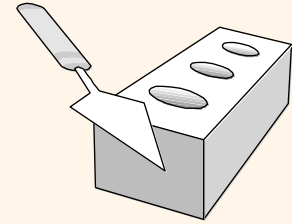


# 数据管理系统实现

## Chapter 1

王晓玲



## 教材

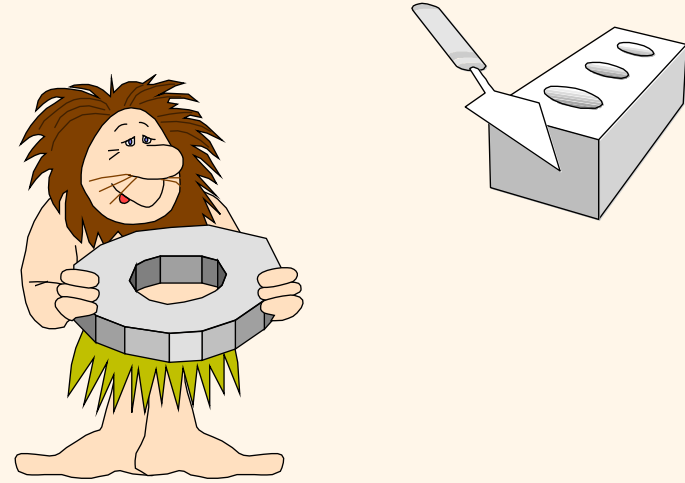
### 《数据库系统实现》

### Database System Implementation

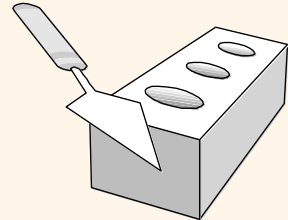
## 参考书

- An Introduction to Database Systems (Seventh Edition)  
C.J.Date, Addison-Wesley
- A First Course in Database Systems (Second Edition)  
Jeffrey D. Ullman and Jennifer Widom, Prentice Hall
- 数据库系统教程 王能斌 电子工业出版社

# What Is a DBMS?



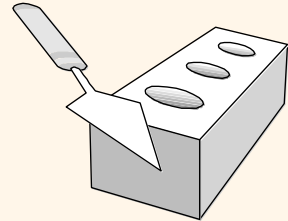
- ❖ 大量的数据.
- ❖ 现实世界的模型
  - 实体 (e.g., 学生, 课程)
  - 关系 (e.g., 张三学修了数据库这门课)
- ❖ Database Management System (DBMS) 数据库管理系统是一个为存储和管理数据库的软件包



# 数据库技术的发展历史

## ❖ (1) 从数据模型的发展来看

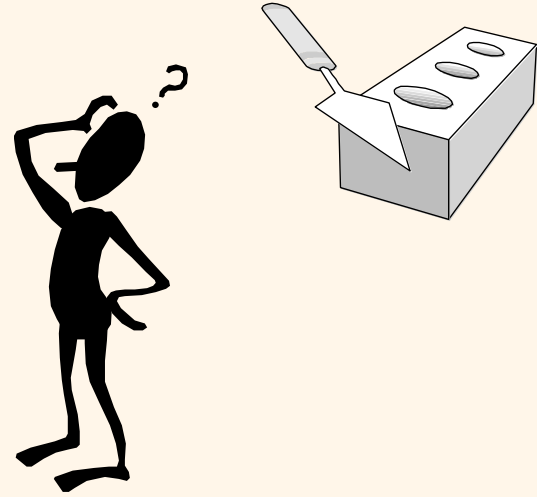
- 无管理(60年代之前): 科学计算
- 文件系统: 简单的数据管理
- 数据管理需求不断增长, 数据库管理系统应运而生。
  - 1964, 美通用电气公司开发出第一个DBMS: IDS, 网状
  - 1969, IBM推出第一个商品化DBMS, 层次
  - 1970, IBM研究员E.F.Codd提出关系模型
  - 其它数据模型: 面向对象、演绎、XML等



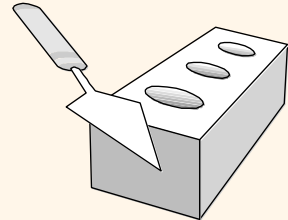
# 文件系统 *vs.* DBMS

- ❖ 应用程序负责数据在内存和二级存储设备之间的数据交换 (e.g., 缓存, 基于页面的存取, etc.)
- ❖ 不同的查询编写不同的代码
- ❖ 应用保证数据的一致性, 特别是多用户的应用环境
- ❖ 恢复机制
- ❖ 安全和存取控制

# 为什么使用DBMS?

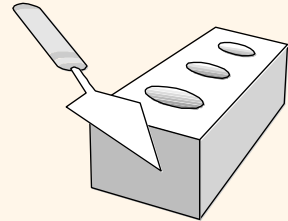


- ❖ 数据的独立性和有效的存取.
- ❖ 减少应用的开发时间.
- ❖ 数据集成和安全.
- ❖ 统一的数据管理.
- ❖ 并发存取和恢复机制.



# 数据库的分类

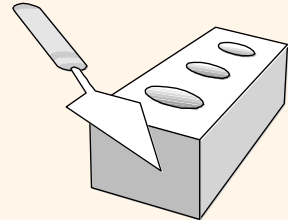
- ❖ (2) 从体系结构的发展来看
  - 集中式：主机+哑终端
  - 分布式数据库
  - Client/Server结构
  - 三层/多层结构
- ❖ (3) 从应用领域的拓展来看
  - OLTP
  - 工程数据库
  - 演绎数据库
  - 多媒体数据库
  - 时态数据库
  - 空间数据库
  - 数据仓库、数据挖掘



- 高级数据库技术

- 分布式数据库
- 多媒体数据库
- 面向对象数据库系统
- WEB数据库
- XML及其在数据管理中的作用
- 数据仓库以及OLAP
- 数据挖掘
- 数据流系统
- ...

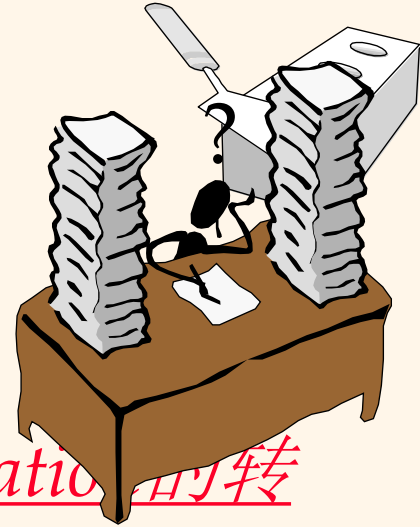




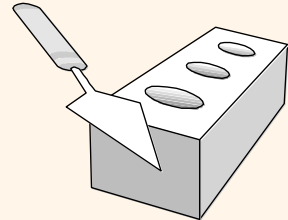
# 数据库研究的趋势

- 数据的来源越来越多，数据量不断增长
  - 大数据
- 提供了统计分析、决策支持等功能
  - OLAP (Online Analytical Processing)和OLTP (Online Transaction Processing)
  - 数据挖掘(Data Mining)
  - 机器学习 (Machine Learning)
- ...

# 为什么学习数据库??

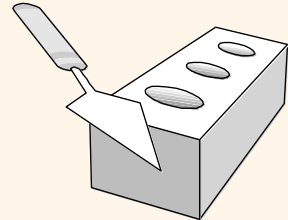


- ❖ 从计算 computation 到信息管理 informatics 的转 变
  - 一般的用户：在大量数据的空间寻找信息
  - 高端应用：科学的应用需求
- ❖ 数据量的增长
  - 数字图书馆，视频点播，基因工程
  - ...需要探讨 DBMS 的功能，满足更多的需求
- ❖ DBMS 包括了计算机领域的很多方面
  - 操作系统 OS, 计算理论, 人工智能 “AI”, 多媒体等
- ❖ 新型应用带给数据库技术哪些新挑战？



# 数据模型

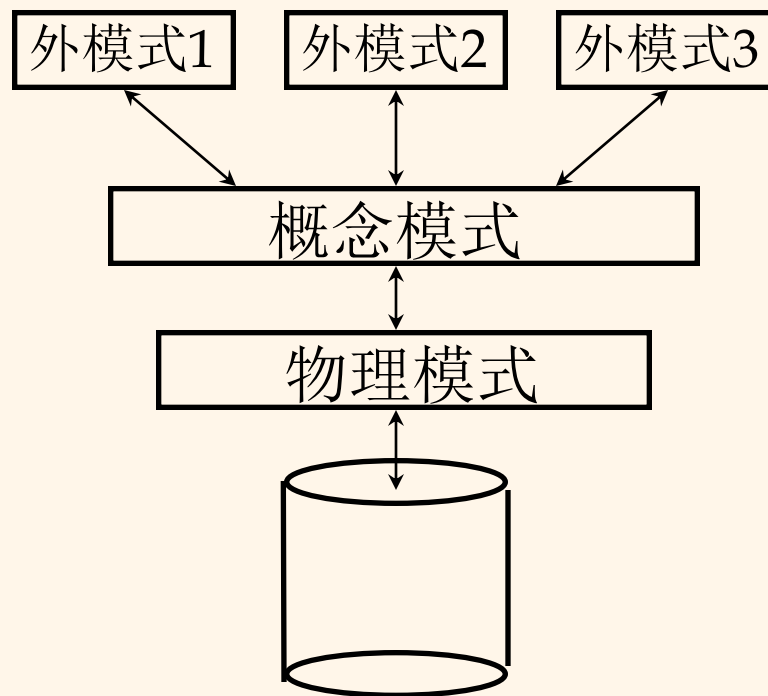
- ❖ 数据模型是描述数据的一组概念和定义。
- ❖ 数据模式是基于数据模型的数据描述。
- ❖ 关系数据模型是目前用得最多的数据模型。
  - 基本概念：关系，简要地说就是存在行和列的表。
  - 每个关系有一个模式 schema, 描述表中的列和相关的信息



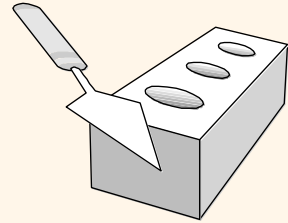
# 多级抽象

❖ 多个外模式 views, 一个概念（逻辑）模式 conceptual (logical) schema, 一个物理模式 physical schema.

- 视图描述用户从不同的角度看这些数据.
- 概念模式定义了数据的逻辑结构
- 物理模式描述了数据的存储细节, 例如索引等.



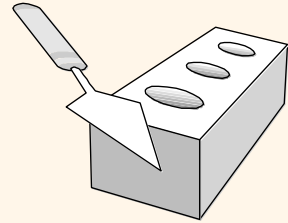
\* 模式通过DDL定义; 数据通过DML进行查询和修改.



# Example: 大学数据库



- *Students*(*sid*: string, *name*: string, *login*: string, *age*: integer, *gpa*: real)
- *Courses*(*cid*: string, *cname*: string, *credits*: integer)
- *Enrolled*(*sid*: string, *cid*: string, *grade*: string)



# Example: 大学数据库

## ❖ 概念模式:

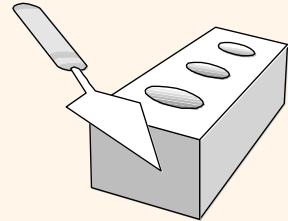
- *Students*(*sid: string, name: string, login: string, age: integer, gpa: real*)
- *Courses*(*cid: string, cname: string, credits: integer*)
- *Enrolled*(*sid: string, cid: string, grade: string*)

## ❖ 物理模式:

- 管理按照文件格式存放.
- 每个学生的第一列上有索引.

## ❖ 外模式 (视图View):

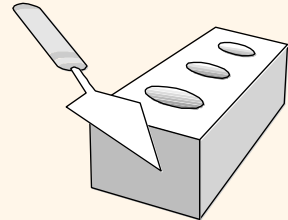
- *Course\_info*(*cid: string, enrollment: integer*)



# 数据的独立性\*

- ❖ 应用独立于数据的存储和数据的结构.
- ❖ 数据的逻辑独立性: 保护用户免受数据逻辑结构变化的影响.
- ❖ 数据的物理独立性: 保护用户免受物理存储变化的影响.

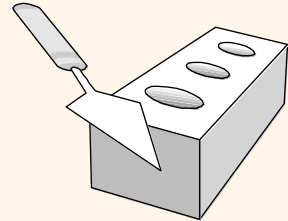
\* 使用 DBMS 的最大好处!



# 并发控制

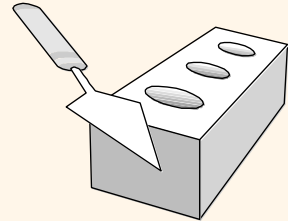
- ❖ 并发执行用户的程序
  - 磁盘存取是经常的，相对来说，比CPU速度慢，所以，可以运行多个程序，保持CPU 高的使用率.
- ❖ 需要消除不同程序之间会产生的不一致.
- ❖ DBMS确保：用户免受系统故障的影响。





# 事务: *DB* 程序的一次执行

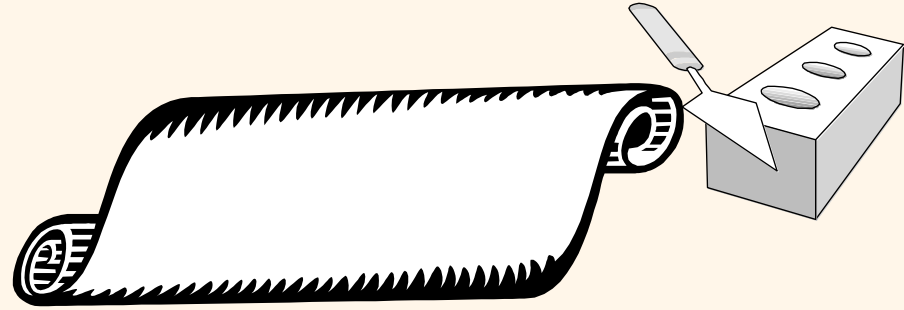
- ❖ 事务是数据库动作的基本单位。例如 (reads/writes).
- ❖ 每个事务必须完全执行, 并且保证数据库的一致性.



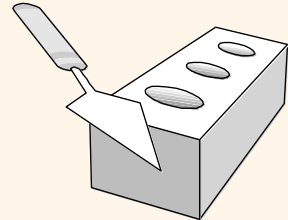
# 并发事务的调度

- ❖ DBMS 确保对事物  $\{T_1, \dots, T_n\}$  的执行等价于按照某种顺序的其他方式执行  $T_1' \dots T_n'$ .
  - 在读写一个数据对象的时候，事务需要对数据对象加锁，直到事务完成，才被解锁 (严格的两阶段加锁协议)
  - 解决方法: 如果一个动作  $T_i$  (写  $X$ ) 影响动作  $T_j$  (要读  $X$ ), 其中的一个，比如  $T_i$ , 将获得  $X$  上的锁， $T_j$  就必须等待，直到事务  $T_i$  完成;
  - 如果  $T_j$  已经对  $Y$  加锁， $T_i$  又请求  $Y$  上的锁，会发生什么情况?
  - (Deadlock!死锁)  $T_i$  或  $T_j$  必须撤销，重新开始!

# 日志



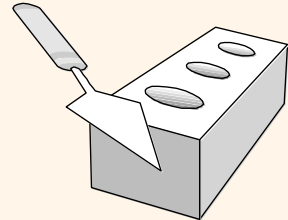
- ❖ 日志中记录的动作:
  - *Ti* 写一个数据对象: 原值和新值.
    - 日志必须在改变数据之前, 把信息记录在磁盘上!
  - *Ti* 提交commits/取消aborts: 日志记录显示了这个动作
- ❖ 日志也记录了Xact标志 id, 所以, 很容易对特定的Xact重做.
- ❖ 日志经常是归档在稳定的存储设备上.
- ❖ 所有的与日志相关的活动被DBMS透明地处理.



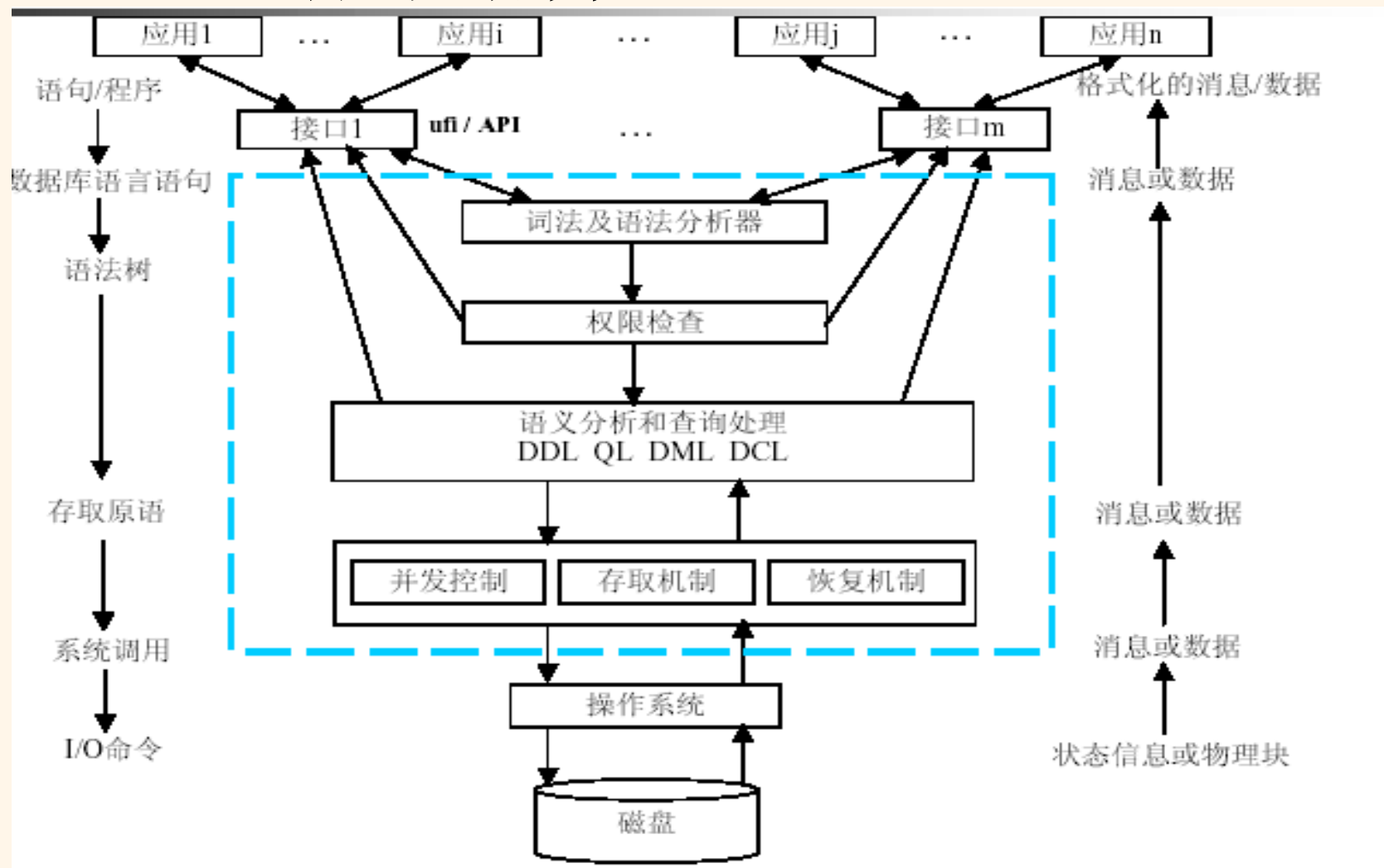
# 数据库的使用者

- ❖ 终端用户
- ❖ 应用开发者
- ❖ 数据库管理员 (DBA)
  - 设计逻辑/物理模式
  - 处理安全和授权问题
  - 数据的可用性，恢复机制
  - 数据库的调整

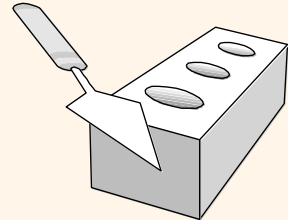
必须理解DBMS工作原理!



# DBMS体系结构



# 总结



- ❖ DBMS 用来管理、查询大量数据.
- ❖ 好处:
  - 恢复机制;
  - 并发存取;
  - 快速的应用开发;
  - 数据集成和安全.
  - 数据独立性.
- ❖ 了解DBMS原理, 是进行数据管理技术研究的基础和关键。
- ❖ DBMS的研发是计算机领域最有应用前景的工作.

