数据库系统概论

AN INTRODUCTION TO DATABASE SYSTEMS

刘 淇
Email: qiliuql@ustc.edu.cn

课程主页：
http://staff.ustc.edu.cn/~qiliuql/DB2021.html

助教： 庄 严 zykb@mail.ustc.edu.cn
毕昊阳 bhy0521@mail.ustc.edu.cn

教材及参考书(1)

教材

- 王珊，萨师煊：数据库系统概论(第五版) 高等教育出版社，**2014.9**
- A First Course in Database Systems

  Jeffrey.D.Ullman, Jennifer Widom

  Dept. Of Computer Science Stanford University

- 数据库系统基础教程 岳丽华 金培权 万寿红等译

教材及参考书(1)

中国人民大学杜小勇王珊团队成果获国家科学技术进步奖二等奖

1月8日 ...正等党和国家领导人出席

信息学 ...库管理系统核心技术的创 ...首次获得国家科学技术奖

该成果由中 ...究，在国产数据库管理系统内核 ...新性研究成果，突破了数据库管 ...41项，软件著作权19项，出版著 ...证，在电子政务、电子党务、 ...个行业和六十多个重大信息化工 ...库管理系统技术领域的发展，全

教材及参考书(2)

上机软件

- MySQL
  - MySQL workbench download:
    http://dev.mysql.com/downloads/workbench/
  - MySQL server download:
    https://dev.mysql.com/downloads/mysql/

安装和使用方式可以参考课程主页：

http://staff.ustc.edu.cn/~qiliuql/DB2021.html

学习方式

- **听课**

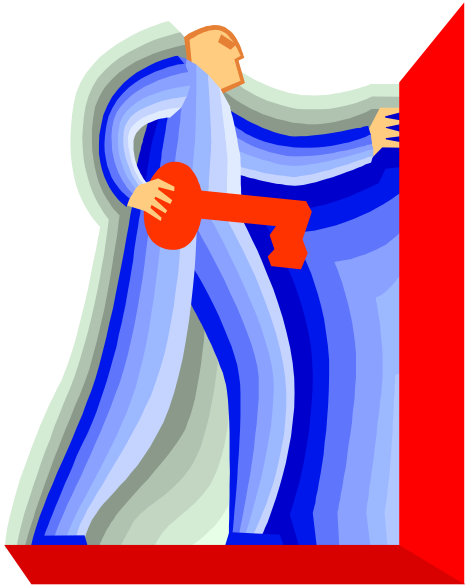
（启发式、讨论式）

- **读书**

（预习、复习）

- **报告**

（课程设计）



考试成绩

- 平时成绩（30%）

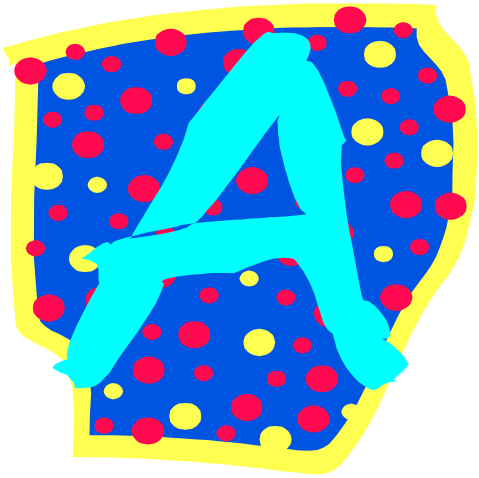
  书面作业+出勤率(三次不到，取消考试成绩)

- 实验成绩（20%）

  课程实验报告

- 期末考试（50%）

  卷面成绩



内容安排(1)

- 基础篇
  - 第一章 绪论
  - 第二章 关系数据库
  - 第三章 关系数据库标准语言SQL
  - 第四章 数据库安全性
  - 第五章 数据库完整性
- 设计与应用开发篇
  - 第六章 关系数据理论
  - 第七章 数据库设计
  - 第八章 数据库编程

内容安排(2)

- **系统篇**
  - 第九章 关系查询处理和查询优化
  - 第十章 数据库恢复技术
  - 第十一章 并发控制
  - * 第十二章 数据库管理系统
- **新技术篇**
  - 第十三章 数据库技术发展概述
  - 第十四章 大数据管理
  - 第十五章 内存数据库系统
  - 第十六章 数据仓库与联机分析处理技术

Any Questions?

- 课程教辅QQ群：526599354

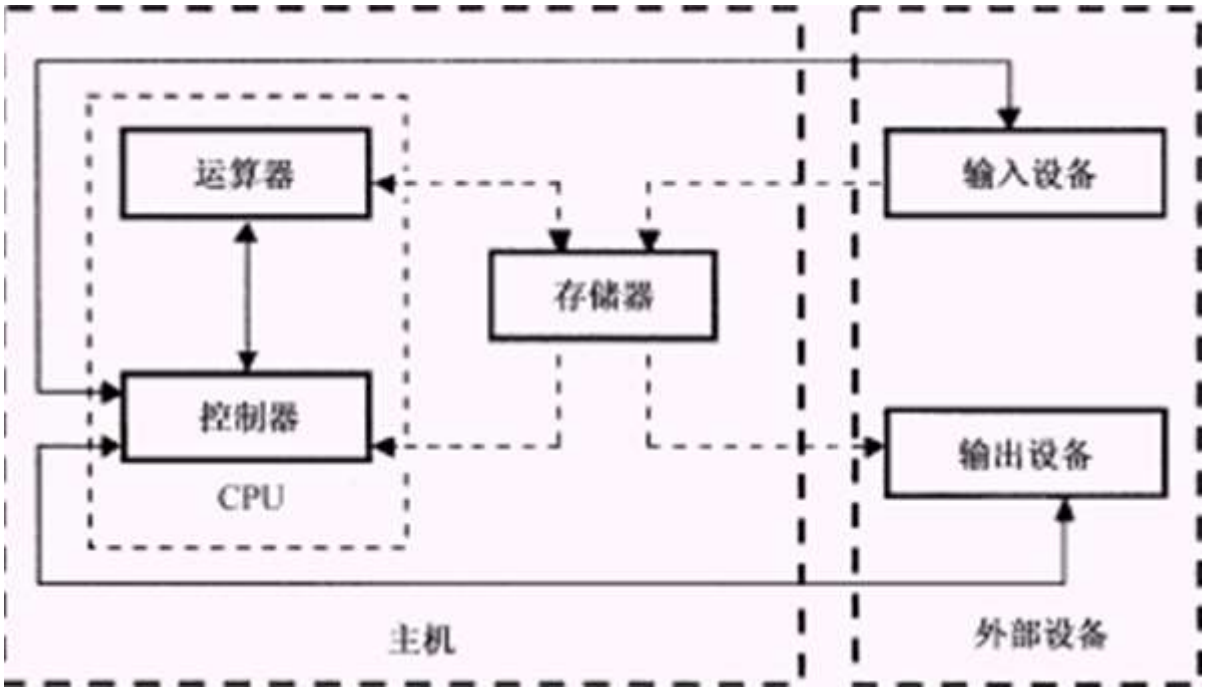
数据库系统概论

An Introduction to Database Systems

第一章　绪论

数据库的位置和作用

- 计算机组成



- 存储层次

  寄存器

  缓存

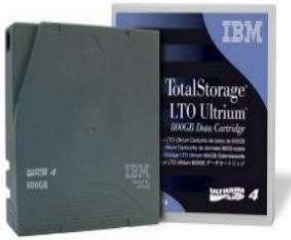
  内存

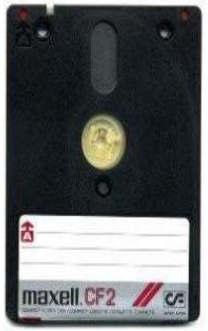
  磁盘

- 存储功能
  - 存储程序
  - 存储数据

数据库的位置和作用

存储器





数据库的位置和作用


BD-R DL
50GB
Blu-ray Disc Recordable
Ver.1.1/2X
COMPACT
disc
DATA STORAGE
DVD
Blu-ray Disc
HD DVD


Digital Media
PC CARD
FLASH STORAGE
1GB
SimpleTech
SanDisk
128MB
CompactFlash
Transcend
4GB
SONY
Memory Stick PRO Duo
4GB
1GB
miniSD
Kingston
FUJIFILM
xD
M2GB
SONY
M2
8GB
SONY
32
SxS PRO


TREK
ThumbDrive
WD
BUFFALO

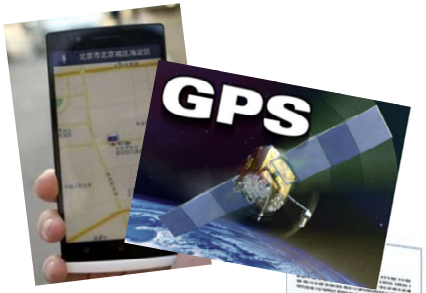
数据库的位置和作用

数据形式的多样:

- 结构化数据，半结构化数据，非结构化数据
- 图像数据，语音数据，文本数据，数字化数据

数据来源的多样性:

- 不同的IT应用系统
- 各种设备（物联网）
- 互联网
- 其它



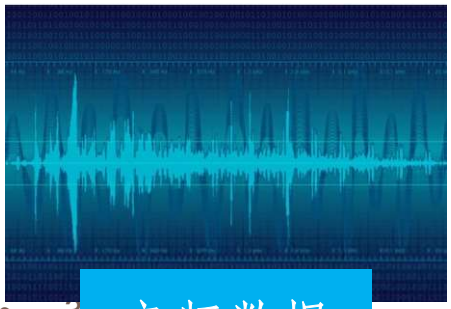
时空数据

文本数据

图像数据

事务数据

视频数据



音频数据

数据库的位置和作用

- 结构化数据（本课程重点）
  - 可以使用关系型数据库表示和存储的数据
- 半结构化数据
  - 弱结构化的数据，虽然不符合关系型数据模型的要求，但是仍然有明确的数据大纲，包括相关的标记，用来分割实体及期属性 （XML，JSON等标记语言）
- 非结构化数据
  - 没有固定数据结构或者很难发现统一数据结构的数据

姓　名	年	性　别
小明	12	男
小白	13	女
小奇	18	男

```
<person>
  <id>1</id>
  <name>小明</name>
  <age>12</age>
  <gender>男</gender>
</person>
```

对比JSON与XML


```
{
   "name": "中国",
   "province": [{
      "name": "黑龙江",
      "cities": {
         "city": ["哈尔滨", "大庆"]
      }
   },
     {
      "name": "广东",
      "cities": {
         "city": ["广州", "深圳", "珠海"]
      }
   },
   .....
   }]
```

对象，成员：键值对

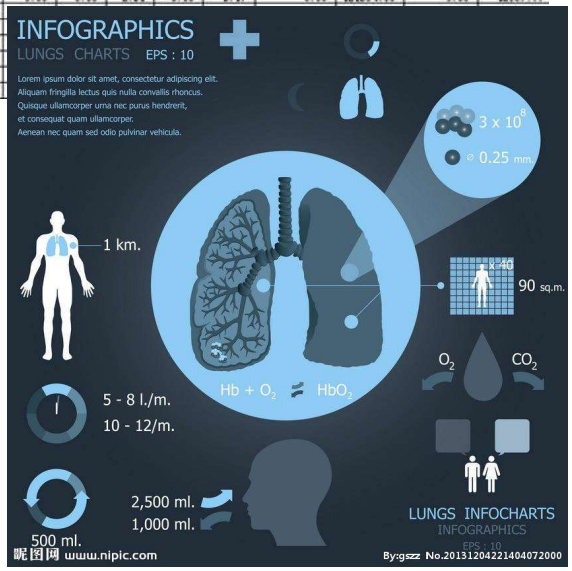
```
<?xml version="1.0" encoding="utf-8"?>
<country>
   <name>中国</name>
   <province>
      <name>黑龙江</name>
      <cities>
         <city>哈尔滨</city>
         <city>大庆</city>
      </cities>
   </province>
   <province>
      <name>广东</name>
      <cities>
         <city>广州</city>
         <city>深圳</city>
         <city>珠海</city>
      </cities>
   </province>
  .........
</country>
```

数据库的位置和作用

- **数据来源的多样性**

财务报表

股票代码	股票名称	流通×股（××年，末期）（万股）	总摊薄（××年，末期）（元）	每股收益摊薄（××年，第三季度）（元）	每股净资产（××年，末期）（元）	每股净资产（××年，第三季度）（元）	净资产收益率（××年，末期）（%）	净资产收益率（××年，第三季度）（%）	主营业务收入（××年，末期）（万元）	主营业务收入（××年，第三季度）（万元）	主营业务利润（××年，末期）（万元）	主营业务利润（××年，第三季度）（万元）
000028	一致药业	5488.56	0.00	0.04	0.00	1.26	0.00	2.84	0.00	176864.45	0.00	30512.72
000153	丰原药业	2500.00	0.00	0.32	0.00	8.55	0.00	3.78	0.00	12576.58	0.00	5271.23
000411	ST 英特	3153.07	0.00	0.01	0.00	0.67	0.00	1.18	0.00	95197.09	0.00	4748.88
000412	ST 五环	10652.17	0.00	-0.12	0.00	0.17	0.00	-68.78	0.00	533.96	0.00	82.51
000416	渤海生物	13831.79	0.62	0.52	1.41	1.37	44.33	37.98	42683.76	35759.51	29050.69	24580.53
000423	东阿阿胶	19175.34	0.32	0.22	2.88	3.04	11.05	7.38	11.00	35775.72	33566.42	21730.09
000513	丽珠集团	11567.23	0.00	0.22	0.00	3.23	0.00	6.95	0.00	113687.21	0.00	50593.88
000522	白云山A	15654.44	0.16	0.14	1.46	1.37	10.72	10.04	195720.59	153176.37	72497.53	56361.19
000538	云南白药	5875.05	0.50	0.38	2.50	2.69	20.20	14.13	110094.25	78594.76	38791.37	27975.84
000545	恒和制药	6809.58	0.00	-0.03	0.00	1.03	0.00	-2.42	0.00	9436.07	0.00	3406.77
		8354.90	0.00	0.05	0.00	-1.18	0.00	-4.09	0.00	17110.21	0.00	6142.17
		6355.39	0.00	-0.02	0.00	1.35	0.00	-1.29	0.00	20188.40	0.00	6634.10
		3828.00	0.00	0.08	0.00	2.51	0.00	3.17	0.00	121504.00	0.00	12067.77
		10881.00	0.00									
		2062.50	0.00									
		12529.66	0.32									
		17111.00	0.01									
		7980.34	0.00									
		8085.37	0.00									



Database is important for everyone!

数据库的位置和作用

背景：铺天盖地的“大数据”字眼

大数据“完全占领”了互联网和IT领域之后，开始进入各行各业，形成了政府大数据、教育大数据、医疗大数据、交通大数据、金融大数据、保险大数据、公安大数据、法院大数据、旅游大数据、……

数据库的位置和作用

- **李德毅院士：大数据本身，既不是科学也不是技术，它反映的是网络时代的一种客观存在**

你们说的大数据到底是啥？**大数据的输入和输出是？**

我不认为数据等同于价值，**哪些数据才有价值？**

大数据到底是**噱头+忽悠**，还是**真金白银**啊？

我没看清楚大数据的价值，但很清楚**大数据的大成本**，真能赚回来吗？

未来真的**不会大数据就不能赢了吗？**

我用SQL Server用的好好的，一定要**现在就转大数据吗？**

所谓的大数据牛的公司，**到底牛在哪？**

大数据就是数据，没什么可神秘的。它是一种原材料，数据库、数据挖掘、云计算、高性能计算、机器学习等都可以看作是对这种原材料进行存储烹饪加工等的手段和技术，目的就是做出各种美食（例如让AlphaGo打败李世石）

数据库的位置和作用

大数据发展的核心动力来源于人类测量、记录和分析世界的渴望

Google 谷歌

Baidu 百度

当文字成为数据

当方位成为数据

当沟通成为数据

一切事物的数据化

数据库的位置和作用

大数据有多大?

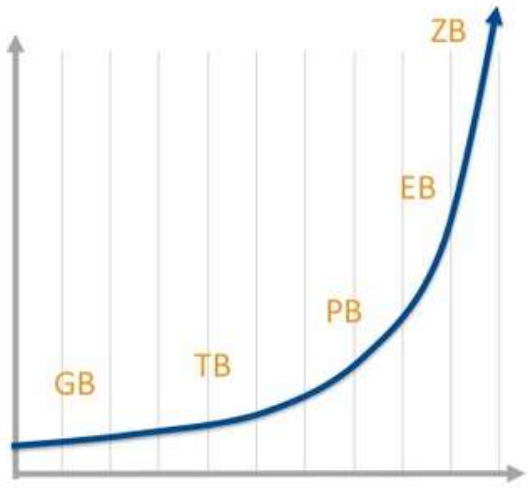
PB是大数据层次的临界点

- 数据量已到ZB等级

KB->MB->GB->TB->PB->EB->ZB->YB->NB->DB

PB以上级别的数据，最有效的传输方式是空运，而不是网络

- 而大数据不仅仅只是量大!



60秒，我们能产生多少数据?

数据库的位置和作用

数据/Data

结构化　半结构化　非结构化

1. 采集、预处理

5. 可视化

应用/App

4. 分析挖掘

3. 计算模式

计算系统

分布式文件系统、数据库

2. 存储管理

6. 隐私和安全

数据库的位置和作用

- 2020年3月4日，中共中央政治局常务委员会召开会议，研究当前新冠肺炎疫情防控和稳定经济社会运行重点工作。会议强调。。。加快5G网络、数据中心等新型基础设施建设进度。

区别于传统“基建”，“新基建”主要发力于科技端。传统基建主要是指铁路、公路、桥梁、水利工程等大建筑,而“新基建”是指立足于科技端的基础设施建设，主要包括5G建设等七大领域。在新冠肺炎疫情冲击中国经济的背景下，启动新一轮基建有助于稳增长、稳就业，释放国内经济增长潜力

数据库的位置和作用

- 大数据抗

要运用大数

【江淮晨报】利用“大数据”破解防疫四大难题

2020年07月13日

“90后”中国科大“后浪”挑战“科技战疫”，斩获冠军

同心抗疫

人民日报海外网

发布时间：03-03 05

- 疫情实时大
  助诊疗体系

例如：使用以下数
移动、联通
（三大运营
迹当作个人的生活

数据库的位置和作用

- 2019的网红“中国知网”

数据库的位置和作用

I Am Legend

简体中文名: 我是传奇

编剧: Mark Protosevich / Akiva Goldsman / Richard Matheson
导演: Francis Lawrence
主演: Will Smith / Alice Braga / Charlie Tahan
官方网站: http://iamlegend.warnerbros.com/
上映年度: 2007
语言: 英语
制片国家/地区: 美国
imdb链接: tt0480249

我看过这部电影 修改 删除

我的评价: ★★★★★力荐

★★★★★ 2851
★★★★☆ 8146
★★★☆☆ 6643
★★☆☆☆ 968
★☆☆☆☆ 140

推荐

放在你的blog里!

增改描述、海报图片

喜欢看"这部电影"的人也喜欢:

机械公敌

全民超人


国家宝藏2：古籍秘辛

通缉令


科洛弗档案

钢铁侠

心灵传输者


300


迷雾


国家公敌

tems

豆瓣猜你可能感兴趣的电影


300 / 300死士 / 300斯巴达勇士

Gerard Butler / Vincent Regan / Lena He

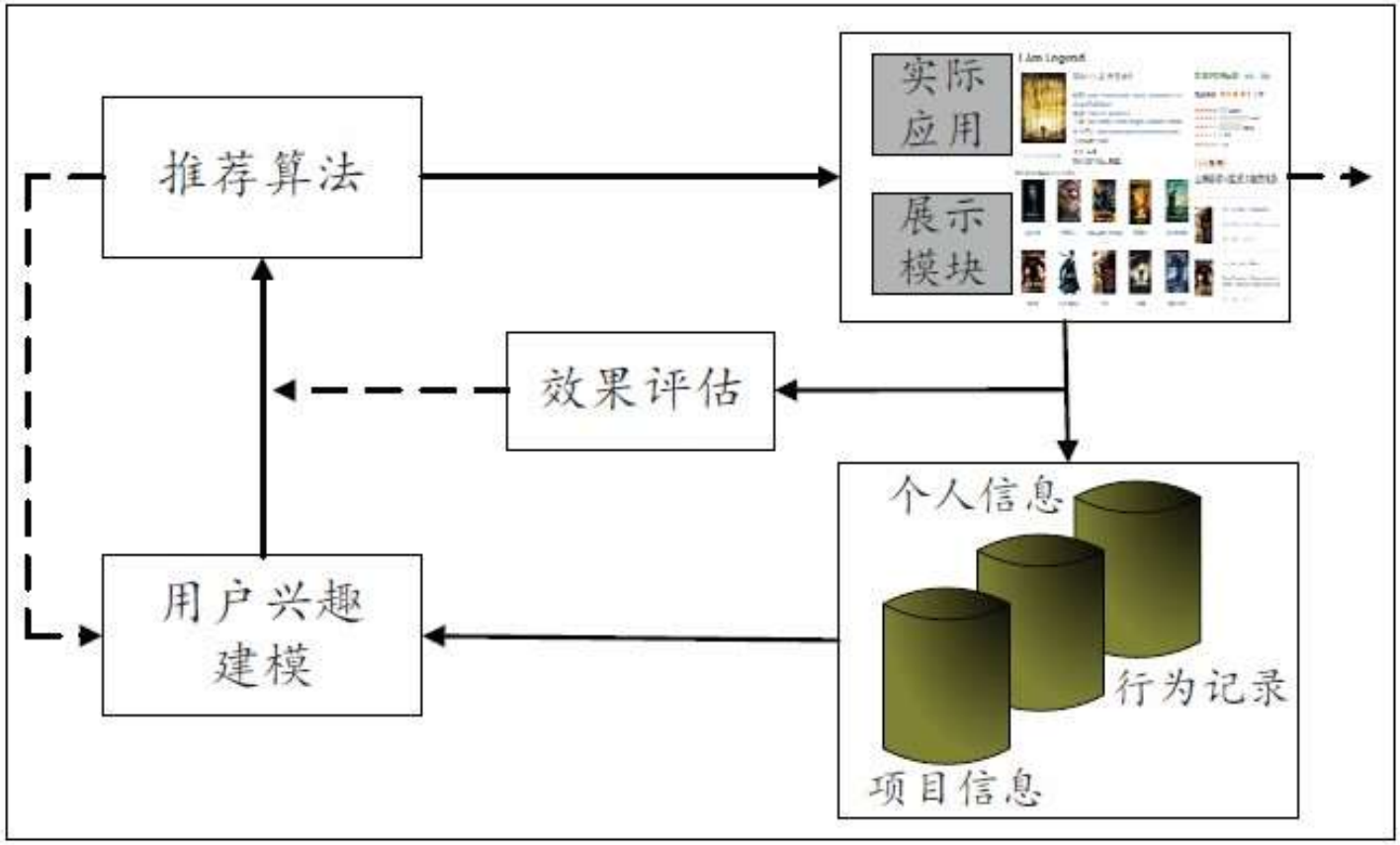
看过 想看 设兴趣

Iron Man / 铁人 / 钢铁侠

Robert Downey Jr. / Terrence Howard /
钢铁侠 / Art Marcum / Matt Holloway / Ma

看过 想看 设兴趣

数据库的位置和作用



国家级教育资源库

关于英语科目考试成绩的说明

[发布时间:2018-11-27 阅读量:1570]

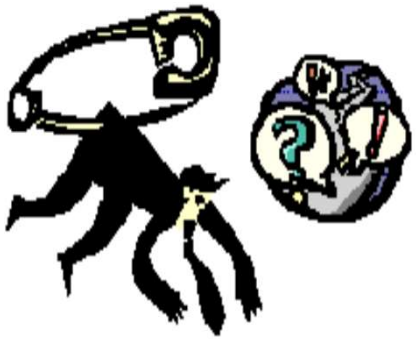
浙江省高考英语科目一年安排2次考试，考生可报考2次，选用其中较高1次的成绩。在2018年11月刚结束的英语科目考试中，根据答卷试评情况，发现部分试题与去年同期相比难度较大。为保证不同次考试之间的试题难度大体相当，浙江省招委组织专家研究论证，在制订评分细则时，决定面向所有考生，对难度较大的第二部分（阅读理解）、第三部分（语言运用）的部分试题进行难度系数调整，实施加权赋分。其他试题未作调整。

浙江省教育考试院

2018年11月27日

数据库与数据挖掘技术的区别

- 数据库技术是从大量数据里找某个具体数据，或是简单的数据统计信息。数据库技术做的事就好比在草堆里去找别针。



- 数据挖掘技术找的不是一个已存在那里的信息。它做的事就好比是要设法搞清楚在草堆里有一根针，会造成什么样的后果。

数据库与数据挖掘国际会议

- 数据库顶级会议介绍：VLDB、SIGMOD、ICDE

中国计算机学会推荐国际学术会议
（数据库，数据挖掘与内容检索）

一、A类

序号	会议简称	会议全称		出版社	网址
1	SIGMOD	ACM Conference on Management of Data	1974	ACM	http://www.sigmod.org
2	SIGKDD	ACM Knowledge Discovery and Data Mining	1995	ACM	http://www.acm.org/sigkdd/
3	SIGIR	International Conference on Research an Development in Information Retrieval	1978	ACM	http://www.acm.org/sigir/
4	VLDB	International Conference on Very Large Data Bases	1975	Morgan Kaufmann/ACM	http://www.vldb.org
5	ICDE	IEEE International Conference on Data Engineering	1984	IEEE	http://www.icde.org/

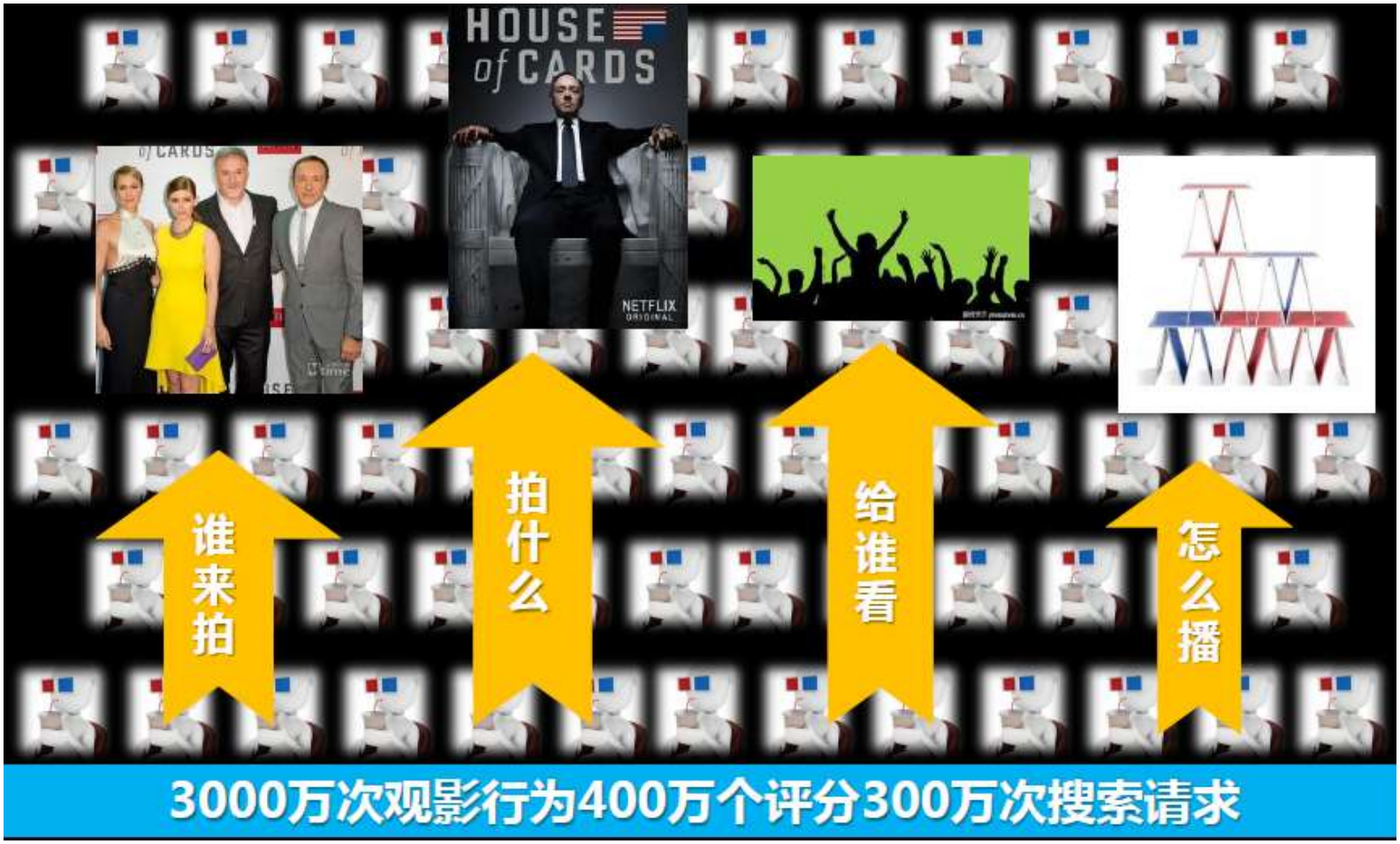
从数据中可以挖到什么？

AlphaGo

AlphaGo不是普通的计算机程序

- 它初始输入了3万多幅专业棋手对局的棋谱数据
- 它能够快速地学习，积累了3000万盘棋局，快速吸取经验
- 与其说人机对弈，不如说是李世石与多名大师之间的对弈

从数据中可以挖到什么？



大卫·芬奇
凯文·史派西

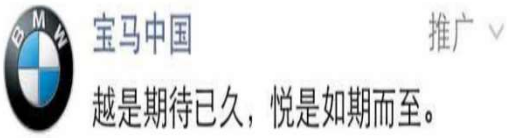
老版《纸牌屋》

喜欢老版纸牌屋及同类剧的用户

13集同时上线

从数据中可以挖到什么？

类似的场景—计算广告：
最大化数据和流量的价值



DMP

(男、28岁、
白领、汽车)

点击率
预测

广告主
平台

广告交
易平台

从数据中可以挖到什么？

- **阿里：以15年“双11”大促为例**
  - **首次实现了全面个性化**
  - **购买预测模型每几个小时重新训练一次**
  - **全局优化几亿商品的相互推荐链**

大促推荐算法-会场个性化

会场入口个性化

会场首图个性化

会场内部个性化

- 利用大数据，从赋能自己，到赋能商家

类似的场景—双11的狂欢

阿里巴巴集团CEO张勇 双11演讲

这次双十一的一大亮点是，我们基于大数据的无线产品和技术的创新，使得整个运营效率有了大幅度提升。在双十一期间，淘系的活跃用户得到了充分的引导和互动，得到了大量个性化的展示和推荐，事实证明了大数据的巨大威力。我们用大数据赋能了双十一，赋能了我们自己的运营能力。我们还要更上一层楼，利用大数据赋能给所有的商家，帮助他们运营好消费者，这样才能让我们在大数据时代践行“让天下没有难做的生意”的使命。

从数据中可以挖到什么？

用户的“纠结”无处不在

- 体现在用户消费（如：点击记录）的过程中

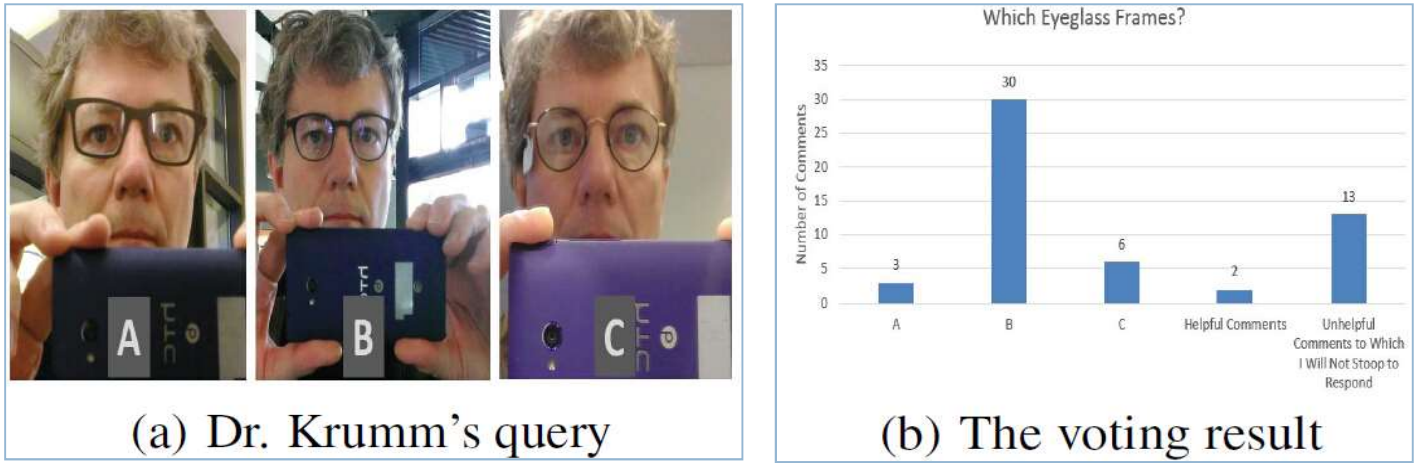


Figure 1: An example of Indecisiveness from Dr. John Krumm.

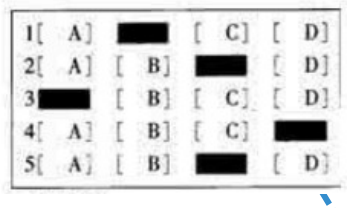
建模纠结心理可以提供更好的服务

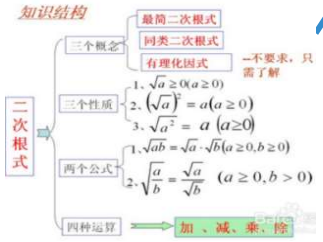
- 量化产品（商家）竞争关系
- 引导用户消费，增加消费成功率

从数据中可以挖到什么？

- 数据蕴含着巨大的价值
  - 教育方面：“因材施教”

学生的**学习行为数据**

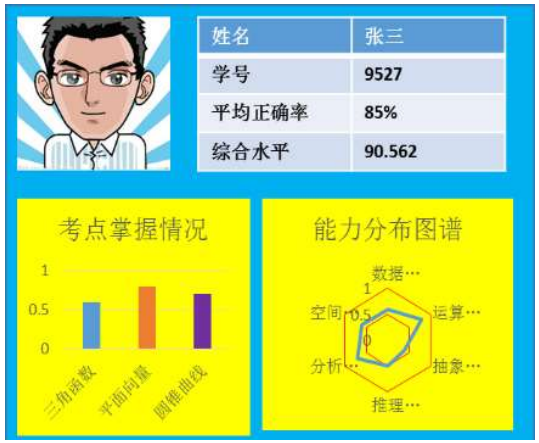




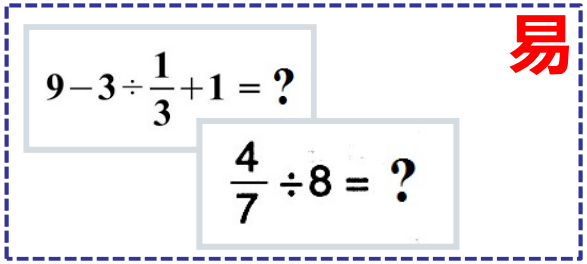
试题-知识点

大数据分析

学生认知水平画像



试题难度等特征的预测

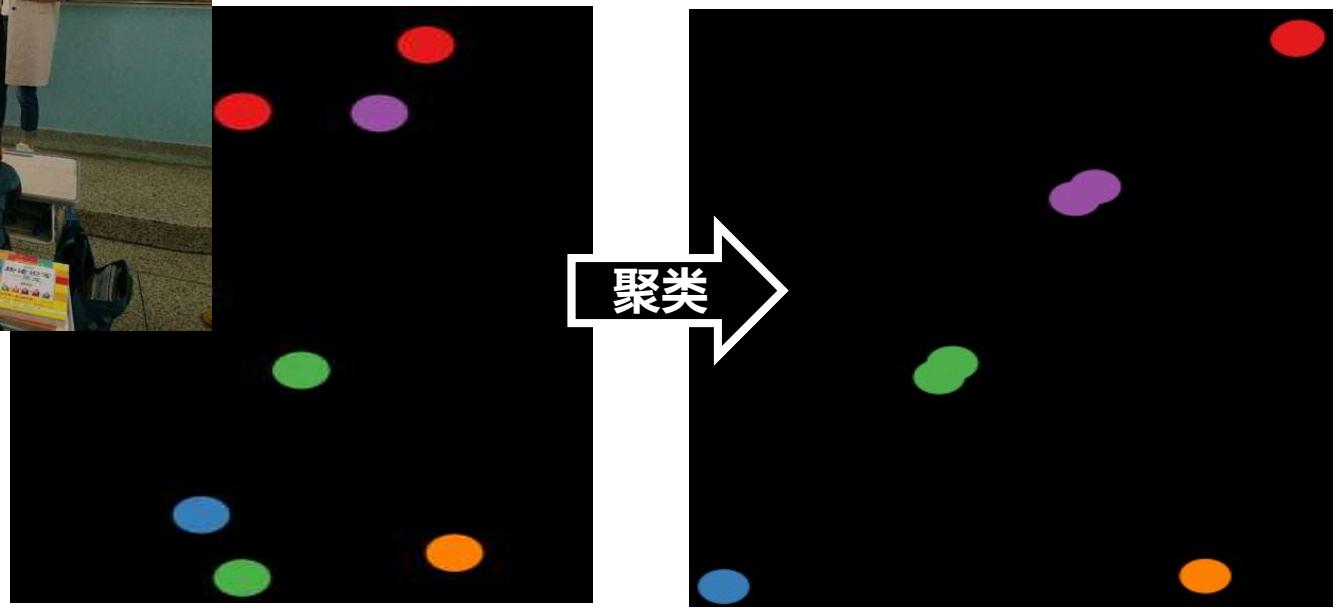


个性化学习推荐

从数据中可以挖到什么？

- 数据蕴含着巨大的价值
  - 教育方面：“优化教师教学”

根据考试数据对班级进行简单聚类，根据聚类结果，发现70%的类里，两个班集是同位授课教师



从数据中可以挖到什么？

- 社会科学方面
  - 社交媒体比问卷调查提供了更有代表性的结果
  - 智能引导社会成员的行为

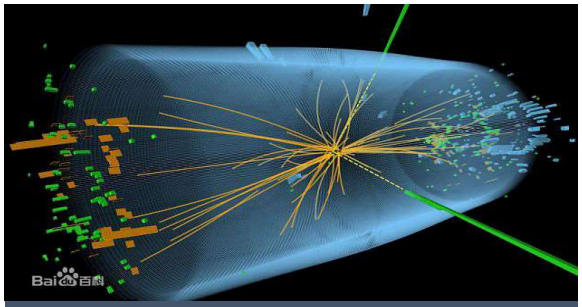
15万名奥巴马支持者在Facebook安装了“奥巴马2012”应用，而通过这个程序，总统竞选团队可以间接得到这些支持者数百万的Facebook好友信息。

有一种说法称，特朗普的团队聘用数据分析公司，做了精准的广告投放，影响了那些徘徊不定的选民，拿下了决定性的关键州选举人票

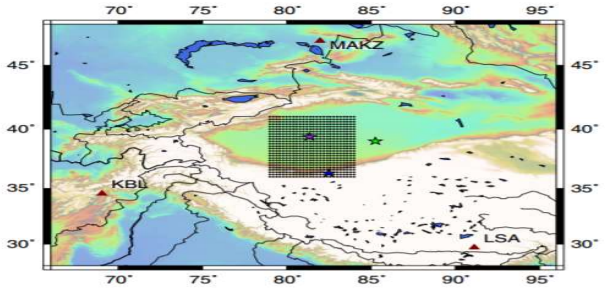
从数据中可以挖到什么？

- 科学技术研究方面
  - 大数据推动科学新技术发现

天文大数据搜索新星



物理大数据预测分子属性



大数据地震速报、余震预测

生物大数据改良基因

科学
大数据

ms

专利数据挖掘保护知识产权

从数据中可以挖到什么？

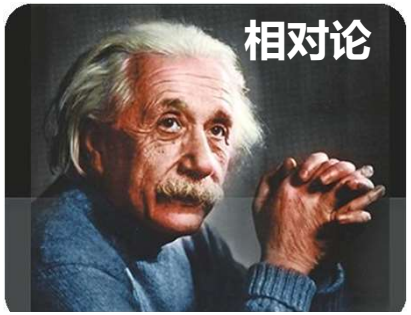
- **科技大数据来自于物理世界**
  - 科学实验数据或传感数据
  - 技术描述型数据—专利、论文
- **集多种特点于一身**
  - 采集的高代价性
  - 复杂性
    - 超高维度
    - 高度计算复杂性
    - 高度的不确定性
  - 学科知识壁垒
  - 信息与通信技术高度集成性

单一学科 → 数据驱动 / 多学科交叉

关系型数据库的鼻祖Jim Gray（右）

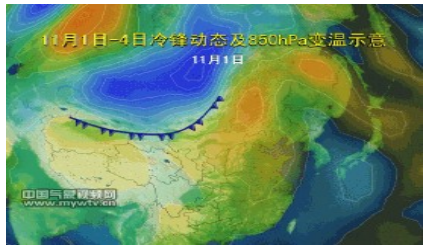
从数据中可以挖到什么？

- **2007年，Jim Gray总结出了四个科学范式**

几千年前	几百年前	几十年前	今天
经验科学 • **第一范式** • 以**归纳法**为主，带有盲目性的观测和实验 • **科学实验**  自由落体实验	**理论科学** • **第二范式** • 以**演绎法**为主，关注理论总结和理性概括 • **数学模型**  相对论	**计算科学** • **第三范式** • 重视**数据模型构建**、**定量分析方法**，利用计算机来分析和解决 • **科学计算**  冯诺依曼计算机	**数据密集型科学** • **第四范式** • 先有了**大量的已知数据**，然后通过计算得出之前未知的理论 • **机器学习**  天文大数据

从数据中可以挖到什么？

数据分析挖掘技术是解决众多国家重大现实需求问题的共性基础
---数据驱动的人工智能

社会媒体、人口流动、居住交通数据	医疗、医保、健康、影像等大数据	环境、气象、交通、社会发展等大数据	交通流、医疗、商业、环境、劳动力等数据
			
突发事件预测、关键人群监测	医疗诊断方案	环境治理	城市智慧管理

数据库是基础中的基础!

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。

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)

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
  - 文本、图形、图像、音频、视频、学生的档案记录、货物的运输情况等
- 数据的特点
  - 数据与其语义是不可分的

数据举例

- 数据的含义称为数据的语义，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：计算机系2003级学生人数

    语义4：。。。。。。

数据举例

- 学生档案中的学生记录

  （李明，男，197205，江苏南京市，计算机系，1990）

  - 语义：学生姓名、性别、出生年月、籍贯、所在院系、入学时间
  - 解释：李明是个大学生，1972年5月出生，江苏南京市人，1990年考入计算机系

请给出另一个解释和语义

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
  - 冗余度较小
  - 数据独立性较高
  - 易扩展

三、数据库管理系统

- 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
- DBMS的用途
  - 科学地组织和存储数据、高效地获取和维护数据

数据库在计算机系统中的位置

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
- 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

DBMS的主要功能（续）

- 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

DBMS的主要功能（续）

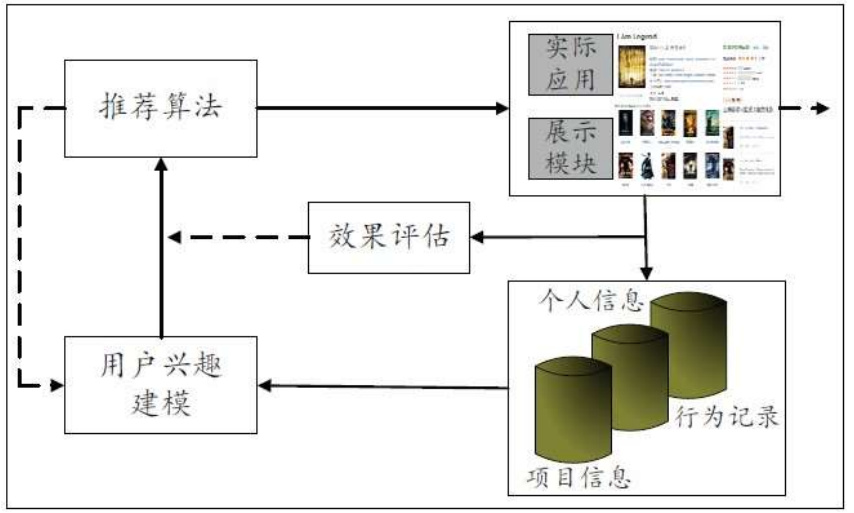
- 数据库的建立和维护功能(实用程序)
  - 数据库初始数据装载转换
  - 数据库转储
  - 介质故障恢复
  - 数据库的重组织
  - 性能监视分析等
- 其它功能
  - DBMS与网络中其它软件系统的通信
  - 两个DBMS系统的数据转换
  - 异构数据库之间的互访和互操作

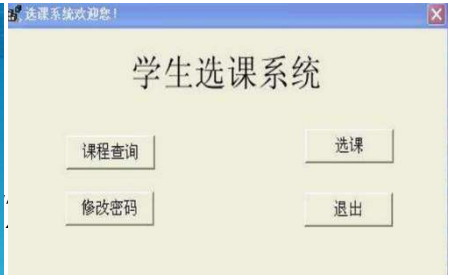
四、数据库系统

- 什么是数据库系统（**Database System，简称DBS**）

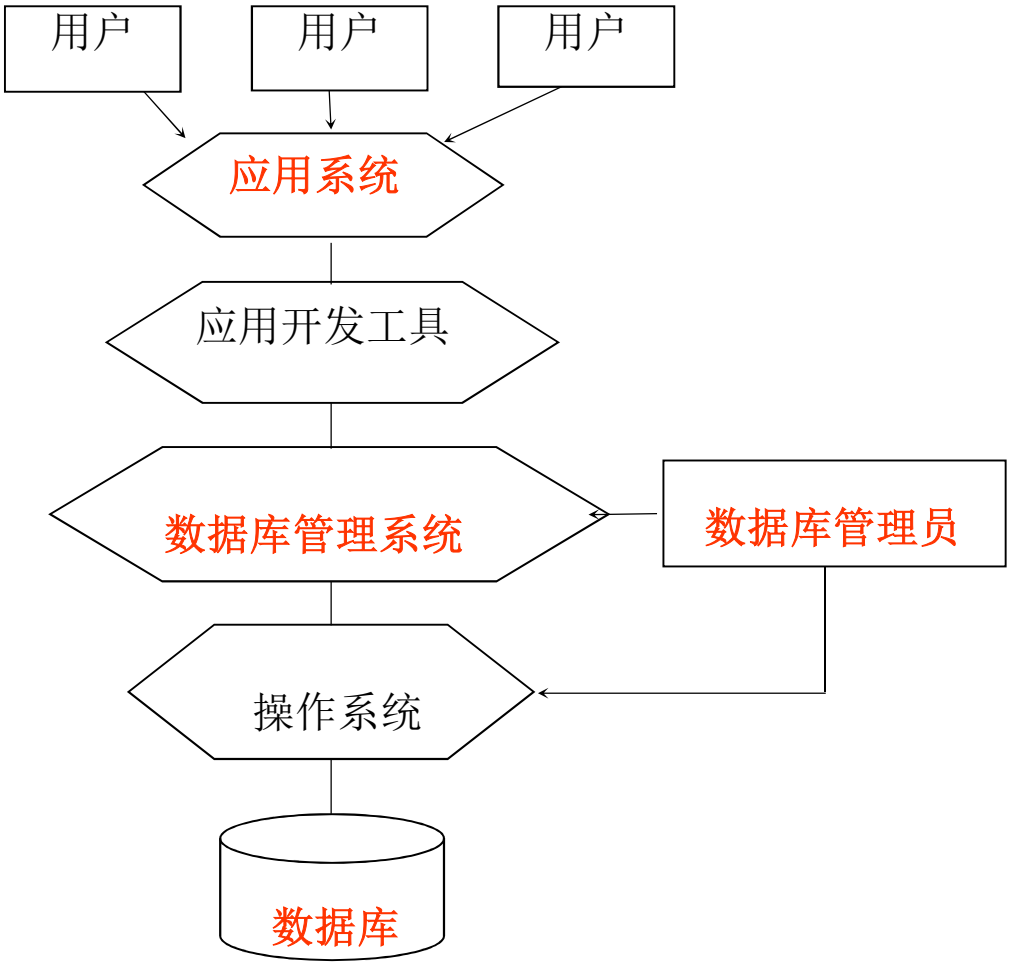
  在计算机系统中引入数据库后的系统构成

- 数据库系统的构成
  - 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





数据库系统



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

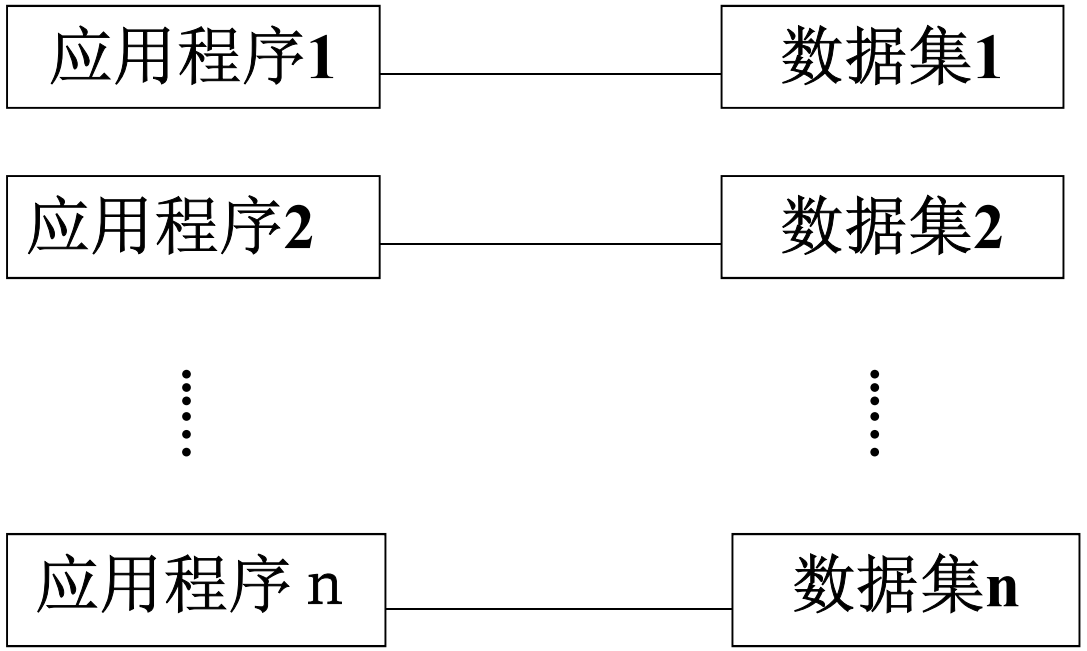
一、人工管理阶段

- 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一一对应关系

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、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

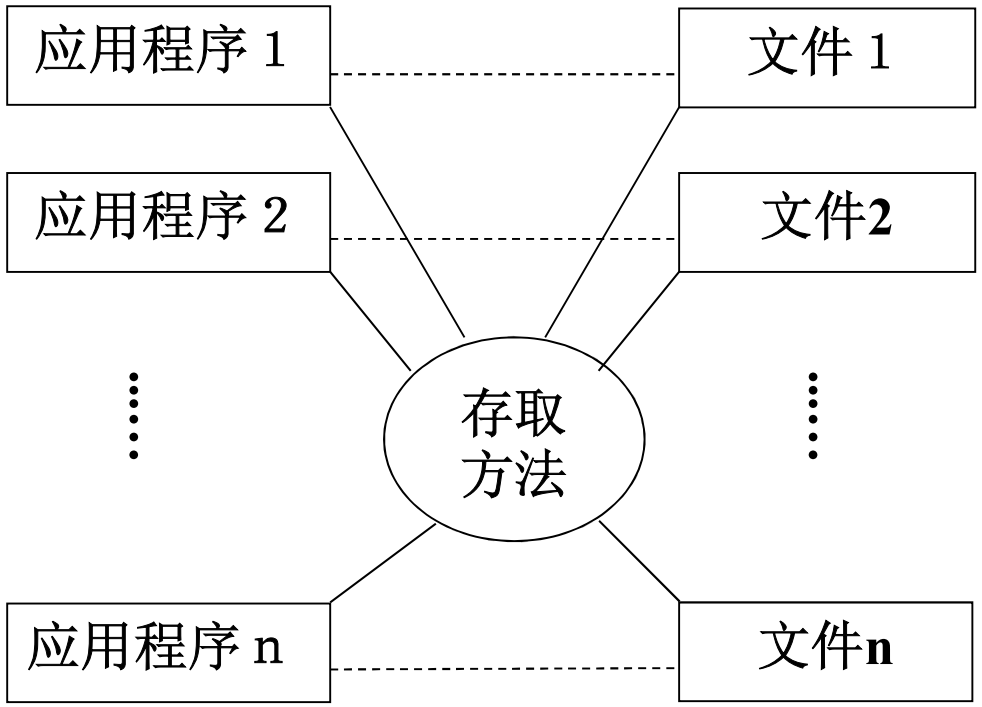
  数据的共享程度：共享性差、冗余度大

  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，数据的逻辑结构改变必须修改应用程序

  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

文件系统中数据的结构

- 记录内有结构。
- 数据的结构是靠程序定义和解释的。
- 数据只能是定长的。
  - 可以间接实现数据变长要求，但访问相应数据的应用程序复杂了。
- 文件间是独立的，因此数据整体无结构。
  - 可以间接实现数据整体的有结构，但必须在应用程序中对描述数据间的联系。
- 数据的最小存取单位是记录。

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

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

数据结构化

- 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

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

数据由DBMS统一管理和控制

- DBMS提供的数据控制功能
  - (1)数据的安全性（Security）保护 （第4章）

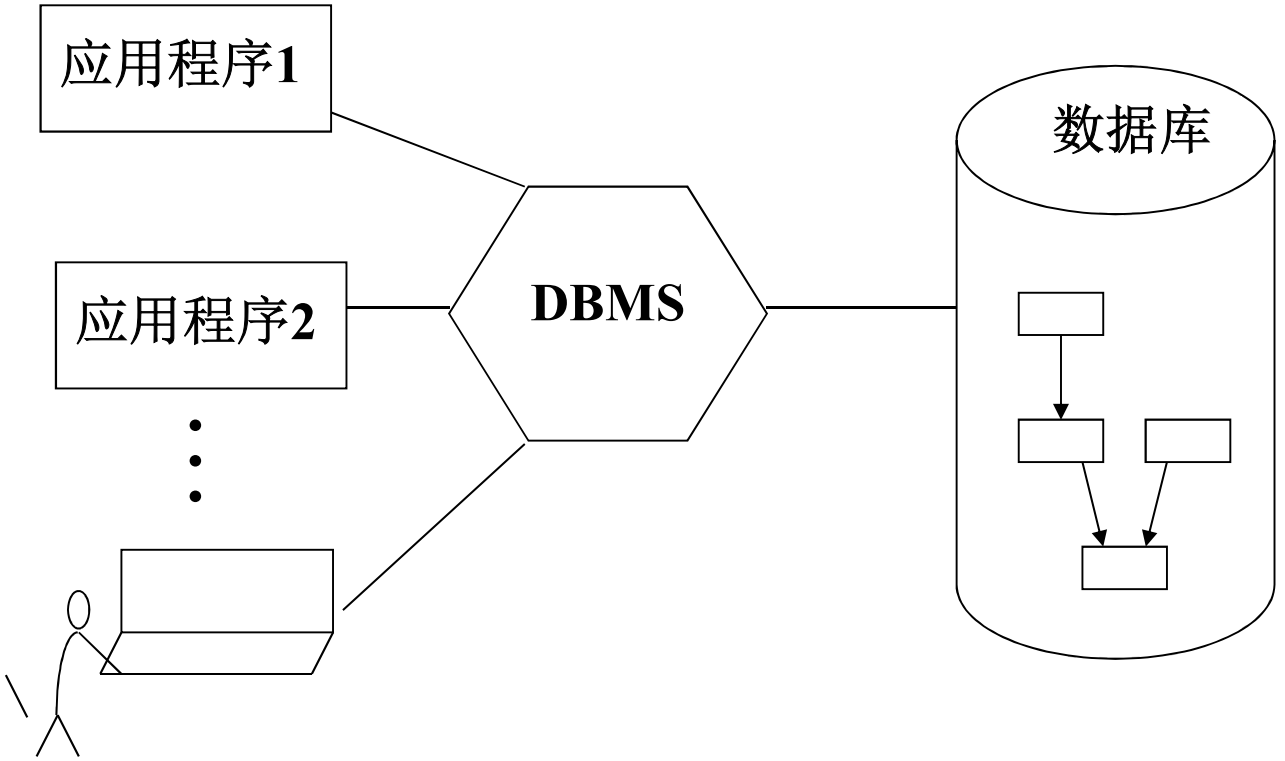
    保护数据，以防止不合法的使用造成的数据的泄密和破坏。
  - (2)数据的完整性（Integrity）检查 （第5章）

    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 （第11章）

    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） （第10章）

    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

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

1.2 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

数据模型

- 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

1.2.1 两大类数据模型

- 数据模型分为两类（分属两个不同的层次）

(1) 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。

(2) 逻辑模型和物理模型

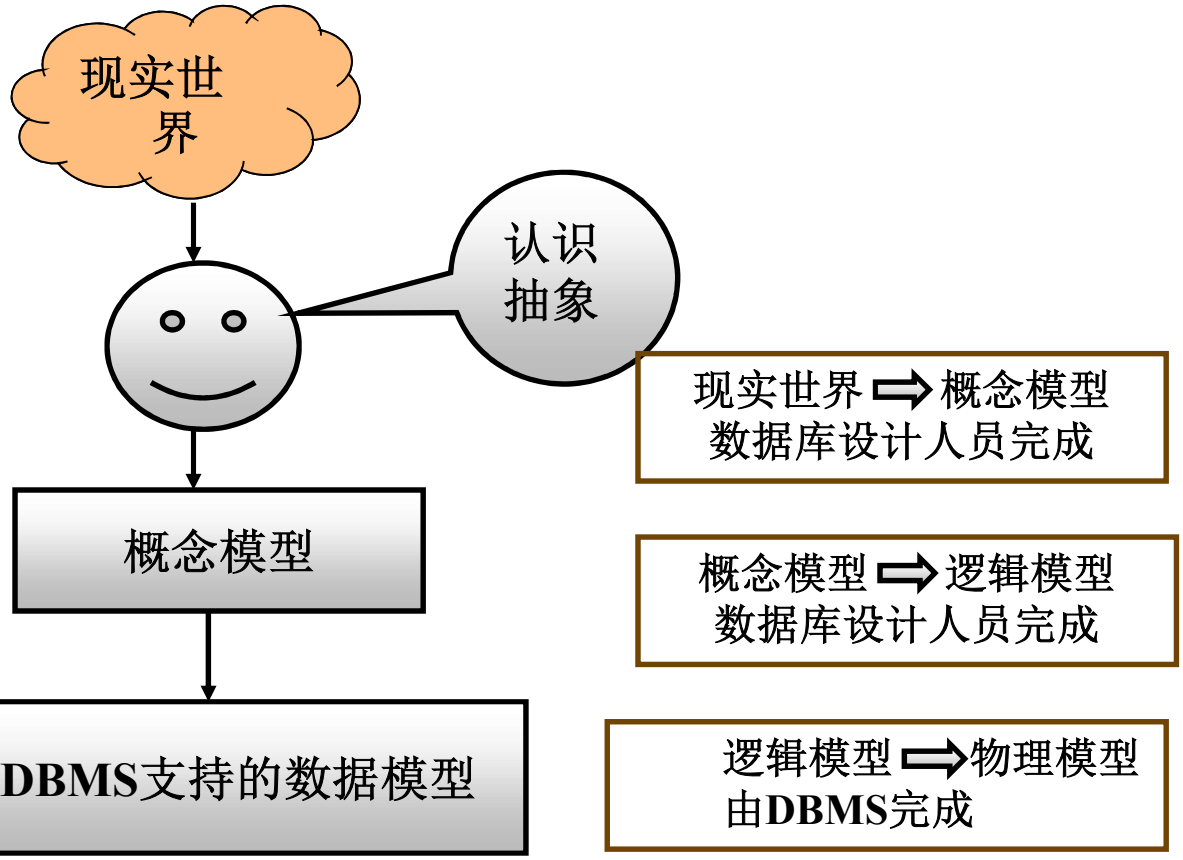
- 逻辑模型主要包括网状模型、层次模型、关系模型、面向对象模型等，按计算机系统的观点对数据建模，用于**DBMS**实现。
- 物理模型是对数据最底层的抽象，描述数据在系统内部的表示方式和存取方法，在磁盘或磁带上的存储方式和存取方法。

两大类数据模型 (续)

- 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；
  - 把概念模型转换为某一**DBMS**支持的数据模型。

两大类数据模型 (续)

现实世界中客观对象的抽象过程



看见一个动物（现实世界），命名为狗（概念模型），怎么在计算机里存（逻辑模型），存在哪里（物理模型）

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

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

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
  - 查询
  - 更新(包括插入、删除、修改)

数据操作(续)

- 数据模型对操作的定义
  - 操作的确切含义
  - 操作符号
  - 操作规则（如优先级）
  - 实现操作的语言
- 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。

数据的完整性约束条件(续)

- 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。
    - 学生必须有学号
    - 学生选的课（课程信息存在另外一个数据表里）必须已经开设
  - 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。

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

1.2.3 概念模型

- 信息世界中的基本概念
- 两个实体型之间的联系
- 概念模型的一种表示方法
- 一个实例

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) 域（Domain）

属性的取值范围称为该属性的域。

(5) 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称为实体型

(6) 实体集（Entity Set）

同一类型实体的集合称为实体集

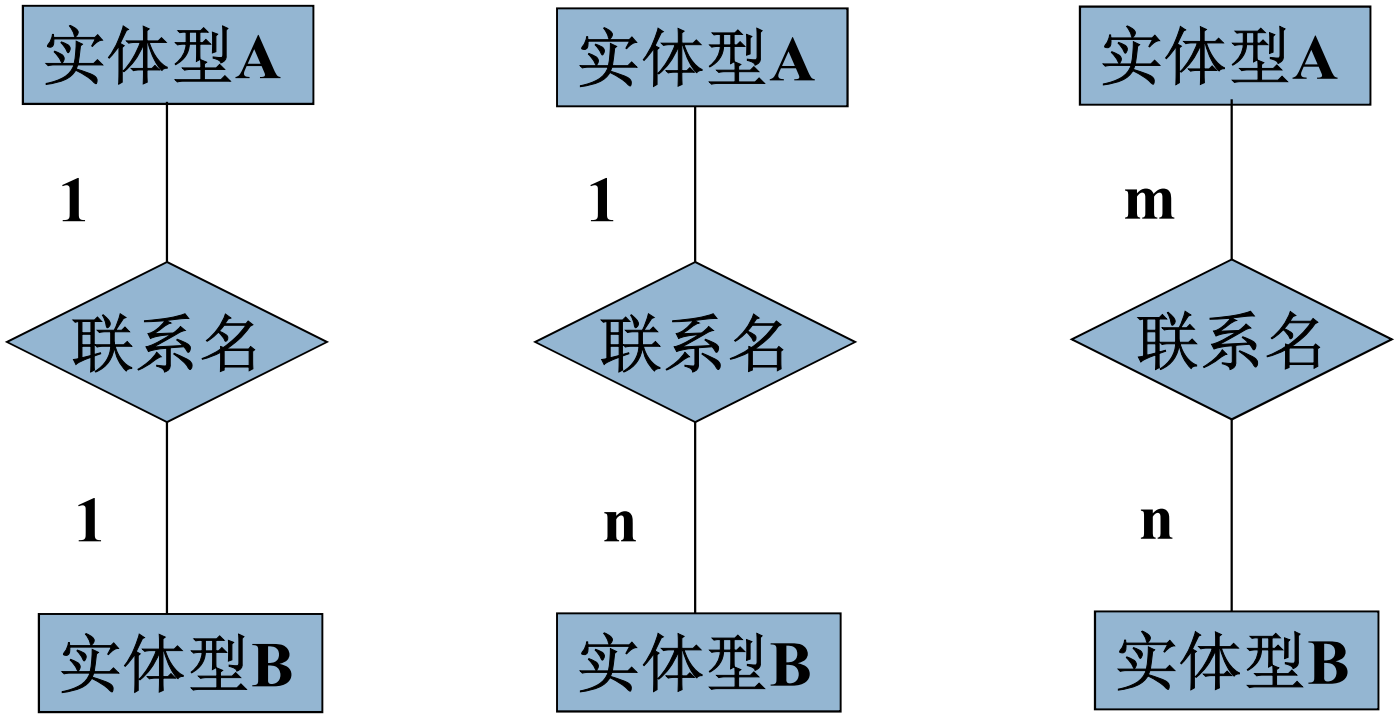
信息世界中的基本概念(续)

(7) 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

二、两个实体型之间的联系

用图形来表示两个实体型之间的这三类联系



1:1联系　　1:n联系　　m:n联系

二、两个实体型之间的联系（续）

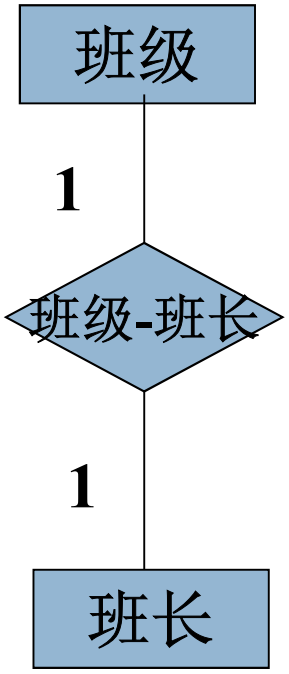
- 一对一联系（1:1）
  - 实例

    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)

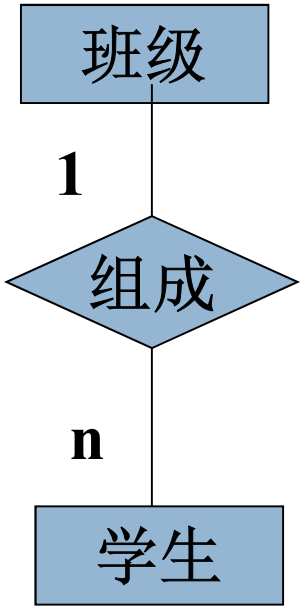
- 一对多联系（1：n）
  - 实例

    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≥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

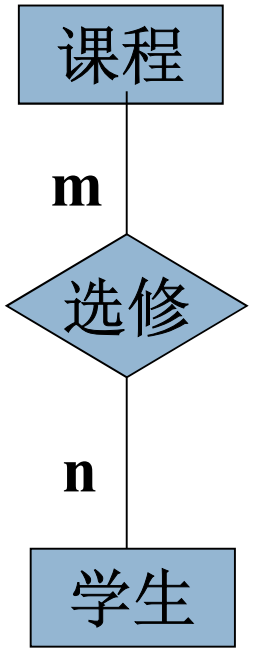
    课程与学生之间的联系：

    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≥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≥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

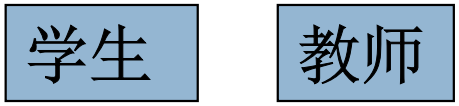
三、 概念模型的一种表示方法

- 实体－联系方法(E-R方法)
  - 用E-R图来描述现实世界的概念模型
  - E-R方法也称为E-R模型

E-R图

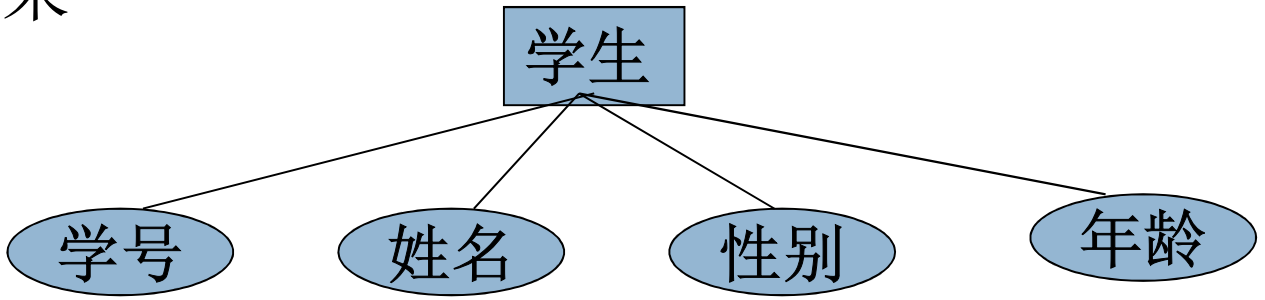
- 实体型

  用矩形表示，矩形框内写明实体名。



- 属性

  用椭圆形表示，并用无向边将其与相应的实体连接起来

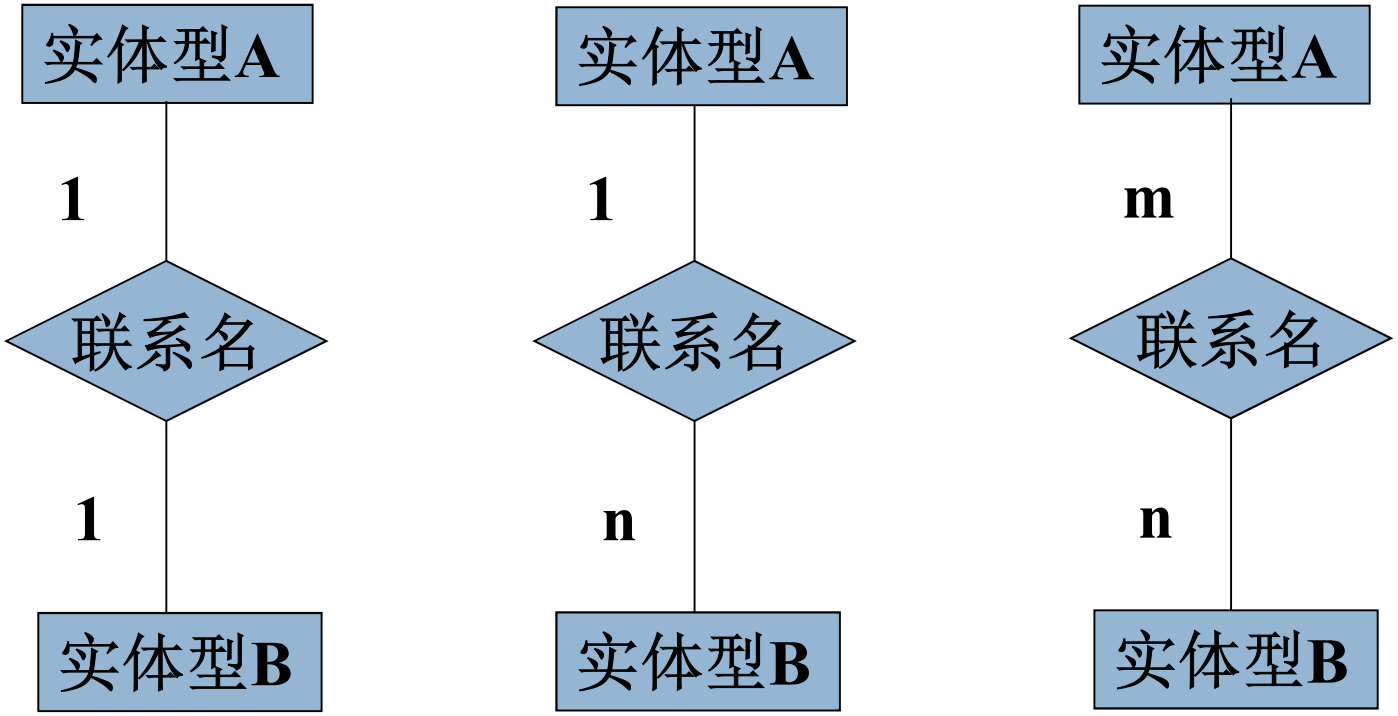


E-R图(续)

- 联系
  - 联系本身：

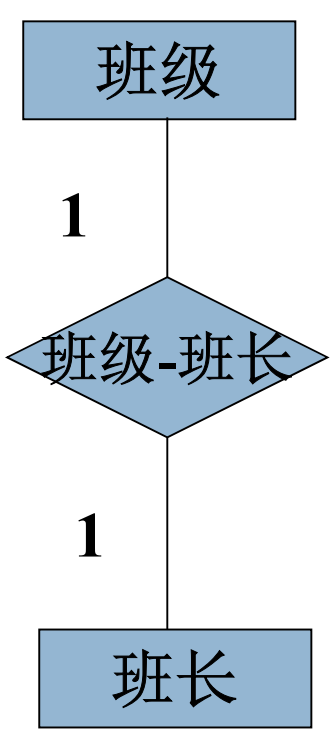
    用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时在无向边旁标上联系的类型（**1:1、1:n或m:n**）

联系的表示方法

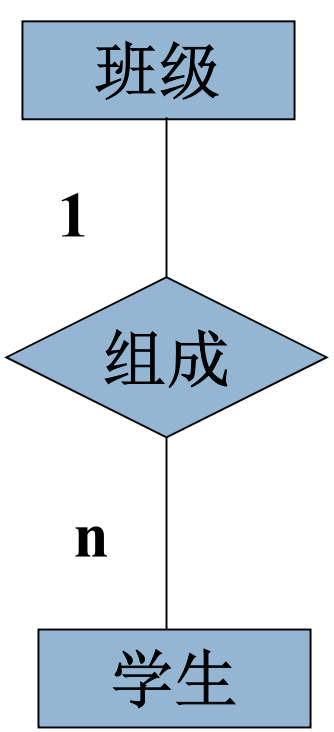


1:1联系　　1:n联系　　m:n联系

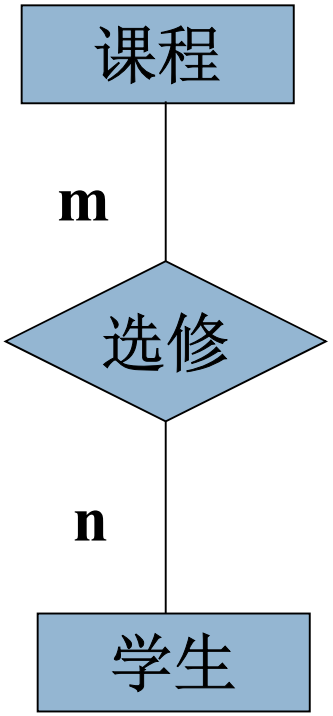
联系的表示方法示例



1:1联系



1:n联系

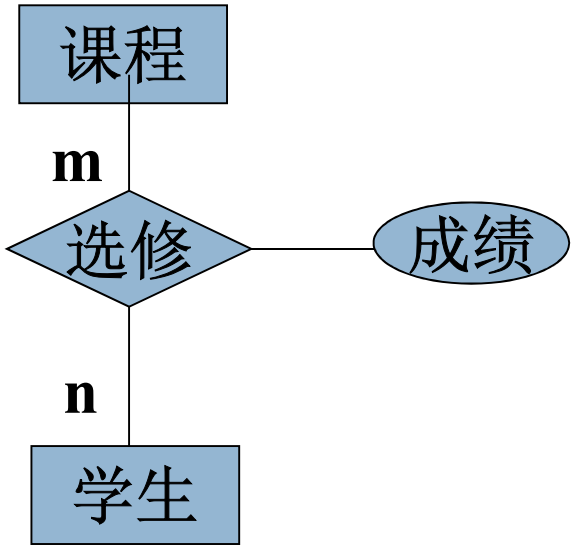


m:n联系

联系的属性

❖联系的属性：

联系本身也是一种实体型，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



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
  - 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：

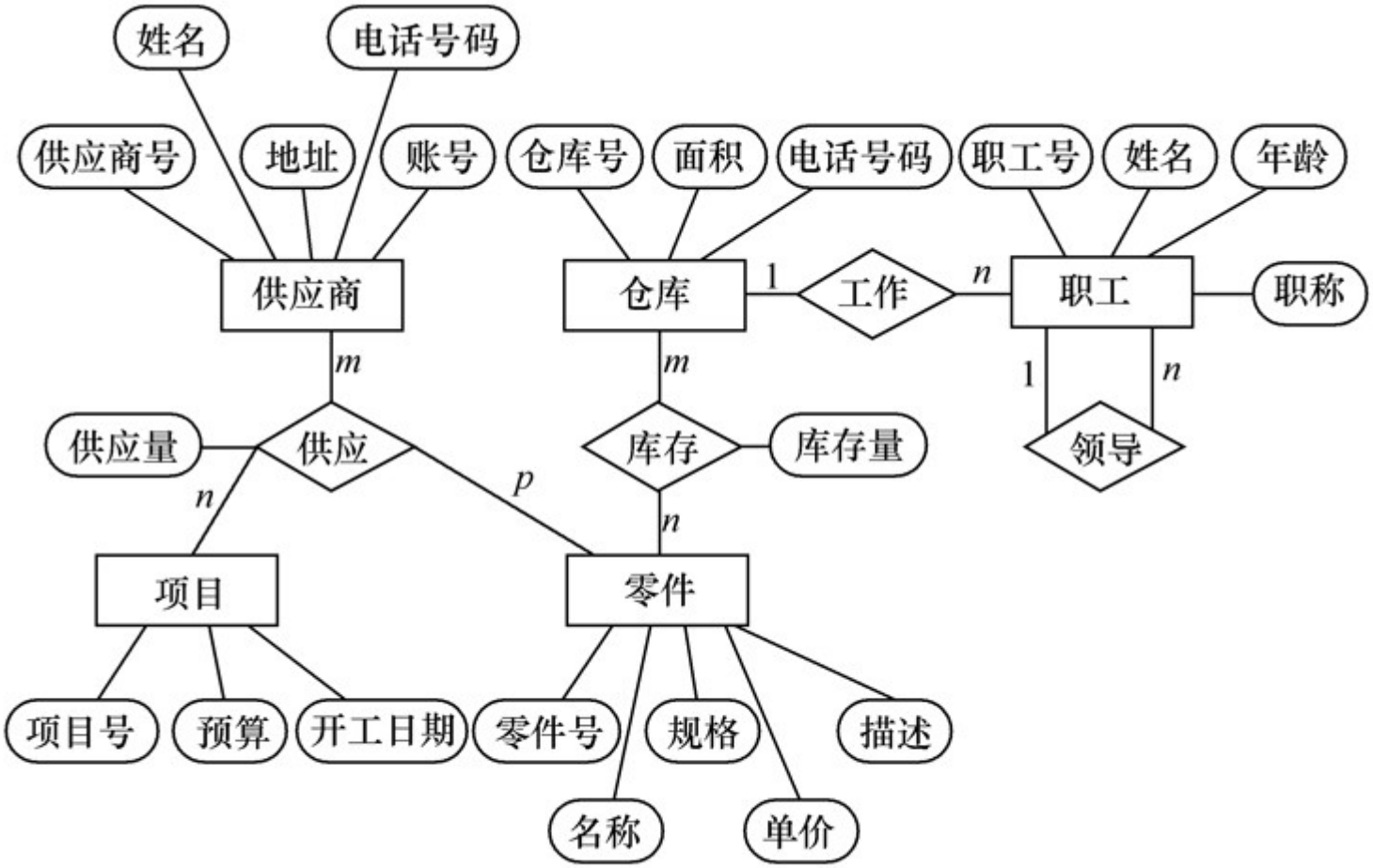
(1)一个仓库可以存放多种零件，一种零件可以存放在多个仓库中。仓库和零件具有多对多的联系。用库存量来表示某种零件在某个仓库中的数量。

(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，仓库和职工之间是一对多的联系。职工实体型中具有一对多的联系

(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

一个实例



(c) 完整的实体-联系图

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

1.2.4 最常用的数据模型

- 非关系模型
  - **层次模型(Hierarchical Model)**
  - **网状模型(Network Model)**
- 关系模型(Relational Model)
- 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（semi-structure data model）

1.2.5 层次模型

- 层次模型是数据库系统中最早出现的数据模型
- 层次数据库系统的典型代表是IBM公司的IMS

  （Information Management System）数据库管理系统
- 层次模型用树形结构来表示各类实体以及实体间的联系

一、 层次数据模型的数据结构

- 基本层次联系：两个实体集之间的一对多联系
- 层次模型

  满足下面两个条件的基本层次联系的集合为层次模型

  1. 有且只有一个结点没有双亲结点，这个结点称为根结点
  2. 根以外的其它结点有且只有一个双亲结点
- 层次模型中的几个术语
  - 根结点，双亲结点，兄弟结点，叶结点

层次数据模型的数据结构(续)

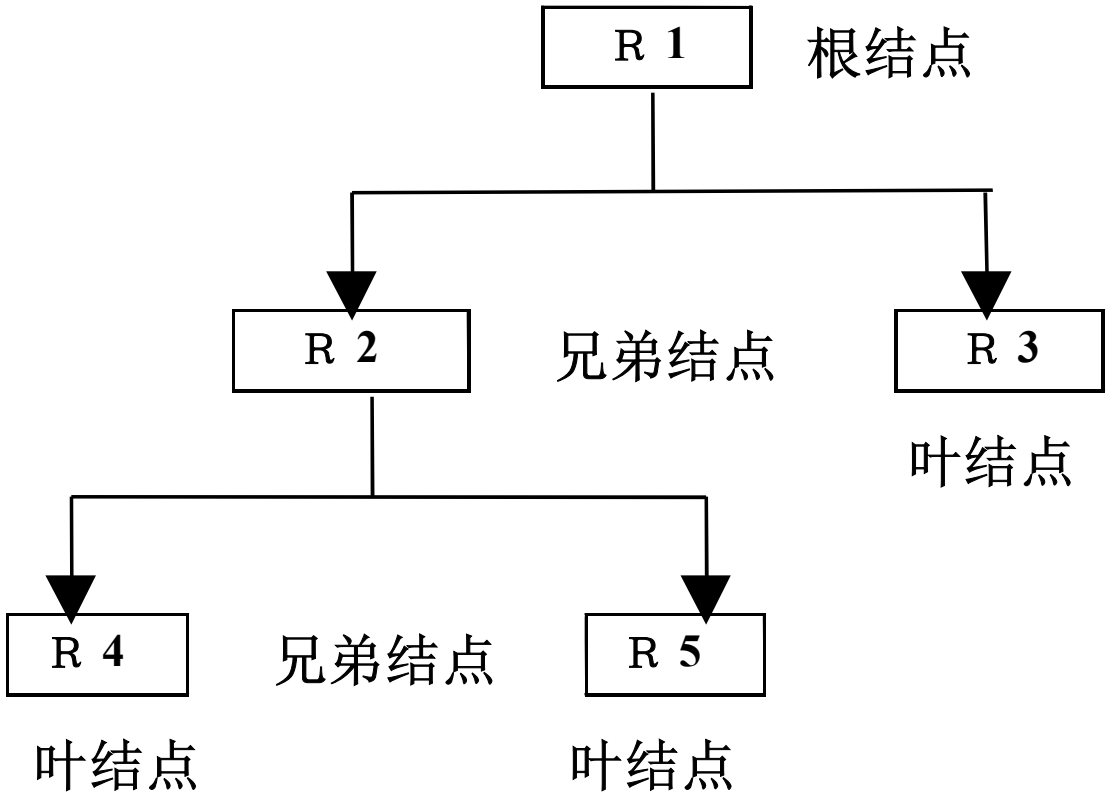


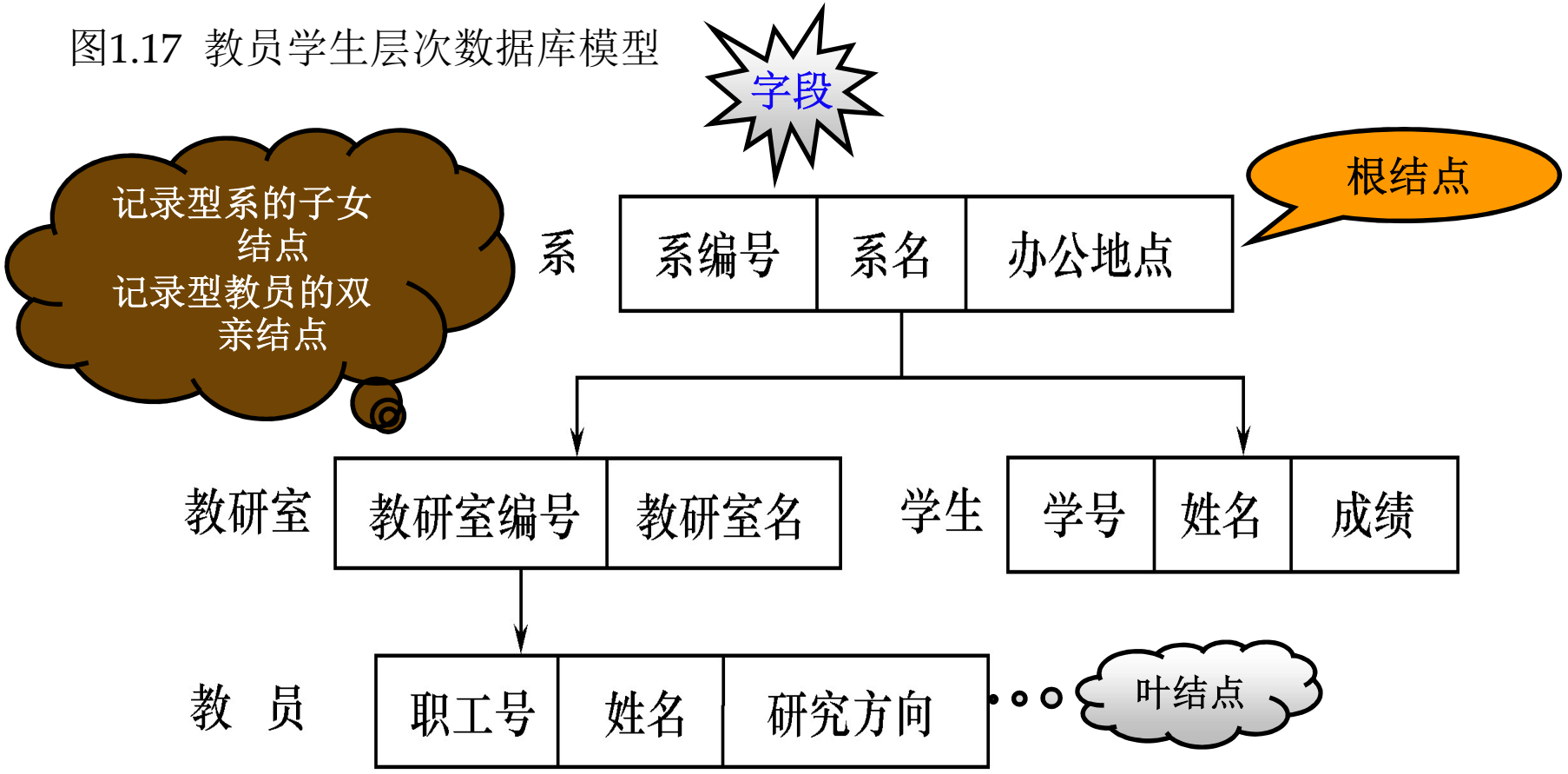
图1.16 一个层次模型的示例

层次数据模型的数据结构(续)

- 层次模型的特点：
  - 结点的双亲是唯一的
  - 只能直接处理一对多的实体联系
  - 每个记录类型可以定义一个排序字段，也称为码字段
  - 任何记录值只有按其路径查看时，才能显出它的全部意义
  - 没有一个子女记录值能够脱离双亲记录值而独立存在

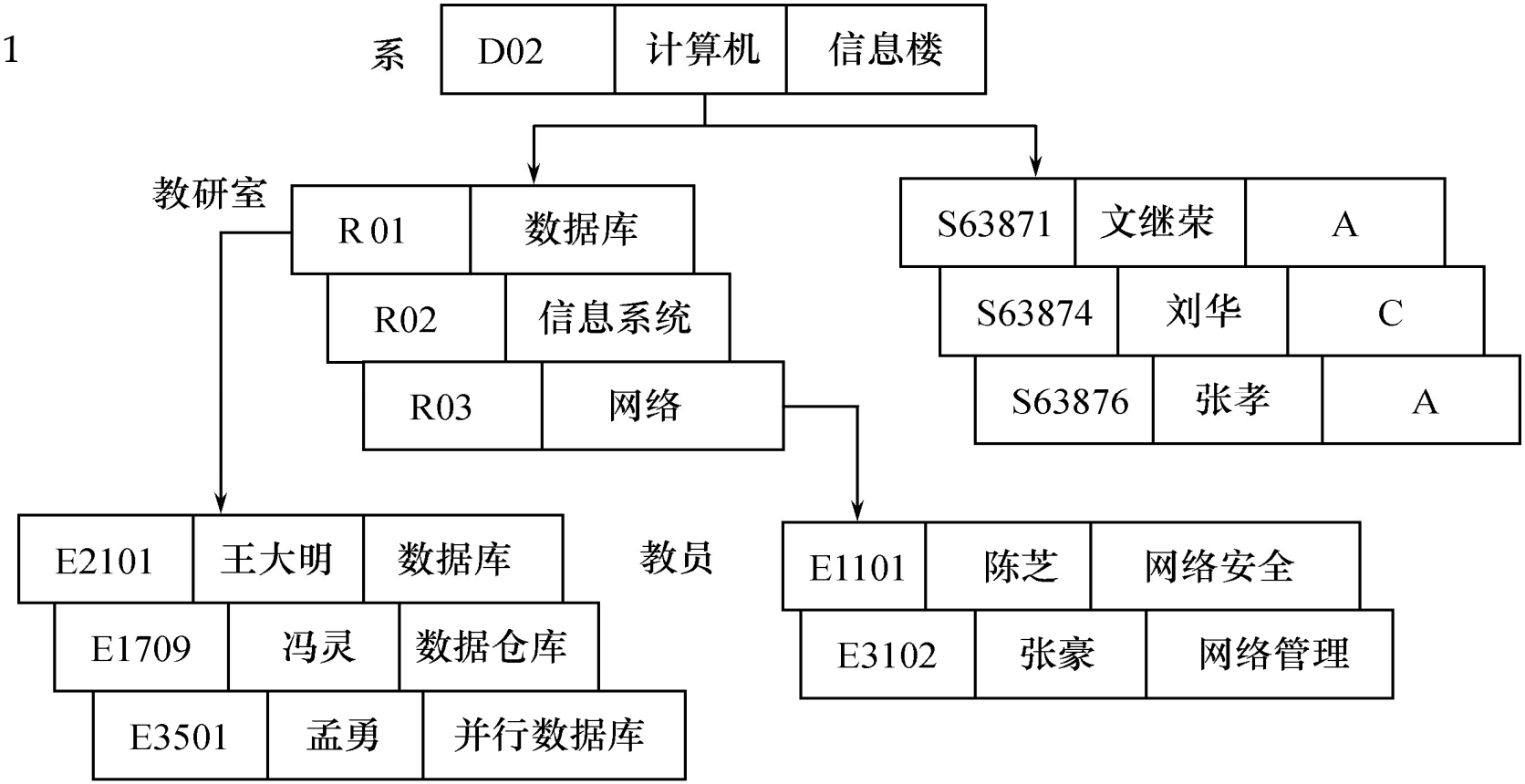
层次数据模型的数据结构(续)

图1.17 教员学生层次数据库模型



层次数据模型的数据结构(续)

图1



层次模型的数据操纵与完整性约束

- 层次模型的数据操纵
  - 查询
  - 插入
  - 删除
  - 更新

层次模型的数据操纵与完整性约束（续）

- 层次模型的完整性约束条件
  - 无相应的双亲结点值就不能插入子女结点值
  - 如果删除双亲结点值，则相应的子女结点值也被同时删除
  - 更新操作时，应更新所有相应记录，以保证数据的一致性

层次模型的优缺点

- 优点
  - 层次模型的数据结构比较简单清晰
  - 查询效率高，性能优于关系模型，不低于网状模型
  - 层次数据模型提供了良好的完整性支持
- 缺点
  - 多对多联系表示不自然
  - 对插入和删除操作的限制多，应用程序的编写比较复杂
  - 查询子女结点必须通过双亲结点
  - 由于结构严密，层次命令趋于程序化

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

1.2.6 网状模型

- 网状数据库系统采用网状模型作为数据的组织方式
- 典型代表是**DBTG**系统：
  - 亦称CODASYL系统
    - 美国数据系统语言委员会CODASYL
  - 70年代由DBTG提出的一个系统方案
    - CODASYL下属的数据库任务组DBTG
  - 奠定了数据库系统的基本概念、方法和技术
- 实际系统
  - Cullinet Software Inc.公司的 IDMS
  - Univac公司的 DMS1100
  - Honeywell公司的IDS/2
  - HP公司的IMAGE

查尔斯·巴赫曼

第一个没有博士学位的图灵奖获得者，第一个工程学背景而不是科学背景的图灵奖，第一个因将计算机应用于工商管理而赢得图灵奖，第一个因一个特定的软件而赢得图灵奖，第一个在职业生涯完全在企业中度过的图灵奖获得者。他的主要贡献不是在学术界任教研工作，而是在工业界开发实际的产品。

1.网状数据模型的数据结构

- 网状模型

  满足下面两个条件的基本层次联系的集合：

  1. 允许一个以上的结点无双亲；

  2. 一个结点可以有多于一个的双亲。

网状数据模型的数据结构（续）

- 表示方法(与层次数据模型相同)

  实体型：用记录类型描述

  每个结点表示一个记录类型（实体）

  属性：用字段描述

  每个记录类型可包含若干个字段

  联系：用结点之间的连线表示记录类型（实体）之间的一对多的父子联系

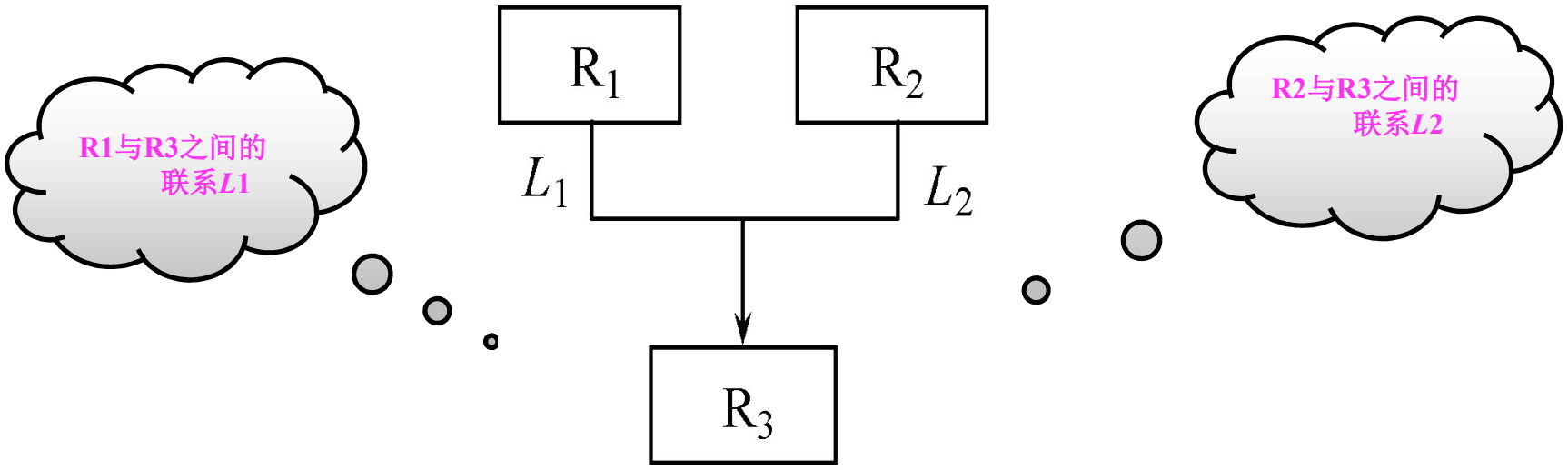
网状数据模型的数据结构（续）

- 网状模型与层次模型的区别
  - 网状模型允许多个结点没有双亲结点
  - 网状模型允许结点有多个双亲结点
  - 网状模型允许两个结点之间有多种联系（复合联系）
  - 网状模型可以更直接地去描述现实世界
  - 层次模型实际上是网状模型的一个特例

网状数据模型的数据结构（续）

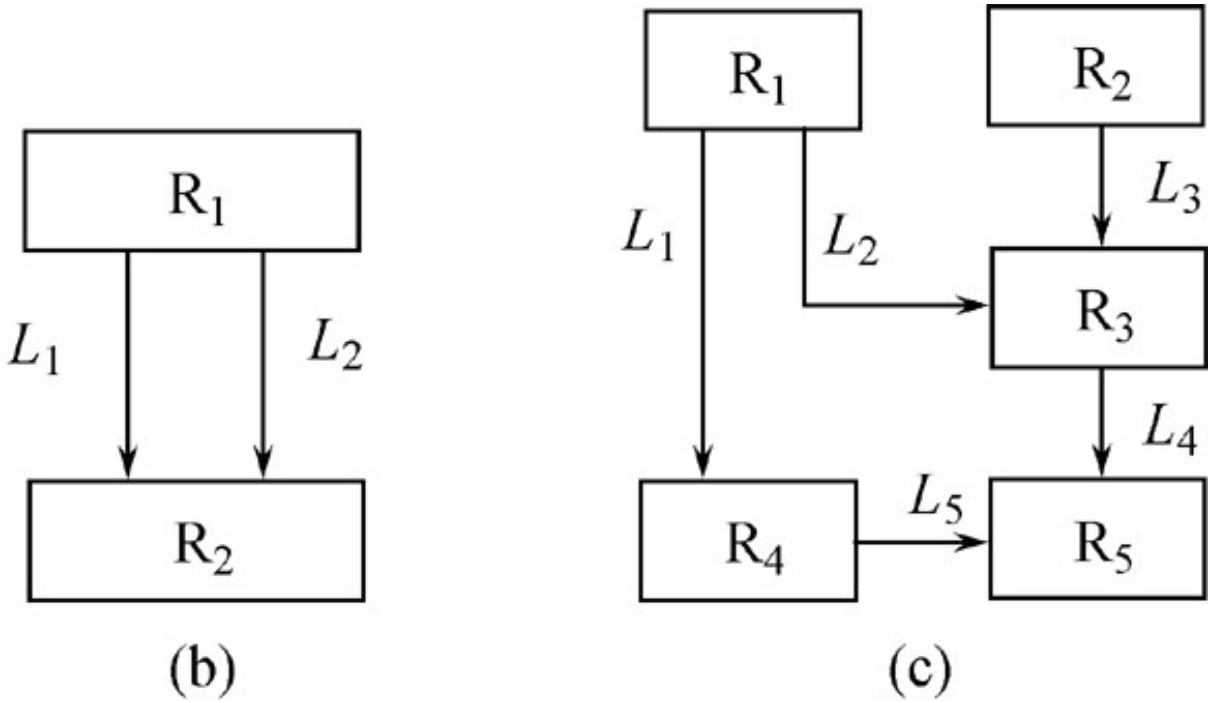
- 网状模型中子女结点与双亲结点的联系可以不唯一

  要为每个联系命名，并指出与该联系有关的双亲记录和子女记录



(a)

网状数据模型的数据结构（续）



网状模型的例子

网状数据模型的数据结构（续）

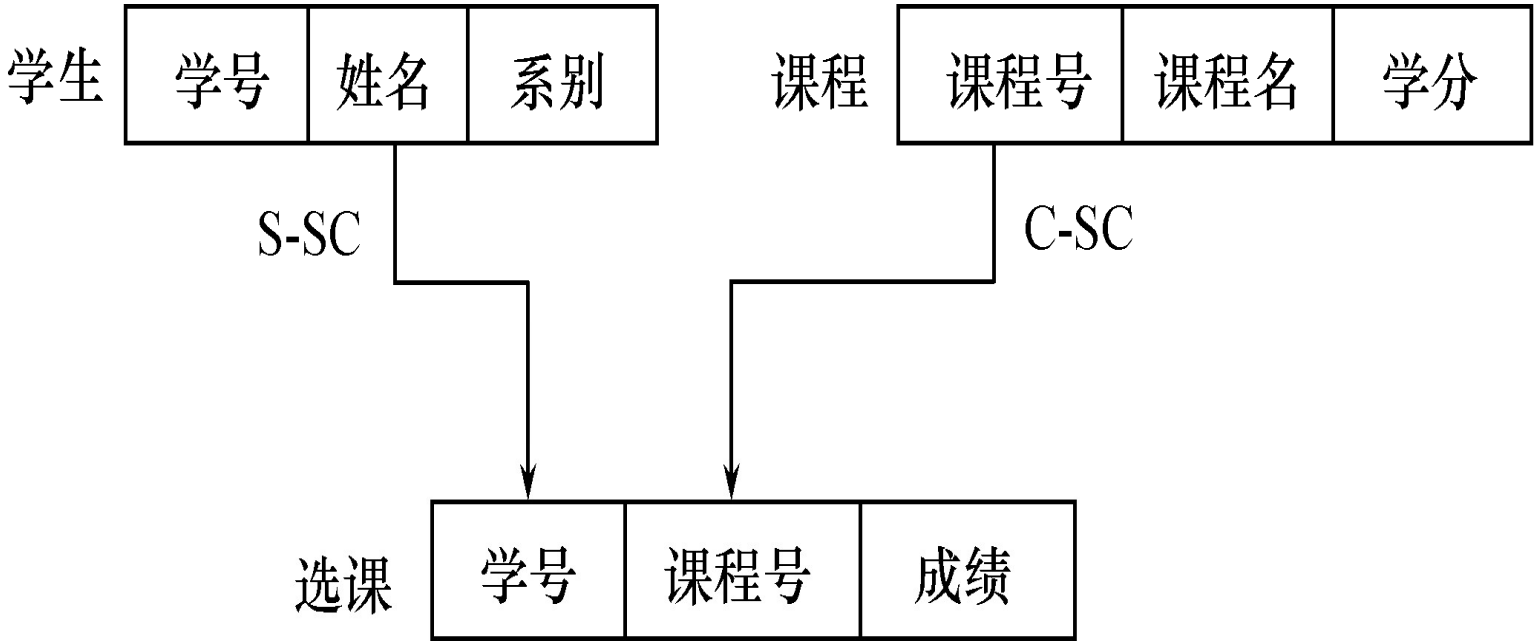
例如：一个学生可以选修若干门课程，某一课程可以被多个学生选修，学生与课程之间是多对多联系

- 引进一个学生选课的联结记录，由3个数据项组成
  - 学号
  - 课程号
  - 成绩

  表示某个学生选修某一门课程及其成绩

网状数据模型的数据结构（续）

图1.24 学生/选课/课程的网状数据模型



网状数据模型的操纵与完整性约束（续）

- 网状数据库系统（如DBTG）对数据操纵加了一些限制，提供了一定的完整性约束
  - 码：唯一标识记录的数据项的集合
  - 一个联系中双亲记录与子女记录之间是一对多联系
  - 支持双亲记录和子女记录之间某些约束条件

网状数据模型的优缺点

- 优点
  - 能够更为直接地描述现实世界，如一个结点可以有多个双亲
  - 具有良好的性能，存取效率较高
- 缺点
  - 结构比较复杂，而且随着应用环境的扩大，数据库的结构就变得越来越复杂，不利于最终用户掌握
  - DDL、DML语言复杂，用户不容易使用
  - 编写应用程序负担比较大

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

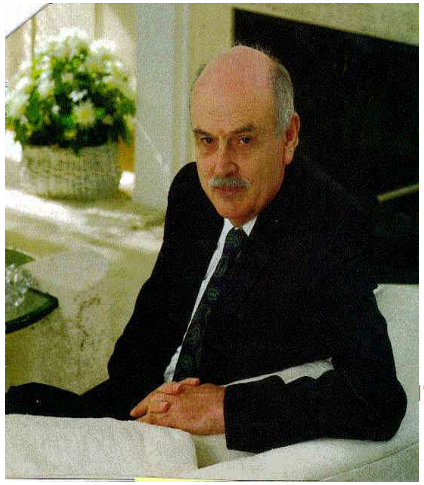
1.2.5 层次模型

1.2.6 网状模型

1.2.7 关系模型

1.2.7 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
- 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



E.F.Codd

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

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
…	…	…	…	…	…

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

关系数据模型的数据结构（续）

- **主码（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。
- **分量**
  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）
  例：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）

例1

学生、系，系与学生之间的一对多联系：

学生（学号，姓名，年龄，性别，系号，年级）

系 (系号，系名，办公地点)

例2

系、系主任，系与系主任间的一对一联系

职工（工号，姓名，年龄，性别）

系 (系号，系名，办公地点，系主任工号)

关系数据模型的数据结构（续）

例3

学生、课程，学生与课程之间的多对多联系：

学生（学号，姓名，年龄，性别，系号，年级）

课程（课程号，课程名，学分）

选修（学号，课程号，成绩）

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

  图1.27中工资和扣除是可分的数据项，不符合关系模型要求

职工号	姓名	职 称	应发工 资			扣 除		实发工资
			基 本	津 贴	职务	房 租	水 电	
86051	陈 平	讲 师	1305	1200	50	160	112	2283
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

图1.27 一个工资表(表中有表)实例

关系数据模型的数据结构（续）

表1.2 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（若干元组的集合）
  - 查询
  - 插入
  - 删除
  - 更新
- 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与完整性约束（续）

- 关系的完整性约束条件（其含义将在第二章介绍）
  - 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

四、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

知识扩展：NoSQL

- NoSQL（Not Only SQL），泛指非关系型的数据库

分类	Examples举例	典型应用场景	数据模型	优点	缺点
键值（key-value）	Tokyo Cabinet/Tyrant, Redis, Voldemort, Oracle BDB	内容缓存，主要用于处理大量数据的高访问负载，也用于一些日志系统等等。	Key 指向 Value 的键值对，通常用hash table来实现	查找速度快	数据无结构化，通常只被当作字符串或者二进制数据
列存储数据库	Cassandra, HBase, Riak	分布式的文件系统	以列簇式存储，将同一列数据存在一起	查找速度快，可扩展性强，容易进行分布式扩展	功能相对局限
文档型数据库	CouchDB, MongoDb	Web应用（与Key-Value类似，Value是结构化的，不同的是数据库能够了解Value的内容）	Key-Value对应的键值对，Value为结构化数据	数据结构要求不严格，表结构可变，不需要像关系型数据库一样需要预先定义表结构	查询性能不高，而且缺乏统一的查询语法。
图形(Graph)数据库	Neo4J, InfoGrid, Infinite Graph	社交网络，推荐系统等。专注于构建关系图谱	图结构	利用图结构相关算法。比如最短路径寻址，N度关系查找等	很多时候需要对整个图做计算才能得出需要的信息，而且这种结构不太好做分布式的集群方案

知识扩展：NoSQL

- NoSQL适用于
  - 数据模型比较简单；
  - 需要灵活性更强的IT系统；
  - 对数据库性能要求较高；
  - 不需要高度的数据一致性；
  - 对于给定key，比较容易映射复杂值的环境
- 比较
  - NoSQL数据库的产生是为了解决大规模数据集合多重数据种类带来的挑战，尤其是大数据应用难题
  - NoSQL结构比较简单，逻辑控制相对较少，同等存量下数据量超过关系型数据库，但是处理能力不一定高
  - 对于数据间有固定模式且紧密联系的，还是建议选择关系型数据库。

图像在数据库中的存储

图像数据一般较大，所以可以考虑以下两种存储方式：

存储图片路径

将图片存在本地，比如windows系统中，在数据库中写入图片的路径，用来索引图片

id	path
1	/image/apple.jpg
2	/image/car.jpg
3	/image/cat.jpg

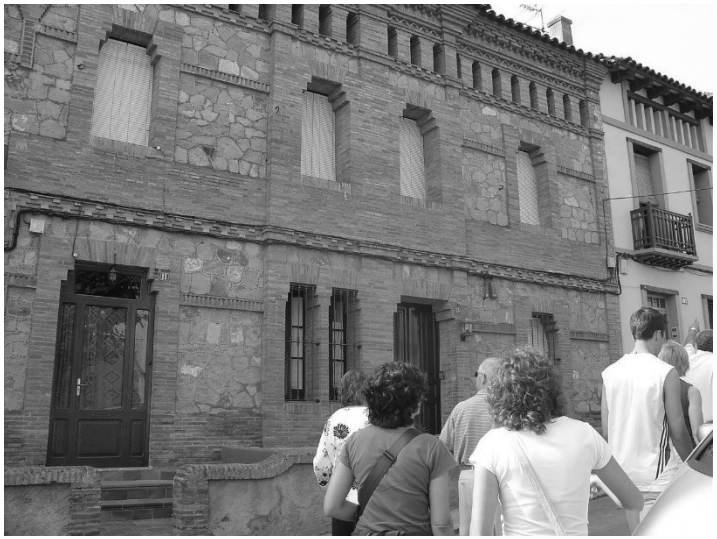
存储图像数据

直接将图像数据存入数据库系统中。可以直接提取图像的像素值，存为numpy, json等格式数据，写入数据库系统中。也可以将其以二进制文件的格式写入。

id	data
1	[[137 124 143 111 62 248 253] [133 116 160 125 133 153 85] [102 122 123 137 142 128 130] [116 52 121 121 56 124 98] [99 116 118 36 127 134 169] [67 119 89 253 158 222 204] [100 54 110 62 202 213 184]]

计算机如何理解图像

height: 960
width: 1280



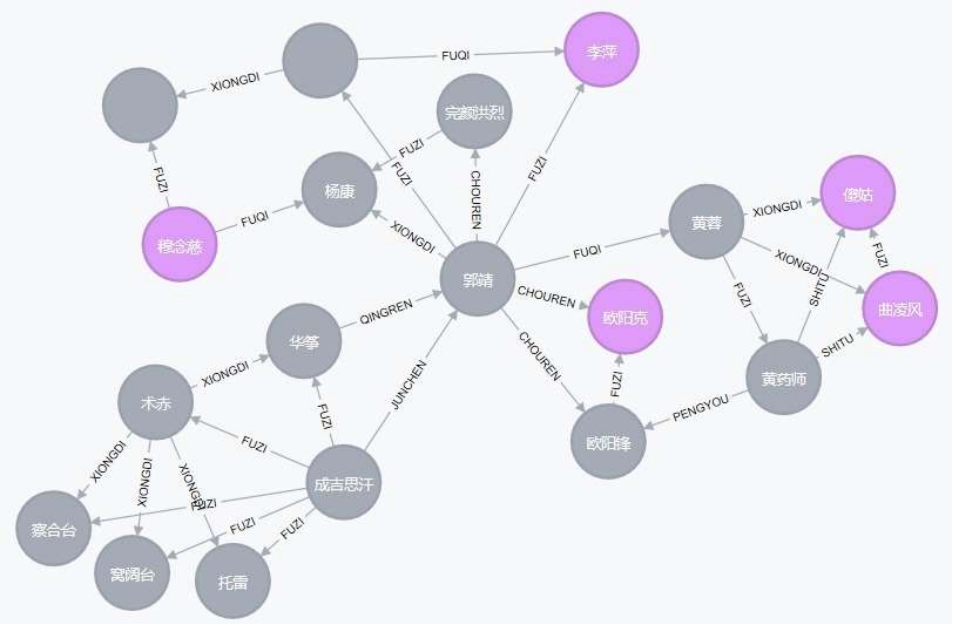
彩色图

灰度图

当计算机看到一张图像时，它看到的是一堆像素值。
对于左边彩色图，它将看到一个960*1280 *3的数组，3指代的是RGB三个通道；
对于右边灰度图，它将看到一个960*1280的数组。
其中，每个数字的值从0到255不等，其描述了对应那一点的像素灰度。
所以，计算机对图像做处理时，实际上就是对这些数组中的像素值做处理。

网络爬虫：存储

- 如何表示关系型数据？
- SQL
  - 冗余数据
  - 大量空白
- Neo4J
  - 符合图数据特性
  - 方便删改



名字	子女	兄弟	父亲	年龄
郭靖	破虏	杨康		33
杨康	\	郭靖	完颜洪烈	
破虏	\	\	郭靖	10

```
Create(n:person{name:”郭靖”,age:”33” })
Create(m:person{name:”破虏”,age:”10” })
create (n)-[:R{type:"父子"}]->(m)
```

网络爬虫：存储

- 删去郭靖

名字	子女	兄弟	父亲	年龄
~~郭靖~~	~~破虏~~	~~杨康~~		33
杨康	\	~~郭靖~~	完颜洪烈	
破虏	\	\	~~郭靖~~	10

match(n:person{name:”郭靖”}) delete n

SQL **Neo4j**

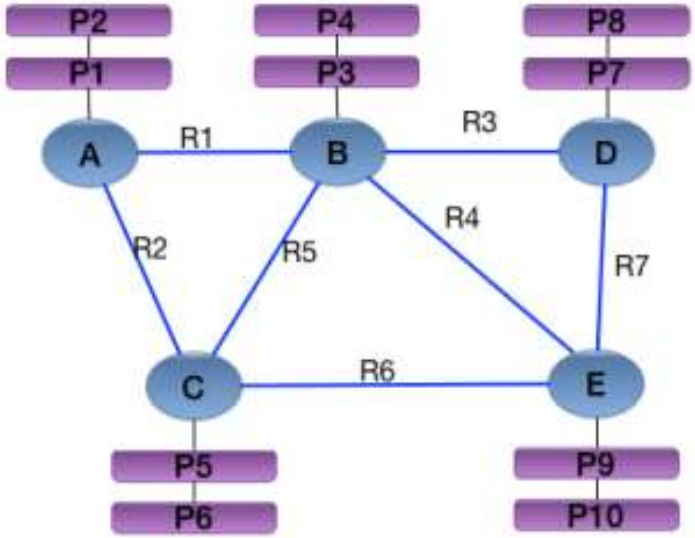
- 杨康和破虏是什么关系？

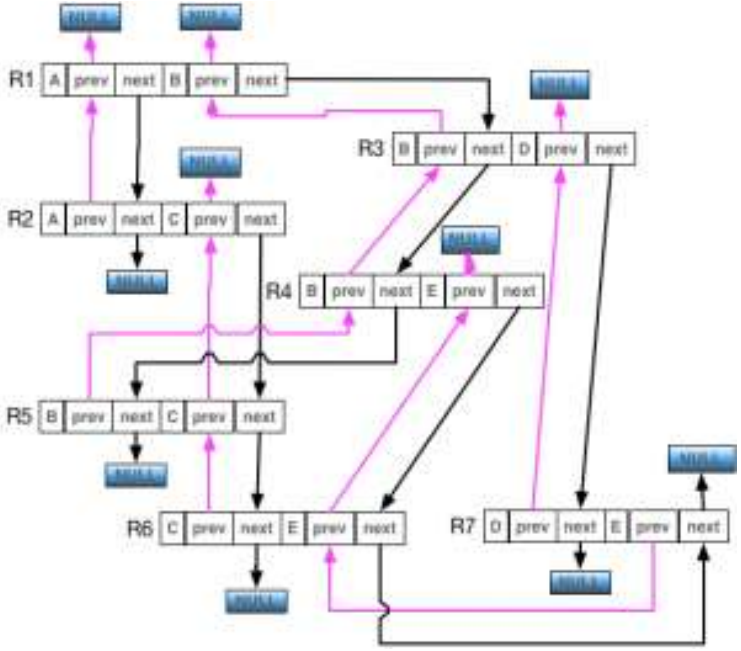
名字	子女	兄弟	父亲	年龄
郭靖	破虏	杨康		33
杨康	\	郭靖	完颜洪烈	
破虏	\	\	郭靖	10

match (n:Person{name:“杨康"}),
(m:Person{name:”破虏”}) ,
p=(n)-[]-(m) ,
return n,m,r

网络爬虫：存储

- 为什么Neo4j适合存储图数据？
  - 使用指针遍历所有节点
  - 符合节点+关系的图结构





通过指针遍历所有节点以及关系

MongoDB

- MongoDB简介
  - 基于分布式文件存储，由 C++ 语言编写，旨在为 WEB 应用提供可扩展的高性能数据存储解决方案。
  - 介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。
  - 将数据存储为一个文档，数据结构由键值(key=>value)对组成，类似于 JSON 对象。字段值可以包含其他文档，数组及文档数组


```
{
  name: "sue",                          ← field: value
  age: 26,                              ← field: value
  status: "A",                          ← field: value
  groups: [ "news", "sports" ]          ← field: value
}
```

MongoDB存储举例——存储字段


```
> db.student.insert({"_id":"BA18011000", "name":"zhang san",
"sex":"male","age":18,"introduction":"Zhang San is a handsome guy!"})
```

```
db.getCollection('student').find({})
```

显示该数据集合

student 0.001 sec.

	_id	name	sex	age	introduction
1	BA18011000	zhang san	male	18.0	Zhang San is a handsome guy!

```
> db.student.insert({ “_id” : “BA18011001” , “name” : “Li Si", "sex":"female",
"age":17,"introduction":"Li Si is a beautiful girl!", "class":["Math","Music","Physics"]})
```

```
db.getCollection('student').find({})
```

document对字段没有强约束，Value可以是各种类型的文档

student 0.001 sec.

	_id	name	sex	age	introduction	class
1	BA18011000	zhang san	male	18.0	Zhang San is a handsome guy!	
2	BA18011001	Li Si	female	17.0	Li Si is a beautiful girl!	[3 elements]

MongoDB存储举例——存储图片


```
file_path = "E:\\USTC\\store_data\\photo"
files = os.listdir(file_path)
# print(files)

pics =[]
#遍历图片目录集合
for index, file in enumerate(files):
    filename = file_path + '\\' + file
    print(filename)
    with open(filename, "rb") as b_image:
        content = b_image.read()
        pics.append(content)
```

二进制读取图片

```
dic = {
    "_id" : "BA18011002",
    "name" : "Wang Wu",
    "sex" : "female",
    "age" : 19.0,
    "introduction" : "Wang Wu studies very hard!",
    "class" : [
        "Math",
        "Physics",
        "chemistry"
    ],
    "picture": pics
}
student.insert_one(dic)
```

构造图片文档

```
db.getCollection('student').find({})
```

student 0.003 sec.

	_id	name	sex	age	introduction	class	picture
1	BA18011000	zhang san	male	18.0	Zhang San i...		
2	BA18011001	Li Si	female	17.0	Li Si is a be...		
3	BA18011002	Wang Wu	female	19.0	Wang Wu s...	[3 element...	[5 element...

```
"picture" : [
    { "$binary" : "/9j/4AAQSkZJRgABAQEASA
    { "$binary" : "/9j/4AAQSkZJRgABAQEASA
    { "$binary" : "/9j/4AAQSkZJRgABAQEASA
    { "$binary" : "/9j/4AAQSkZJRgABAQEASA
    { "$binary" : "/9j/4AAQSkZJRgABAQEASA
```

MongoDB VS MySQL

对比角度	MongoDB	MySQL
数据库模型	非关系型	关系型
存储方式	虚拟内存+持久化	根据引擎有不同
查询语句	独特的mongodb查询语句	传统的sql语句
架构特点	通过副本集和分片可实现高可用	单点，M-S，MHA，MMM，Cluster等
数据处理方式	将热数据存储于内存，高速读写	根据引擎有不同
成熟度	新兴，成熟度较低	较为成熟的体系
广泛度	在Nosql中较为完善，使用人群也在不断增长	开源数据库的份额在增加，mysql的份额也在增长
优势	在适量级内存的Mongodb的性能是非常迅速的；高扩展性，存储的数据格式是json格式；非常适合日志、博客等比较杂乱的系统的存储	拥有较为成熟的体系，成熟度很高，支持事务
劣势	不支持事务，而且开发文档不是很完全，完善	在海量数据处理的时候效率会显著变慢