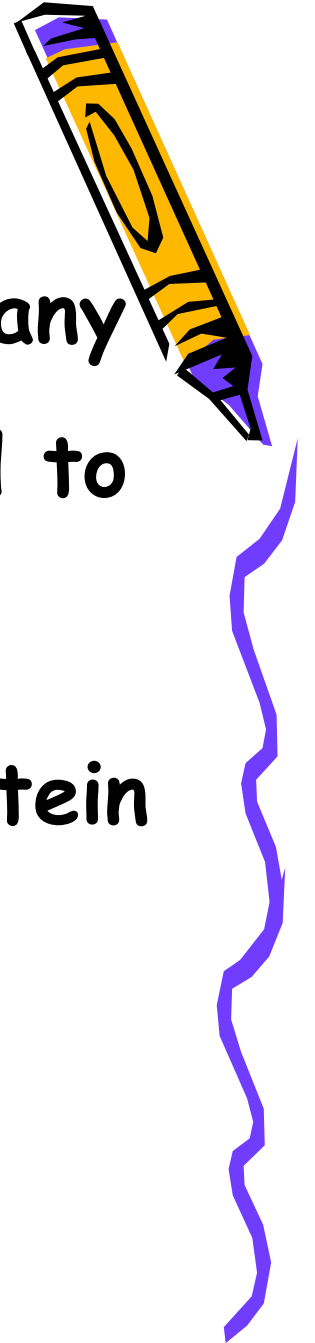
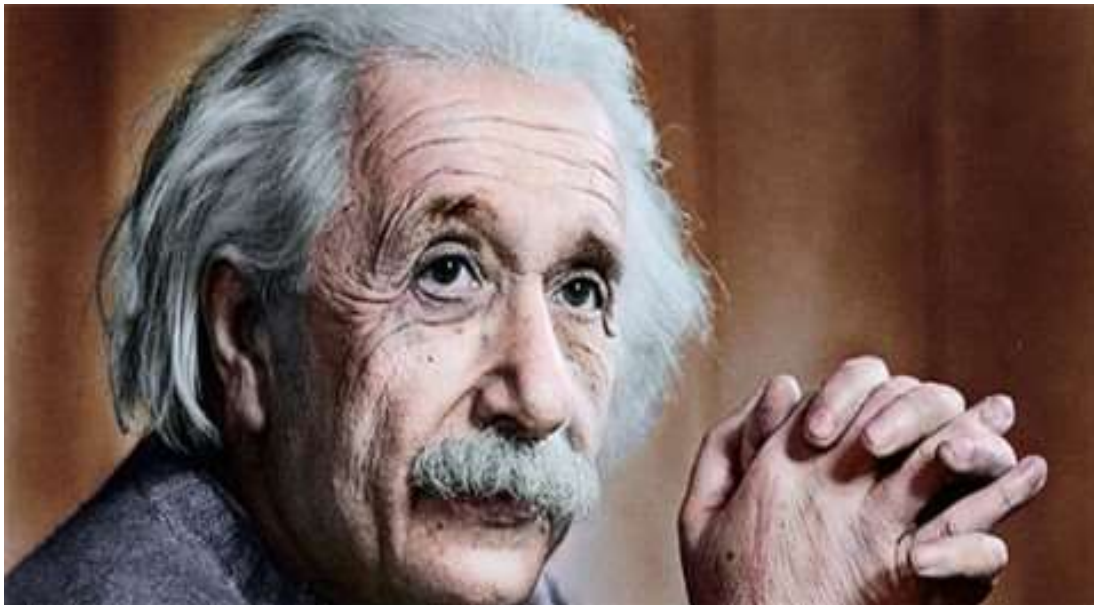




“The value of a college education is not the learning of many facts but the training of the mind to think”



-Albert Einstein





《数据库原理》

课 时: 40+8 学时

任课教师: 马力

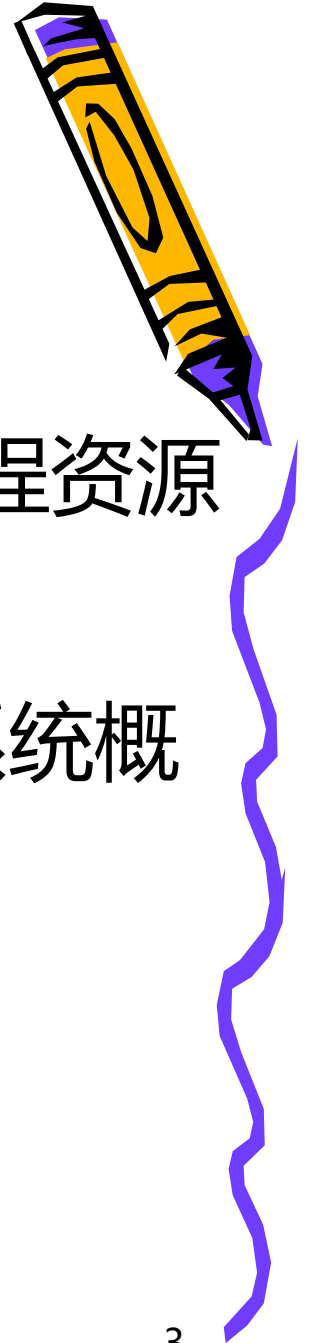
mali@upc.edu.cn

计算机科学系





- **课程性质：** 计算机-必修； 软件-限选
- **课程网站：** 石大云课堂->该课程->课程资源中
- **MOOC：** 中国人民大学 王珊 数据库系统概论





2020-2021-1 数据库...

群号: 1087048291



扫一扫二维码，加入群聊。





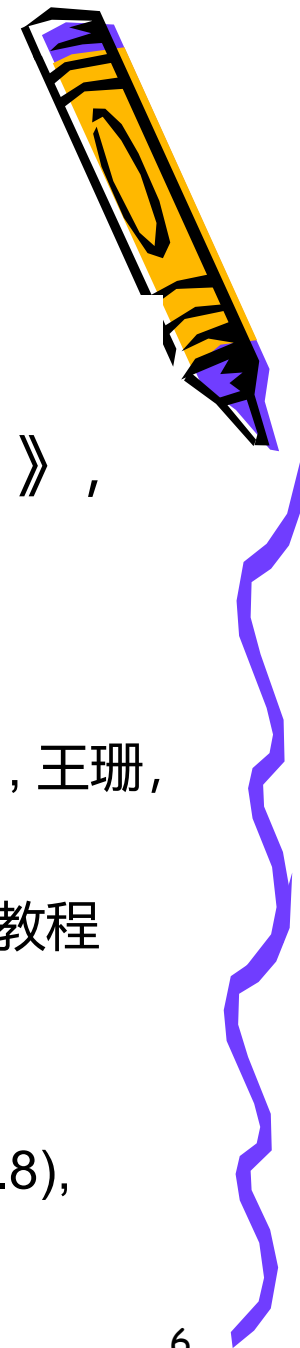
毕业要求与该课程相关指标点

- 目标1：理解数据库系统的基本概念、关系数据库的基本理论，对具体应用需求抽象并运用数据建模工具设计概念模型，为毕业要求指标点2.2的达成提供支持。
- **目标2：掌握规范化理论，理解Armstrong公理和最小函数依赖集的概念，理解查询处理及优化思想，设计并优化关系数据模型，构建并评价物理模型，提高数据库应用系统的性能，为毕业要求指标点2.4的达成提供支持。**
- 目标3：理解结构化分析方法和概念模型的设计方法，能够针对具体问题进行基于关系数据库的应用系统总体设计，为毕业要求指标点3.1的达成提供支持。
- **目标4：掌握数据的安全性保护和完整性检查，熟练使用一种数据库管理系统工具，通过运用关系数据库标准语言SQL与数据库进行交互操作，为毕业要求指标点5.1的达成提供支持。**





教材与参考书



- **选用教材**

- 王珊, 萨师煊 《数据库系统概论 (第五版) 》, 高等教育出版社;

- **参考书**

- 《数据库系统概论 (第五版) 习题解析与实验指导》, 王珊, 张俊, 高等教育出版社, 2015.7
- 詹英等, 《数据库技术与应用——SQL Server 2012教程 (第2版) 》, 清华大学出版社, 2014
- 施伯乐等, 《数据库系统教程》, 高等教育出版社
- C.J Date, An Introduction to Database Systems (Ed.8), Addison-Wesley, 2003





考试成绩评定

- ◆ 平时表现：10%（上课前要**刷卡**）
- ◆ 课内测试：20%（2次），内容：作业相关知识
点。
- ◆ 上机：10%（测试）
- ◆ 期末闭卷考试：60%





内容安排(1)

基础篇

第一章 绪论 (6学时)

第二章 关系数据库 (4学时)

第三章 关系数据库标准语言SQL (8学时)

第四章 数据库安全性 (2学时)

第五章 数据库完整性 (2学时)





内容安排(2)

设计与应用开发篇

- 第六章 关系数据理论 (6学时)
- 第七章 数据库设计 (6学时)
- 第八章 关系查询处理和查询优化 (2学时)





内容安排(3)

系统篇

第九章 数据库恢复技术 (2学时)

第十章 并发控制 (2学时)





内容安排 (4)

上机篇 (8学时)

– 选用数据库:

- SQLServer 任意版本 (机房)
- MySQL任意版本 (自己安装)

– 完成有关**SQL操作**的实验内容



“大数据”的诞生

半个世纪以来，随着计算机技术全面融入社会生活，信息爆炸已经积累到了一个开始引发变革的程度。它不仅使世界充斥着比以往更多的信息，而且其增长速度也在加快。信息爆炸的学科如天文学和基因学，创造出了“大数据”这个概念。如今，这个概念几乎应用到了所有人类发展的领域中。

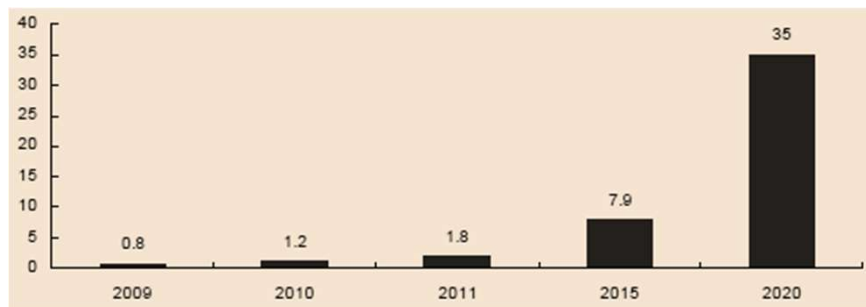


21世纪是数据信息大发展的时代，移动互联、社交网络、电子商务等极大拓展了互联网的边界和应用范围，各种数据正在迅速膨胀并变大。

互联网（社交、搜索、电商）、移动互联网（微博）、物联网（传感器，智慧地球）、车联网、GPS、医学影像、安全监控、金融（银行、股市、保险）、电信（通话、短信）都在疯狂产生着数据。

大数据时代到来

数据量增加

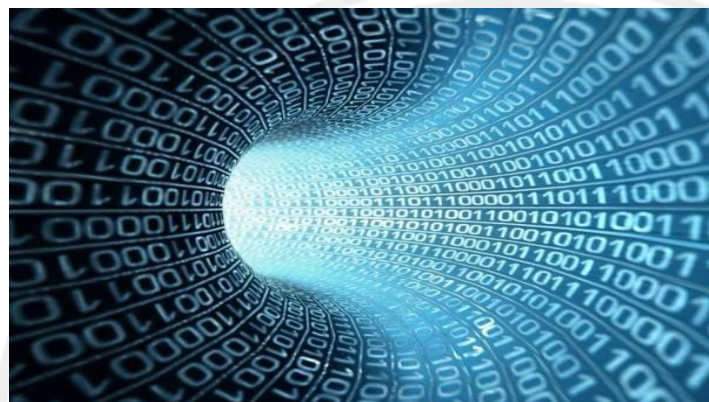


根据IDC 监测，人类产生的数据量正在呈指数级增长，大约每两年翻一番，这个速度在2020 年之前会继续保持下去。这意味着人类在最近两年产生的数据量相当于之前产生的全部数据量

TB \Rightarrow PB \Rightarrow EB \Rightarrow ZB

数据结构日趋复杂

大量新数据源的出现则导致了非结构化、半结构化数据爆发式的增长



- 这些由我们创造的信息背后产生的这些数据早已经远远超越了目前人力所能处理的范畴
- 大数据时代来临..

大数据时代到来

大数据是指社会生产生活和管理服务过程中形成的，依托现代信息技术采集、传输、汇总的，超过传统数据系统处理能力的数据，具有数据量大、数据类型多、处理速度快的特点，通过整合共享、交叉复用、提取分析可获取新知识，创造新价值。

大数据包括：
交易数据和交互数据
集在内的所有数据集

大数据 = 海量数据 + 复杂类型的数据处理方法

海量交易数据：

企业内部的经营交易信息主要包括联机交易数据和联机分析数据，是结构化的、通过关系数据库进行管理和访问的静态、历史数据。通过这些数据，我们能了解过去发生了什么。

海量交互数据：

源于Facebook、Twitter、LinkedIn及其他来源的社交媒体数据构成。它包括了呼叫详细记录CDR、设备和传感器信息、GPS和地理定位映射数据、通过管理文件传输Manage File Transfer协议传送的海量图像文件、Web文本和点击流数据、科学信息、电子邮件等等。可以告诉我们未来会发生什么。

海量数据处理：

大数据的涌现已经催生出了设计用于数据密集型处理的架构。例如具有开放源码、在商品硬件群中运行的Apache Hadoop。





- 现实世界信息无处不在、数据无处不用，数据库技术是各业务数据处理系统的核心。
- 著名未来学家阿尔文·托夫勒指出：“谁掌握了信息，谁控制了网络，谁就将拥有整个世界”。
- 数据库的建设规模和应用深度已成为衡量一个国家信息化程度重要标志,世界各国高度重视数据资源和数据库技术,并纳入重要优先发展战略。

数据库技术及应用重要且广泛！



数据库在计算机系统中的位置





数据库技术发展回顾



- **发展了一门计算机基础学科**
 - 以数据模型和DBMS核心技术为主，内容丰富、领域宽广
- **经历了三代演变**
 - 层次/网状系统、关系系统、新一代数据库系统**家族**（面向对象DBMS、分布式DBMS、流DBMS、图DBMS、时空DBMS、众包DBMS等）
- **造就了四位图灵奖得主**
 - C.W.Bachman、E.F.Codd、James Gray和Michael Stonebraker
- **形成了一个巨大的软件产业**
 - DBMS及相关工具产品、应用解决方案





四位图灵奖得主





Charles.W.Bachman (查尔斯·巴赫曼)

网状数据库之父

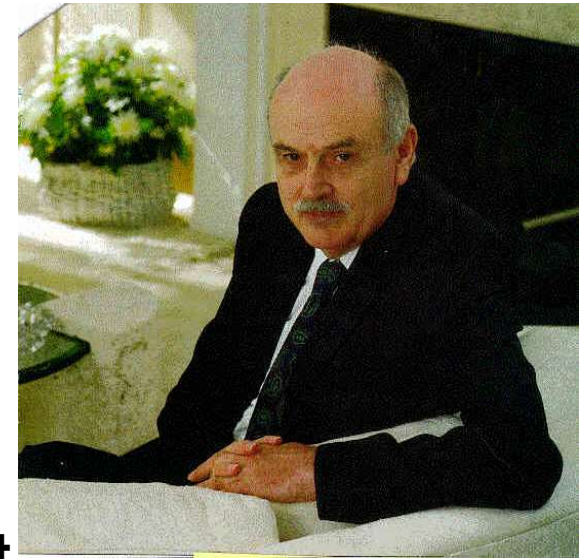


- 1924.11出生在美国，2017.07去世
- 由于他在数据库方面的杰出成就**1973获图灵奖**，获奖演说“作为导航员的程序员”。
- **在数据库方面主要贡献：**
 - 1960年为通用电气制造了世界上第一个网状数据库系统IDS
 - 积极推动与促成了数据库标准的制定：DBTG报告（DBTG提出的网状数据库模型以及数据定义和数据操纵语言的规范说明）
- 1983年成立自己的公司
Bachman Information System





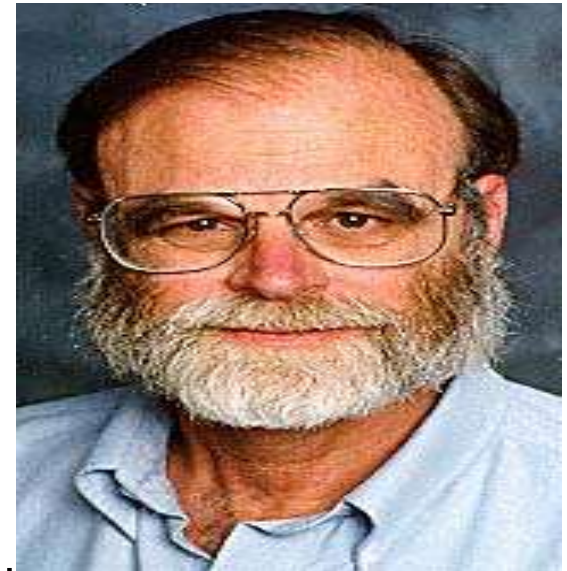
Edgar F.Codd 博士 (埃德加·科德) 关系数据库之父、美国工程院院士



- 原是英国人，1923生于英格兰中部波特兰，2003年4月去世
- 第二次世界大战时应征入伍，在皇家空军服役。1942-1945年间任机长，参与了许多惊心动魄的空战。
- 英国牛津大学数学专业理学士及硕士学位，毕业后到IBM公司工作从事操作系统和自动机理论研究
- 年近40重返密歇根大学进修计算机与通信专业，1963年获得硕士学位，1965年又获得博士学位。
- 60年代后期开始数据库研究, 1970年E.F.Codd 博士提出关系模型概念(CACM, Vol.13, Vol.6, 1970)
- 1981年获图灵奖，获奖演说“关系数据库：提高生产率的实际基础”。84年从IBM公司退休
- 还创办了一个研究所：关系研究所 (The Relational Institute) 和一个公司：Codd



James Gray (詹姆斯·格雷) 数据库技术和事务处理专家



- 1944年生，美国加州大学伯克利分校，计算机科学系博士。
- 先后在贝尔实验室、IBM、Tandem、DEC等工作，研究方向转向数据库领域。
- 由于他在数据库和事务处理研究方面（解决用户共享、安全性、完整性和恢复等重大技术问题）的原创性贡献以及在将研究原型转化为商业产品的系统实现方面的技术领袖地位，**1998年获**
奖(时任微软研究员)，获奖演说“信息技术今后的目标”。
- 2007年1月去世。



Michael Stonebraker (迈克尔·斯通布雷克)

——现代主流数据库系统架构的奠基人



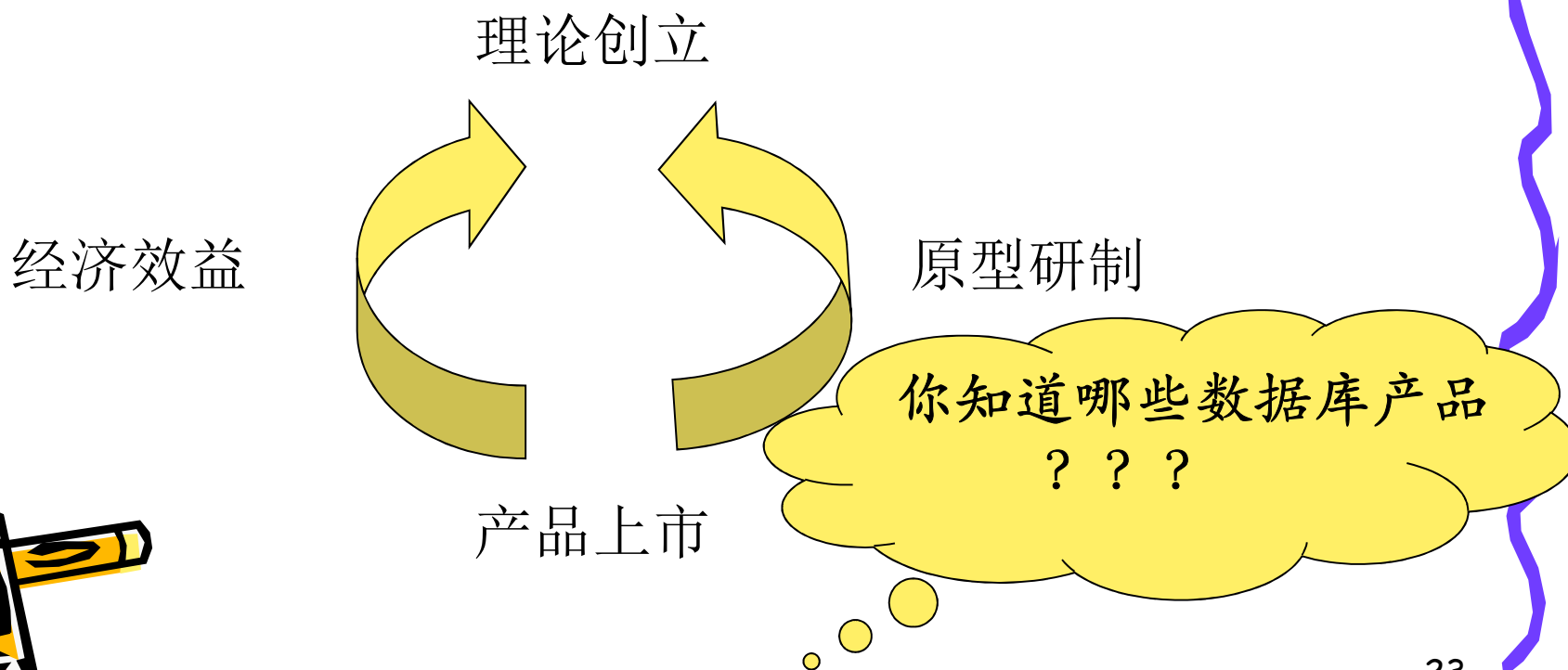
- 1943年10月出生，1971年密歇根大学计算机信息与控制专业博士学位。
- 2014年图灵奖获得者，主要贡献是不断设计、实现和推广新的数据库架构和管理技术，并在工业界和商业市场中连续取得了巨大成功。
- 研究工作主要分为三个阶段：
 - 1971-2000年，从事关系数据库的体系架构与实现技术研究。
 - 2001-2008年，在**one size does not fit all**观念下进行了一系列新型数据库系统的体系结构设计与产品开发。
 - 2009-，大数据系统的体系架构设计与实践。





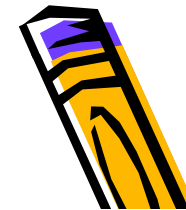
数据库：一个巨大的软件产业

- 已形成一个巨大的软件产业
- 是理论成果转化为产品的成功范例





国产数据库现状



- 在国家科技计划支持下，通过产学研合作，诞生了国产数据库企业人大金仓（1999）、武汉达梦（2000）、南大通用（2004）等，其产品已在多个行业使用；
- 阿里云自研的数据库PolarDB当选了世界互联网领先科技成果；
- 蚂蚁金服的OceanBase数据库（支持淘宝）登顶了TPC-C排行榜的榜首；
- 华为将GaussDB定位于AI-Native数据库，通过异构计算创新框架，在权威标准测试集TPC-DS上，性能比业界提升50%；
- 腾讯TDSQL、中兴GOLDDDB、百度Cloud RDS、TiDB、SequoiaDB等已形成百花齐放的大好局面。



引自彭智勇-武汉大学教授报告



国产数据库未来发展方向

- 理论创新引领数据库发展
- 应用驱动突破数据库技术
- 生态建设拓广数据库市场

该你们隆重出场了!!!





第一章 绪论

1.1 引言

1.2 数据模型 (重点)

1.3 数据库系统的结构 (重点)

1.4 数据库系统的组成



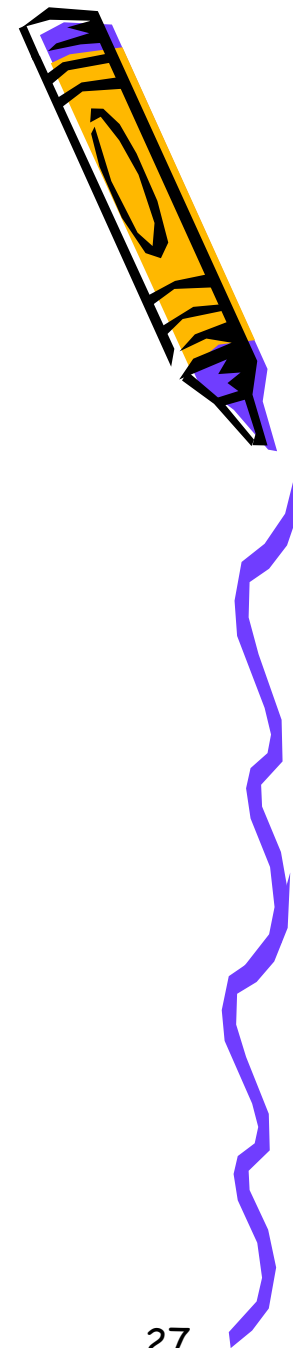


1.1 引言

1.1.1 数据库的四个基本概念

1.1.2 数据管理技术的产生与发展

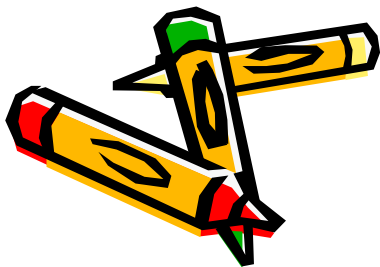
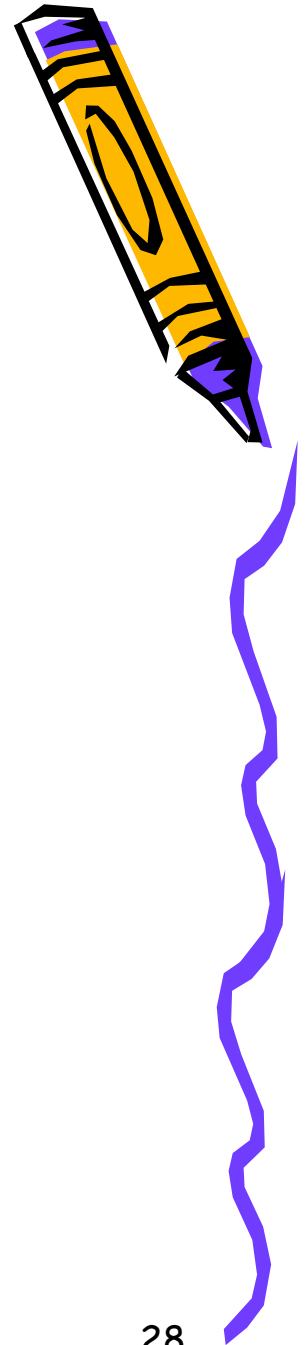
1.1.3 数据库技术的研究领域（了解）





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，是信息的具体表现形式，是客观世界中的事物与现象在计算机中的表示。

◆ **组成数据的符号种类**

— 数字、字符串、日期、逻辑值、文本、图形、图象、声音

◆ **数据的特点**

— 数据与其语义是不可分的





◆ 数据的含义

☞ 称为数据的语义（**信息**），数据与其语义是不可分的。

☞ 例如： 90，是一个数据

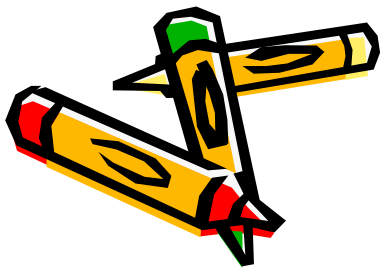
- 语义1：学生某门课的成绩
- 语义2：某人的体重
- 语义3：计算机专业12级的学生人数
- 语义4：请同学给出。

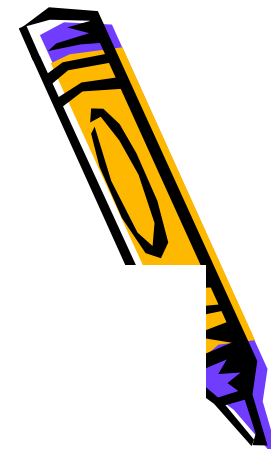




信息是一种被加工为特定形式的数据，但这种数据形式对接收者来说是有意义的，而且对当前和将来的决策具有明显的或实际的价值。

如“2021年某名校硕士研究生将扩招30%”，对接受者有意义，使接受者据此作出决策。





◆ 数据与信息的联系

- ✓ 数据与信息相互依存又有区别；
- ✓ 数据是信息的符号表示或载体，信息则是数据的内涵，是对数据的语义解释；
- ✓ 信息可用不同的数据形式来表示，不随数据表现形式而改变；而数据的表示方式却可以选择。

如上例中的数据2021、30%被赋予了特定的语义，它们就具有了传递信息的功能。





◆ 数据处理

数据处理是将数据转换成信息的过程，包括对数据的收集、存储、加工、检索、传输等一系列活动。其目的是从大量的原始数据中抽取和推导出有价值的信息，作为决策的依据。其实质是信息处理。



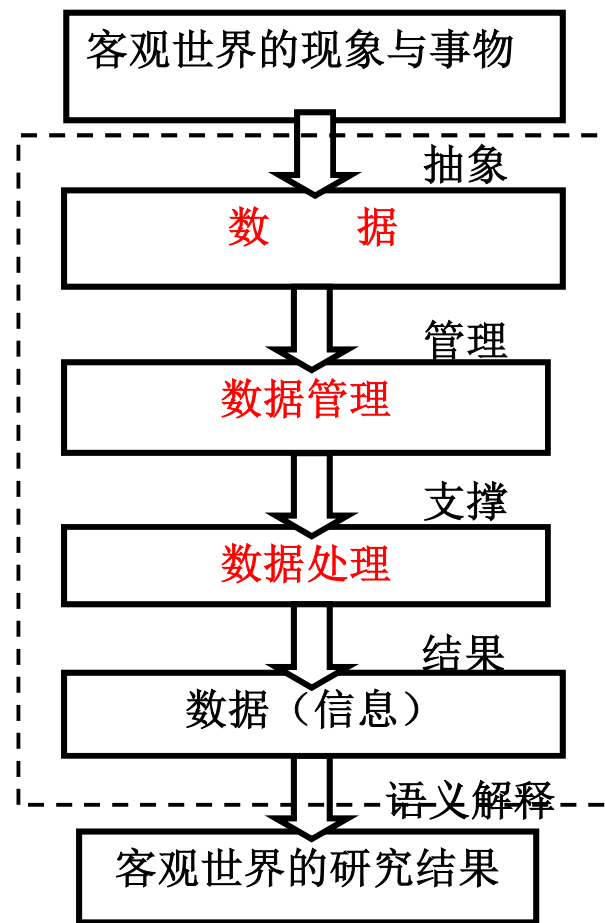
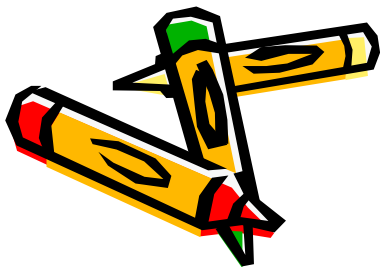


数据处理流程

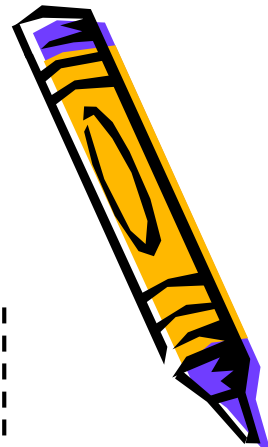
信息=数据+处理

如 年龄 = 现日期
- 出生日期

即： 数据库处理数据
过程就是把数据处
理成信息的过程



数据处理流程

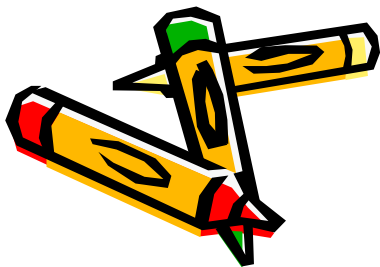
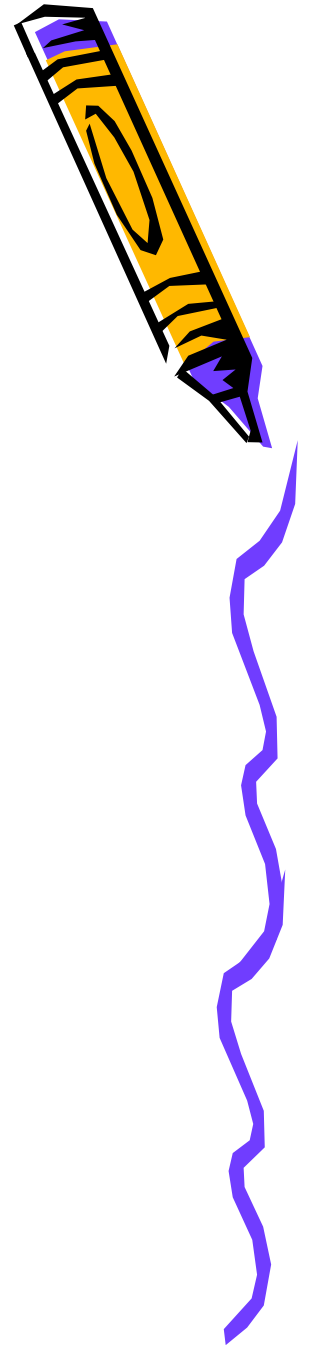




数据处理两个方向：

事务处理：数据 \longrightarrow 数据（信息）

分析处理：数据 \longrightarrow 规则（知识）





二. 数据库

◆ 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

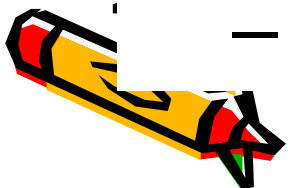


◆ 定义

- ∞ (Database Management system, DBMS)
- ∞ 位于用户与操作系统之间的一层**数据管理软件**。是数据库系统的核心。是处理数据库访问的软件，提供数据库的用户接口。
 - 是基础软件，是一个大型复杂的软件系统。

◆ DBMS的用途

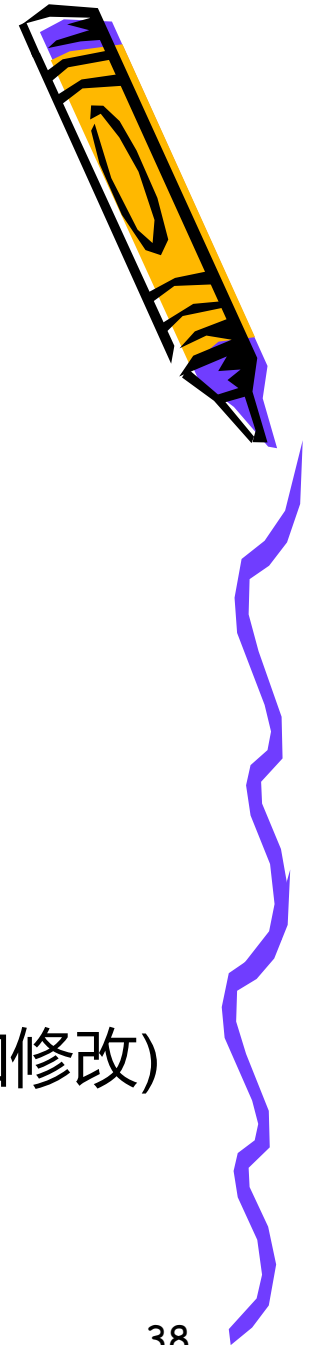
- 科学地组织和存储数据
- 高效地获取和维护数据





DBMS的主要功能

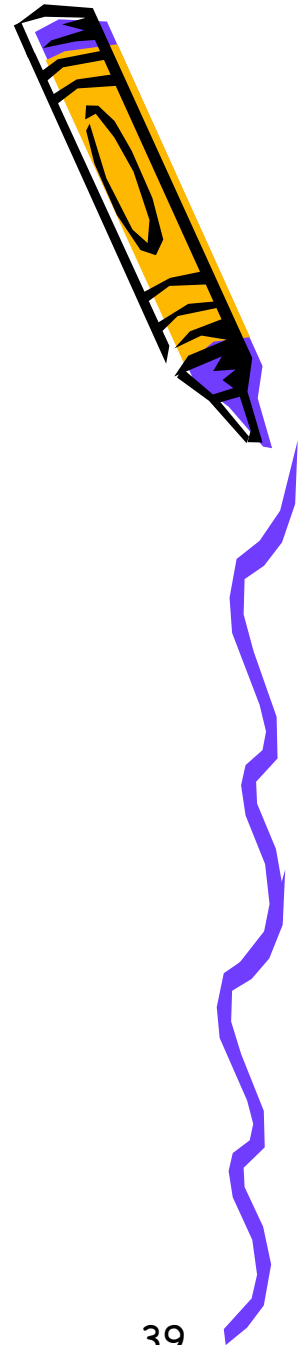
- **数据定义功能**
 - 提供数据定义语言(DDL)
 - 定义数据库中的数据对象
- **数据操纵功能**
 - 提供数据操纵语言(DML)
 - 实现对数据库的基本操作 (查询、插入、删除和修改)





DBMS的主要功能

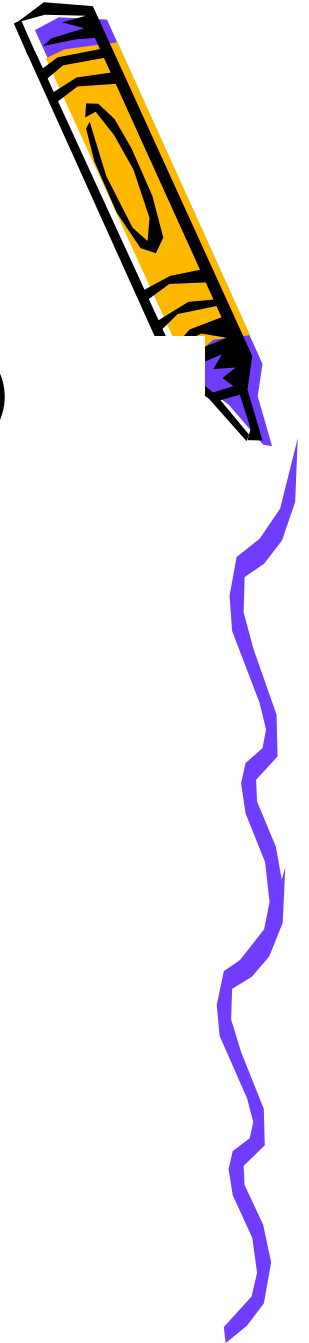
- **数据库的事务管理和运行管理**
 - 保证数据的安全性、完整性
 - 多用户对数据的并发使用
 - 发生故障后的系统恢复
- **数据组织、存储和管理**
 - 分类组织、存储和管理各种数据
 - 确定组织数据的文件结构和存取方式
 - 实现数据之间的联系
 - 提供多种存取方法提高存取效率





DBMS的主要功能

- **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
 - 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



◆ 定义

∞ (DataBase System, 简称DBS)

∞ 是指在计算机系统中引入数据库后的系统, 即采用了数据库技术的计算机系统。

– 在不引起混淆的情况下常常把数据库系统简称为数据库。

◆ 数据库系统的构成

– 数据库

– 数据库管理系统 (及其开发工具)

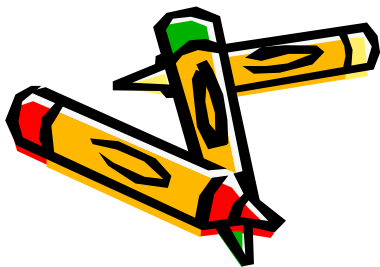
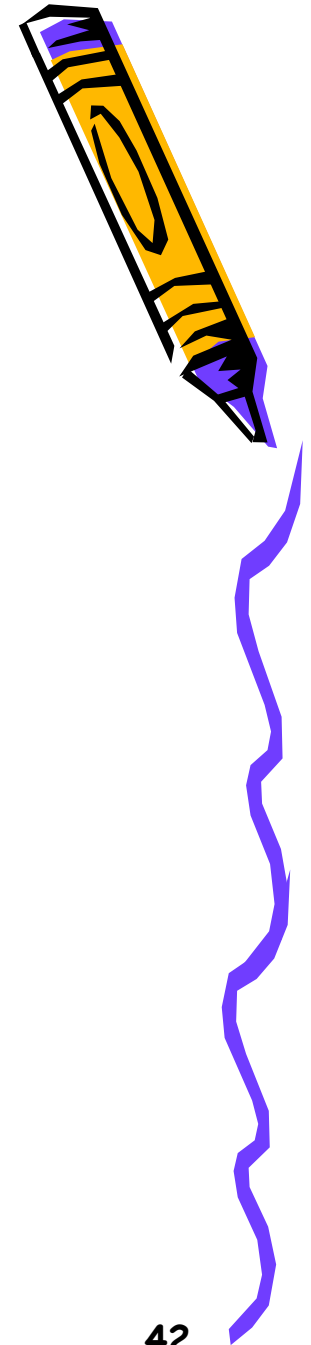
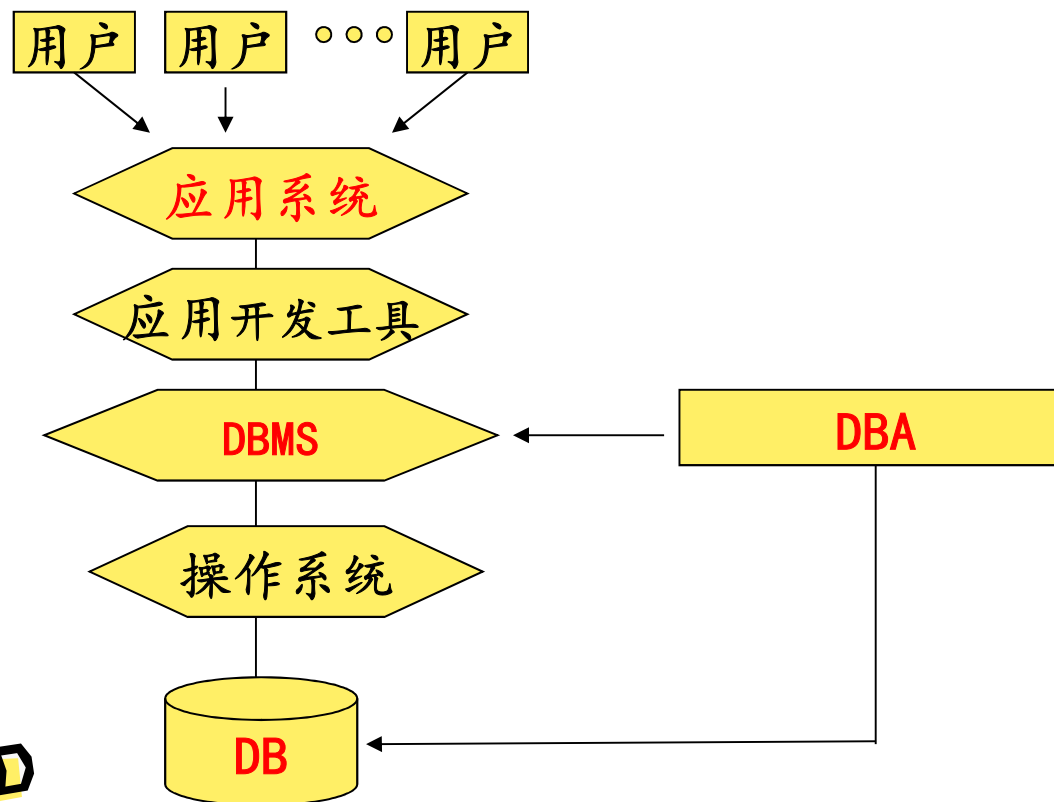
– 应用系统

– 数据库管理员





数据库系统构成图示



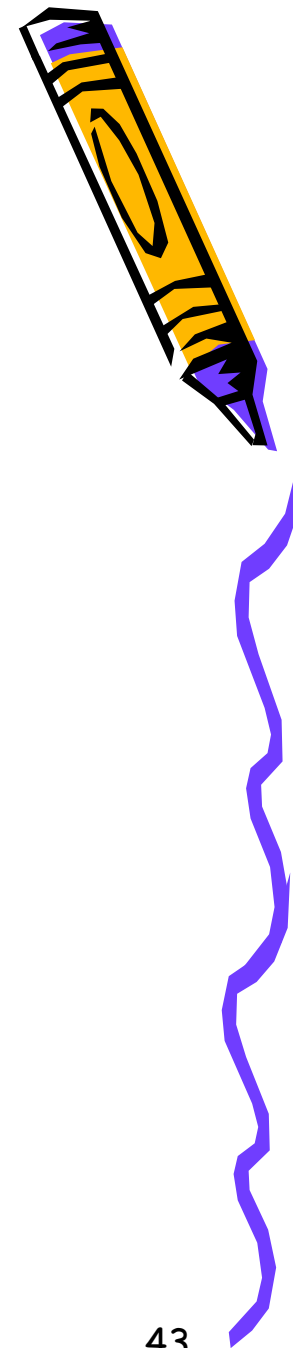


1.1 引言

1.1.1 数据库的四个基本概念

1.1.2 数据管理技术的产生与发展

1.1.3 数据库技术的研究领域





1.1.2 数据管理技术的产生和发展

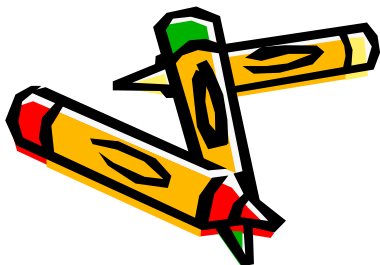
◆ 什么是数据管理

- 以数据的管理为目的
- 对数据进行分类、组织、编码、存储、检索和维护
- 是数据处理的中心问题。

◆ 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

数据处理是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。例如：





• 数据管理技术的发展过程

- 人工管理阶段(40年代中--50年代中)
- 文件系统阶段(50年代末--60年代中)
- 数据库系统阶段(60年代末--现在)





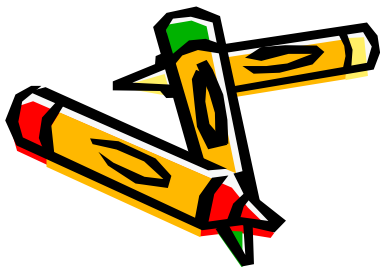
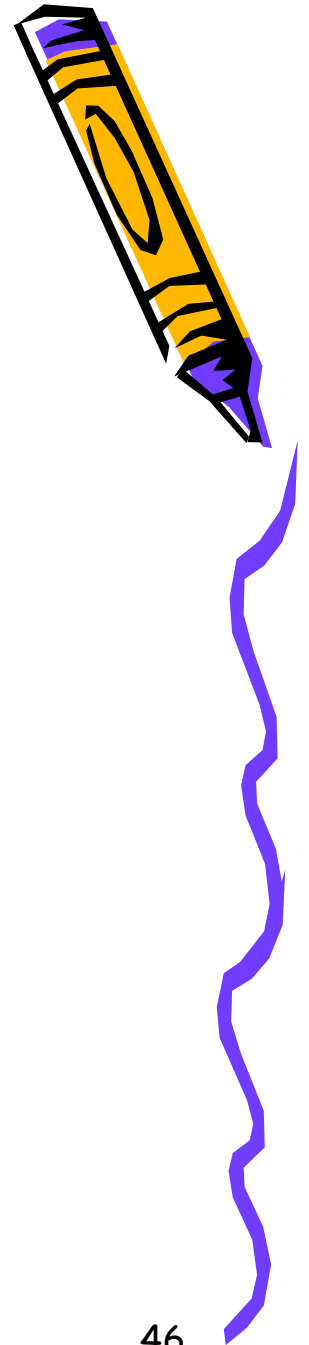
一. 人工管理

- **时期**

- 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





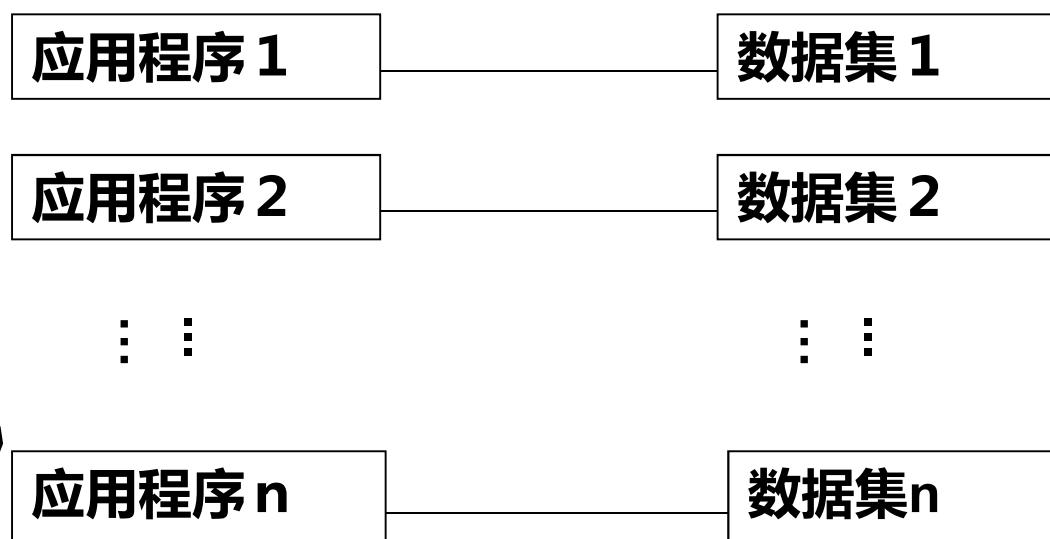
特点1

- 数据的管理者：应用程序，数据不保存。

特点2

- 数据面向的对象：某一应用程序

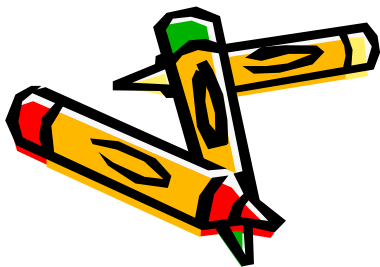
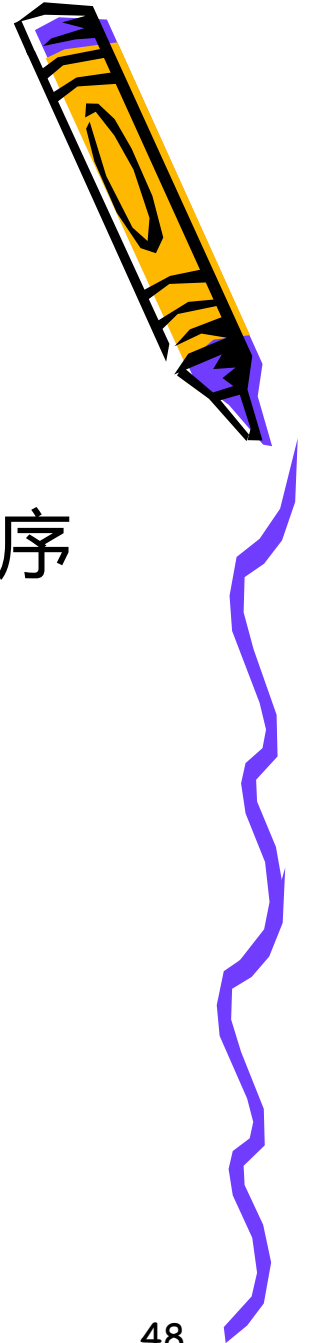
应用程序与数据的对应关系图





特点3

- 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





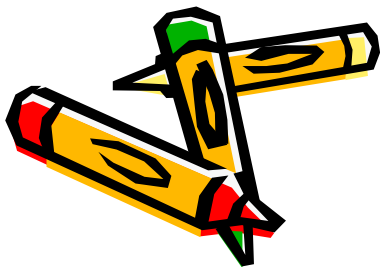
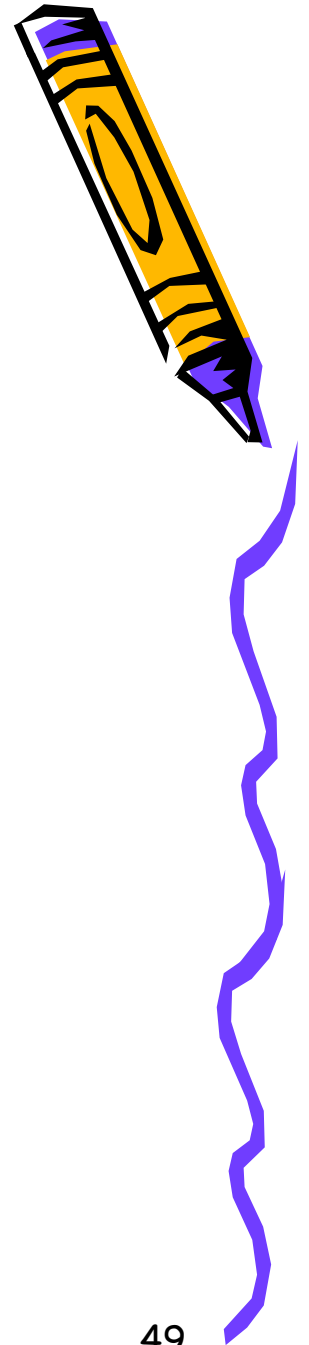
二．文件系统

- **时期**

- 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





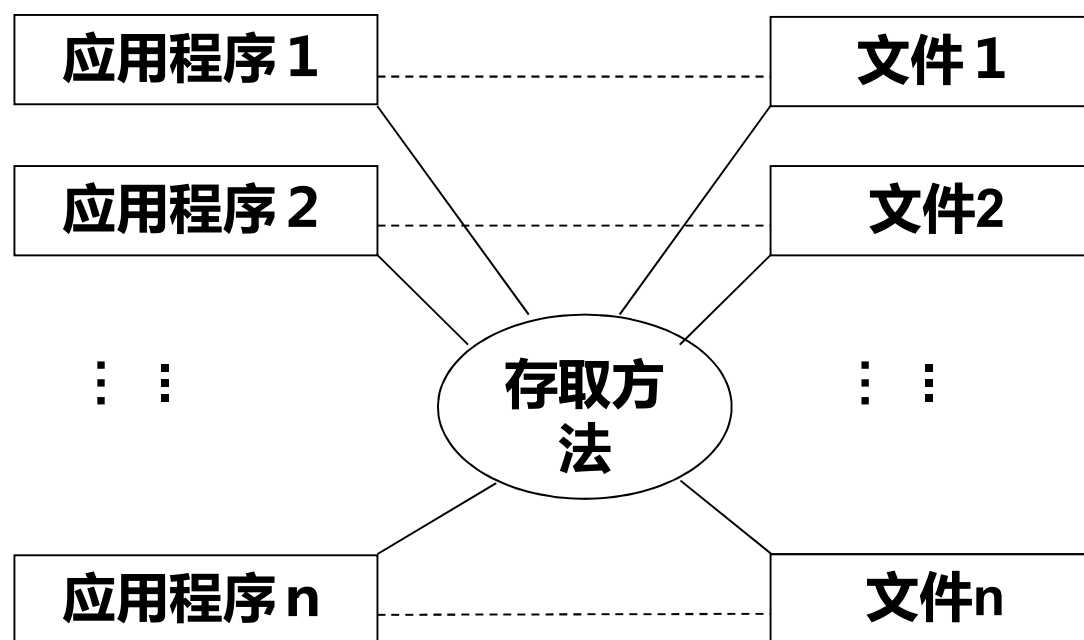
特点1

- 数据的管理者：文件系统，数据可长期保存

特点2

- 数据面向的对象：某一应用程序

应用程序与数据的对应关系



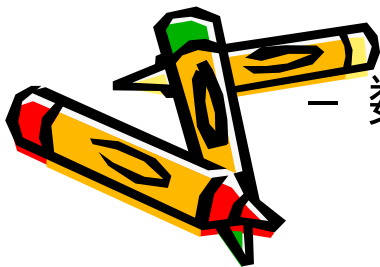


特点3

- 数据的共享程度：共享性差、冗余度大

特点4

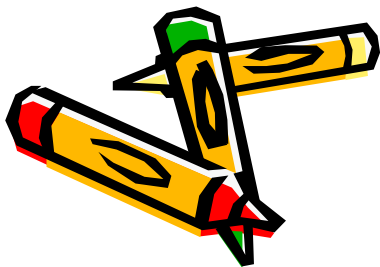
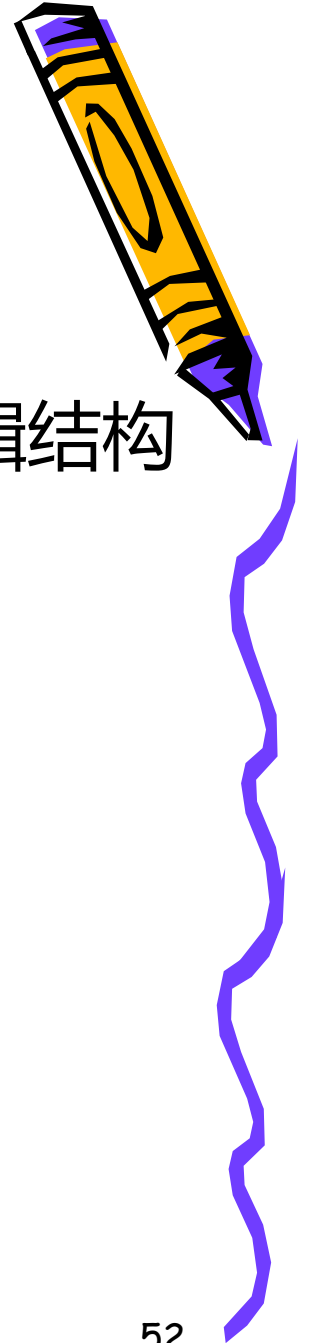
- 数据的结构化：记录内有结构,整体无结构
- 文件系统中数据的结构
 - 记录**内有结构**。
 - **数据的结构是靠程序定义和解释的。**
 - 数据只能是定长的。
 - 可以间接实现数据变长要求，但访问相应数据的应用程序复杂了。
 - 文件间是独立的，因此数据整体无结构。
 - 可以间接实现数据整体的有结构，但必须在应用程序中对描述数据间的联系。
 - 数据的最小存取单位是记录。





特点5

- 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

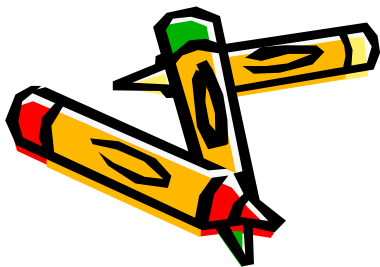
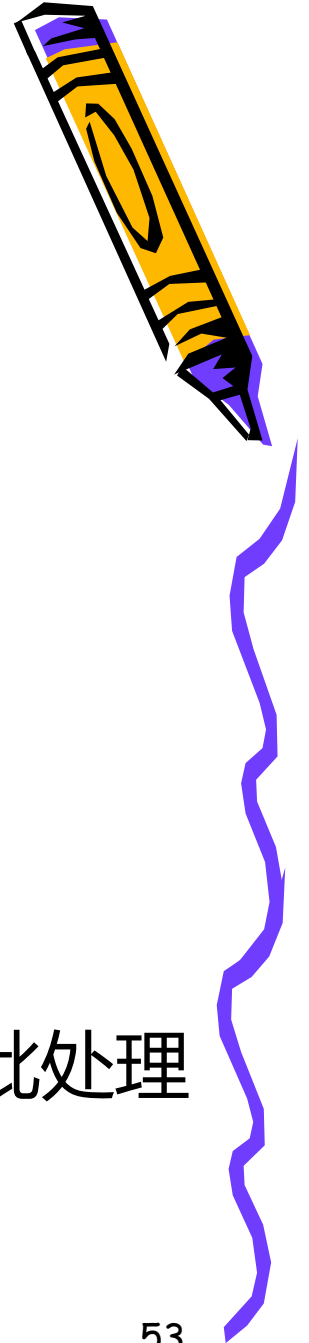




三. 数据库系统

- **时期**
 - 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
– 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





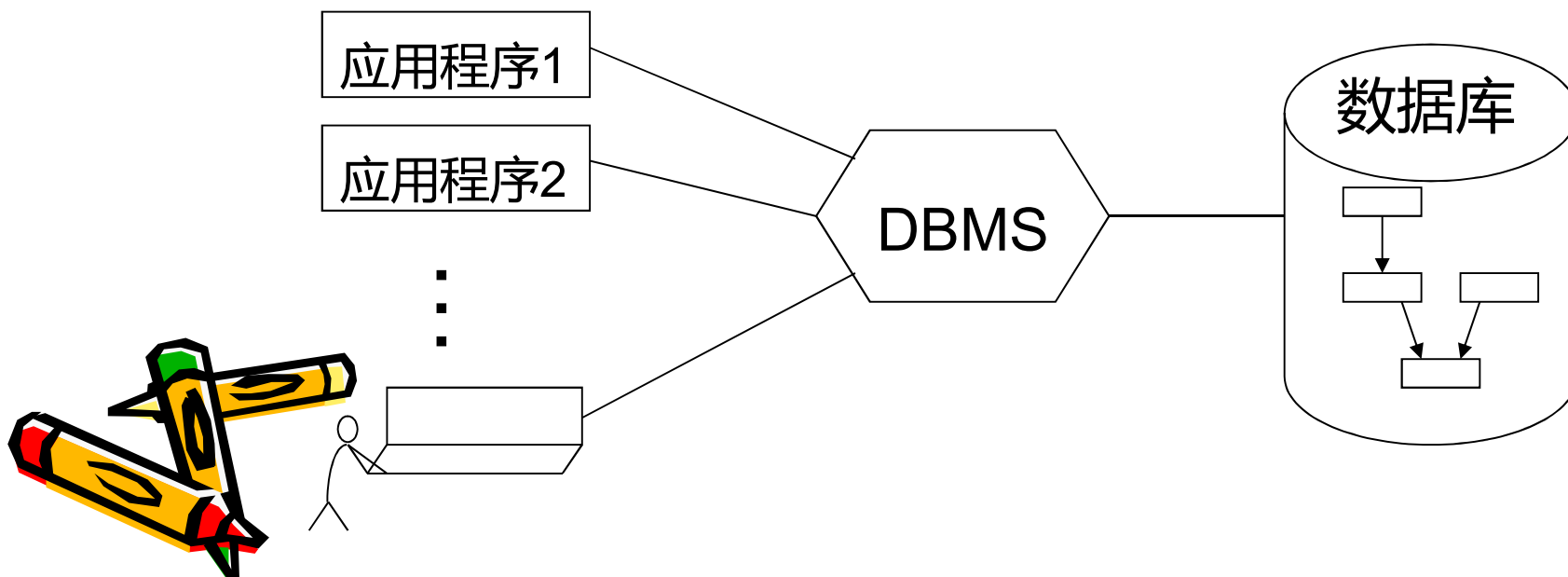
特点1

- 数据的管理者：DBMS

特点2

- 数据面向的对象：现实世界

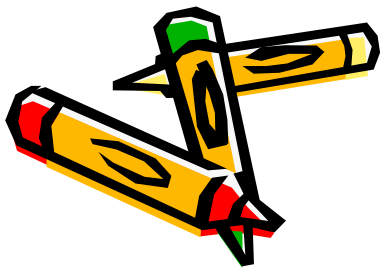
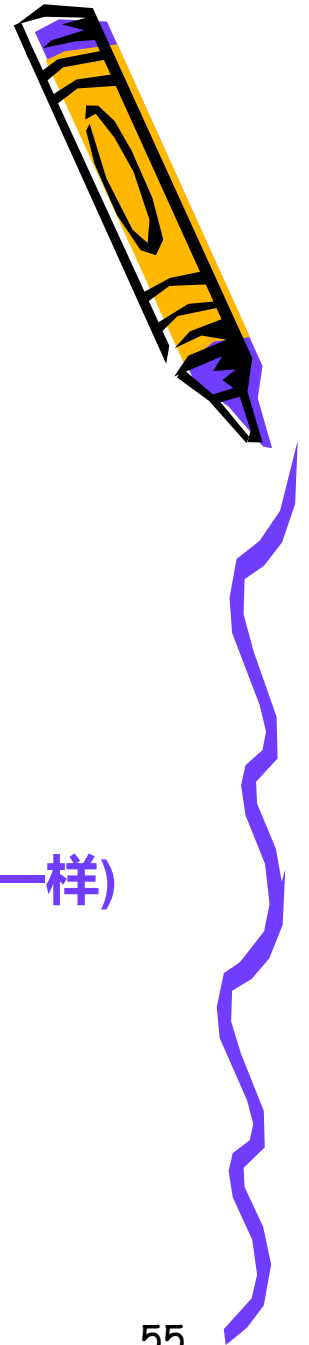
应用程序与数据的对应关系





特点3

- 数据的共享程度：共享性高
- 数据的高共享性的好处
 - 降低数据的冗余度，节省存储空间
 - 避免数据间的**不一致性**(同一数据不同拷贝的值不一样)
 - 使系统易于扩充





特点4

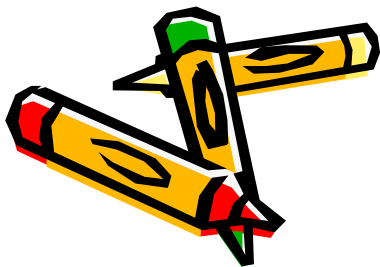
- **数据的独立性：高度的物理独立性和一定的逻辑独立性**

- **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。





特点5

- 数据的结构化：整体结构化。
- 数据结构化

— 整体数据的结构化是数据库的主要特征之一。

— 数据库中实现的是数据的真正结构化。按照某种数据模型，将全组织的各种数据组织到一个结构化的数据库中，整个组织的数据不是一盘散沙，可表示出数据之间的有机关联。

- 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。





例：要建立**学生成绩管理系统**，系统包含：

学生（学号、姓名、性别、系别、年龄）、

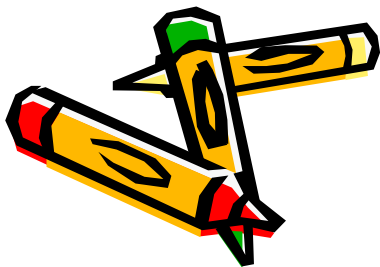
课程（课程号、课程名）、

成绩（学号、课程号、成绩）等数据，分别对应三个文件。



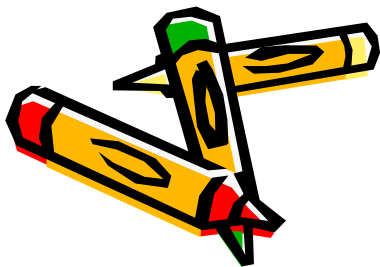
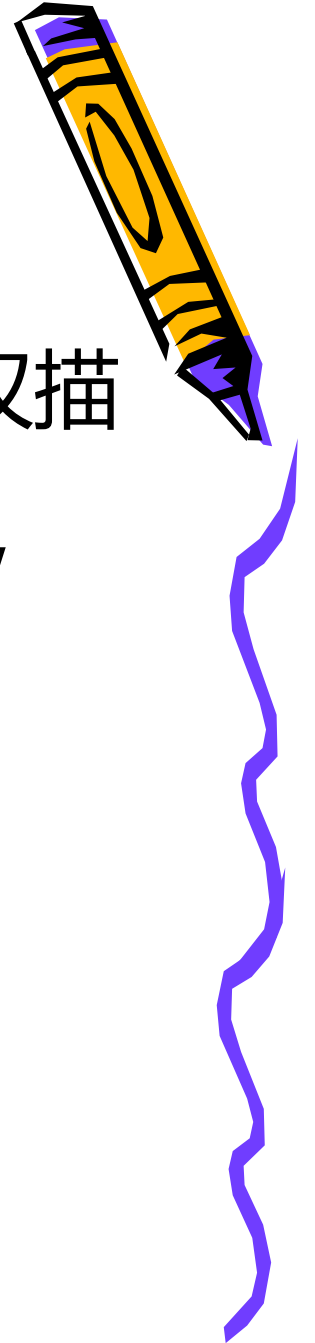


若采用文件处理方式，因为文件系统只表示记录内部的联系，而不涉及不同文件记录之间的联系，要想查找某个学生的学号、姓名、所选课程的名称和成绩，必须编写一段不很简单的程序来实现。





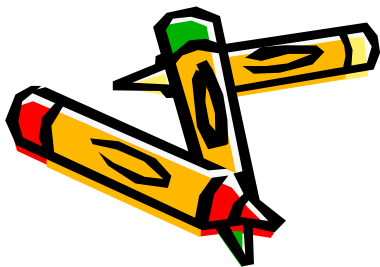
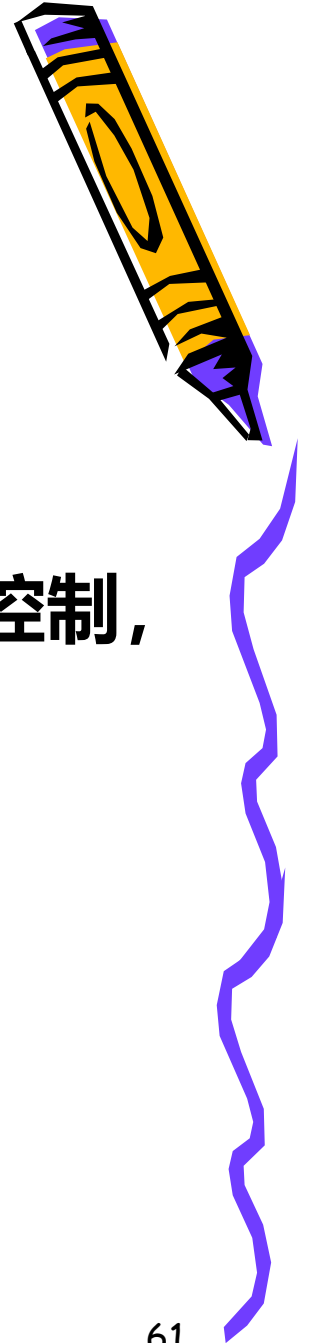
而采用数据库方式，数据库系统不仅描述数据本身，还描述数据之间的联系，上述查询可以非常容易地连接查到。





特点6

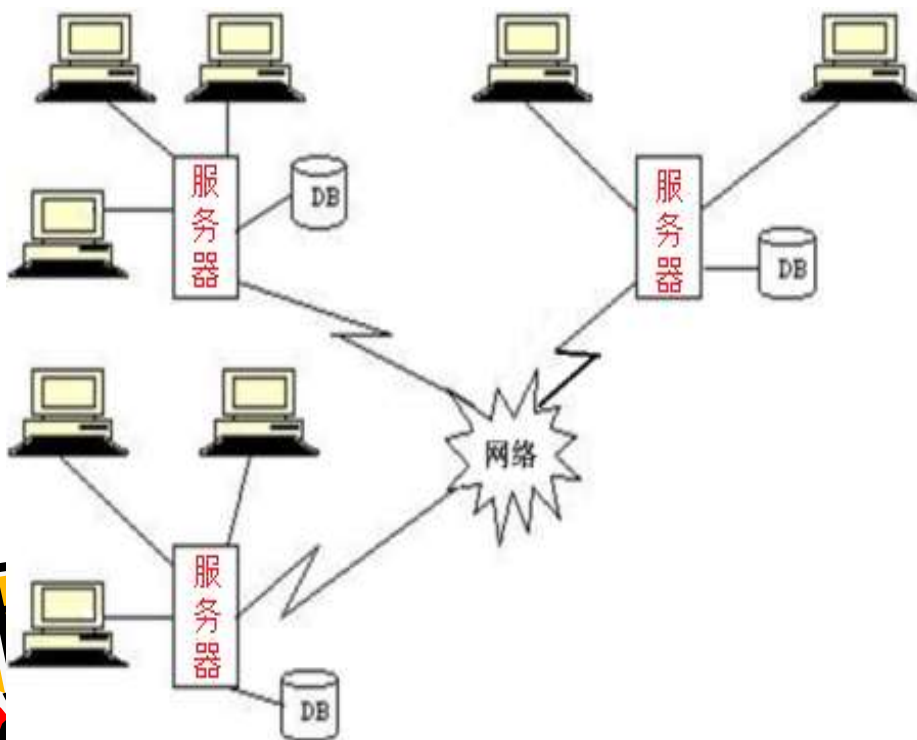
- 数据控制能力（安全性，完整性，并发控制，恢复）：由DBMS统一管理和控制

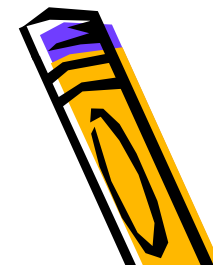




四. 高级数据库阶段

- 从20世纪80年代后期,数据库技术在商业领域取得巨大成功,激发了其他领域对其需求的快速增长,开辟了新的应用领域。





(1) 分布式数据库技术

具有如下5个主要特点:

- 1) 以本地为主处理大部分数据。本地分布处理各种业务数据,提高系统处理效率和可靠性,并通过数据复制技术实现网络数据共享。
- 2) 各地终端数据通过网络互联。
- 3) 降低中心数据库及数据传输负载。
- 4) 系统安全可靠性能更高,局部系统故障,其他仍可继续。
- 5) 系统分布扩展便捷。



■ 分布式数据库系统兼顾集中管理和分布处理两项任务,



(2) 面向应用领域的专用数据库

为了适应应用多元化的需求，结合各应用领域的特点，将数据库技术应用到特定领域，产生了工程数据库、地理数据库、统计数据库、科学数据库、空间数据库等多种数据库，同时也出现了数据仓库和数据挖掘等技术，使数据库领域中的新技术不断涌现。

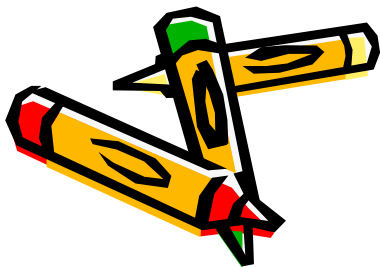




(3) 面向对象数据库技术

主要有2个特点:

- 1) 对象数据模型能完整地描述现实世界的数据结构,表达数据间嵌套、递归的联系。
- 2) 具有面向对象技术的**封装性**(数据与操作定义一起)和**继承性**(继承数据结构和操作)的特点,提高软件可重用性。





数据库技术的发展趋势

- (1) 云数据与云平台
- (2) 数据（人机物）融合
- (3) 区块链
- (4) 大数据技术,分布式文件系统
- (5) 数据挖掘
- (6) 知识库与知识发现



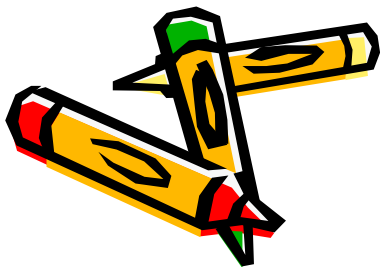
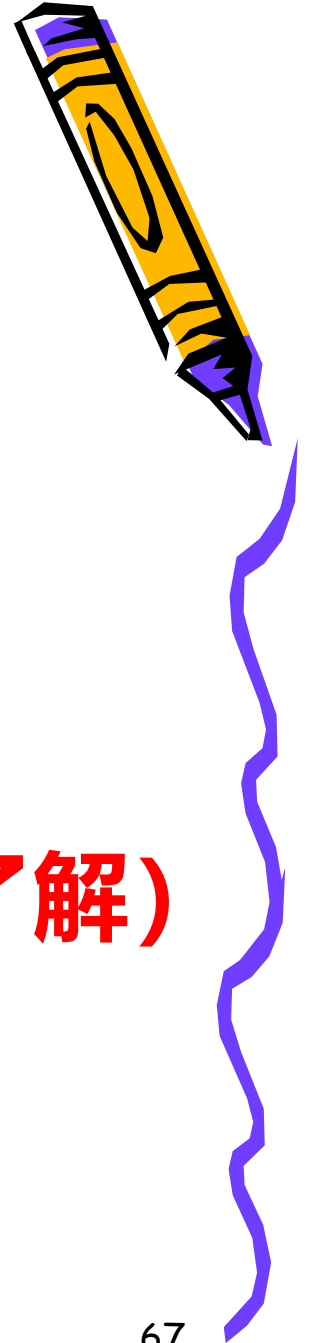


1.1 引言

1.1.1 数据库的四个基本概念

1.1.2 数据管理技术的产生与发展

1.1.3 数据库技术的研究领域（了解）





1.1.3 数据库技术的研究领域

- **数据库管理系统软件的研制**

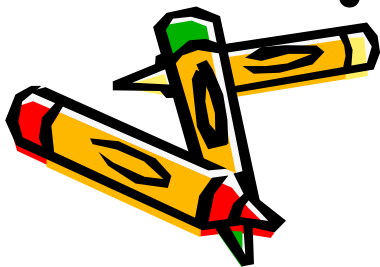
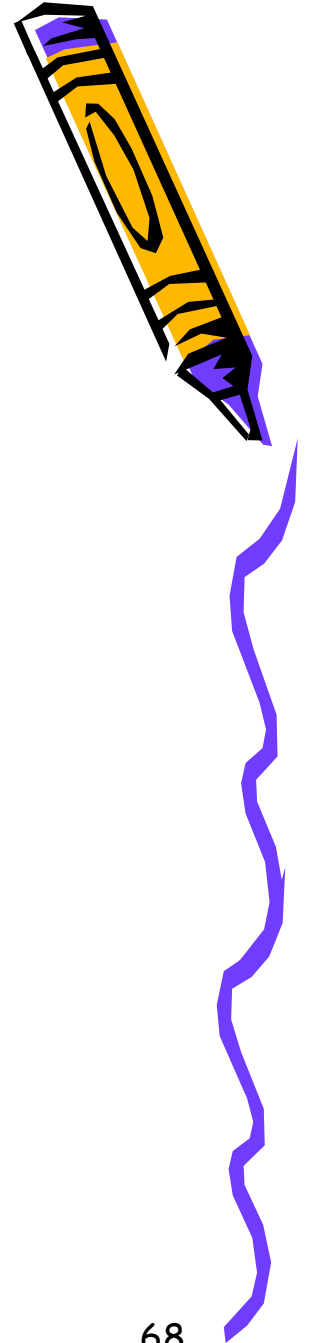
- DBMS本身
- 以DBMS为核心的一组相互联系的软件系统
 - 工具软件
 - 中间件

- **数据库设计**

- 数据库设计方法
- 设计工具
- 设计理论
- 数据模型和数据建模

- **数据库理论**

- 关系的规范化理论
- 关系数据理论





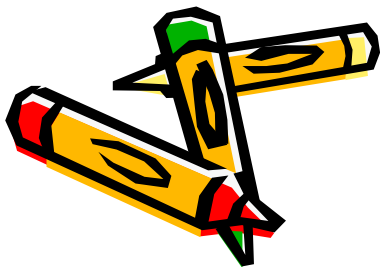
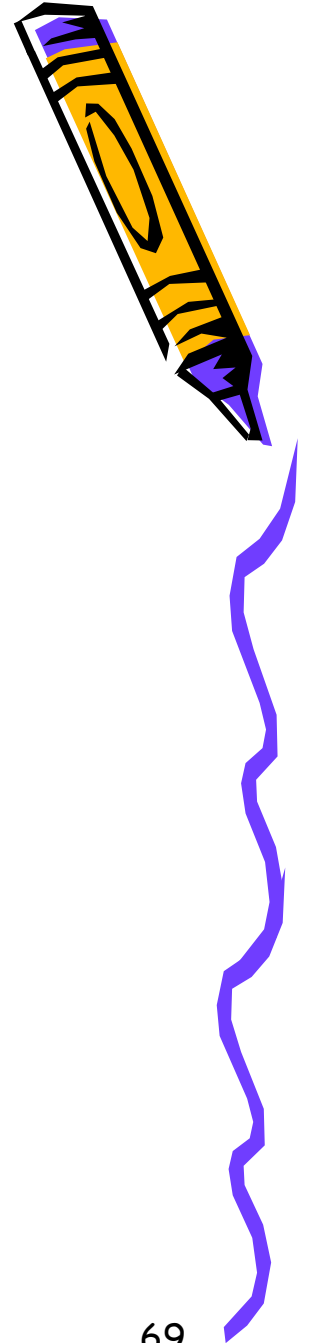
第一章 绪 论

1.1 引 言

1.2 数据模型

1.3 数据库系统的结构

1.4 数据库系统的组成

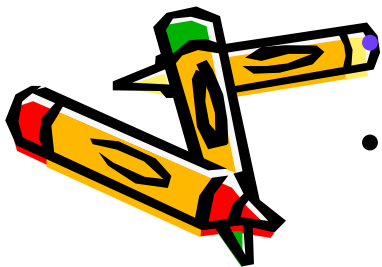




数据模型(data model)



- ◆ 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。
- ◆ 数据模型是数据库中数据的存储方式和操作方式，是数据库系统的基础。
- ◆ 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





- 根据不同模型的应用层次，数据模型分成两个不同的层次

- (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，主要用于数据库设计

- (2) **逻辑模型和物理模型**

- 逻辑模型主要包括层次模型、网状模型、关系模型、面向对象模型、半结构化模型等，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，或在磁盘或磁带上的存储方式和存取方法，是面向计算机系统的。

- **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。

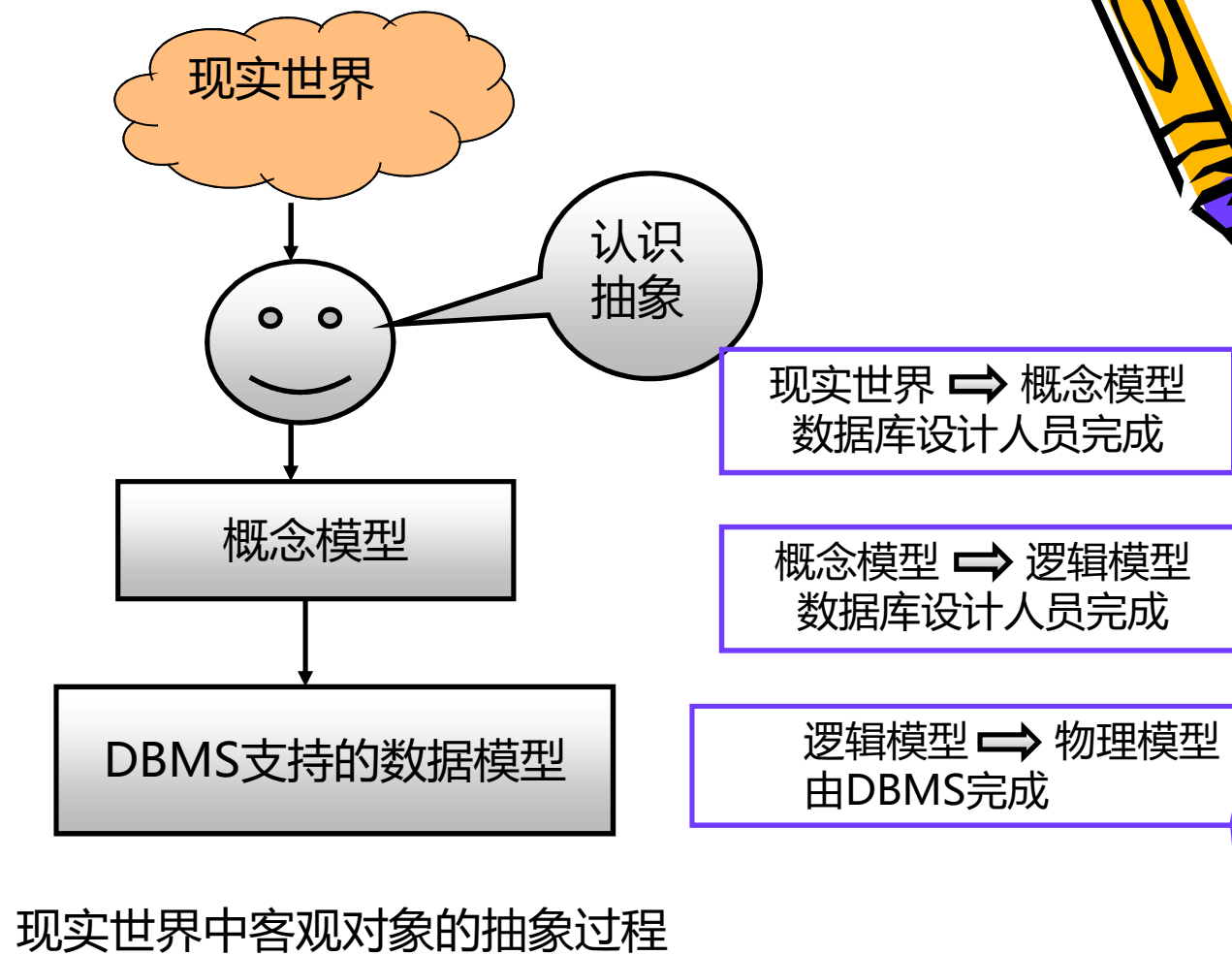
- **概念模型是现实世界到机器世界的一个中间层次。**





信息世界

机器世界





根据不同模型的类型可分为:

<元素名 属性名="属性值"/>
例:
<Student ID="S100">
 <Name>Tom</Name>
</Student>



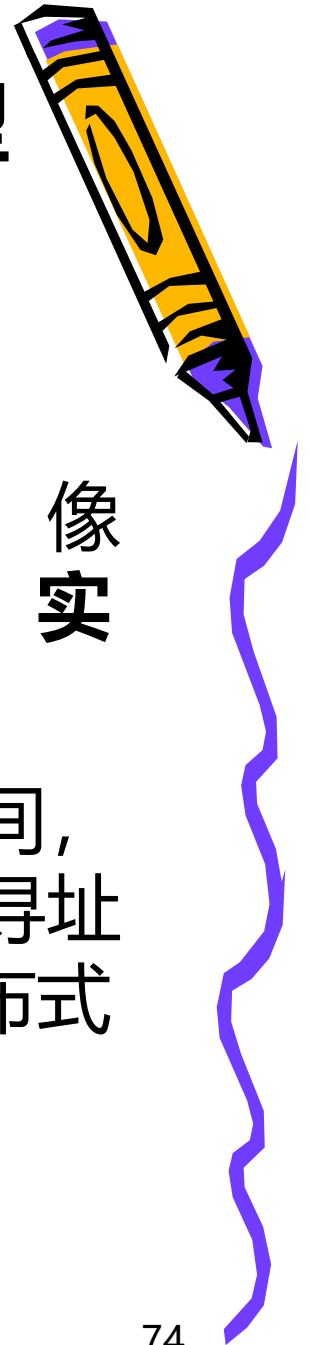
- **XML模型** (extensible markup language) : 一种分层次自描述模型, 是由若干带有标签节点组成的有向树;
- **RDF模型** (resource description framework) : 一种基于XML编写的元数据 (描述数据的数据), 用于描述Web资源的标记语言;
- **超模型**: 一组超实体以及定义在它们上面的关系和约束组成, 为复杂的实体模型建模提供了快捷的方法。





根据不同模型的应用场景，数据模型可分为：

- **离线模型**：主要代表为OLAP模型；
- **在线模型**：可以可靠地处理无限的数据流，像Hadoop（离线计算）批处理大数据一样，**实时处理数据**模型，主要代表为Storm等；
- **近线模型**：定位于在线存储和离线存储之间，是指那些并不经常用到但要求对这些数据寻址迅速、传输率高。很多是基于Hadoop分布式文件系统建立来的。





根据数据模型的发展历程，数据模型可分为：

- **结构化模型**：主要包括层次模型、网状模型、关系模型、E-R模型和面向对象模型等；
- **半结构化模型**：XML模型、JSON模型、RDF模型、图模型和超模型等；
- **OLAP分析模型**：ROLAP模型（如星型模型、雪花模型）、MOLAP模型（如数据立方体）和Storm模型等；
- **大数据模型**：NoSQL模型（key-value模型、key-column模型和key-document模型等）和NewSQL模型。



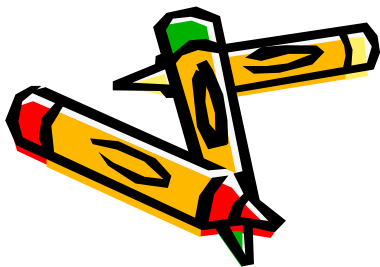
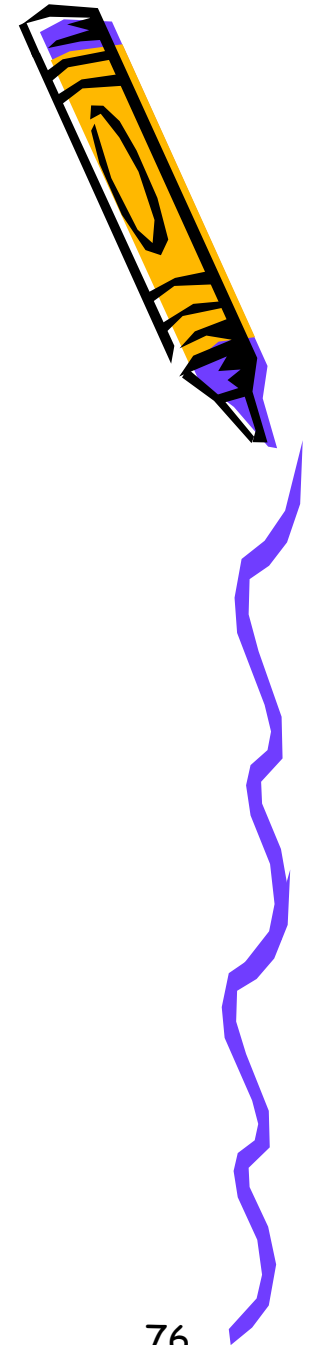


1.2 数据模型

1.2.1 数据模型的要素

1.2.2 概念模型

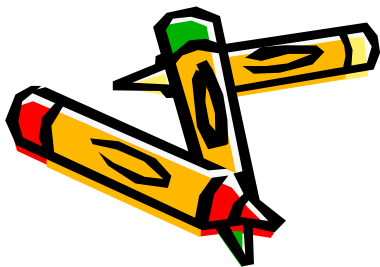
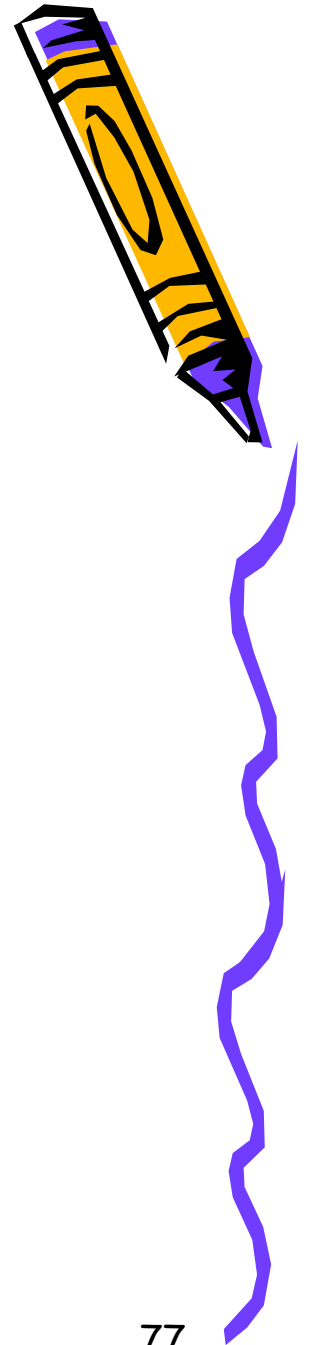
1.2.3 常用的数据模型





1.2.1 数据模型的要素

- ◆ 数据结构
- ◆ 数据操作
- ◆ 完整性约束条件





1. 数据结构

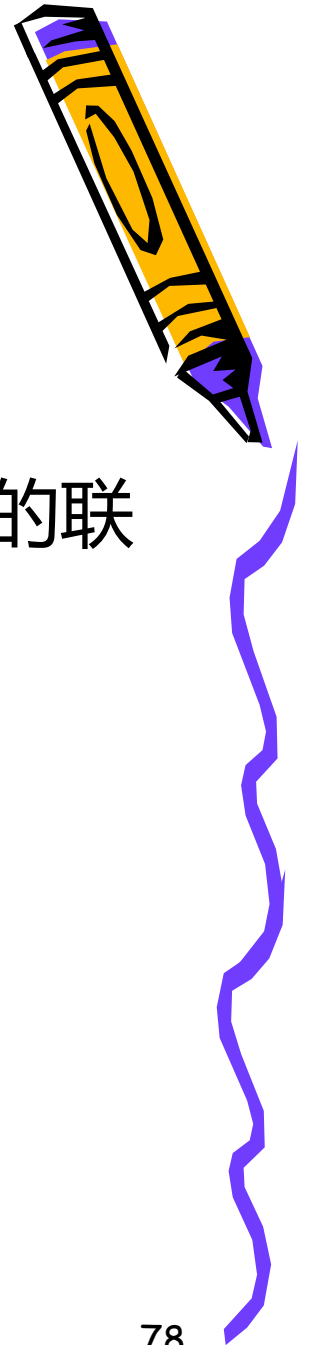
- **什么是数据结构**

- 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**





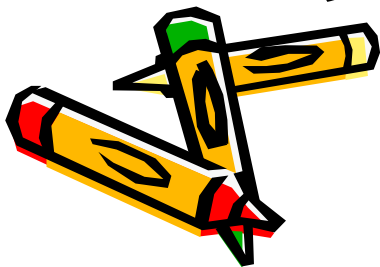
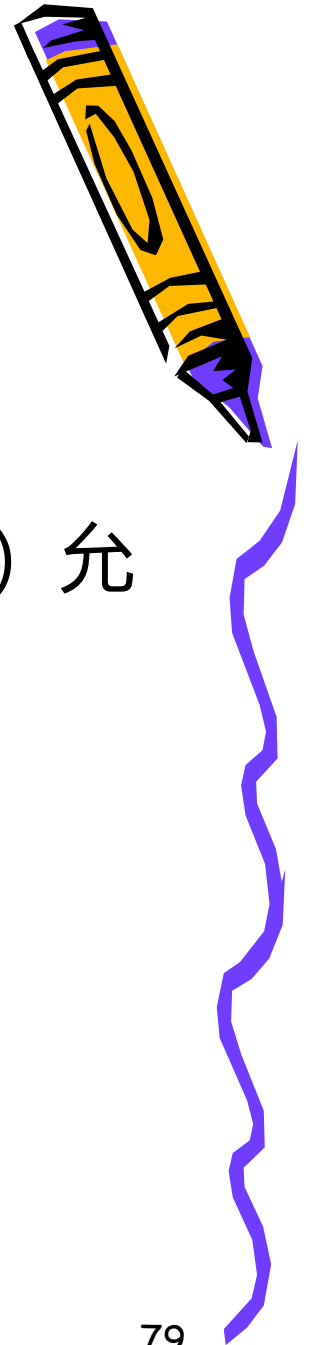
2. 数据操作

- 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**

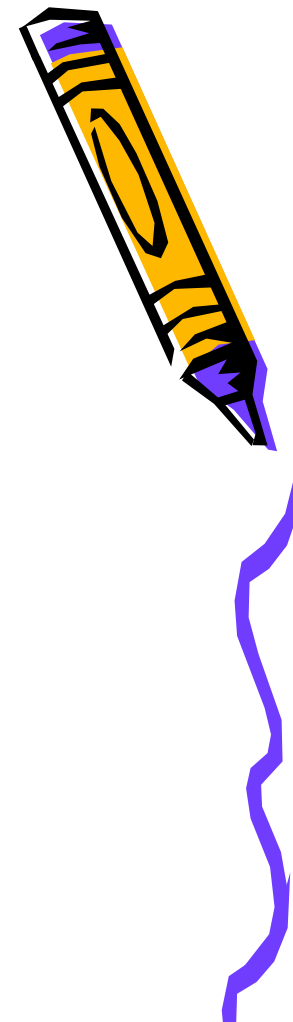
- 数据操作的类型

- 查询
- 更新（包括插入、删除、修改）





- **数据模型必须定义**
 - 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。**

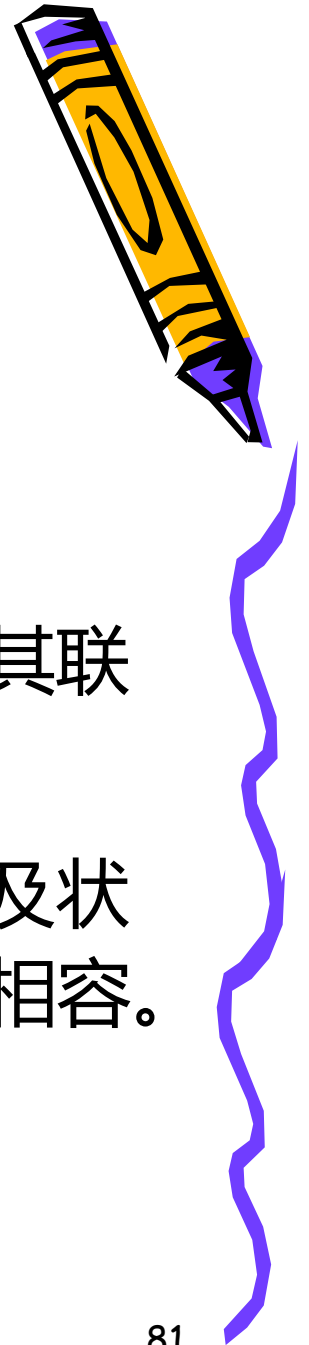




3. 数据的完整性约束条件

- 数据的完整性约束条件

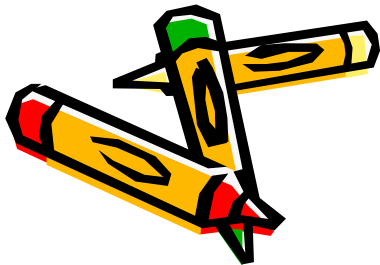
- 一组完整性规则的集合。
- 完整性规则是给定的数据模型中数据及其联系所具有的制约和依存规则，
- 用以限定符合数据模型的数据库状态以及状态的变化，以保证数据的正确、有效、相容。





• 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。
- 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 数据模型的要素

1.2.2 概念模型

1. 概念模型
2. 信息世界中的基本概念
3. 概念模型的表示方法

1.2.3 传统数据模型





1. 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





2. 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。





(4) 域 (Domain)

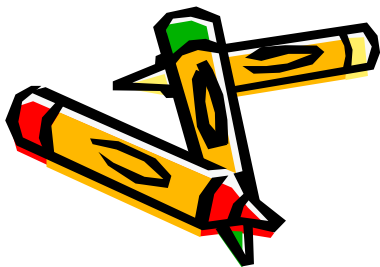
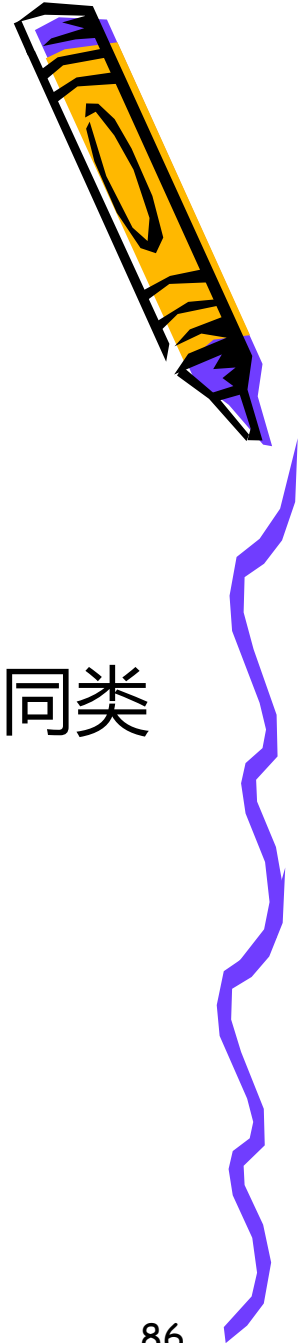
- 属性的取值范围称为该属性的域。

(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)

- 同型实体的集合称为实体集。





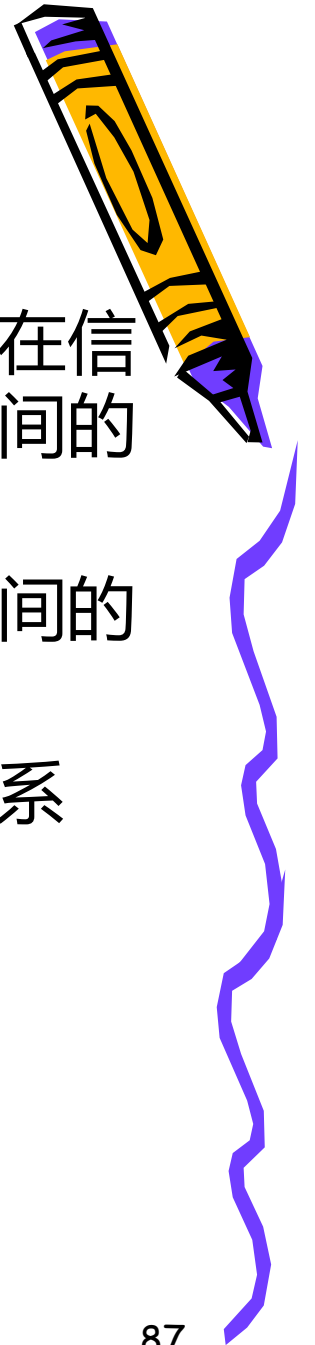
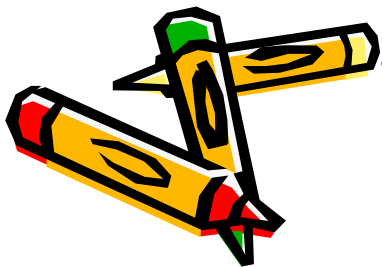
(7) 联系 (Relationship)

– 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：组成实体的各属性之间的联系
- 实体之间的联系：不同实体集之间的联系

– 三类实体型间联系

- 一对一联系 (1:1)
- 一对多联系 (1:n)
- 多对多联系 (m:n)





两个实体型间的联系

- 一对一联系

- 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。

- 实例

班级与班长之间的联系：一个班级只有一个正班长，而一个班长只在一个班中任职



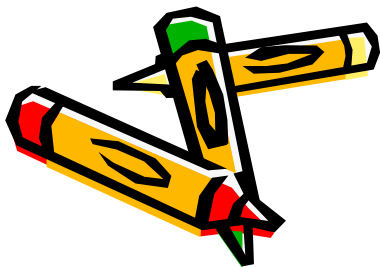


• 一对多联系

– 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B有一对多联系。记为1:n。

– 实例

班级与学生之间的联系：一个班级中有若干名学生，而每个学生只在一个班级中学习



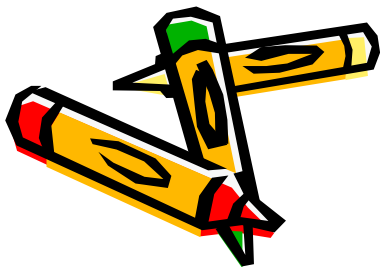


• 多对多联系 (m:n)

- 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。

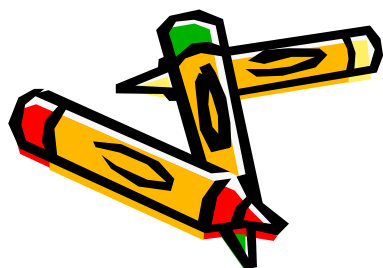
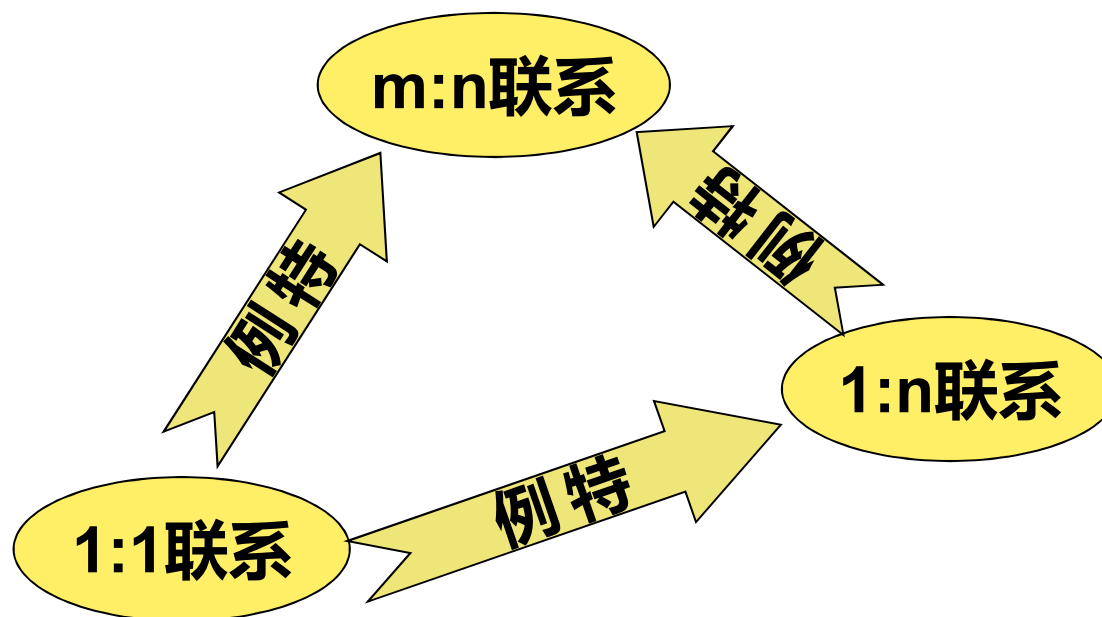
– 实例

课程与学生之间的联系：一门课程同时有若干个学生选修，而一个学生可以同时选修多门课程





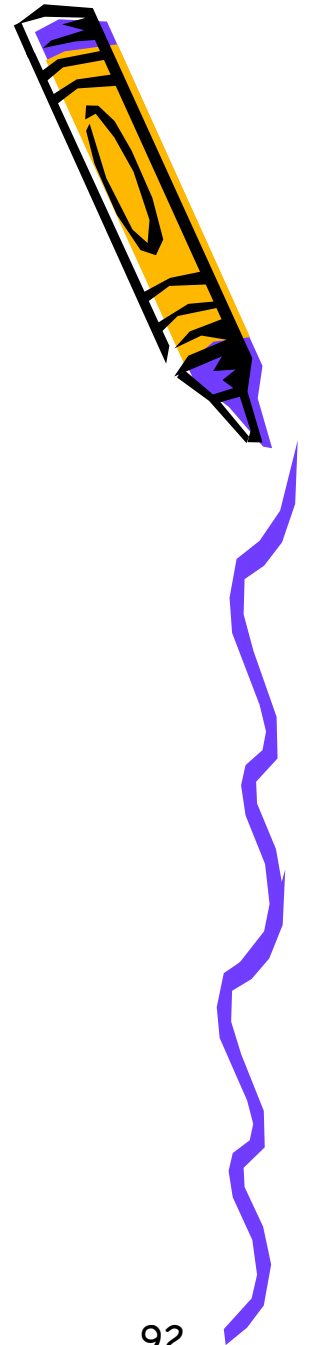
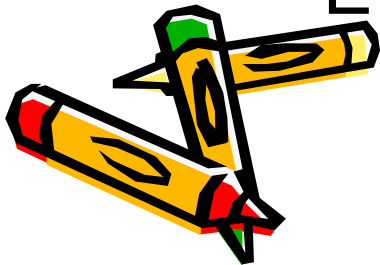
• 三类联系之间的关系





3. 概念模型的表示方法

- 概念模型的表示方法很多
- 实体 - 联系方法(E-R方法)
 - 是最为常用的概念模型表示方法。
 - 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。





1.2 数据模型

1.2.1 数据模型的要素

1.2.2 概念模型

1.2.3 传统数据模型

一、层次模型

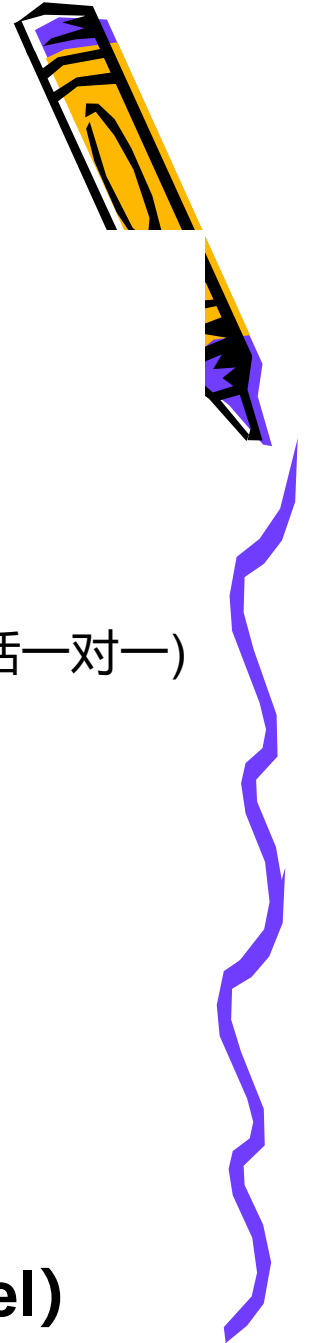
二、网状模型

三、关系模型





最常用的数据模型



- **格式化（非关系）模型**

- 种类

- 层次模型 (Hierarchical Model)
 - 网状模型(Network Model)

- 数据结构：以基本层次联系为基本单位

- 基本层次联系：两个记录以及它们之间的一对多（包括一对一）的联系

- **关系模型(Relational Model)**

- 数据结构：表

- **面向对象模型(Object Oriented Model)**

- 数据结构：对象

- **对象关系模型(Object Relational Model)**

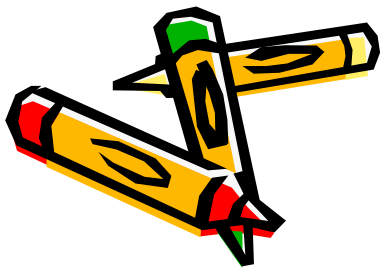
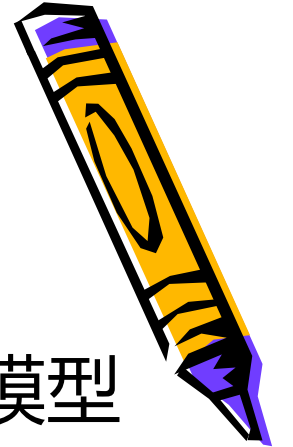
- **半结构化数据模型 (Semistructure data model)**





一、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 (1968年推出)
- 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、层次模型

1. 层次数据模型的数据结构
2. 层次数据模型的优缺点





1. 层次数据模型的数据结构

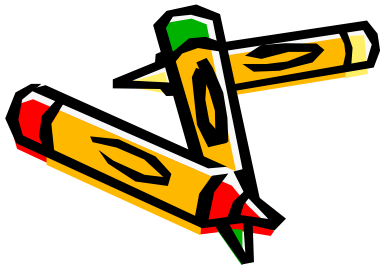
- **层次模型**

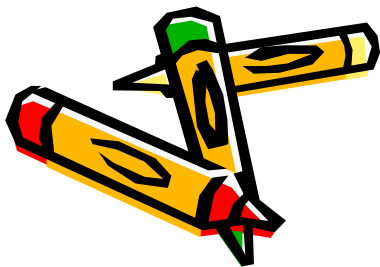
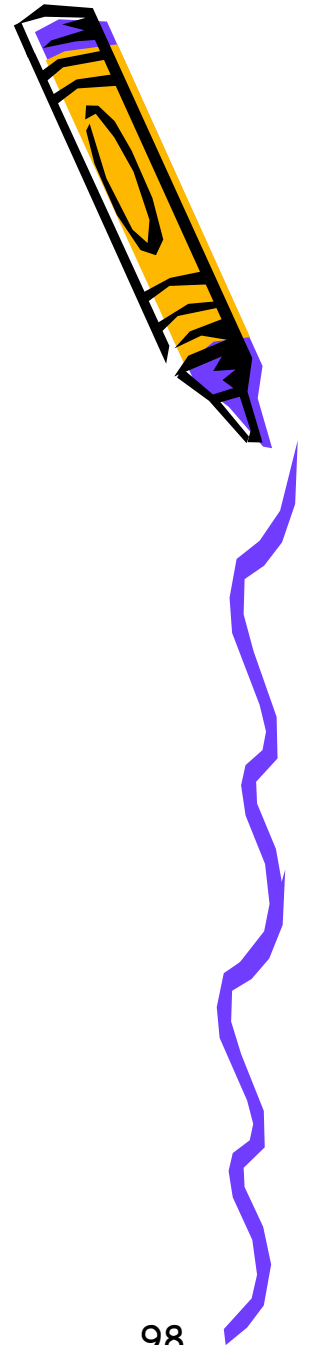
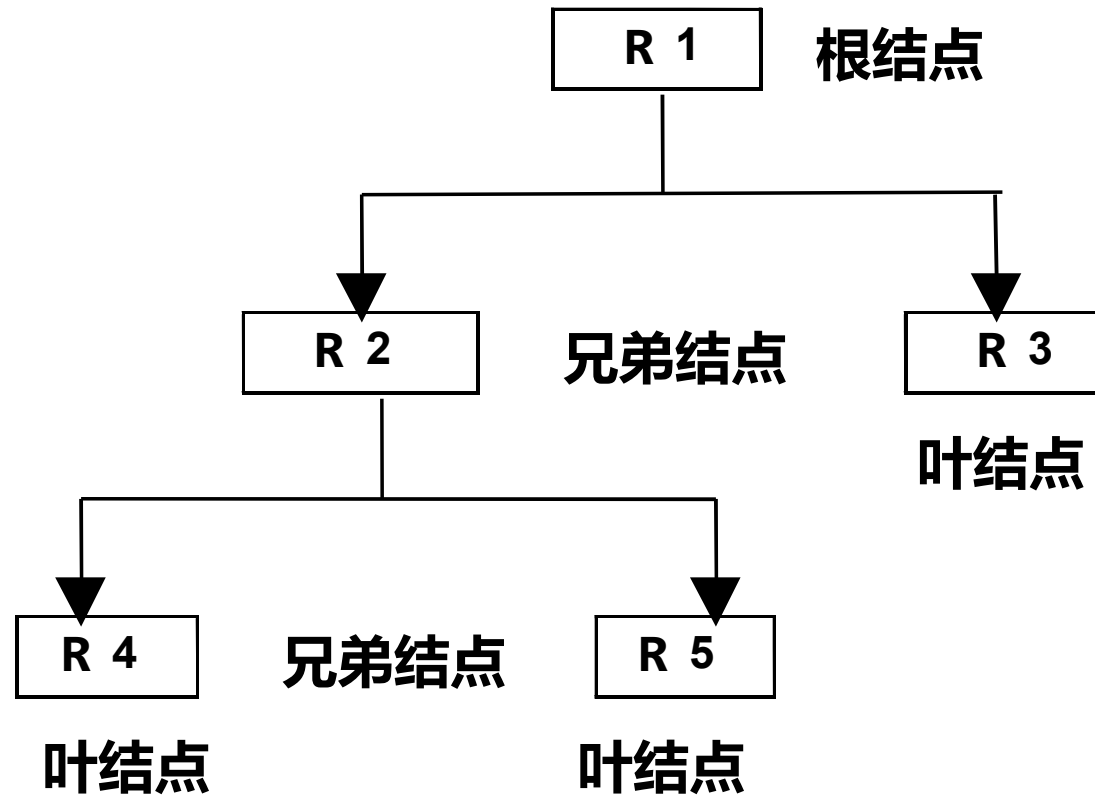
满足下面两个条件的基本层次联系的集合为层次模型。

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**

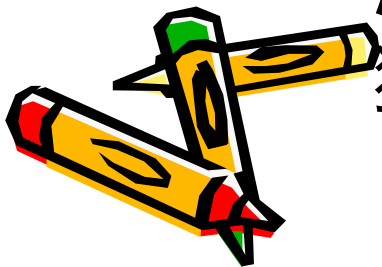






• 特点

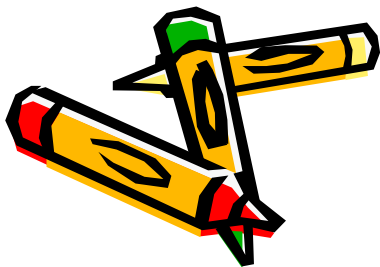
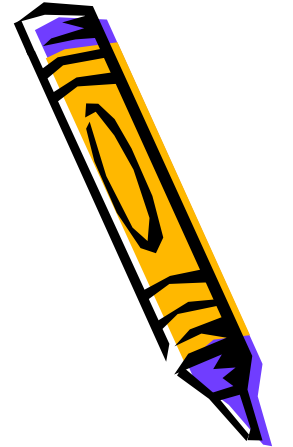
- 结点的双亲是唯一的
- 只能直接处理一对多的实体联系
- 每个记录类型定义一个排序字段，也称为码字段
- 任何记录值只有按其路径查看时，才能显出它的全部意义
- 没有一个子女记录值能够脱离双亲记录值而独立存在

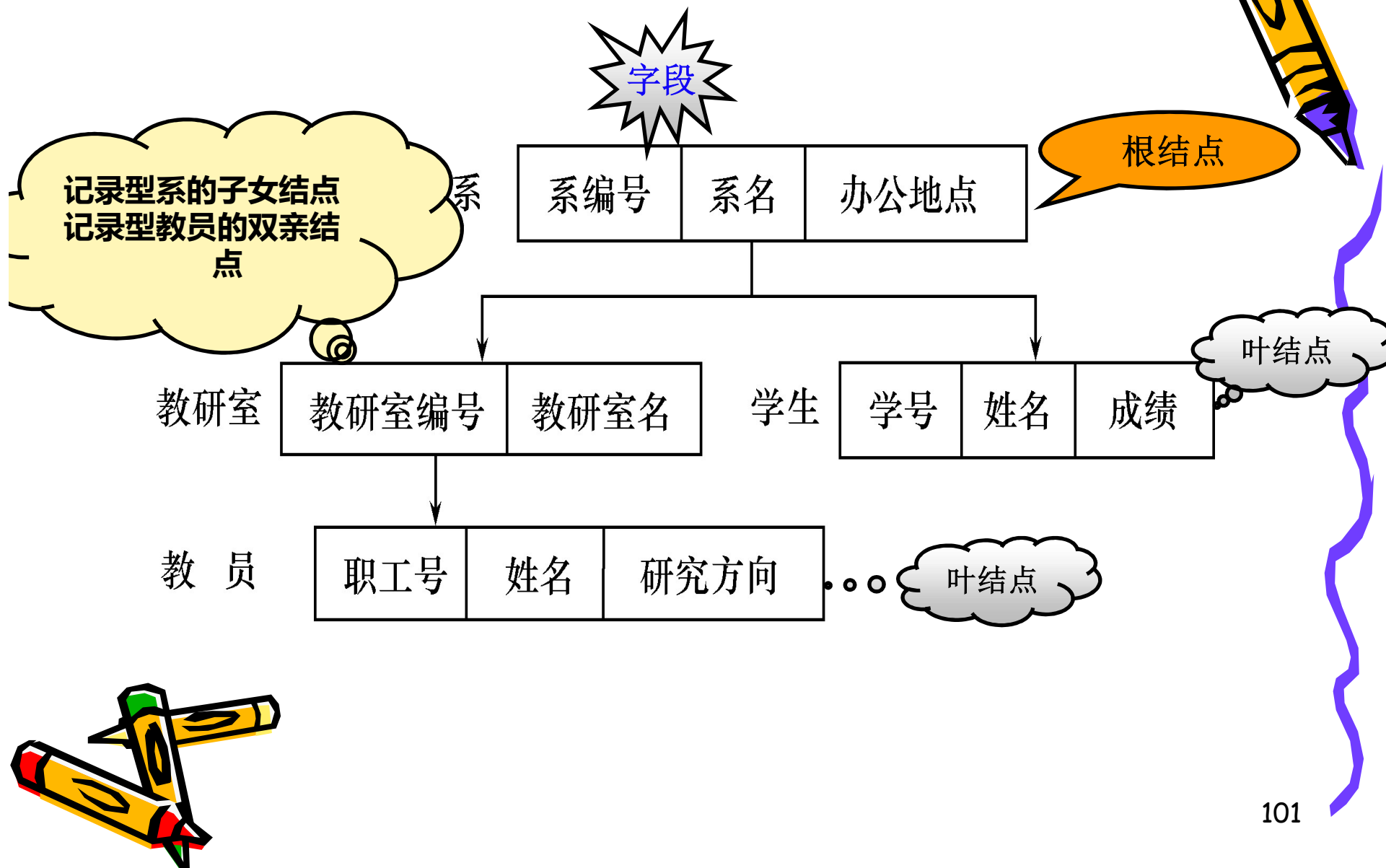


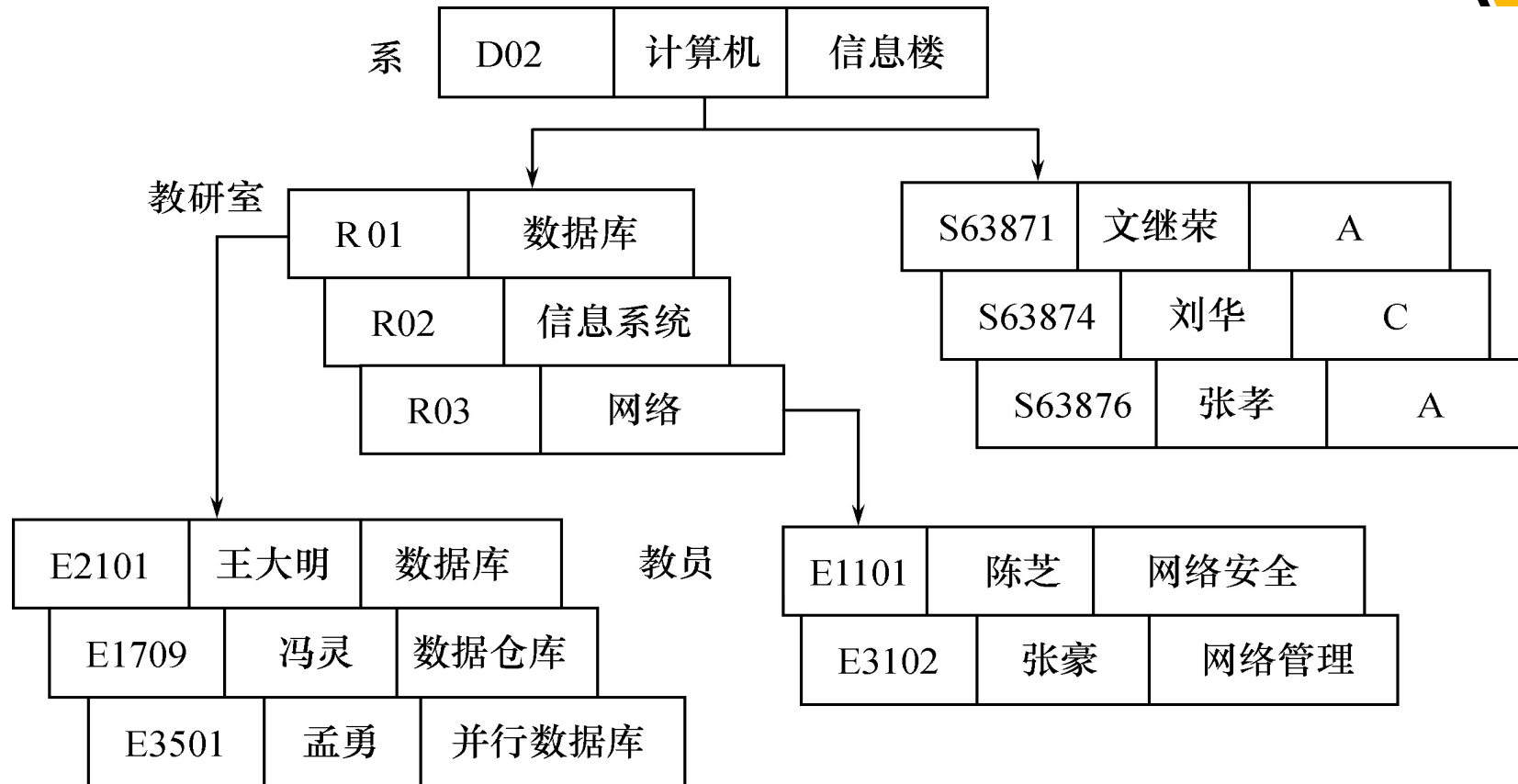
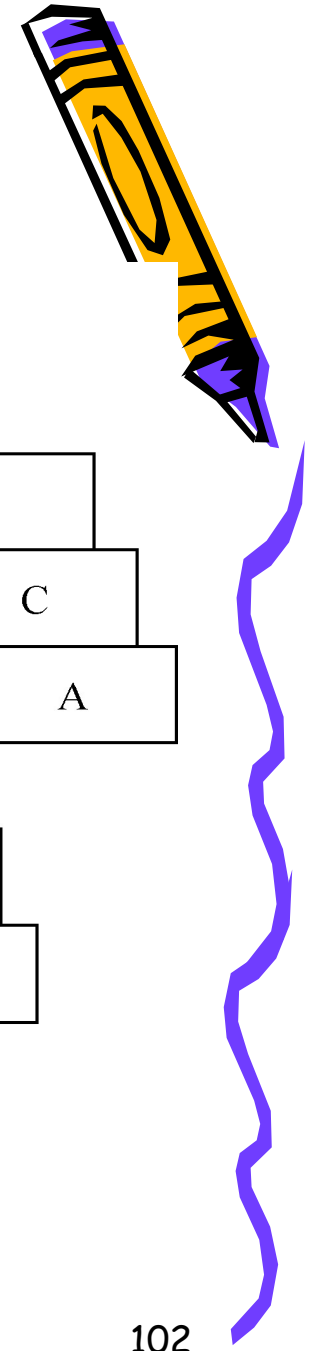


• 表示方法

- 实体型：用记录类型描述。每个结点表示一个记录类型。
- 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（有向边）表示记录（类）型之间的一对多的联系。









• 多对多联系在层次模型中的表示

– 用层次模型**间接**表示多对多联系

– 表示方法

- 将多对多联系**分解**成一对多联系

– 分解方法

- 冗余结点法

- 虚拟结点法





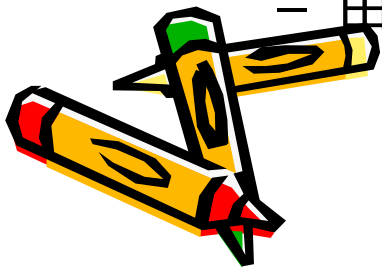
2. 层次模型的优缺点

- 优点

- 层次数据模型简单，对具有一对多的层次关系的部门描述自然、直观，容易理解
- 对于实体间联系是固定的，且预先定义好的应用系统，其性能优于关系模型，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





1.2.3 逻辑模型

一、层次模型

二、网状模型

三、关系模型





二、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
 - 69-70年代初由DBTG提出的一个系统方案
 - 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



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为网状模型。

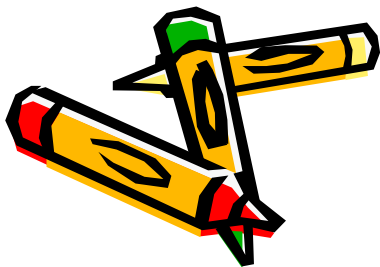
1. 允许一个以上的结点无双亲；
2. 一个结点可以有多个的双亲。





• 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。
- **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。





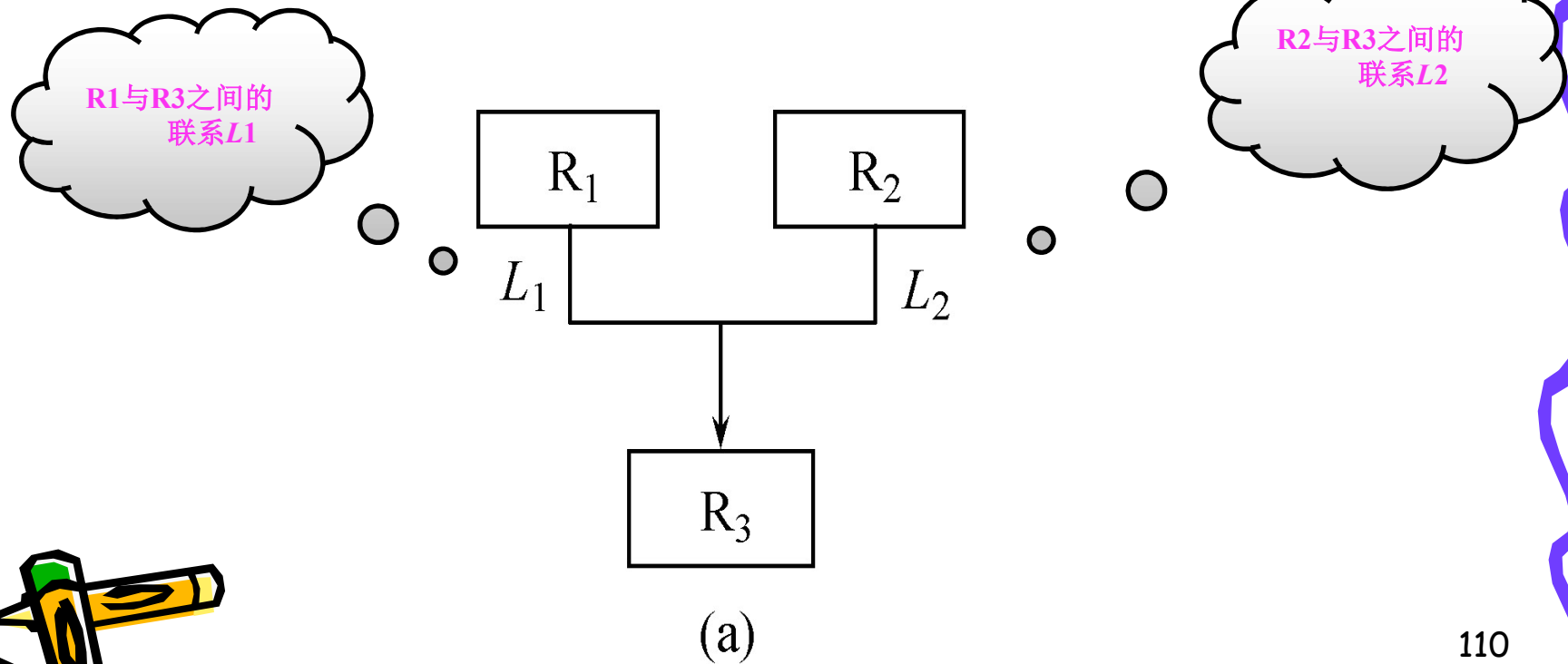
• 网状模型与层次模型的区别

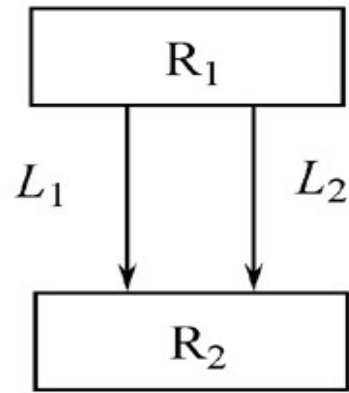
- 网状模型允许多个结点没有双亲结点
- 网状模型允许结点有多个双亲结点
- 网状模型允许两个结点之间有多种联系
(复合联系)
- 网状模型可以更直接地去描述现实世界
- 层次模型实际上是网状模型的一个特例



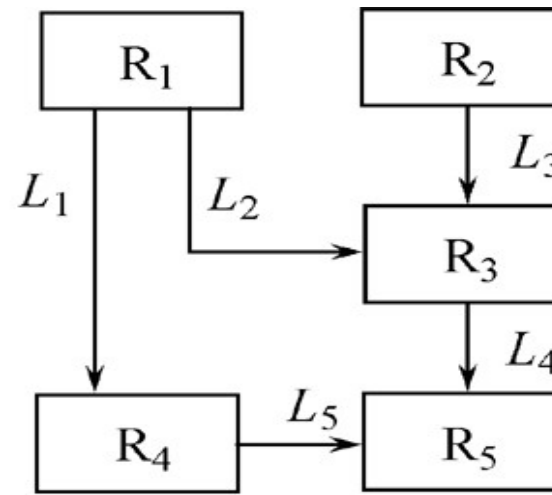


- ❖ 网状模型中子女结点与双亲结点的联系可以不唯一
要为每个联系命名，并指出与该联系有关的双亲记录和子女记录



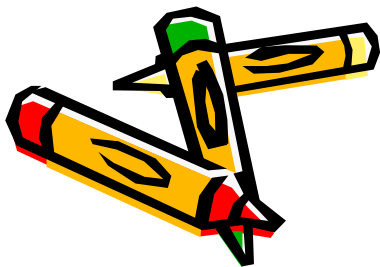
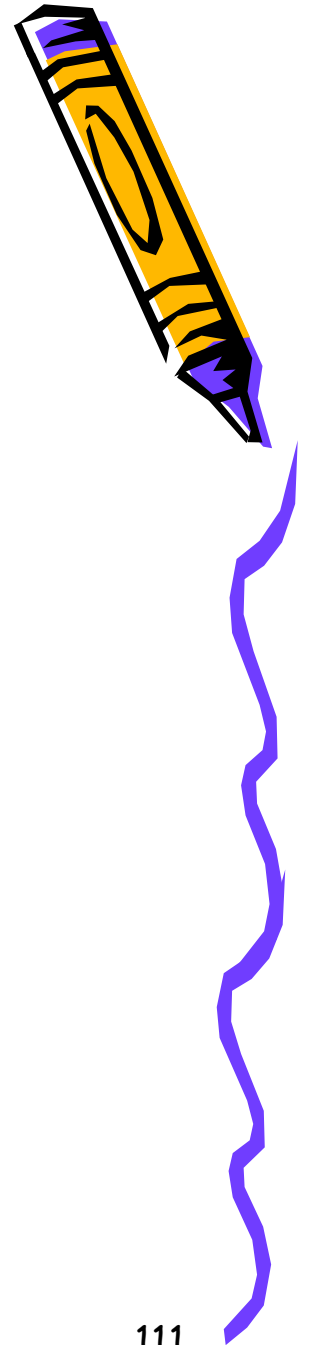


(b)



(c)

网状模型的例子





• 多对多联系在网状模型中的表示

- 用网状模型间接表示多对多联系
- 表示方法

- 将多对多联系直接分解成一对多联系

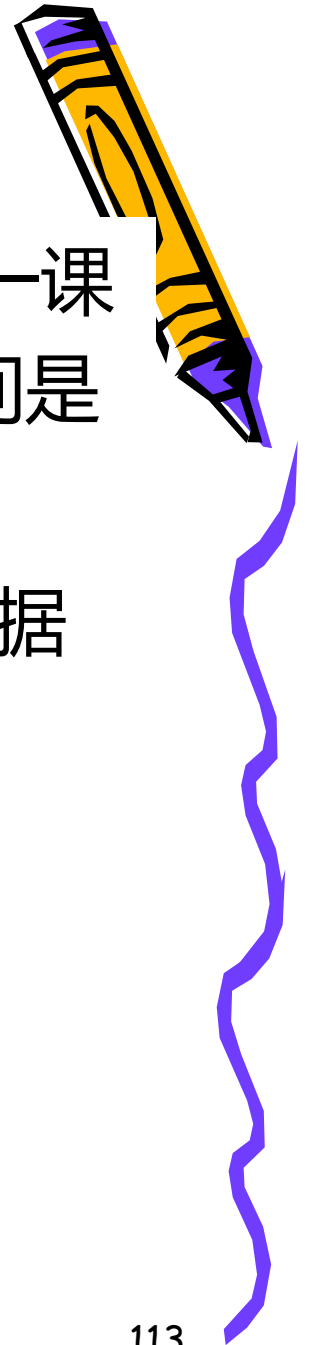




例如：一个学生可以选修若干门课程，某一课程可以被多个学生选修，学生与课程之间是多对多联系

■ 引进一个学生选课的联系记录，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



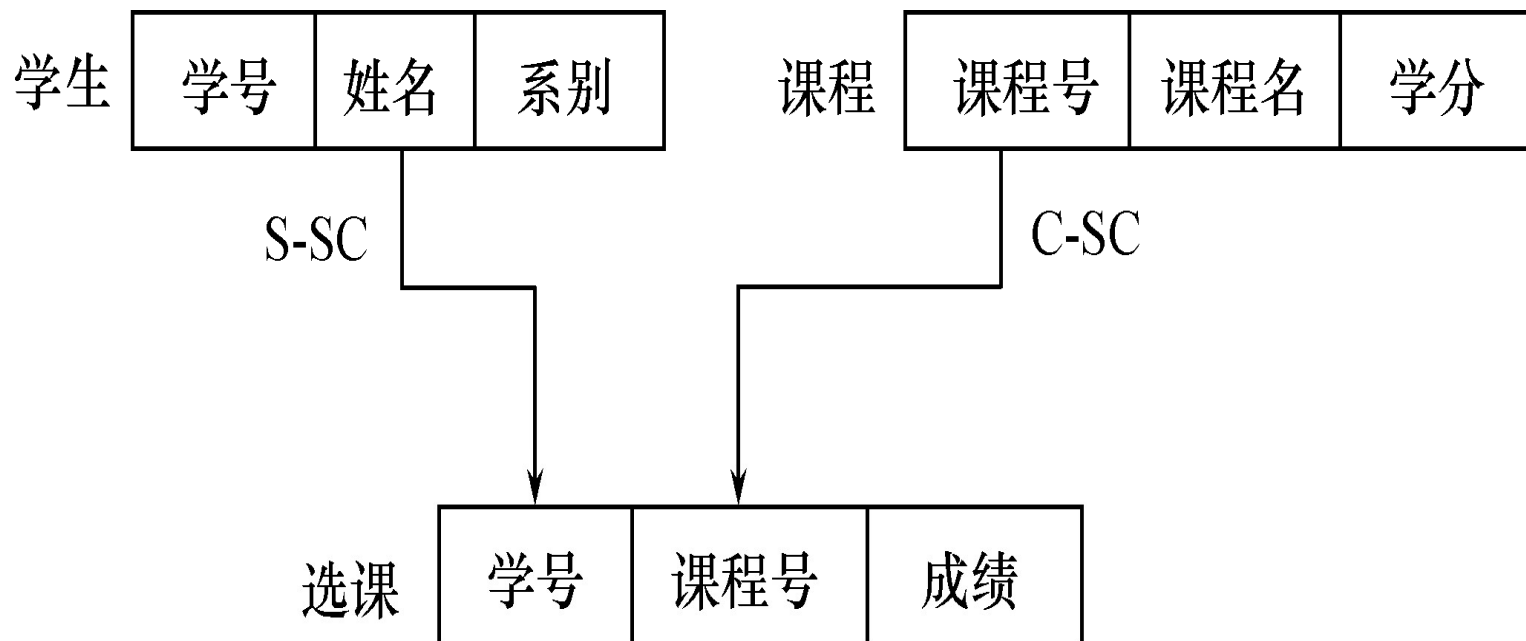


图 学生/选课/课程的网状数据模型





2. 网状模型的优缺点

- **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





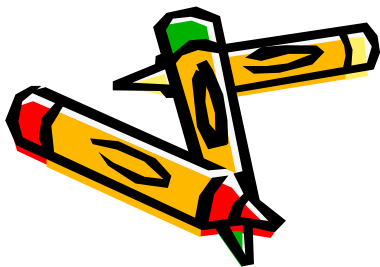
1.2.3 传统数据模型



一、层次模型

二、网状模型

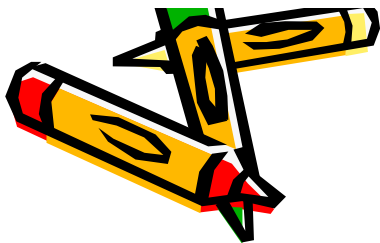
三、关系模型





三、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





三、关系模型

1. 关系数据模型的数据结构
2. 关系数据模型的操纵
3. 关系数据模型的完整性约束
4. 关系数据模型的存储结构
5. 关系数据模型的优缺点





1. 关系数据模型的数据结构

- 关系模型的基本数据结构
- 关系模型的基本概念
- 实体及实体间的联系的表示方法
- 关系必须是规范化的





• 关系模型的基本数据结构

- 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 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





• 关系模型的基本概念

– 关系 (Relation)

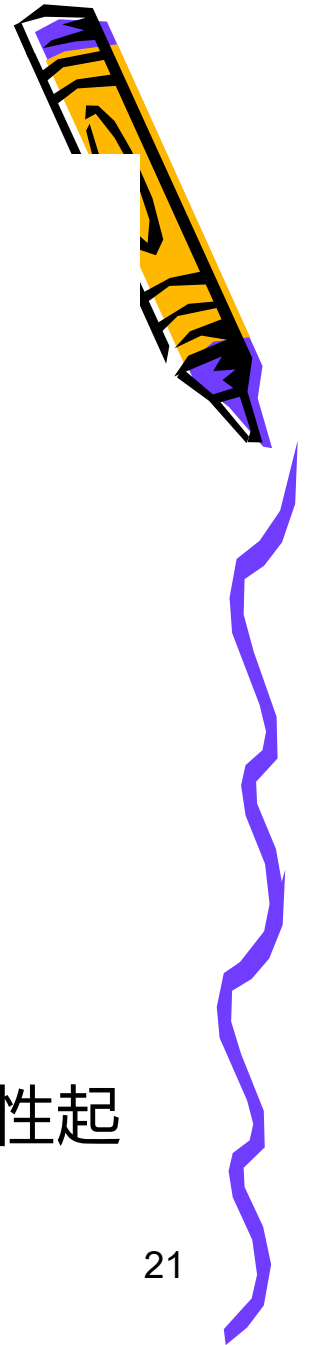
- 一个关系对应通常说的一张表。

– 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。





– 码 (Key)

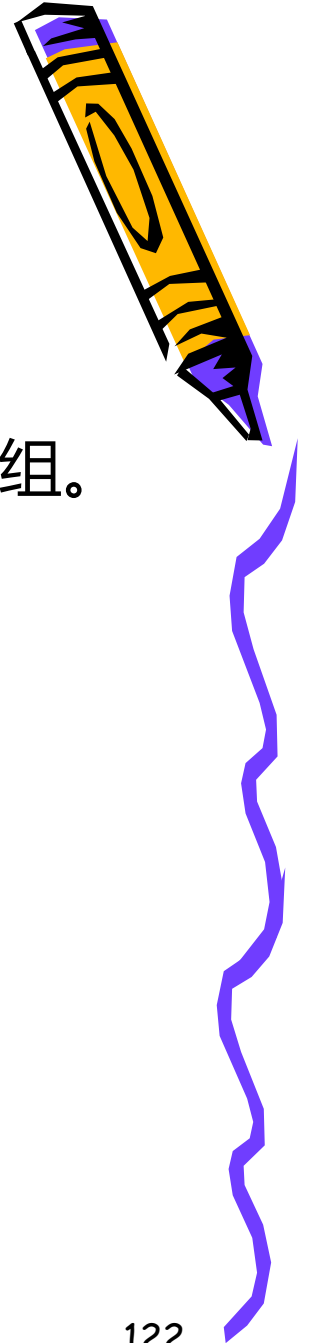
- 表中的某个属性组，它可以唯一确定一个元组。

– 域 (Domain)

- 属性的取值范围。

– 分量

- 元组中的一个属性值。





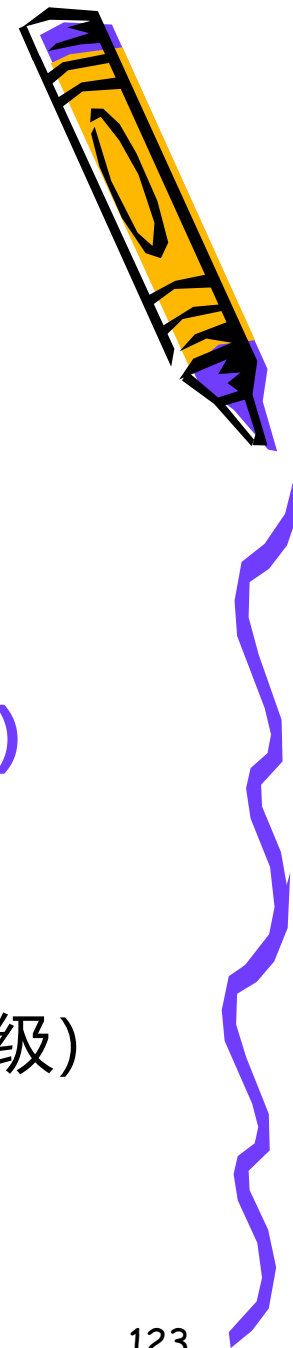
- 关系模式

- 对关系的描述
- 表示方法

关系名 (属性1, 属性2, ..., 属性n)

例如:

学生 (学号, 姓名, 年龄, 性别, 系, 年级)

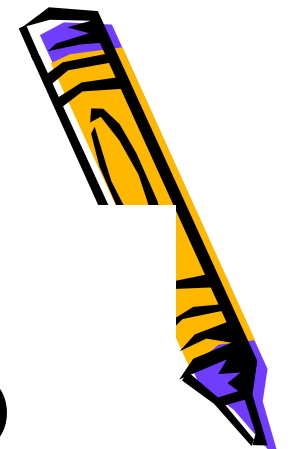




• 实体及实体间的联系的表示方法

- 实体型：直接用关系（表）表示
- 属性：用属性名表示
- 一对一联系：隐含在实体对应的关系中
- 一对多联系：隐含在实体对应的关系中
- 多对多联系：直接用关系表示





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

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）。

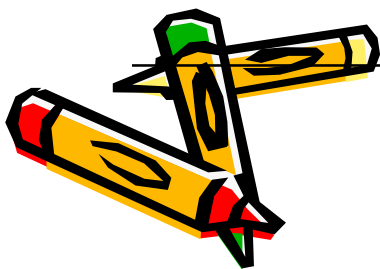
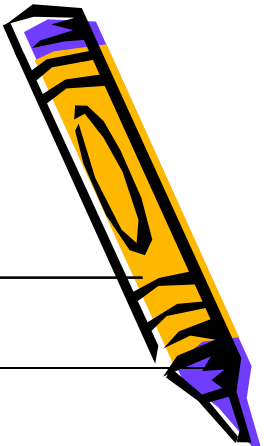
职工号	姓名	职称	工资			扣除		实发
			基本	工龄	职务	房租	水电	
86051	陈平	讲师	105	9.5	15	6	12	115.5





表1.2 术语对比

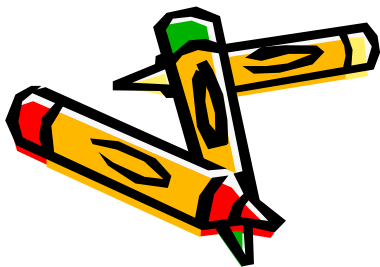
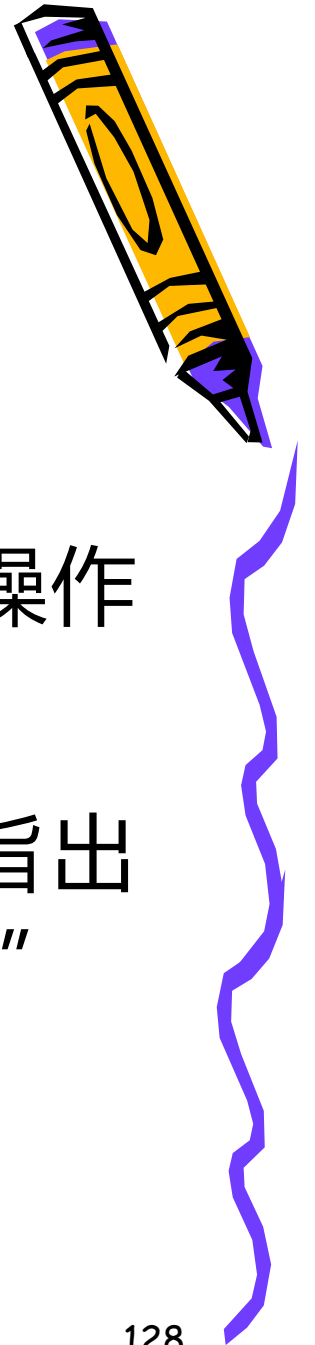
关系术语	一般表格的术语
关系名	表名
关系模式	表头（表格的描述）
关系	（一张）二维表
元组	记录或行
属性	列
属性名	列名
属性值	列值
分量	一条记录中的一个列值
非规范关系	表中有表（大表中嵌有小表）





2. 关系模型的数据操纵

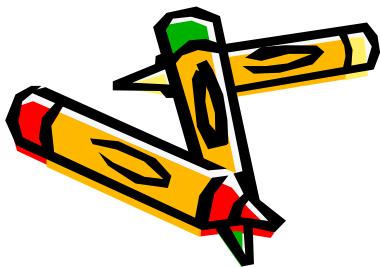
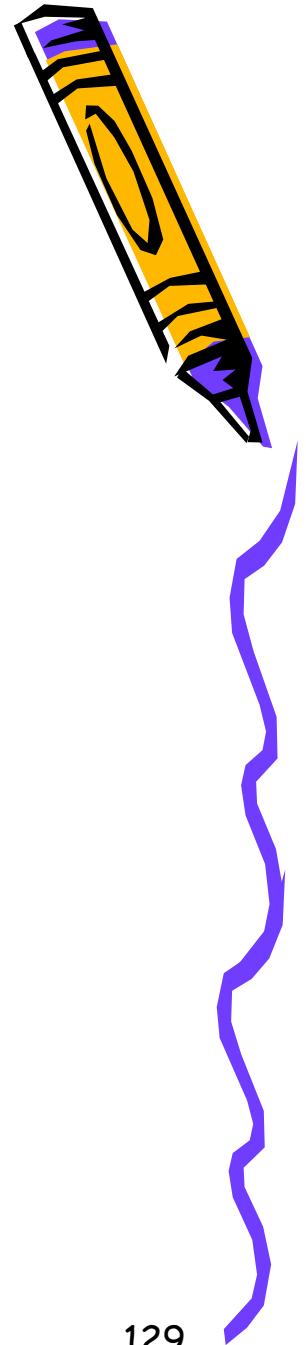
- 查询、插入、删除、修改
- 数据操作是集合操作，操作对象和操作结果都是关系，即若干元组的集合
- 存取路径对用户隐蔽，用户只要指出“干什么”，不必详细说明“怎么干”





3. 关系模型的完整性约束

- 实体完整性
- 参照完整性
- 用户定义的完整性

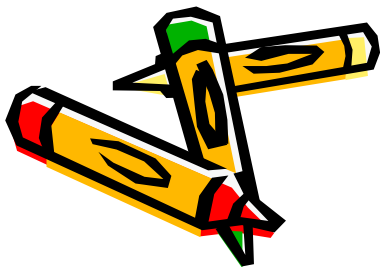




4. 关系数据模型的存储结构

- 表以文件形式存储

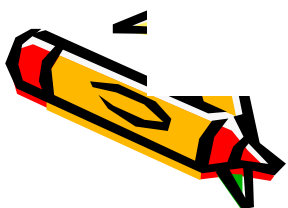
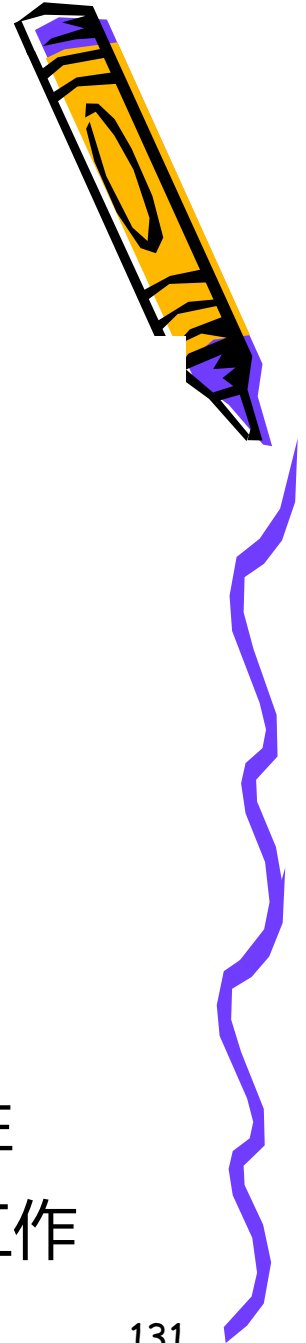
- 有的DBMS一个表对应一个操作系统文件
- 有的DBMS自己设计文件结构





5. 关系模型的优缺点

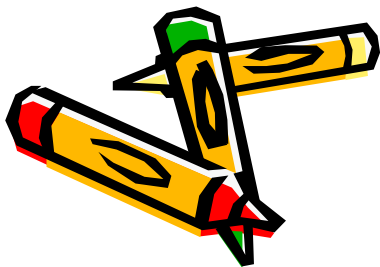
- 优点
 - 建立在严格的数学概念的基础上
 - 概念单一
 - 实体和各类联系都用关系来表示。
 - 对数据的检索结果也是关系。
 - 关系模型的存取路径对用户透明
 - 具有更高的数据独立性，更好的安全保密性
 - 简化了程序员的工作和数据库开发建立的工作

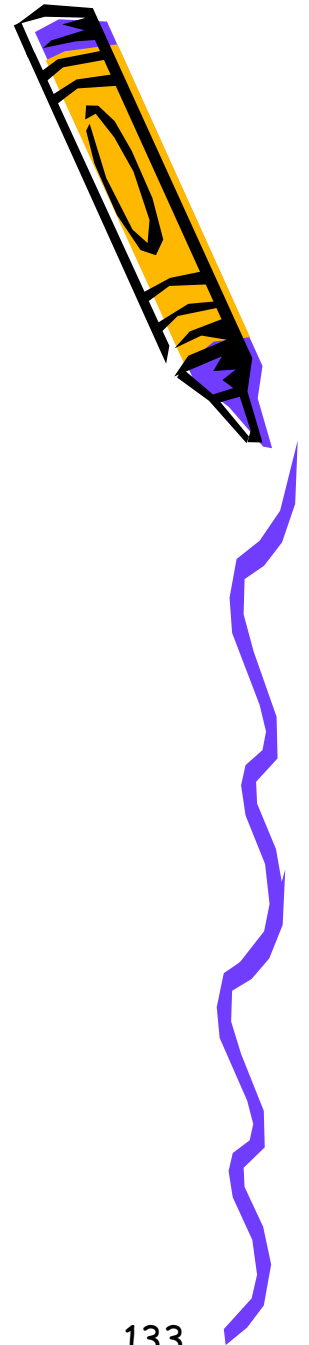
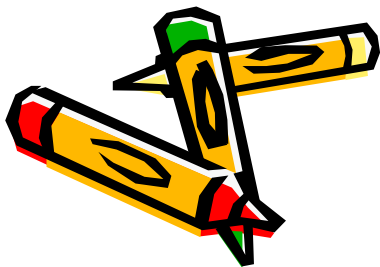




• 缺点

- 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 引言

1.2 数据模型

1.3 数据库系统的结构

1.4 数据库系统的组成





1.3 数据库系统的结构

1.3.1 数据库系统的模式结构

— 从数据库管理系统角度看

1.3.2 数据库系统的体系结构

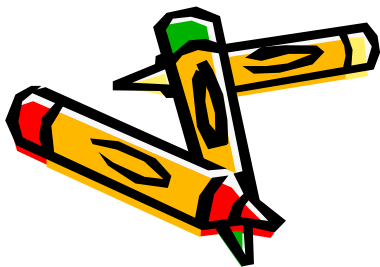
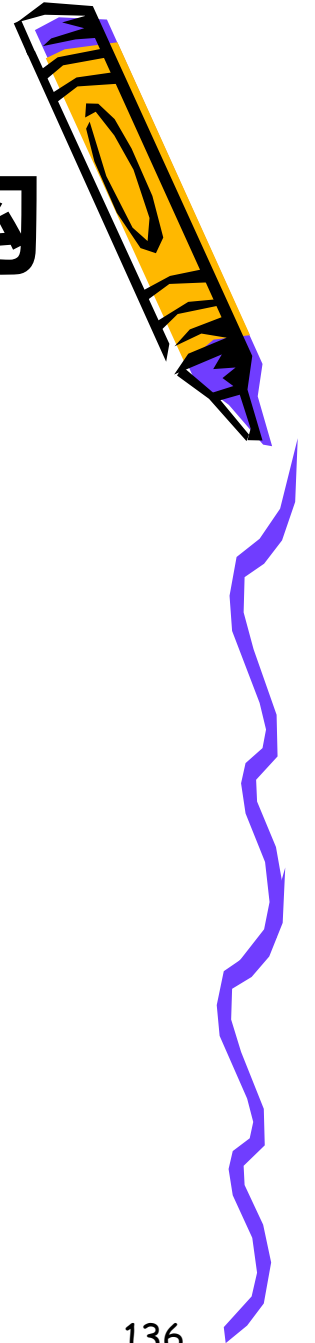
— 从数据库最终用户角度看





1.3.1 数据库系统的模式结构

- 数据库系统模式的概念
- 数据库系统的三级模式结构
- 数据库的二级映象功能与数据独立性





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

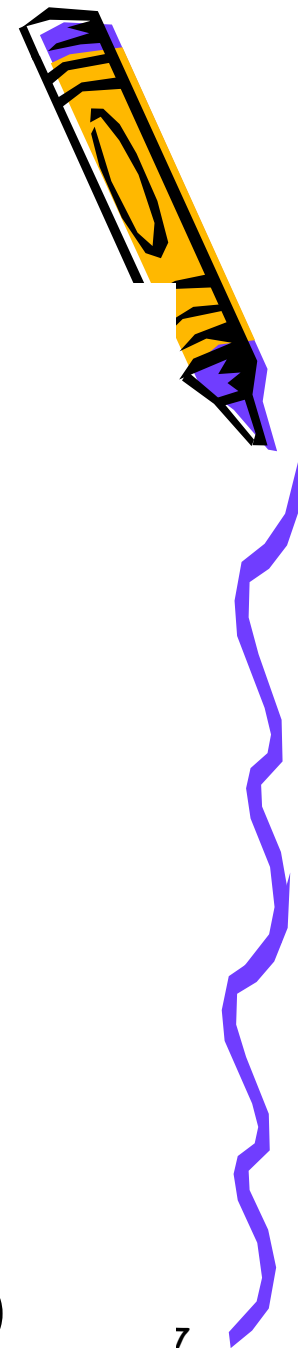
例如：学生记录

记录型：

(学号，姓名，性别，系别，年龄，籍贯)

该记录型的一个记录值：

(900201，李明，男，计算机，22，江苏)



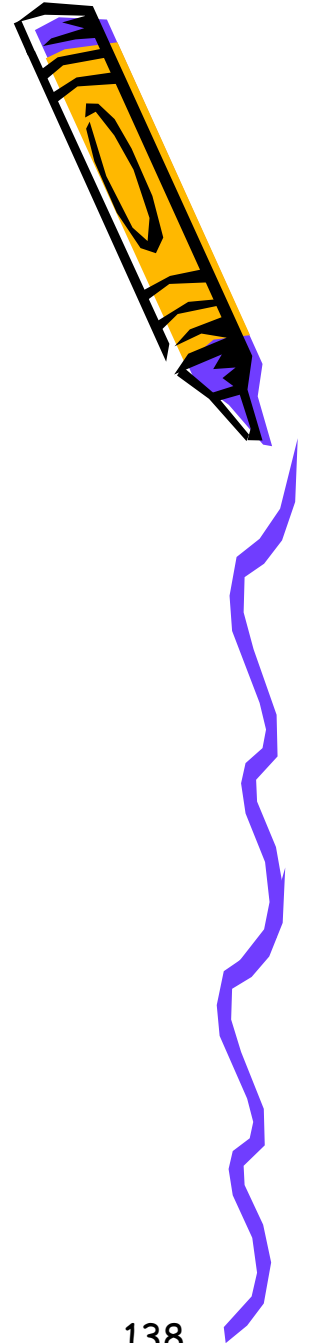
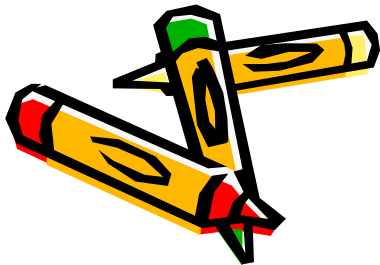


– 模式 (Schema)

- 数据库逻辑结构和特征的描述
- 是型的描述
- 反映的是数据的结构及其联系
- 模式是相对稳定的

– 模式的一个实例 (Instance)

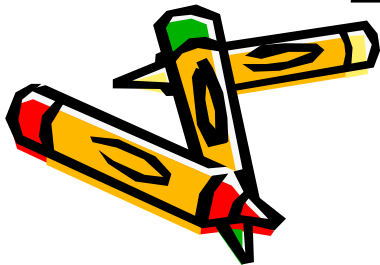
- 模式的一个具体值
- 反映数据库某一时刻的状态
- 同一个模式可以有很多实例
- 实例随数据库中的数据更新而变动





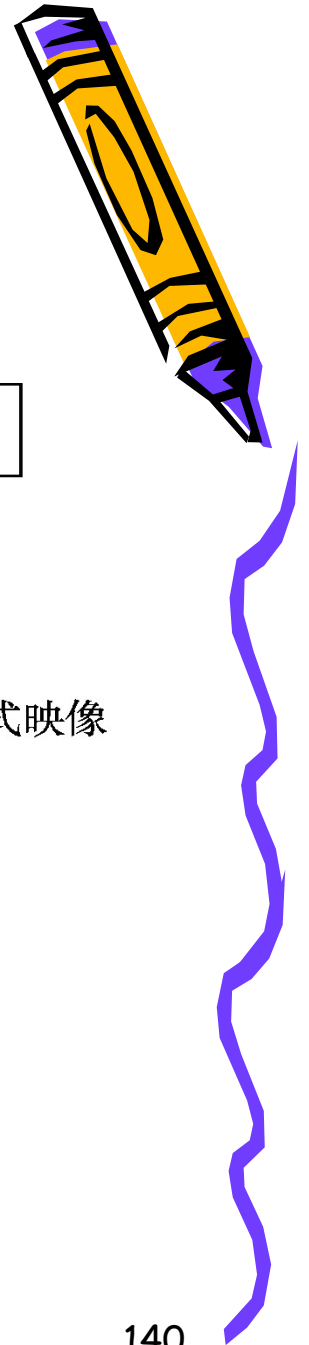
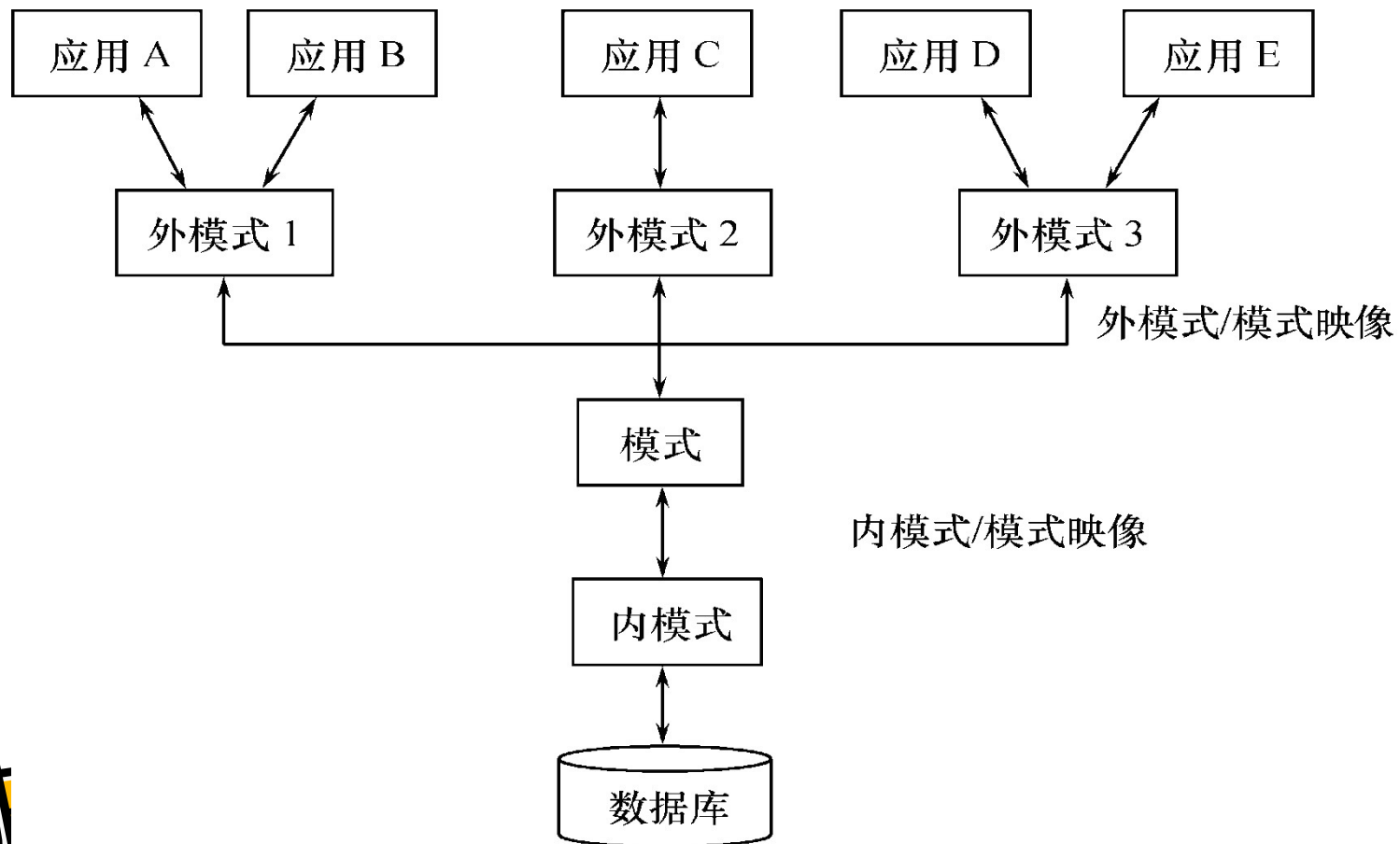
例如：在学生选课数据库模式中，包含学生记录、课程记录和学生选课记录

- 2016年的一个学生数据库实例，包含：
 - 2016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
- 2015年度学生数据库模式对应的实例与2016年度学生数据库模式对应的实例是**不同**的





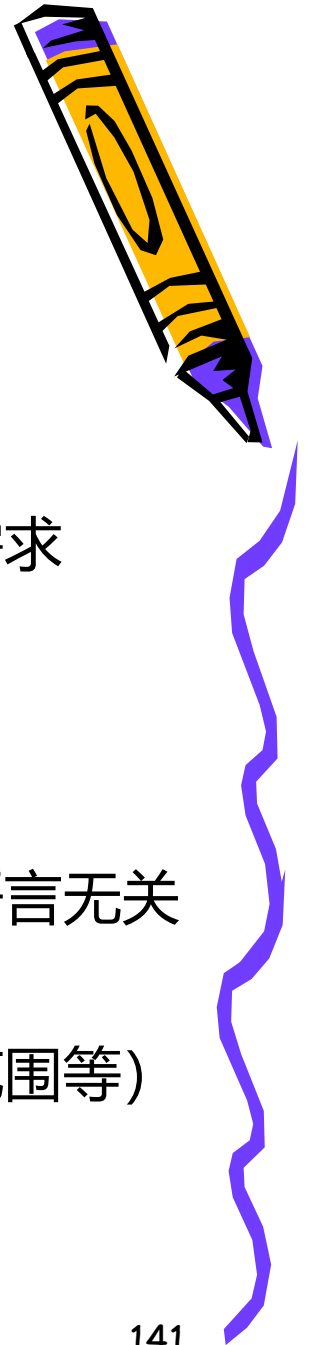
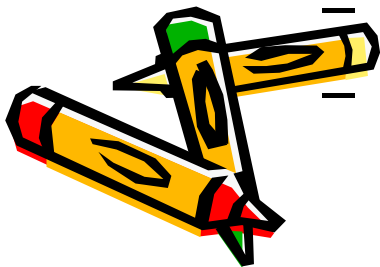
数据库系统的三级模式结构





1. 模式(Schema)

- **模式（也称逻辑模式）**
 - 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





2. 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

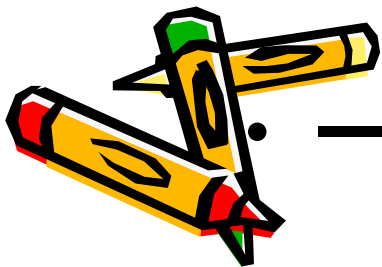




3. 内模式(Internal Schema)

- **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（顺序存储，按照B树结构存储，按hash方法存储）
 - 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（如变长或定长结构，一个记录是否能跨物理页存储）

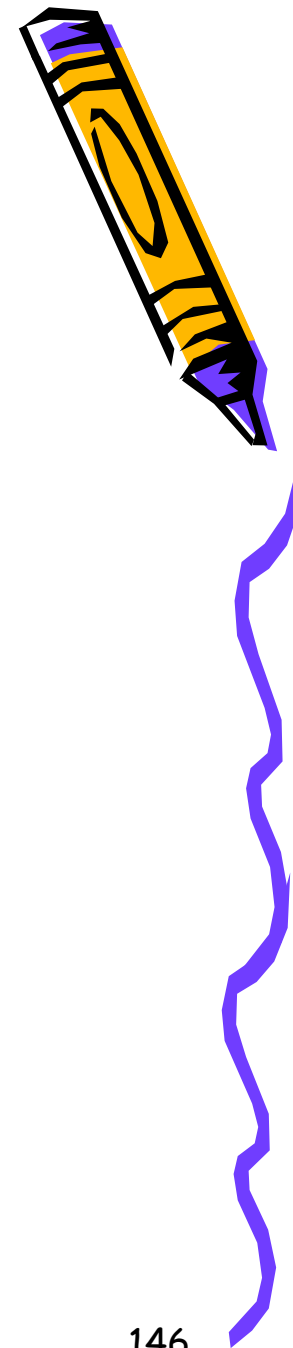
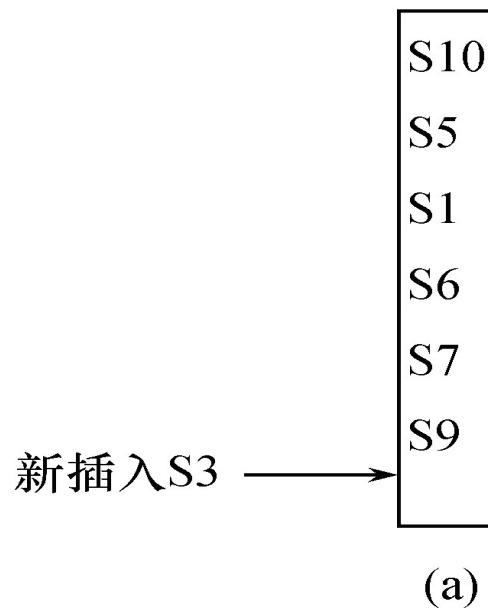


- **一个数据库只有一个内模式**





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





- 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（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（c）所示

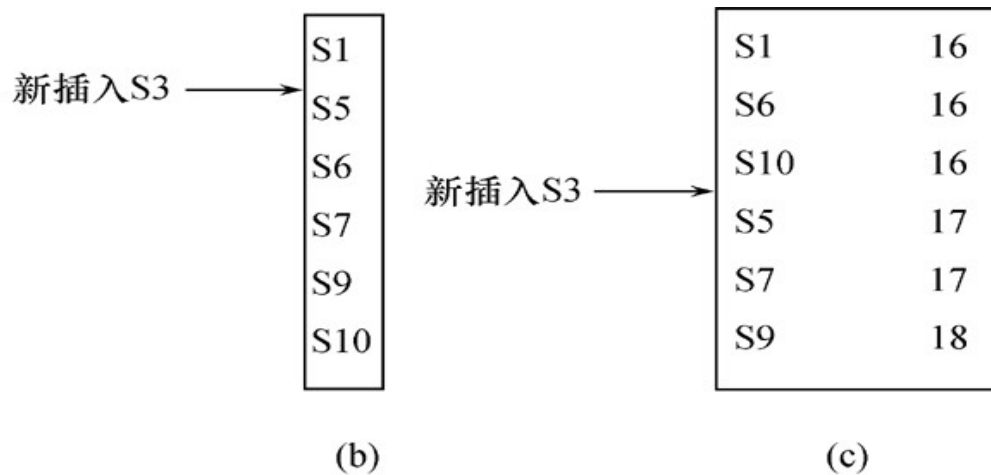
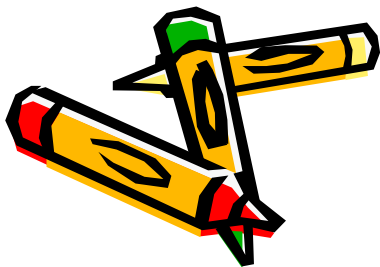
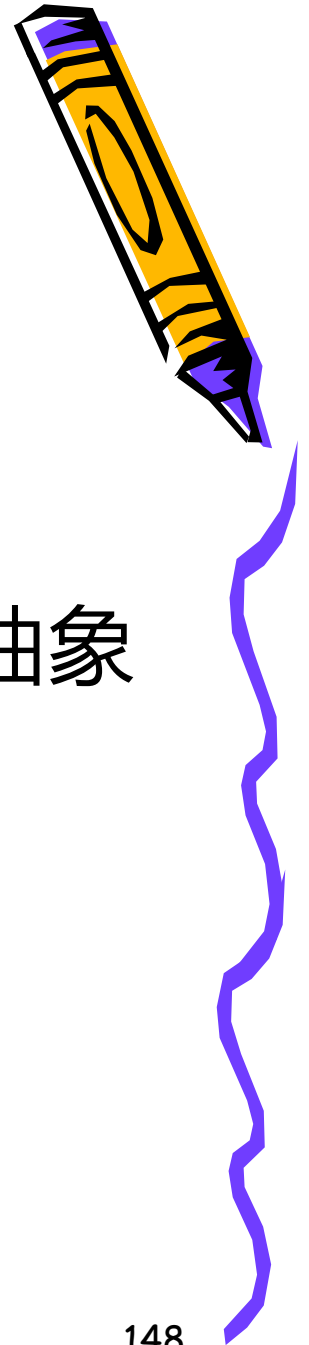


图1.29 记录不同的存储方式示意图



三级模式与二级映象

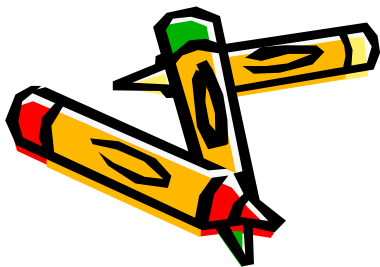
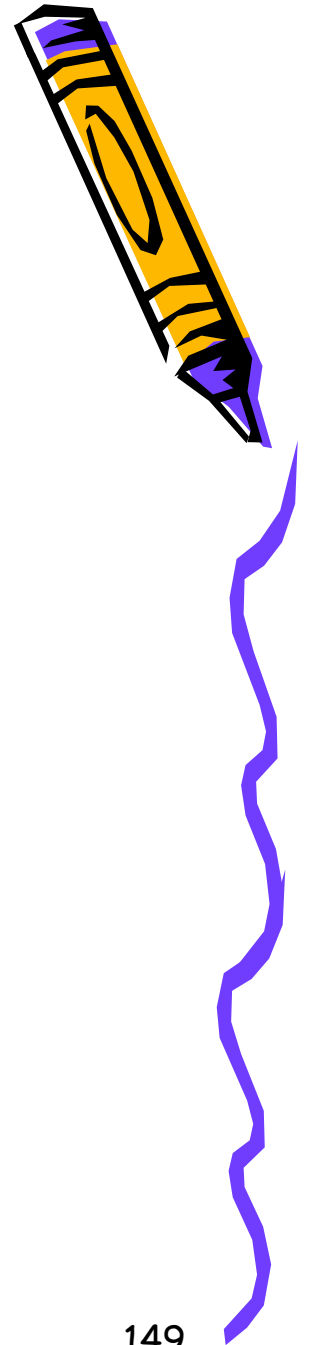
- 三级模式是对数据的三个抽象级别
- 二级映象在DBMS内部实现这三个抽象层次的联系和转换





二级映象功能

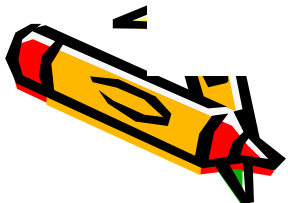
1. 外模式 / 模式映象
2. 模式 / 内模式映象





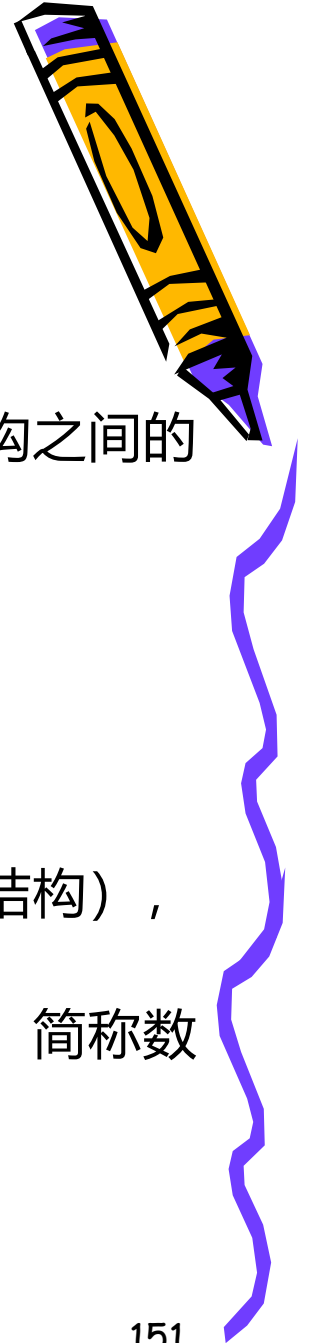
1. 外模式/模式映象

- **什么是外模式 / 模式映象**
 - 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
 - 映象定义通常包含在各自外模式的描述中
- **外模式 / 模式映象的用途：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。





2. 模式/内模式映象



- **什么是模式 / 内模式映象**
 - 模式 / 内模式映象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 / 内模式映象是唯一的
 - 该映象定义通常包含在模式描述中
- **模式/内模式映象的用途：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。





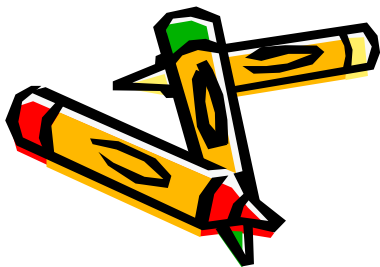
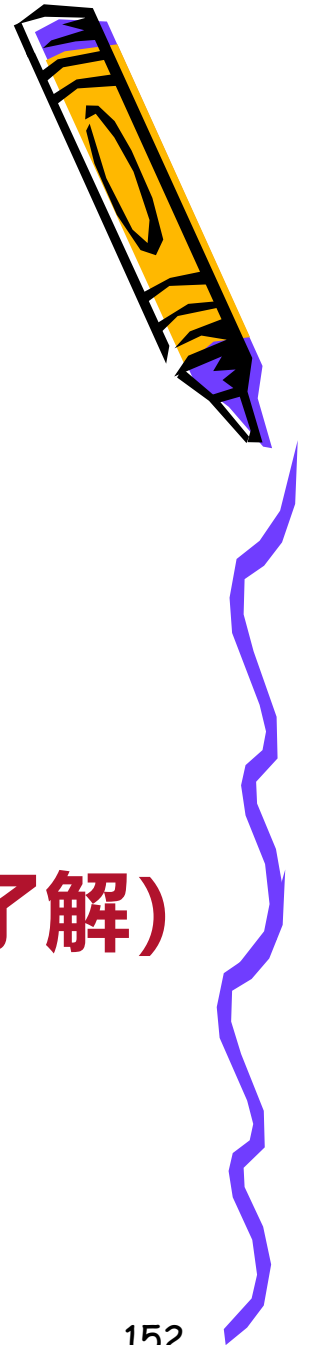
1.3 数据库系统的结构

1.3.1 数据库系统内部的模式结构

— 从数据库管理系统角度看

1.3.2 数据库系统外部的体系结构（了解）

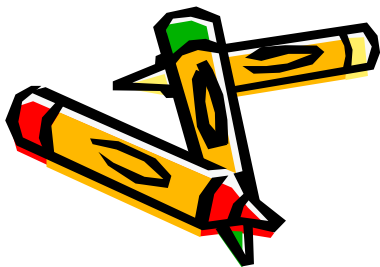
— 从数据库最终用户角度看





1.3.2 数据库系统外部的体系结构

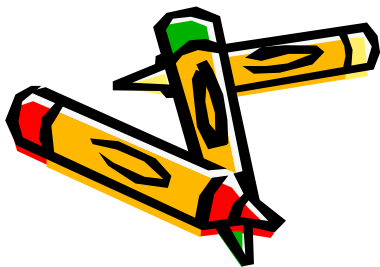
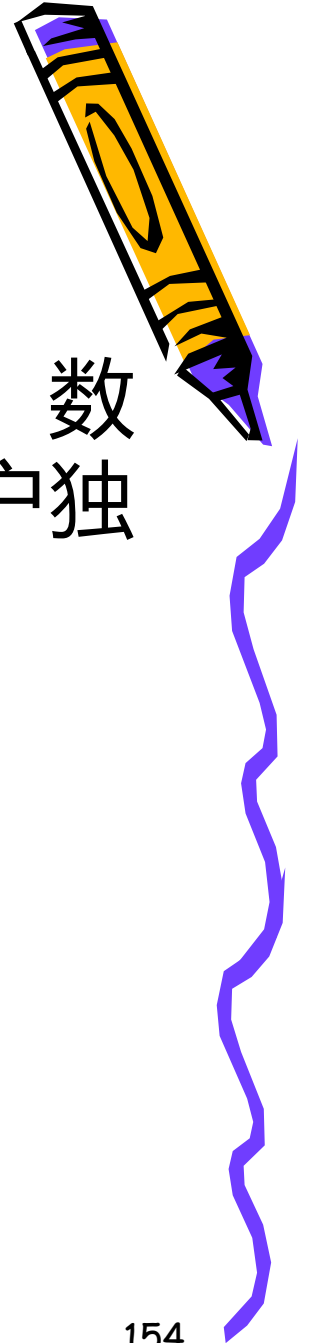
- 单用户结构
- 客户/服务器结构(主从式结构)
- 浏览器 / 应用服务器 / 数据库服务器多层结构等
- 分布式结构





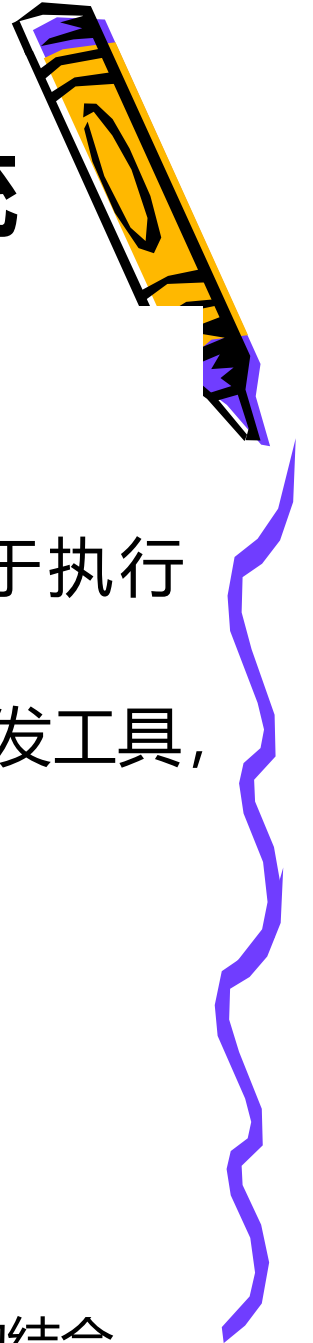
1. 单用户数据库系统

- 整个数据库系统(应用程序、DBMS、数据)都装在一台计算机上，为一个用户独占，不同机器之间不能共享数据。
- 早期的最简单的数据库系统





2. 客户/服务器结构的数据库系统



- 主从式
- 把DBMS功能和应用分开
 - 网络中某个（些）结点上的计算机专门用于执行DBMS功能，称为数据库服务器，简称服务器
 - 其他结点上的计算机安装DBMS的外围应用开发工具，支持用户的应用，称为客户机
- 客户 / 服务器数据库系统的种类
 - 集中的服务器结构
 - 在网络中仅有一台数据库服务器，而客户机是多台
 - 分布的服务器结构
 - 在网络中有多台数据库服务器
 - 分布的服务器结构是客户 / 服务器与分布式数据库的结合



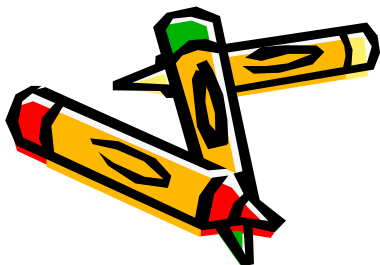


• 优点

- 客户端的用户请求被传送到数据库服务器，数据库服务器进行处理后，只将结果返回给用户，从而显著减少了数据传输量
- 数据库更加开放
 - 客户与服务器一般都能在多种不同的硬件和软件平台上运行
 - 可以使用不同厂商的数据库应用开发工具
 - 应用程序具有更强的可移植性，同时也可以减少软件维护开销

• 缺点

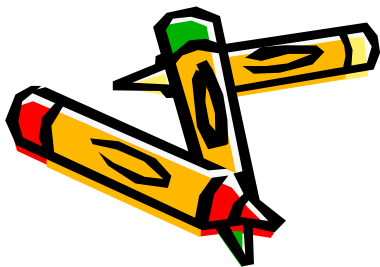
- 在集中的服务器结构中，一个数据库服务器要为众多的客户服务，往往容易成为瓶颈，制约系统的性能（与主从式结构相似）
- 在分布的服务器结构中数据分布在不同的服务器给数据的处理、管理与维护带来困难（与分布式结构相似）





3. 浏览器 / 应用服务器 / 数据库服务器多层结构

- 大型Web应用中
- Web容器装在应用服务器上，DB装在DB服务器上，以分担压力





4. 分布式结构的数据库系统



- **数据库中的数据在逻辑上是一个整体，但物理地分布在计算机网络的不同结点上。**
 - 网络中的每个结点都可以独立处理本地数据库中的数据，执行局部应用
 - 同时也可以同时存取和处理多个异地数据库中的数据，执行全局应用
- **优点**
 - 适应了地理上分散的公司、团体和组织对于数据库应用的需求。
- **缺点**
 - 数据的分布存放给数据的处理、管理与维护带来困难。
 - 当用户需要经常访问远程数据时，系统效率会明显地受到网络交通的制约。





第一章 绪论

1.1 引言

1.2 数据模型

1.3 数据库系统的结构

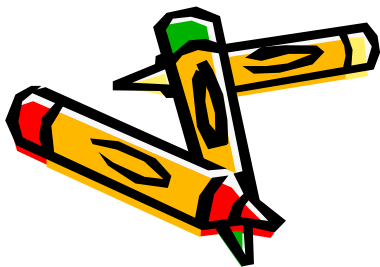
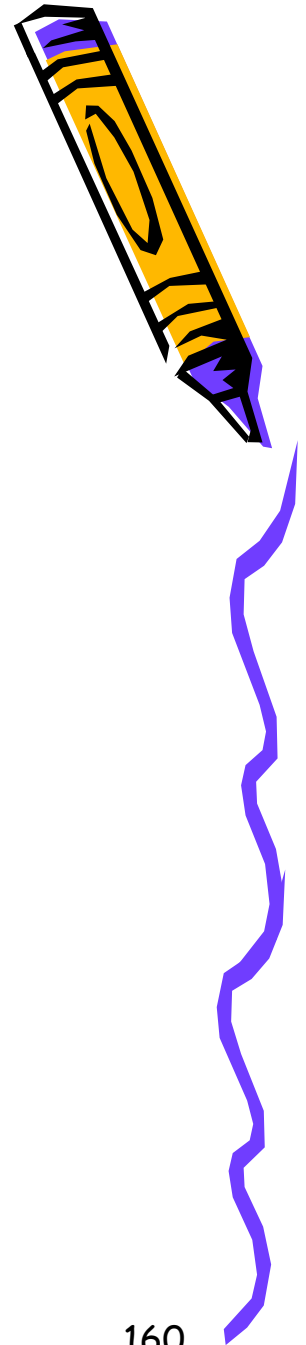
1.4 数据库系统的组成(自学)





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

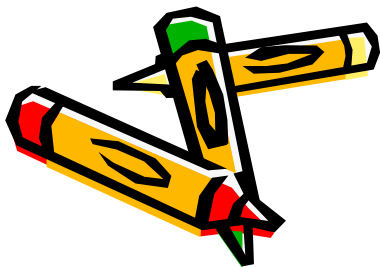
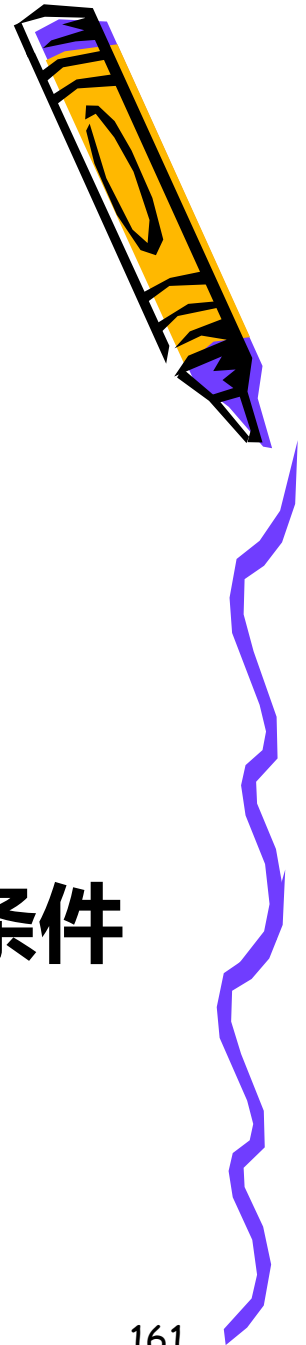




数据库管理员(DBA)

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**





(4) 监控数据库的使用和运行

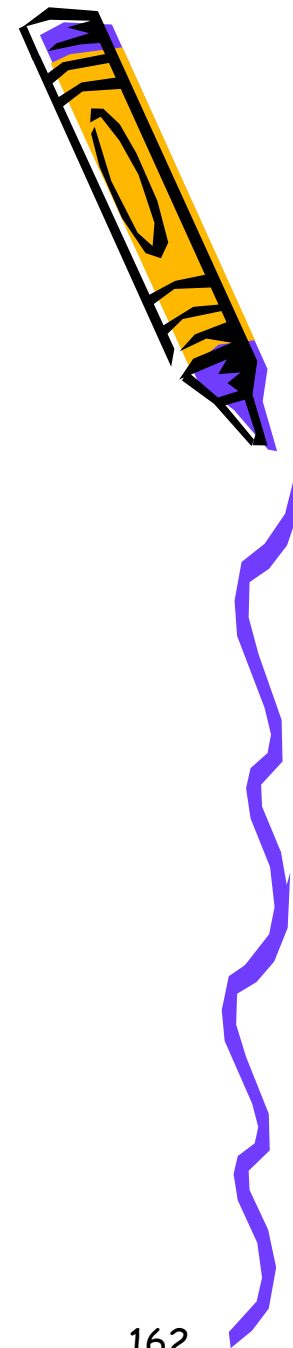
— 周期性转储数据库

- 数据文件
- 日志文件

— 系统故障恢复

— 介质故障恢复

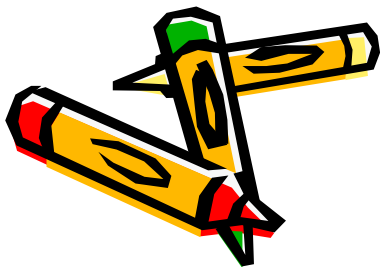
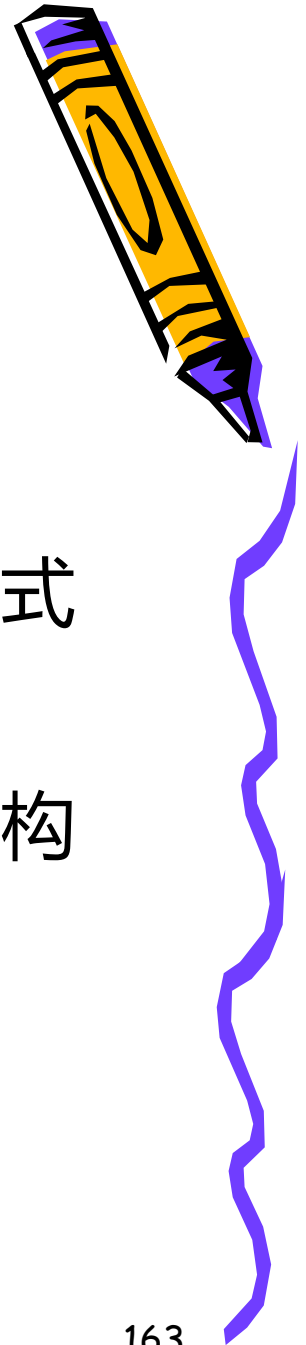
— 监视审计文件





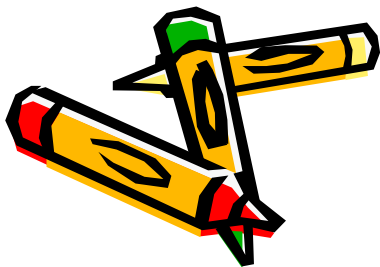
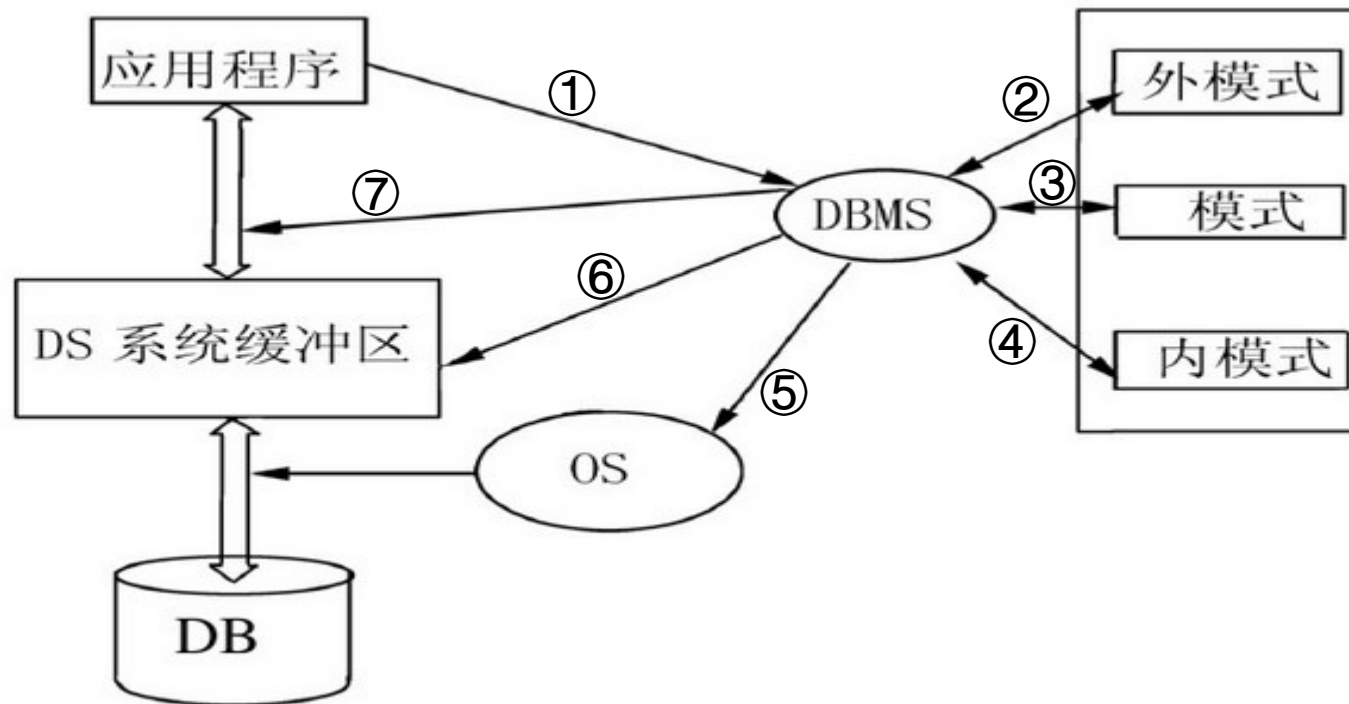
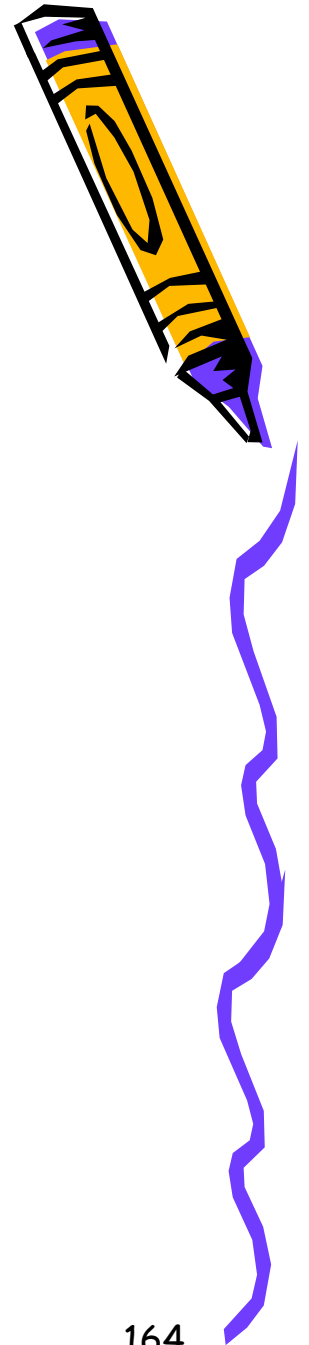
(5) 数据库的改进和重组

- 性能监控和调优
- 定期对数据库进行重组织（数据库模式不变），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（数据库模式简单变更）





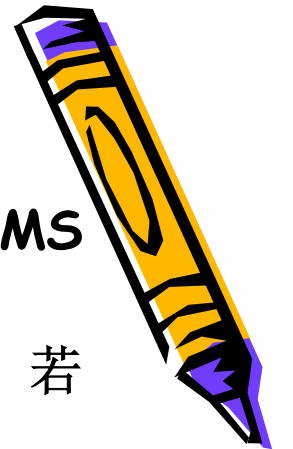
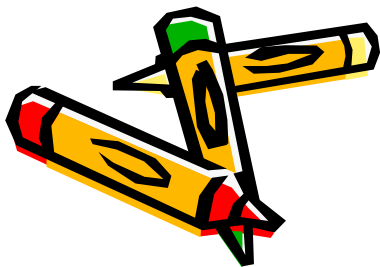
例：以查询为例，概述用户访问数据库的主要步骤





- ① 当执行应用程序中的一条查询数据库的记录时，就会向**DBMS**发出读取相应记录的命令，并指明外模式。
- ② **DBMS**接到命令后，调出所需的外模式，并进行权限检查：若合法，则继续执行，否则向应用程序返回错误信息。
- ③ **DBMS**访问模式，并根据外模式/模式映像，确定所需数据在模式上的有关信息（逻辑记录型）。
- ④ **DBMS**访问内模式，并根据模式/内模式映像，确定所需数据在内模式上的有关信息（读取的物理记录及存取方面）。
- ⑤ **DBMS**向操作系统发出读相应数据的请求（读取记录）。
- ⑥ 操作系统执行读命令，将有关数据从外存调入系统缓冲区上。
- ⑦ **DBMS**把数据按外模式的形式送入用户工作区，返回正常执行的信息。

由此可见，**DBMS**是数据库系统的核心，且与操作系统有关。





作业

