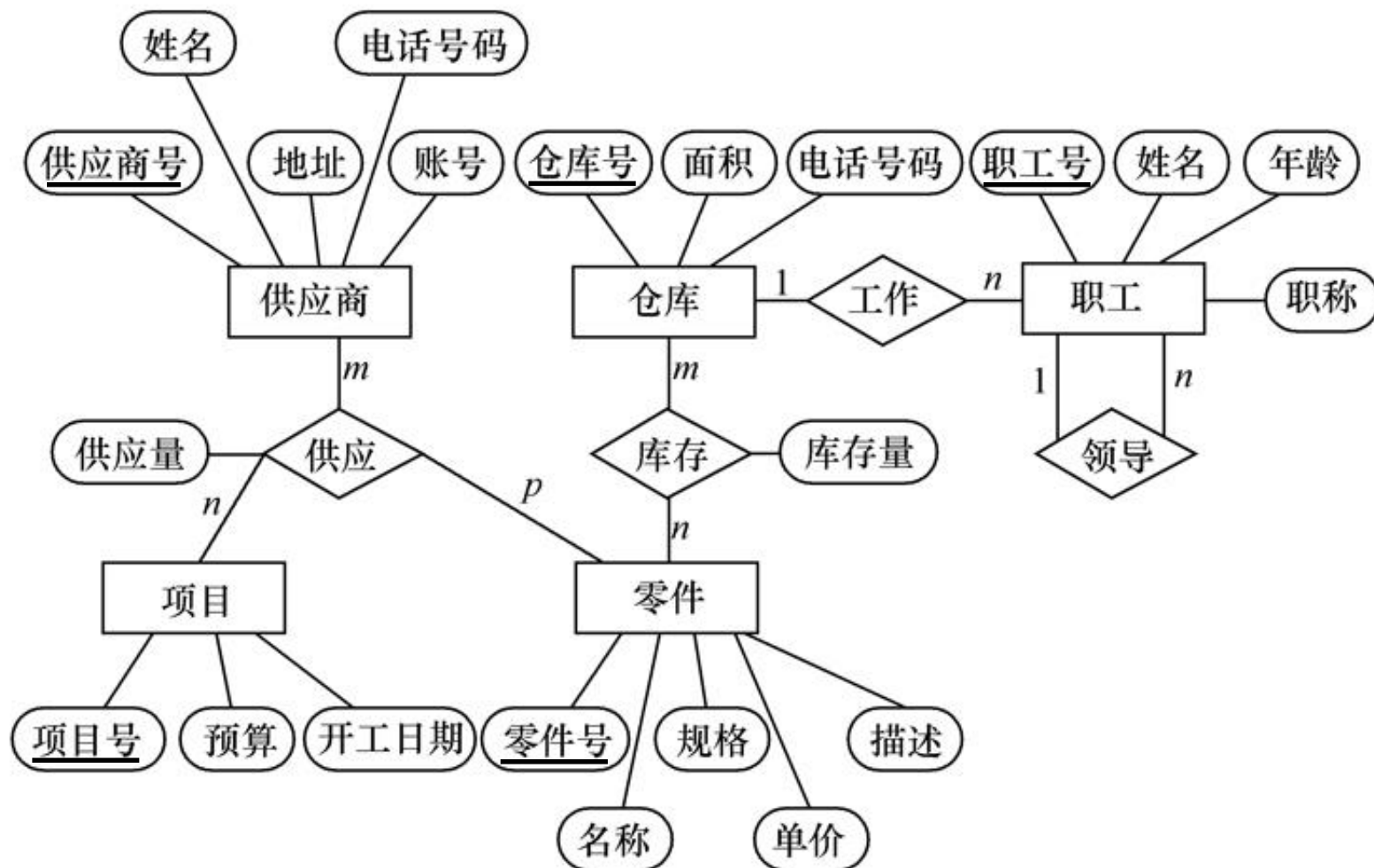
- 仓库： 仓库号、面积、电话号码
- 零件： 零件号、名称、规格、单价、描述
- 供应商： 供应商号、姓名、地址、电话号码、帐号
- 项目： 项目号、预算、开工日期
- 职工： 职工号、姓名、年龄、职称

一个实例

● 实体型之间的联系如下：

- (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。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。
- (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。用**供应量**来表示某种零件由某个供应商为某个项目提供的数量。

一个实例



(c) 完整的实体-联系图

习题

为某百货公司设计一个**ER**模型。

百货公司管辖若干个连锁商店，每家商店经营若干商品，同一种商品可在不同商店销售。每家商店有若干职工，但每个职工只能服务于一家商店。商店的属性有：商店编号，店名，店址。商品的属性有：商品编号，商品名，单价。职工的属性有：职工编号，姓名。职工到某商店工作有“聘期”和“工资”，商店销售商品有“销量”。

根据上述语义画出**ER**图，在**ER**图中需注明实体的属性、联系的类型及实体的标识符。

1.2 数据模型

1.2.1 两大类数据模型

1.2.2 数据模型的组成要素

1.2.3 概念模型

1.2.4 常用的数据模型

1.2.5 层次模型

1.2.6 网状模型

1.2.7 关系模型

1.2.4 常用的数据模型

- 格式化模型

- 层次模型(Hierarchical Model)

- 网状模型(Network Model)

- 关系模型(Relational Model)

- 面向对象模型(Object Oriented Model)

- 对象关系模型(Object Relational Model)

- 半结构化数据模型(Semistructured Data Model)

- 非结构化数据模型(Unstructured Data Model)



1.2 数据模型

1.2.1 两大类数据模型

1.2.2 数据模型的组成要素

1.2.3 概念模型

1.2.4 常用的数据模型

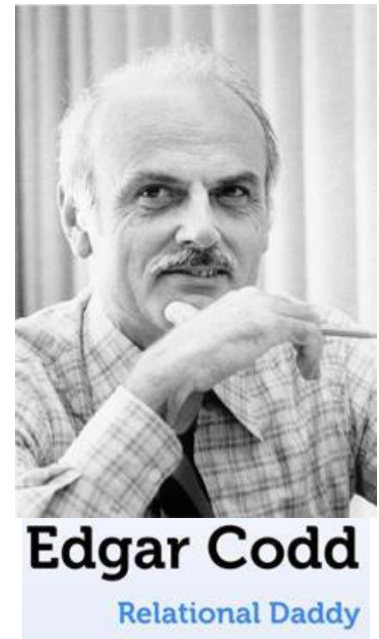
1.2.5 层次模型

1.2.6 网状模型

1.2.7 关系模型

1.2.7 关系模型

- 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
- 回忆：数据模型的三个要素



一、关系数据模型的数据结构

- 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。

学生登记表

学 号	姓 名	年 龄	性 别	系 名	年 级
2014004	王小明	19	女	社会学	2014
2014006	黄大鹏	20	男	商品学	2014
2014008	张文斌	18	女	法律	2014
...

属性

元组

关系数据模型的数据结构（续）

○关系（**Relation**）

一个关系对应通常说的一张表

○元组（**Tuple**）

表中的一行(**row**)即为一个元组

○属性（**Attribute**）

表中的一列(**column**)即为一个属性，给每一个属性起一个名称即属性名

关系数据模型的数据结构（续）

○码（Key）

表中的某个属性组，它可以唯一确定一个元组。

○域（Domain）

属性的取值范围，比如：整数，{'男', '女'}。

○分量

2014004	王小明	19	女	社会学	2014
---------	-----	----	---	-----	------

元组中的一个属性值。 ↗

○关系模式

对关系的描述

关系名（属性1，属性2，...，属性n）

Ex: 学生（学号，姓名，年龄，性别，系，年级）

关系数据模型的数据结构（续）

- 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**

比如:工资和扣除是可分的,**不符合关系模型要求**

职工号	姓名	职 称	工 资			扣 除		实 发
			基 本	津 贴	绩 效	房 租	水 电	
86051	陈 平	讲 师	1305	1200	1850	160	112	4083
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

关系数据模型的数据结构（续）

表1.5 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）

二、关系数据模型的操纵与完整性约束

- 数据操作是集合操作，操作对象和操作结果都是关系，即若干元组的集合
 - 查询、插入、删除、更新
- 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

三、关系数据模型的优缺点

● 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

- 关系模型的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

关系数据模型的优缺点（续）

● 缺点

- 存取路径对用户透明导致查询效率往往不如非关系数据模型
- 为提高性能，必须对用户的查询请求进行优化增加了开发DBMS的难度

第一章 绪论



1.1 数据库系统概述

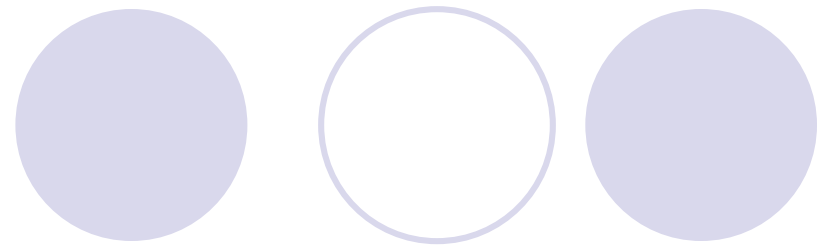
1.2 数据模型

1.3 数据库系统结构

1.4 数据库系统的组成

1.5 小结

1.3 数据库系统结构



- 从数据库应用开发人员角度看，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

数据库系统结构（续）



1.3.1 数据库系统模式的概念

1.3.2 数据库系统的三级模式结构

1.3.3 数据库的二级映像功能与数据独立性

1.3.1 数据库系统模式的概念

- “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 型:

(学号, 姓名, 性别, 系别, 年龄, 籍贯)

一个 值:

(1001, 李明, 男, 软工, 22, 福建)

数据库系统模式的概念（续）

- 模式（**Schema**）

- 数据库**全体数据**的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

数据库系统模式的概念（续）

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

数据库系统结构（续）



1.3.1 数据库系统模式的概念

1.3.2 数据库系统的三级模式结构

1.3.3 数据库的二级映像功能与数据独立性

1.3.2 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**）

数据库系统的三级模式结构（续）

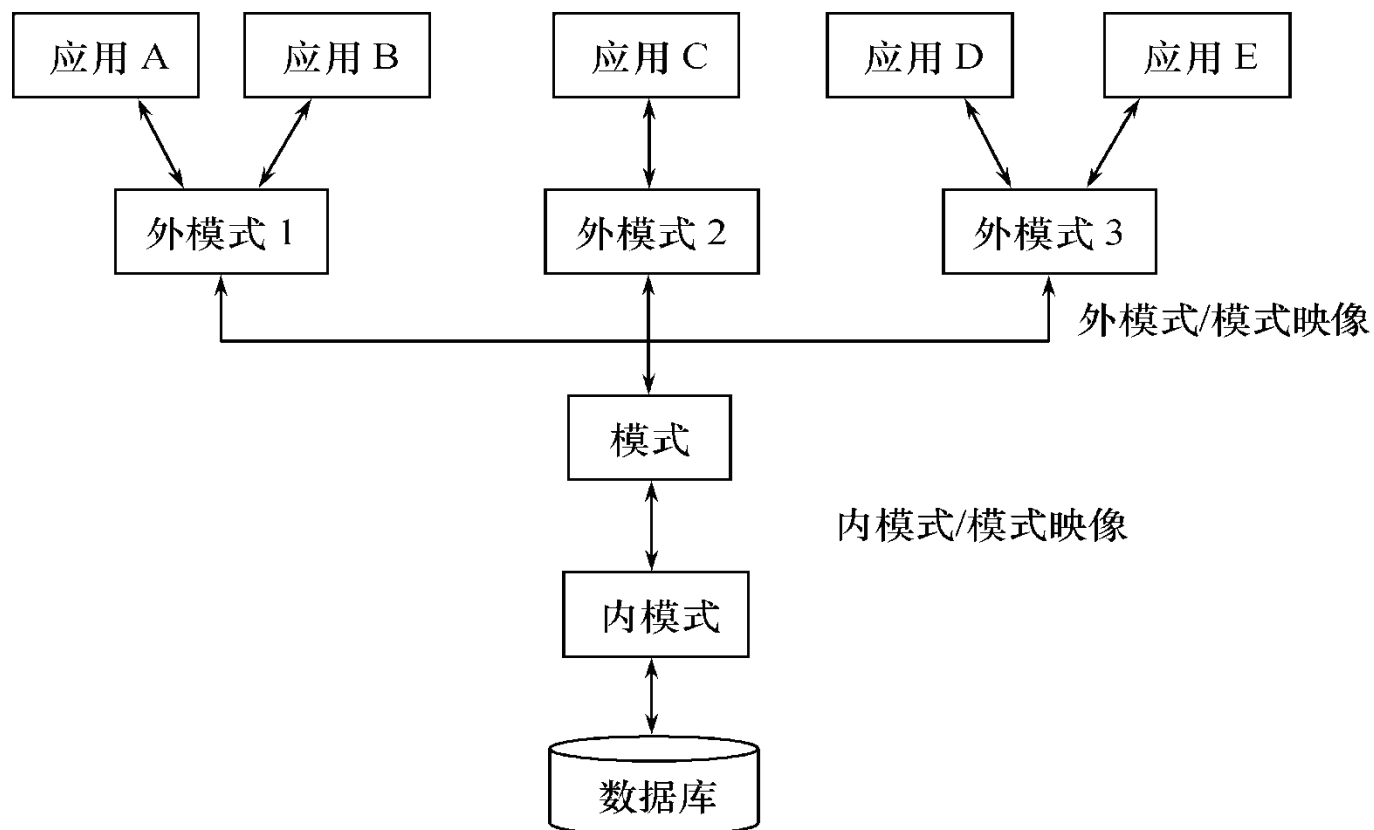
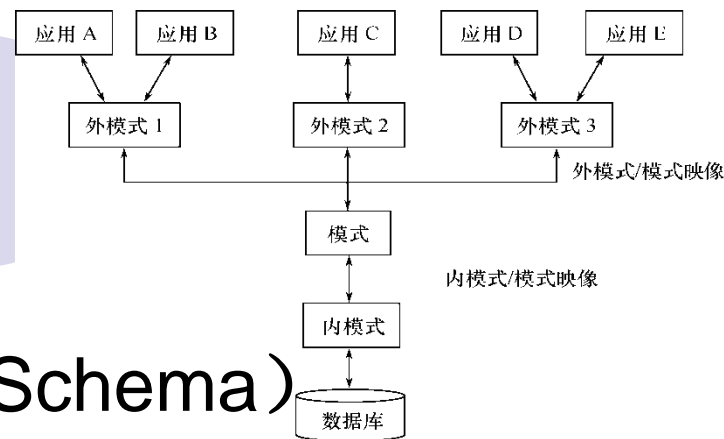


图1.16 数据库系统的三级模式结构

一、模式（Schema）



- 模式（也称逻辑模式 Logical Schema）
 - 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（续）

● 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
- 使用模式描述语言(模式**DDL**)来定义

二、外模式（External Schema）

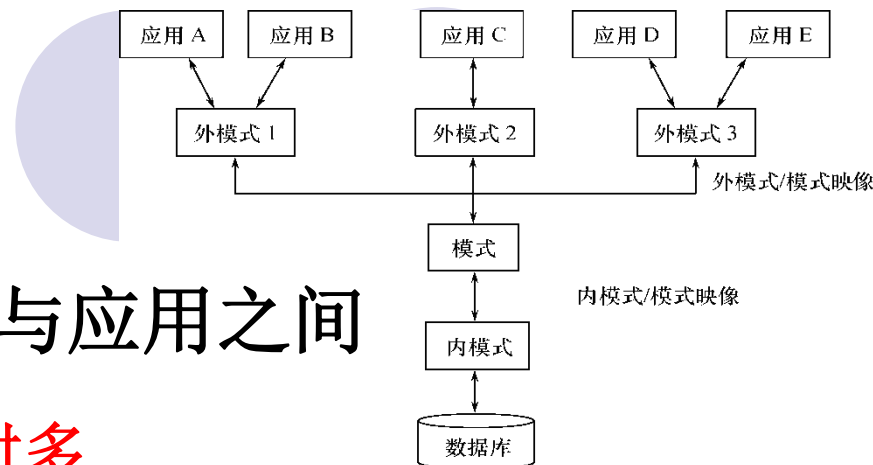
- 外模式（也称子模式或用户模式）

- 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 (续)



- 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

外模式（续）

- 外模式的用途

- 保证数据库安全性的一个有力措施
- 每个用户只能看见和访问所对应的外模式中的数据
- 数据库中的其余数据对其不可见

- 使用外模式数据定义语言来定义

三、内模式（Internal Schema）

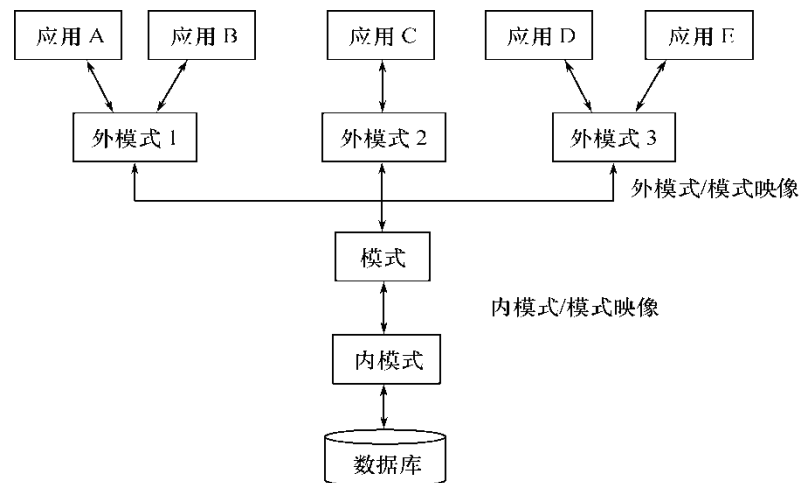
- 内模式（也称存储模式）

- 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
 - 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

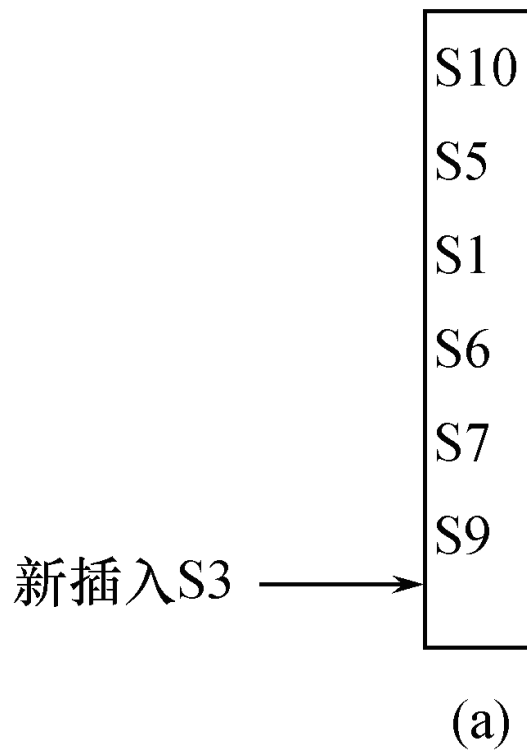
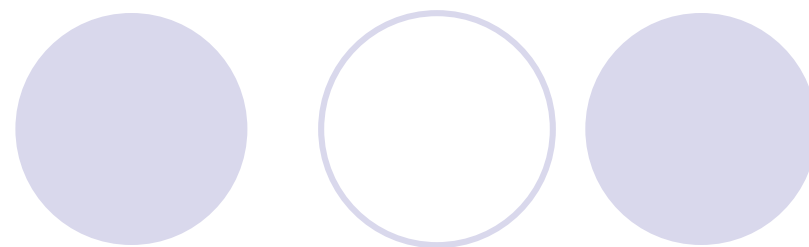
- 一个数据库只有一个内模式



内模式（续）

- 例如学生记录，如果按堆存储，则插入一条新记录总是放在学生记录存储的最后，如右图所示

- 回忆堆排序



内模式（续）

- 如果按学号升序存储，则插入一条记录就要找到它应在的位置插入，如图1.29（b）所示
- 如果按照学生年龄聚簇存放，假如新插入的**S3**是**16**岁，则应插入的位置如图1.29（c）所示

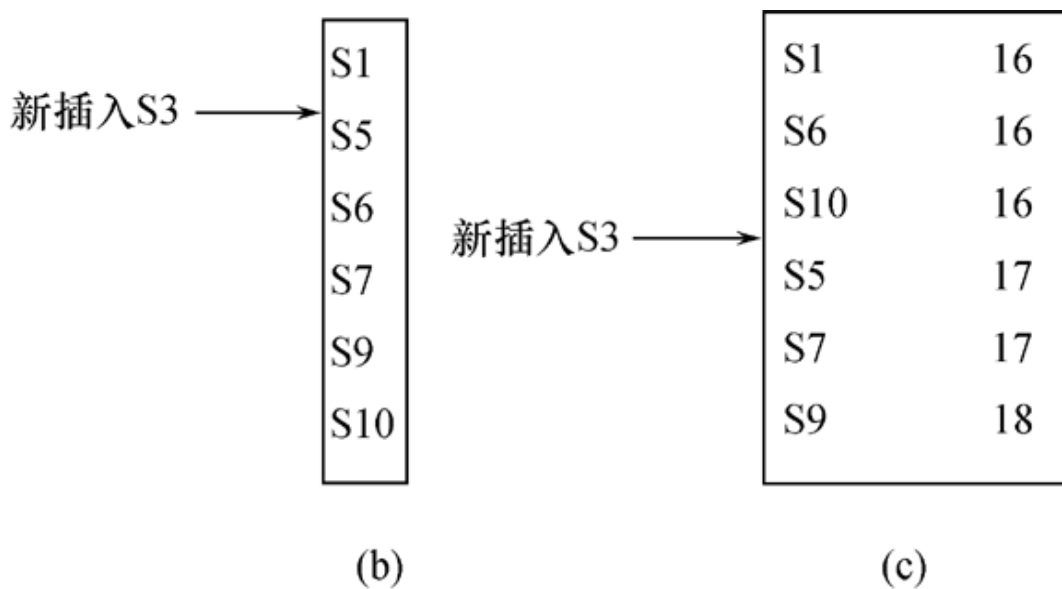


图1.29 记录不同的存储方式示意图

数据库系统结构（续）



1.3.1 数据库系统模式的概念

1.3.2 数据库系统的三级模式结构

1.3.3 数据库的二级映像功能与数据独立性

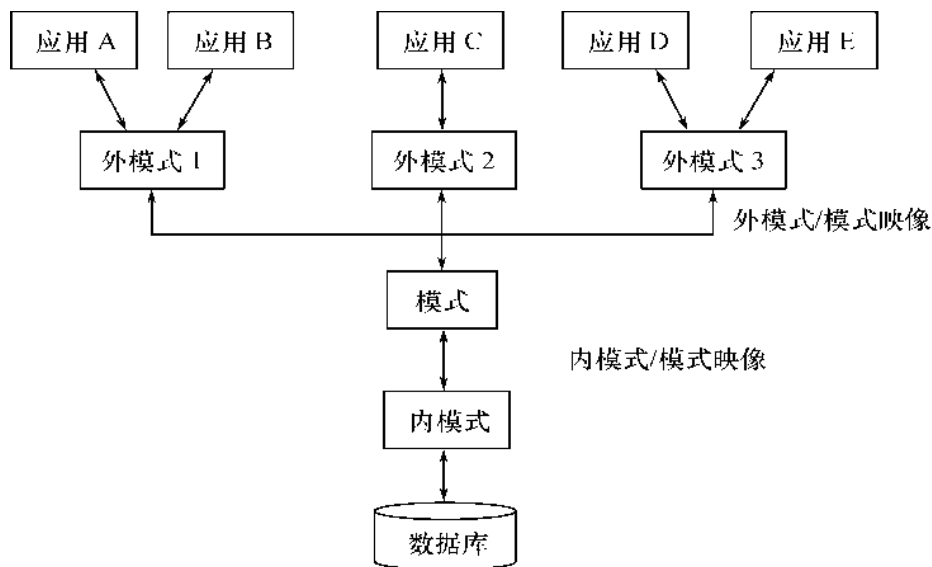
1.3.3 数据库的二级映像功能与数据独立性

- 三级模式是对数据的三个抽象级别
- 二级映像是在 **DBMS** 内部实现这三个抽象层

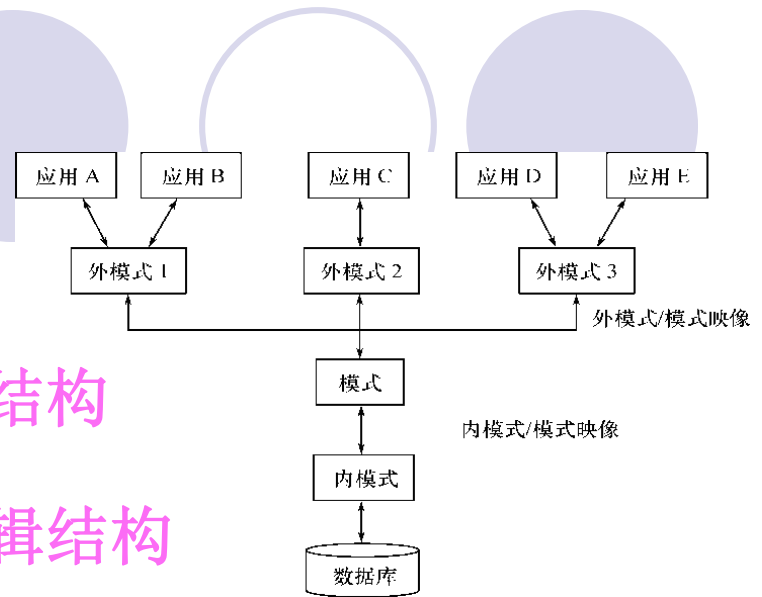
次的联系和转换

○ 外模式 / 模式映像

○ 模式 / 内模式映像



一、外模式 / 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 / 模式映像，定义外模式与模式之间的对应关系
- **映像定义通常包含在各自外模式的描述中**

外模式 / 模式映象（续）

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。

二、模式 / 内模式映像

- 模式 / 内模式映像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 / 内模式映像是**唯一**的
- 该映像定义通常**包含在模式描述中**

模式 / 内模式映象（续）

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**。

模式 / 内模式映象（续）

● 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**

模式 / 内模式映象（续）

● 数据库的内模式

- 依赖于它的全局逻辑结构（模式）
- 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

模式 / 内模式映象（续）

● 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**

模式 / 内模式映像（续）

- 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

模式 / 内模式映象（续）

- 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

1.5 小结

1.4 数据库系统的组成

- 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

数据库系统的组成（续）



- 硬件平台及数据库
- 软件
- 人员

一、硬件平台及数据库

- 数据库系统对硬件资源的要求

(1) 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

硬件平台及数据库（续）

(2) 足够大的外存

- 磁盘或磁盘阵列

- 数据库

- 光盘、磁盘

- 数据备份

(3) 较高的通道能力，提高数据传送率

二、软件

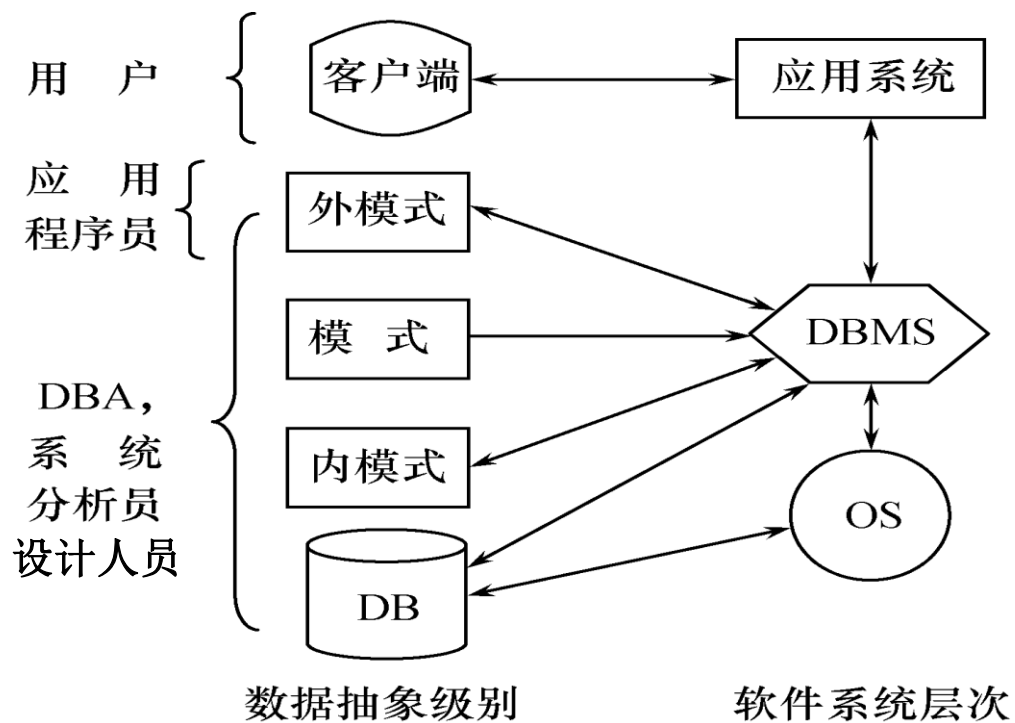
- **DBMS**
- 支持**DBMS**运行的操作系统
- 与数据库接口的高级语言及其编译系统
- 以**DBMS**为核心的应用开发工具
- 为特定应用环境开发的数据库应用系统

三、人员

- 数据库管理员(**DBA**)
- 系统分析员和数据库设计人员
- 应用程序员
- 用户

人员（续）

❖ 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



1. 数据库管理员(DBA)

具体职责：

- 1.决定数据库中的信息内容和结构
- 2.决定数据库的存储结构和存取策略
- 3.定义数据的安全性要求和完整性约束条件

数据库管理员(续)

● 4. 监控数据库的使用和运行

○ 周期性转储数据库

- 数据文件

- 日志文件

○ 系统故障恢复

○ 介质故障恢复

○ 监视审计文件

数据库管理员(续)

- 5. 数据库的改进和重组
 - 性能监控和调优
 - 定期对数据库进行重组织，以提高系统的性能
 - 需求增加和改变时，数据库须需要重构造

2. 系统分析员和数据库设计人员

● 系统分析员

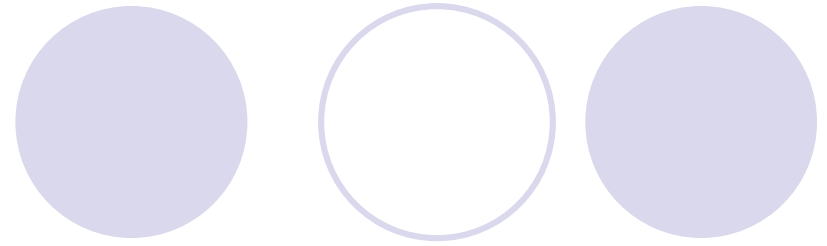
- 负责应用系统的**需求分析**和**规范说明**
- 与用户及**DBA**协商，确定系统的硬软件配置
- 参与数据库系统的概要设计

系统分析员和数据库设计人员（续）

- 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

3. 应用程序员



- 设计和编写应用系统的程序模块
- 进行调试和安装

4. 用户

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。

● 1. 偶然用户

- 不经常访问数据库，但每次访问数据库时往往需要不同的数据库信息
- 企业或组织机构的高中级管理人员

用户（续）

- 2. 简单用户

- 主要工作是查询和更新数据库
- 银行的职员、机票预定人员、旅馆总台服务员

- 3. 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

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

1.5 小结

1.5 小结

- 数据库系统概述

- 四个基本概念(**Data,DB,DBMS,DBS**)
- 数据管理的发展过程(三个阶段)
- 数据库系统特点(结构化,共享性,数据独立性,统一管控)

- 数据模型

- 数据模型的三要素(数据结构,数据操作,完整性约束)
- 概念模型, **E-R** 模型
- 关系模型

小结(续)

- 数据库系统的结构

- 数据库系统三级模式结构(外模式,模式,内模式)
- 数据库系统两层映像系统结构

- 数据库系统的组成