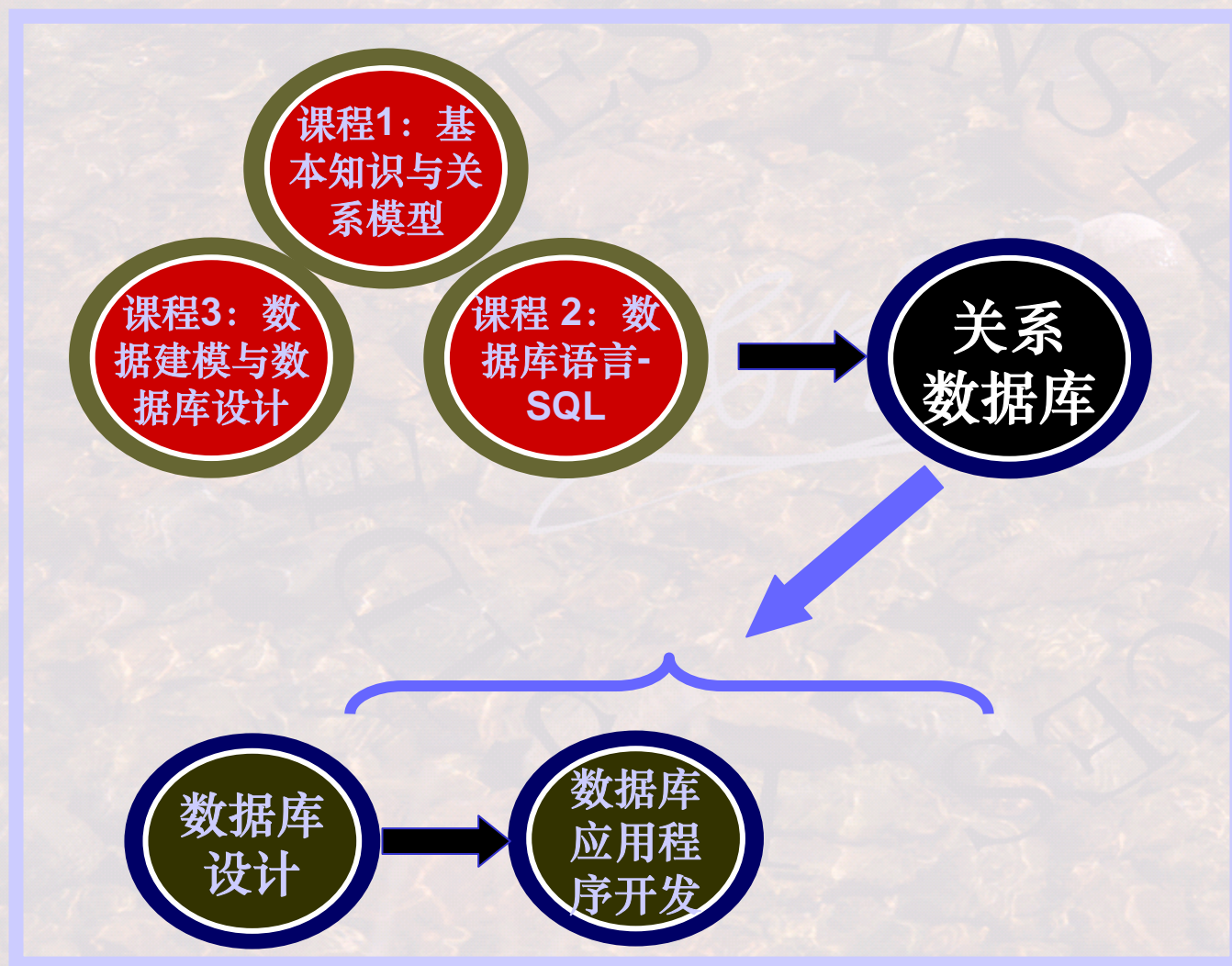


# 数据库系统之四

## --数据库管理系统实现技术









### 一、基本知识与关系模型

第1讲 初步认识数据库系统

第2讲 数据库系统的结构抽象与演变

第3讲 关系模型之基本概念

第4讲 关系模型之关系代数--操纵数据之集合思维训练

第5讲 关系模型之关系演算--操纵数据之逻辑思维训练

### 二、数据库语言-SQL

第6讲 概览SQL语言

第7讲 SQL语言之查询与视图

第8讲 SQL语言与数据库完整性和安全性

第9讲 嵌入式SQL语言之基本技巧

第10讲 嵌入式SQL语言之动态SQL

数据库系  
统(上)

数据库系  
统(中)

### 三、数据建模与数据库设计

第11讲 数据建模：思想与方法

第12讲 数据建模：工程化方法及案例分析

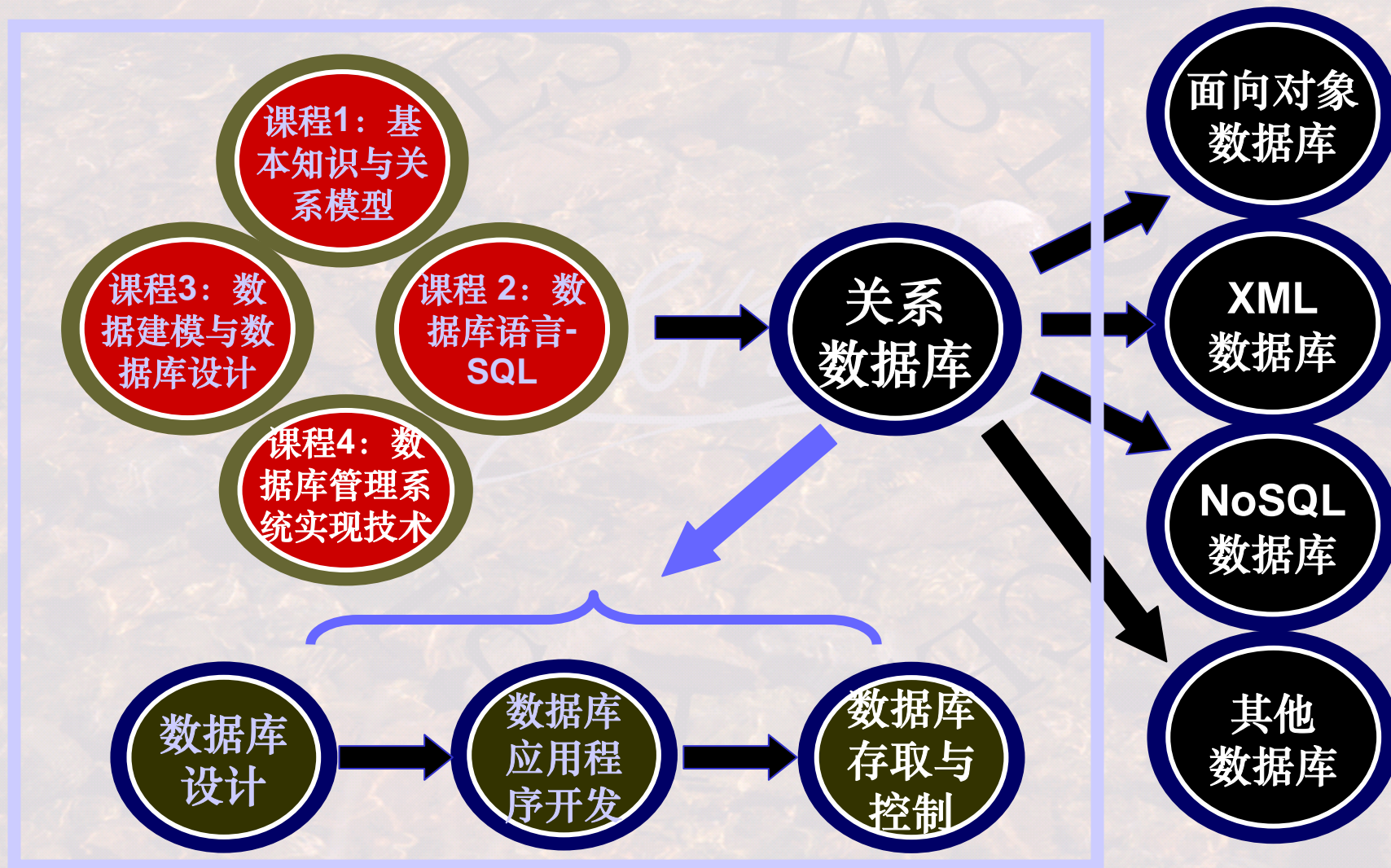
第13讲 数据库设计过程

第14讲 数据库设计理论之数据依赖

第15讲 数据库设计理论之模式分解

第16讲 数据库设计理论之关系范式

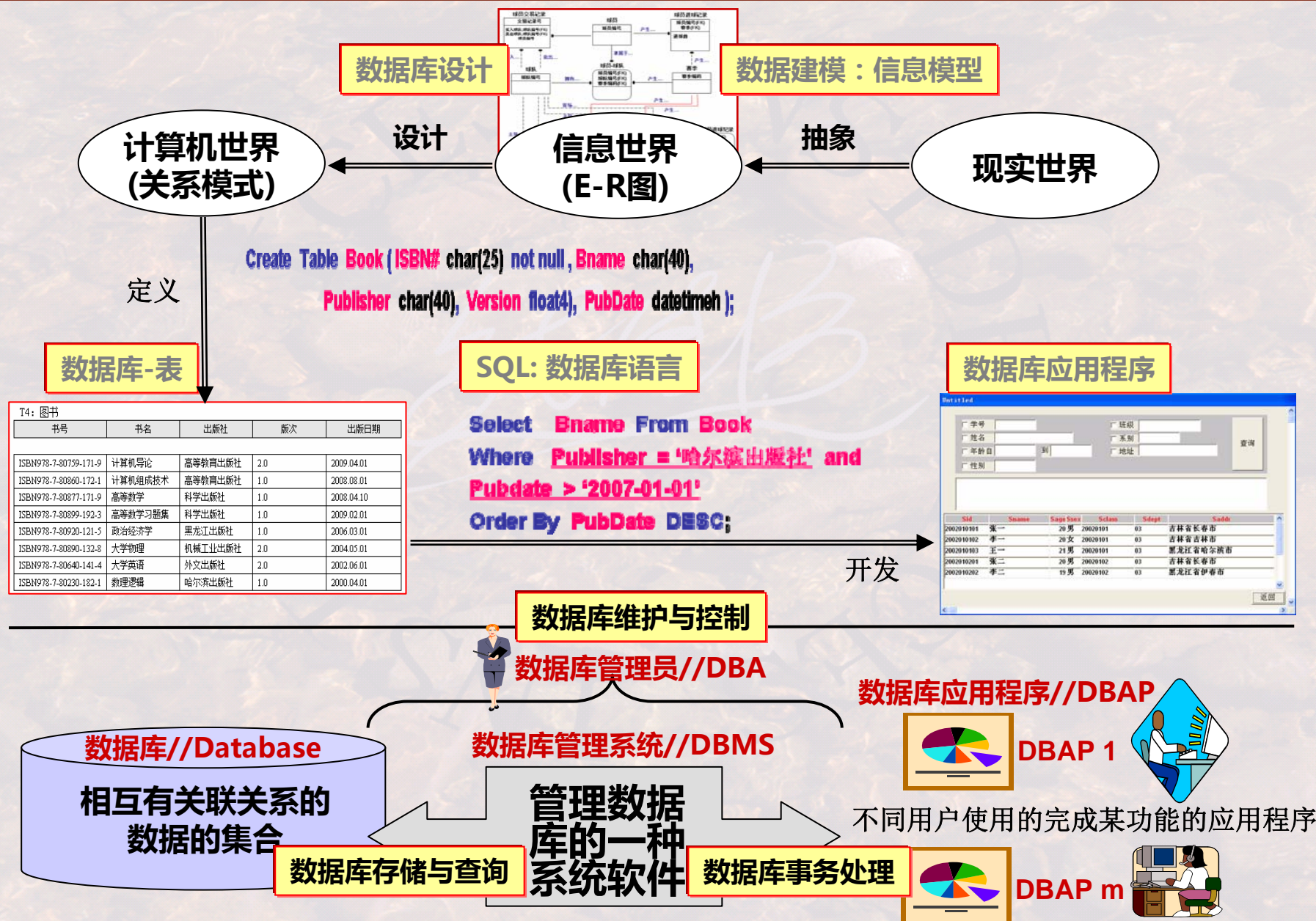






# 数据库管理系统实现技术

## (2)在《数据库系统》课程中的位置





## •教材

–**Database System Concepts**, 6th Edition, 机械工业出版社, 2012

--**Database System Implementation**, 2nd Edition, 机械工业出版社, 2010

## •课程特点与教材阅读要求

- **抽象**----概念和原理是抽象的, 要通过具体的实例加以理解
- **思维**----数据库的思维(集合、逻辑与对象思维)
- **应用**----结构化/面向对象程序设计语言的补充 → 数据库语言
- **管理**----数据存储与数据查询: **算法与效率**
- **案例**----通过案例理解, 听案例讲解, 自己做案例



# 第17讲 数据库物理存储

战德臣

哈尔滨工业大学 教授·博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent  
**C**omputing for **E**nterprises & **S**ervices,  
**H**arbin **I**nstitute of **T**echnology



# 本讲学习什么？



## 基本内容

1. 基础回顾-计算机系统的存储体系
2. 磁盘的结构与特性
3. DBMS数据存储与查询实现的基本思想
4. 数据库之表和记录与磁盘块的映射
5. 数据库之文件组织方法？

## 重点与难点

- 理解利用磁盘组织大规模数据的基本思维
- 初步了解数据存储与查询实现的基本思想
- 理解三种文件组织方法及其特性：堆文件、顺序文件和散列文件
- 理解数据库重组的概念和作用



# 基础回顾-计算机系统的存储体系

战德臣

哈尔滨工业大学 教授·博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent  
**C**omputing for **E**nterprises & **S**ervices,  
**H**arbin **I**nstitute of **T**echnology



# 基础回顾-计算机系统的存储体系

## (1)数据库的存储与检索问题

数据

数据

数据库

bit & Byte, 1KB  
1MB(Megabyte) =  $2^{10}$ KB  
1GB(Gigabyte) =  $2^{10}$ MB  
1TB(Trillionbyte) =  $2^{20}$ MB

大数据

1TB(Trillionbyte) =  $2^{20}$ MB  
1PB(Petabyte) =  $2^{30}$ MB  
1EB(Exabyte) =  $2^{40}$ MB  
1ZB(Zettabyte) =  $2^{50}$ MB  
1YB(Yottabyte) =  $2^{60}$ MB  
1BB(Brontobyte) =  $2^{70}$ MB

性能评估  
与改进

既是DBA的职责—  
借助于软件管理与  
维护，又是计算机  
科学家需要研究和  
解决的问题

算法设计  
与实现

### 两个基本问题如何解决？

- 如何高效率的存储？--数据组织与索引
  - 如何快速的检索？--查询实现与查询优化
- 面向大规模用户, 又如何解决？



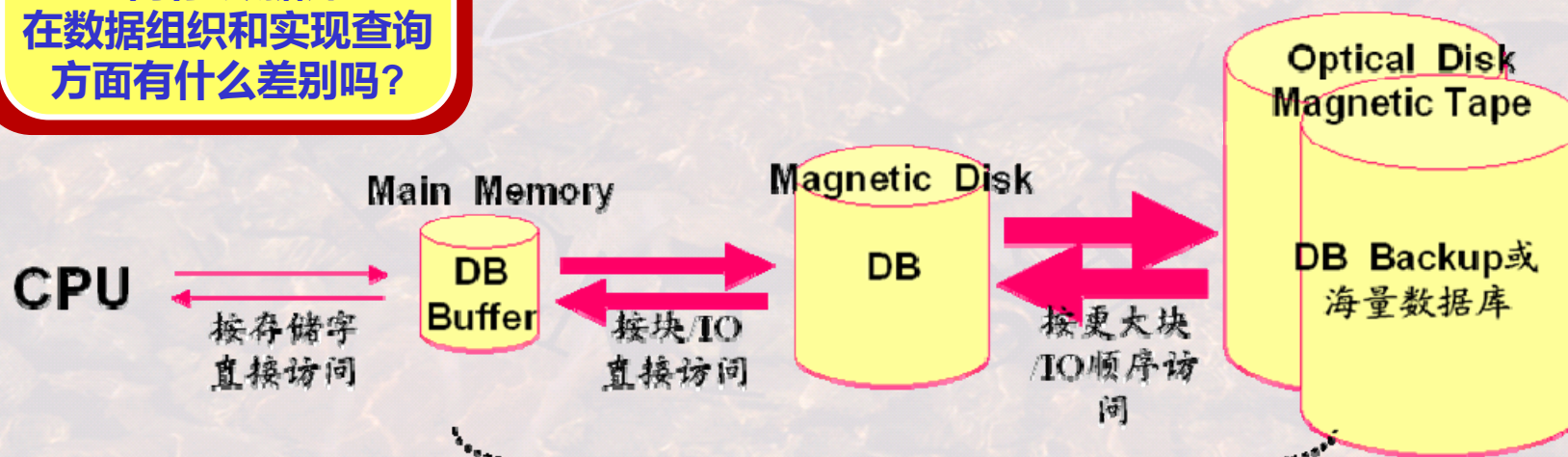
# 基础回顾-计算机系统的存储体系

## (2)什么是存储体系

### ➤ 数据组织的基础--存储体系

- 将不同性价比的存储器组织在一起，满足高速度、大容量、低价格 需求
- CPU与内存直接交换信息，按存储单元(存储字)进行访问
- 外存按存储块进行访问，其信息需先装入内存，才能被CPU处理

• 磁带-数据库  
• 磁盘-数据库  
• 内存-数据库  
在数据组织和实现查询  
方面有什么差别吗？



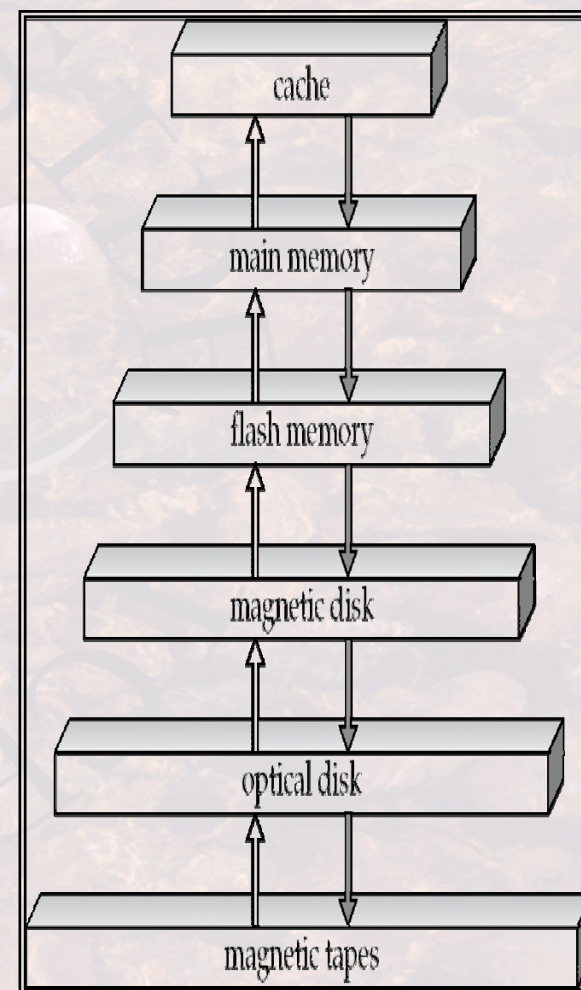
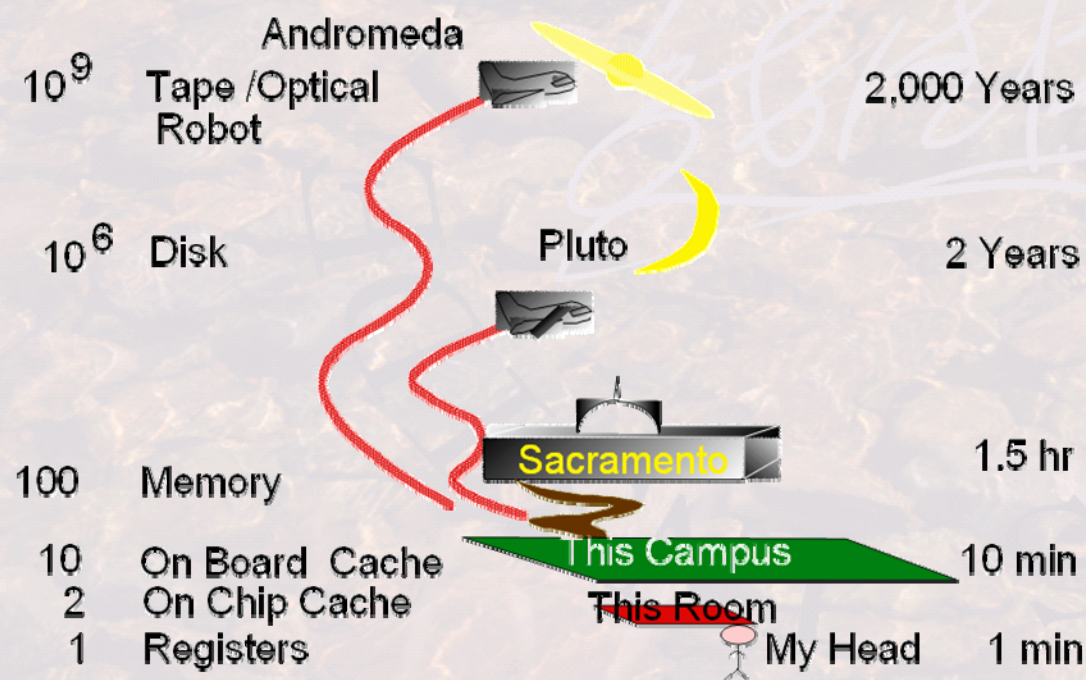
DBMS及操作系统管理着存储体系  
实现数据库的透明访问



# 基础回顾-计算机系统的存储体系

## (3)不同层次存储的访问时间上的差异

### Jim Gray's Storage Latency Analogy: How Far Away is the Data?



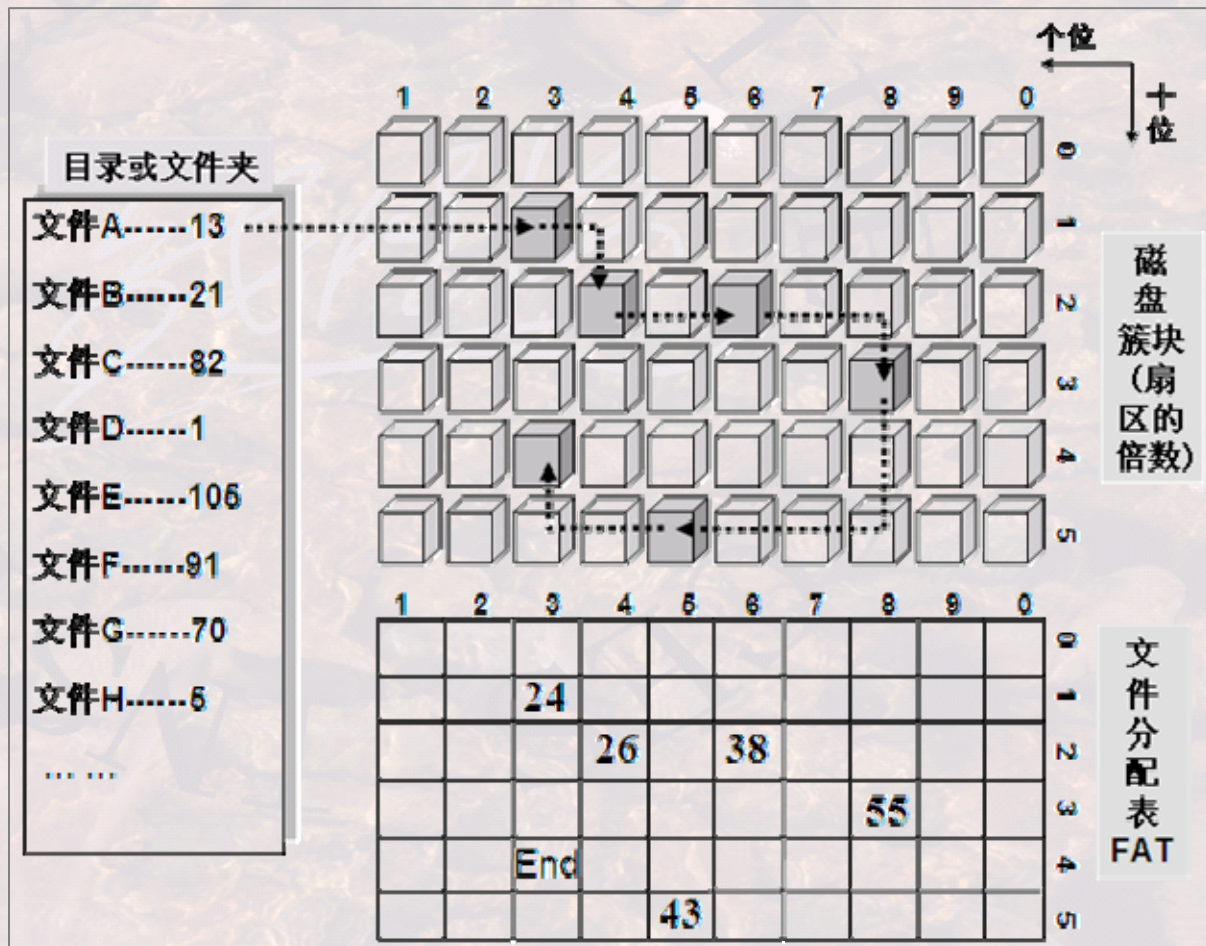
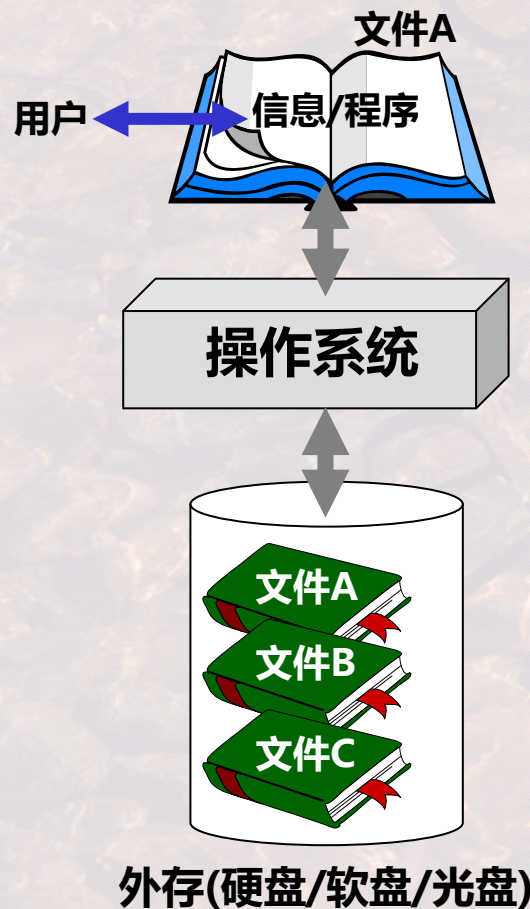


# 基础回顾-计算机系统的存储体系

## (4)操作系统如何管理磁盘和数据

### 操作系统对数据的组织：FAT-目录(文件夹)-磁盘块/簇

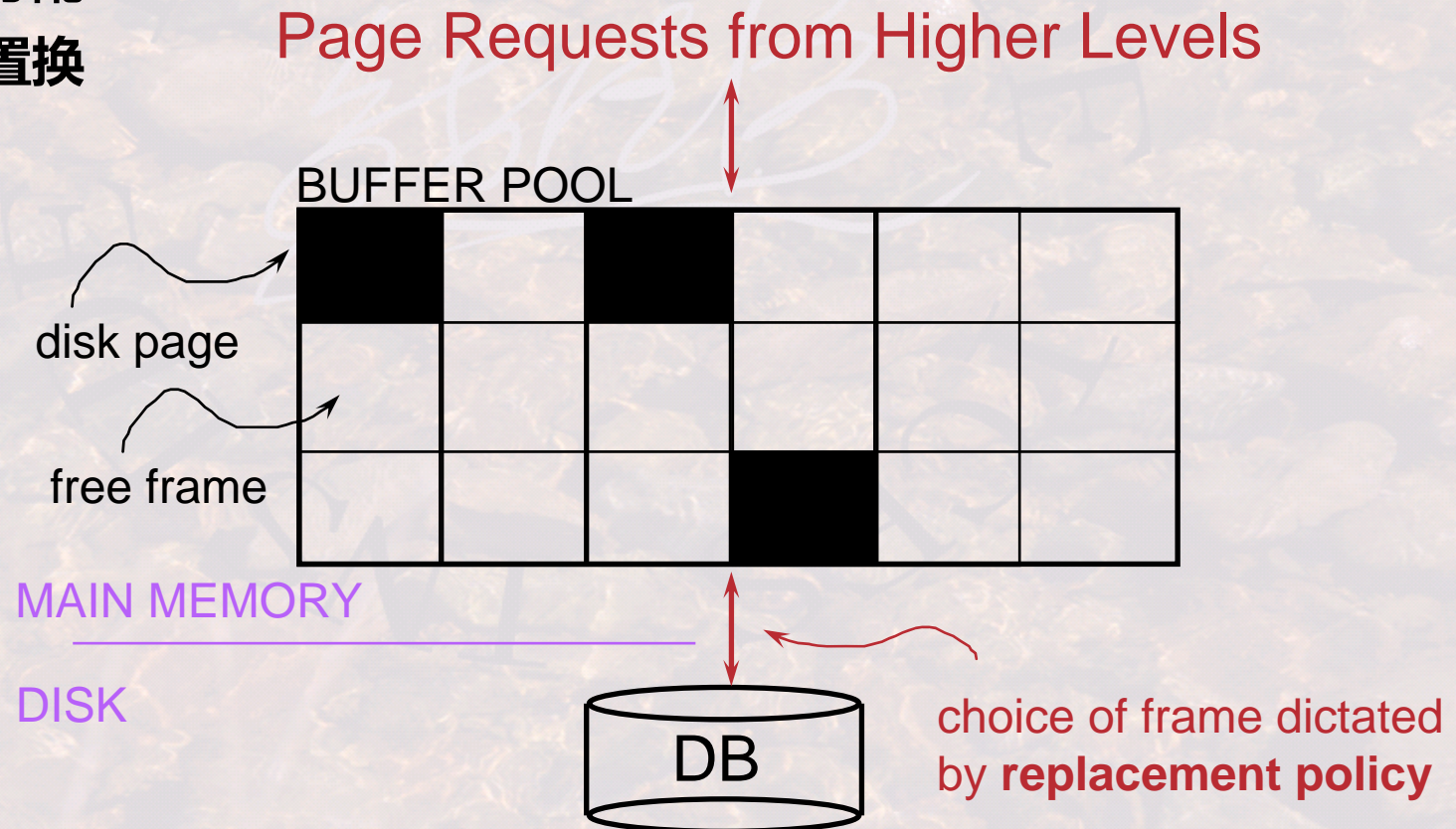
#### ➤FAT(文件分配表-File Allocation Table)





## 内存管理

- 一条记录的地址 = 存储单元的地址 = 内存地址 = 页面 : 页内偏移量
- 页面(Page) = 块(Block)
- 内存页面的分配
- 内存页面的置换





# 磁盘的结构与特性

战德臣

哈尔滨工业大学 教授·博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent  
Computing for Enterprises & Services,  
Harbin Institute of Technology



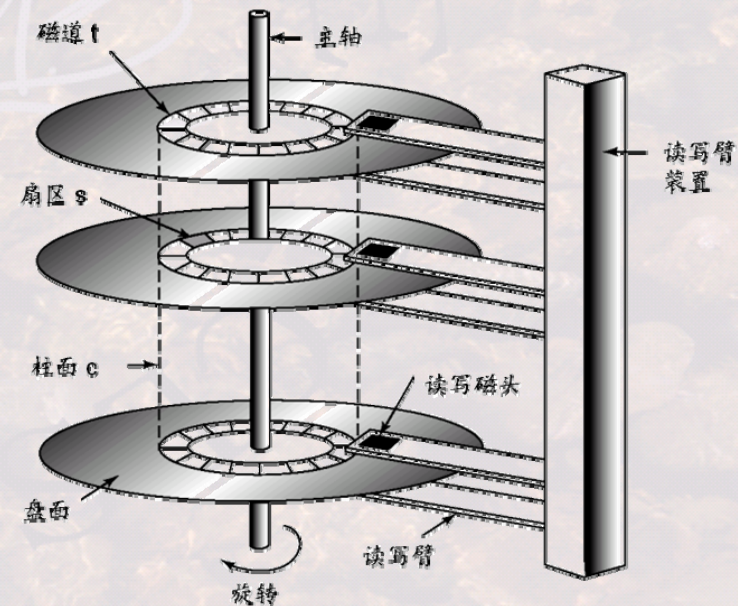
### 盘面: 磁道: 扇区

➤ 磁盘读写单位: Sector → 簇Cluster/块Block: 连续的若干个扇区

**示例:** 一个磁盘的基本信息

- 8个圆盘, 16个盘面
- 每个盘面有 $2^{16}$ 或65536个磁道
- 每个磁道(平均)有 $2^8=256$ 个扇区
- 每个扇区有 $2^{12}=4096$ 个字节

**磁盘的容量** =  $2^4 * 2^{16} * 2^8 * 2^{12} = 2^{40}$  字节





### 磁盘数据读写时间

- 寻道时间(约在1-20ms)
- 旋转时间(约0-10ms)
- 传输时间(每4KB页<1ms)

#### 示例：一个磁盘的基本信息

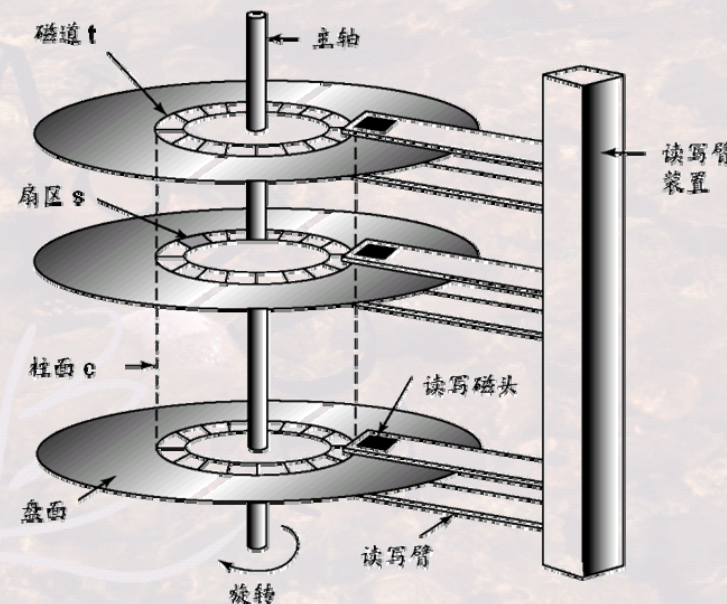
- 磁盘以7200转/min旋转，则：8.33ms内旋转一周
- 柱面之间移动磁头组合从启动到停止花费1ms,每移动4000个柱面另加1ms. 即磁头在0.00025ms内移动一个磁道，则：从最内圈移动到最外圈，移动65536个磁道大约用17.38ms.
- 一个磁道中扇区间的空隙大约占10%的空间
- 一个磁盘块 = 4个扇区 = 16384个字节

**读一个磁盘块(=4个扇区16384个字节)：**

**最小时间**=传输时间大约是0.13ms

**最长时间** = 寻道时间+旋转时间+传输时间  
= 17.38+8.33+0.13=25.84ms

**平均时间** = 6.36+4.17+0.13=10.76ms



物理存取算法考虑的关键:

- 降低I/O次数
- 降低排队等待时间
- 降低寻道/旋转延迟时间：
  - 同一磁道连续块存储；
  - 同一柱面不同磁道并行块存储；
  - 多个磁盘并行块存储



## RAID技术: Redundant Array of Independent Disk

- 并行处理**: 并行读取多个磁盘

- 可靠性**: 奇偶校验与纠错

(a) **块级拆分但无冗余**

(b) **镜像处理**: 每一个磁盘有一个镜像磁盘C

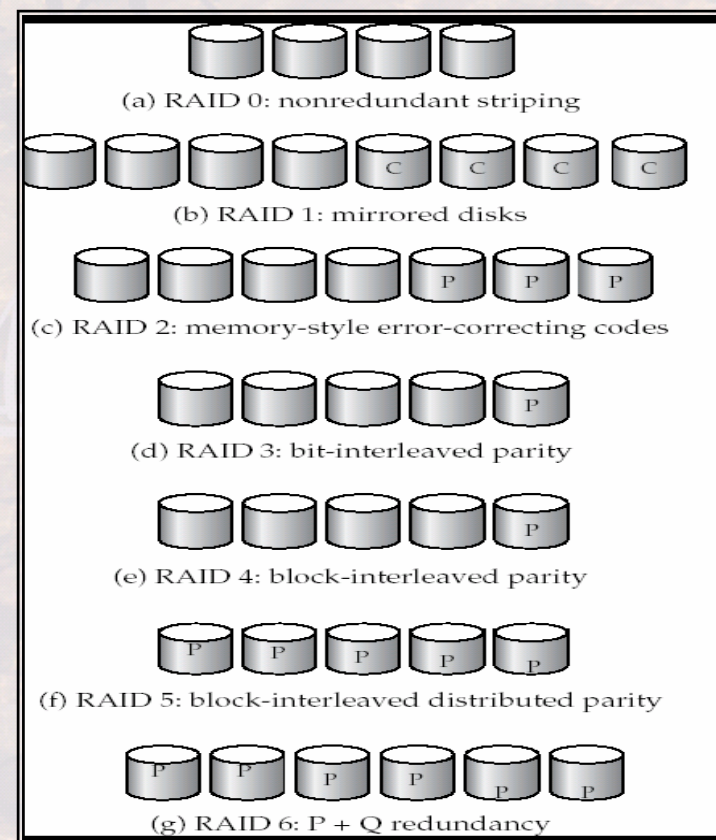
(c) **位交叉纠错处理**: 4个磁盘存储4位+3个校验盘P存储3校验位

(d) **位交叉校验**: 4个磁盘存储4位+1个校验盘存储1校验位, 位拆分存储(借助于扇区读写校验判断出错磁盘, 再依据校验盘进行纠错)

(e) **块交叉校验**: 块拆分存储, 其他同。

(f) **块交叉分布式校验**: 块拆分存储, 互为校验盘

(g) **更为复杂的冗余处理**(略)



- 比特级拆分**: 一个字节被拆分成8个比特位, 不同比特位存储于不同磁盘.

- 块级拆分**: 一个文件由多个块组成, 不同块存储于不同磁盘.

- 扇区/块读写校验**: 对一个扇区/块读写做校验.

- 磁盘间读写校验**: 多个磁盘间共同构成的信息读写做校验.



# DBMS数据存储与查询实现的基本思想

战德臣

哈尔滨工业大学 教授·博士生导师

黑龙江省教学名师

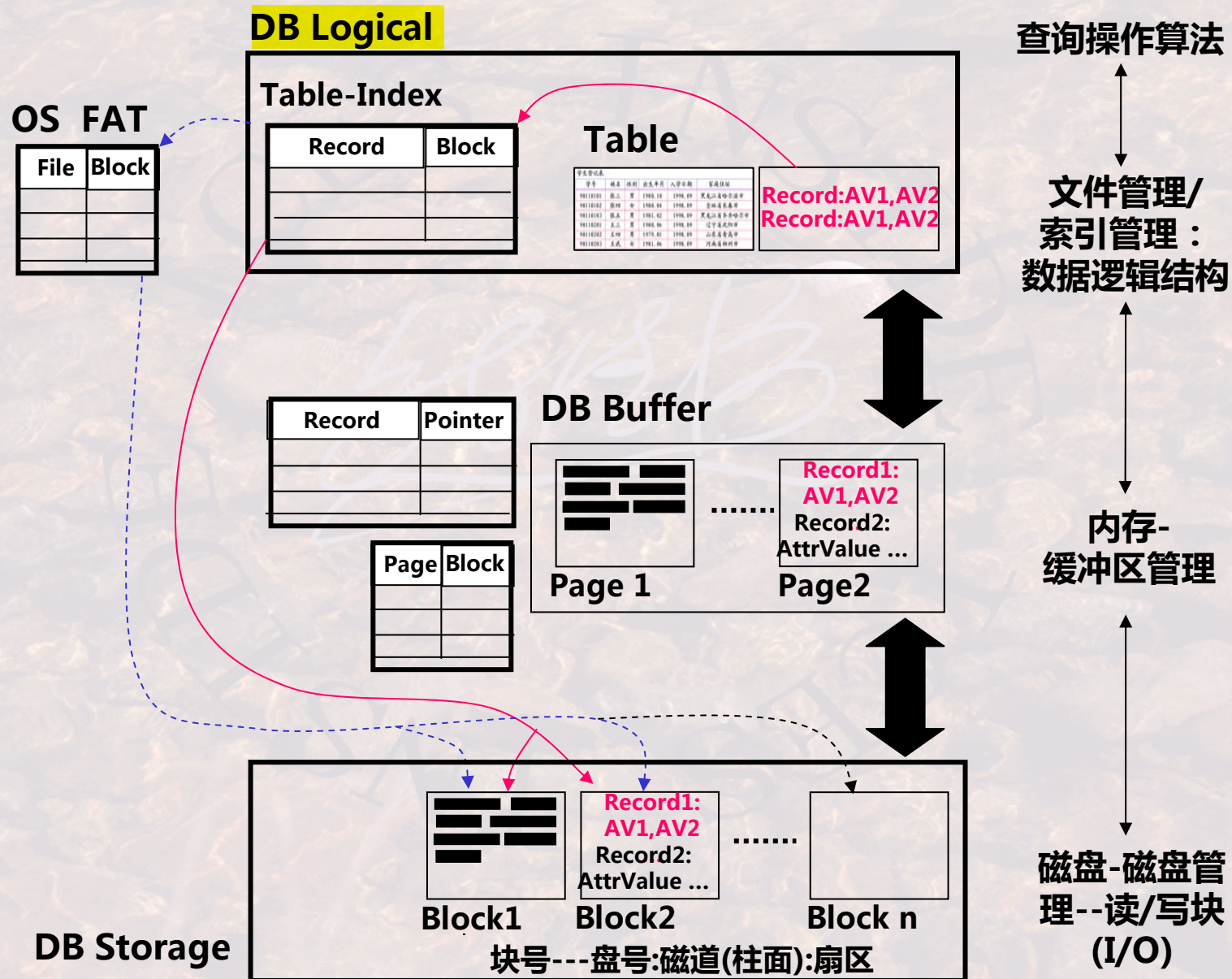
教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent  
**C**omputing for **E**nterprises & **S**ervices,  
**H**arbin **I**nstitute of **T**echnology



# DBMS数据存储与查询实现的基本思想

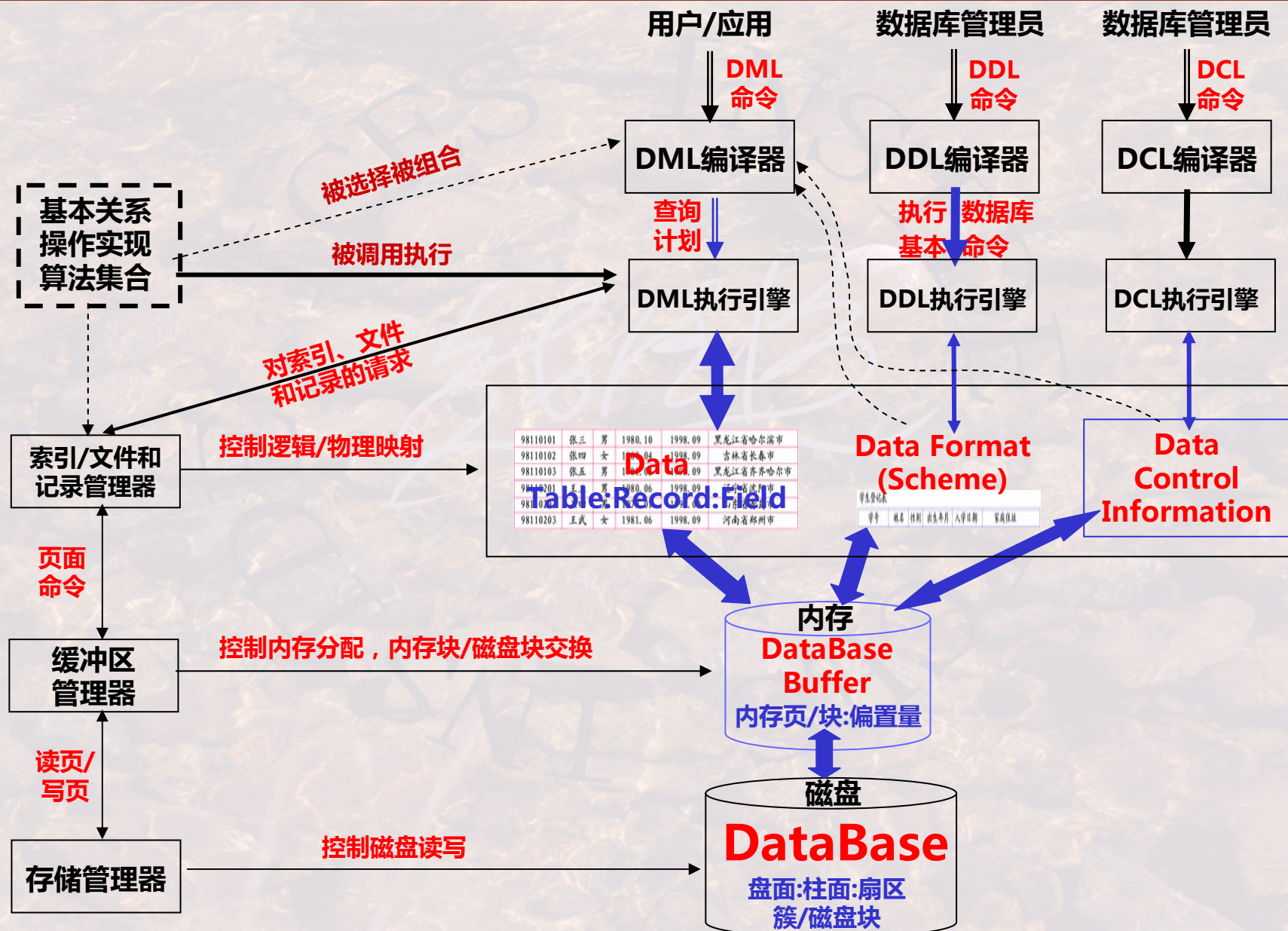
## (1)数据存储的映射关系示意





# DBMS数据存储与查询实现的基本思想

## (2)数据存储与查询实现的基本框架示意





# 数据库之表-记录与磁盘块的映射

战德臣

哈尔滨工业大学 教授·博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent  
**C**omputing for **E**nterprises & **S**ervices,  
**H**arbin **I**nstitute of **T**echnology



# 数据库之表-记录与磁盘块的映射

## (1)数据库概念与磁盘相关概念的映射示意

SQL—表:记录:属性值

学生登记表

学号	姓名	性别	出生年月	入学日期	家庭住址
98110101	张三	男	1980.10	1998.09	黑龙江省哈尔滨市
98110102	张四	女	1980.04	1998.09	吉林省长春市
98110103	张五	男	1981.02	1998.09	黑龙江省齐齐哈尔市
98110201	王三	男	1980.06	1998.09	辽宁省沈阳市
98110202	王四	男	1979.01	1998.09	山东省青岛市
98110203	王武	女	1981.06	1998.09	河南省郑州市

Storage—盘面:磁道:扇区

Block1

010101010101010101001	98110101	张三	男	1980.10	1998.09	黑龙江省哈尔滨市
01010010101010101001010	98110102	张四	女	1980.04	1998.09	吉林省长春市
100110000101010101010	98110103	张五	男	1981.02	1998.09	黑龙江省齐齐哈尔市

Block5

01010101010101010101001	98110101	张三	男	1980.10	1998.09	黑龙江省哈尔滨市
01010010101010101001010	98110102	张四	女	1980.04	1998.09	吉林省长春市
100110000101010101010	98110103	张五	男	1981.02	1998.09	黑龙江省齐齐哈尔市

Block8

01010101010101010101001	98110101	张三	男	1980.10	1998.09	黑龙江省哈尔滨市
01010010101010101001010	98110102	张四	女	1980.04	1998.09	吉林省长春市
100110000101010101010	98110103	张五	男	1981.02	1998.09	黑龙江省齐齐哈尔市

逻辑  
层面

物理  
层面

表

记录:属性值

Storage Schema

文件

FAT

磁盘块

块内0/1数据

盘面:磁道:扇区

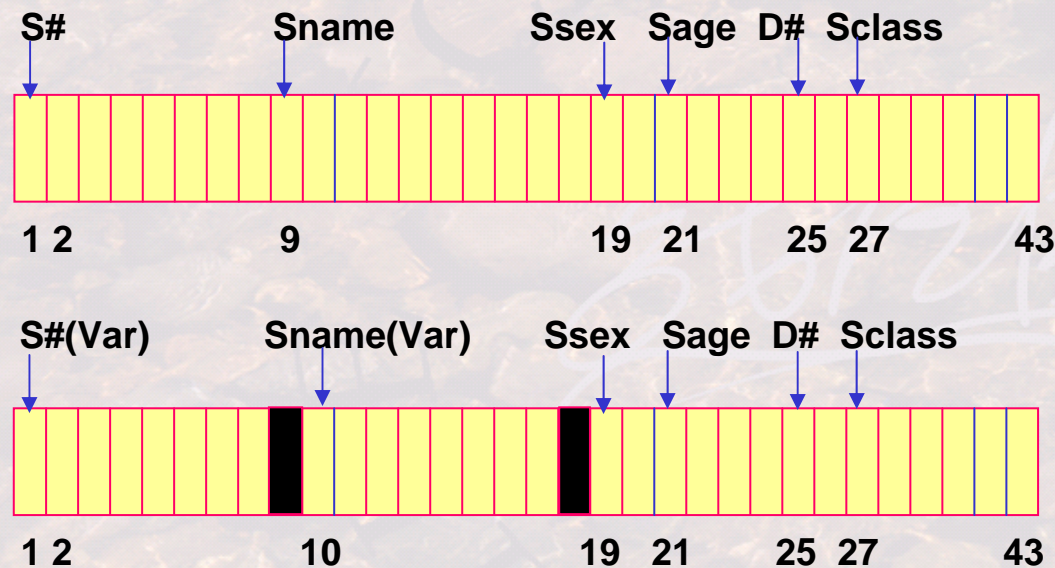


# 数据库之表-记录与磁盘块的映射

## (2)数据库中记录的区分及记录内属性值的区分

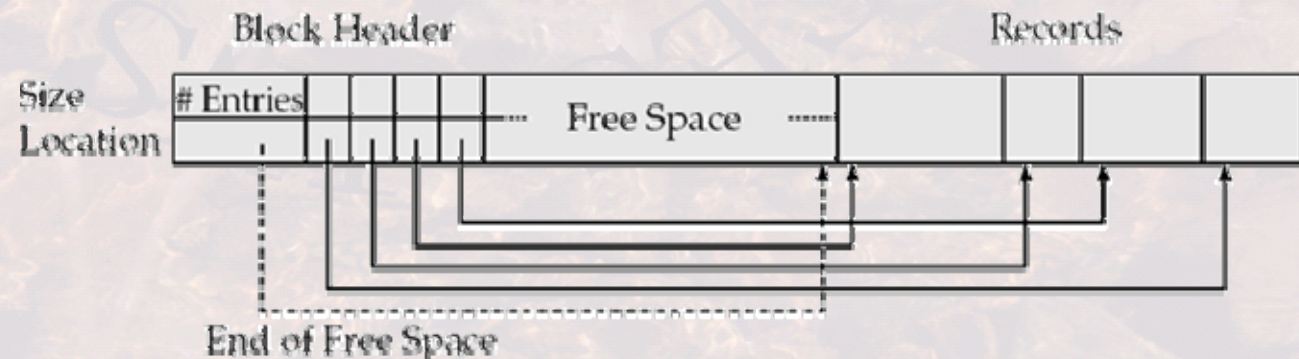
### 数据库记录在磁盘上的存储

□定长记录，还是变长记录(靠分隔符区分开始与结束)



•按长度区分记录  
•按指针(或标志)  
区分记录

•块头如何设计



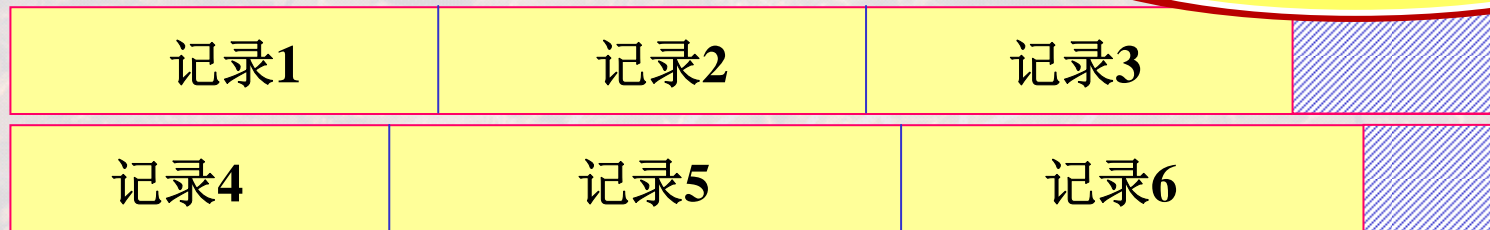


# 数据库之表-记录与磁盘块的映射

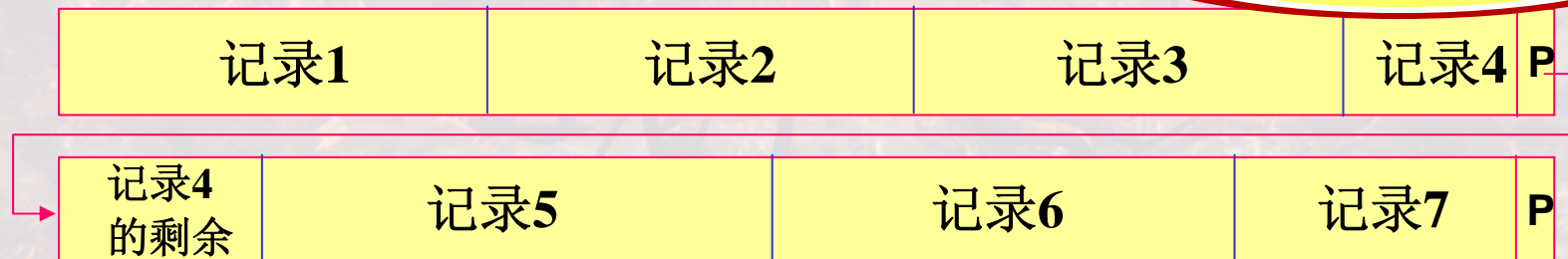
## (3)数据库中的记录 vs. 磁盘块

□记录是非跨块存储，还是跨块存储(靠指针连接)

- 浪费一些存储空间
- 磁盘块间无关联可并行



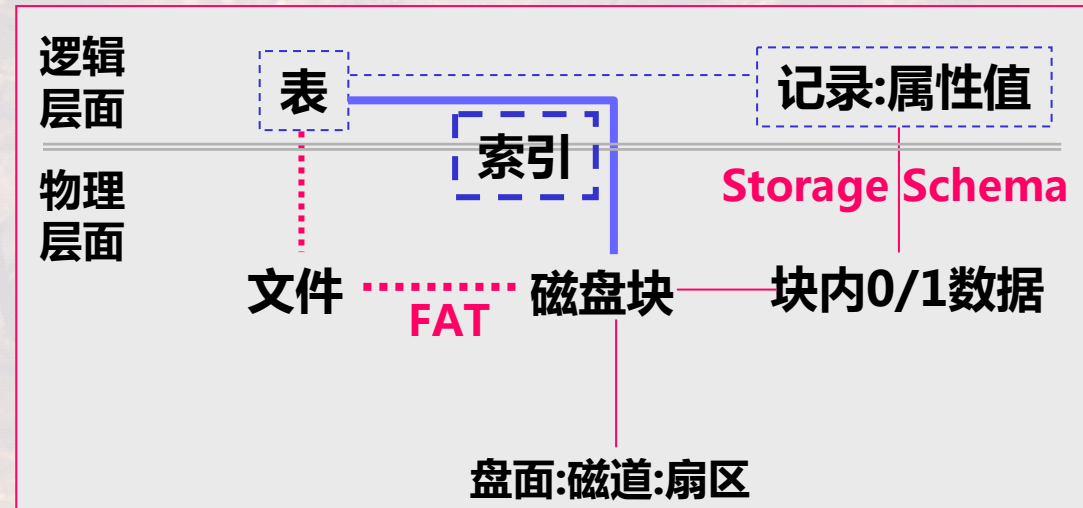
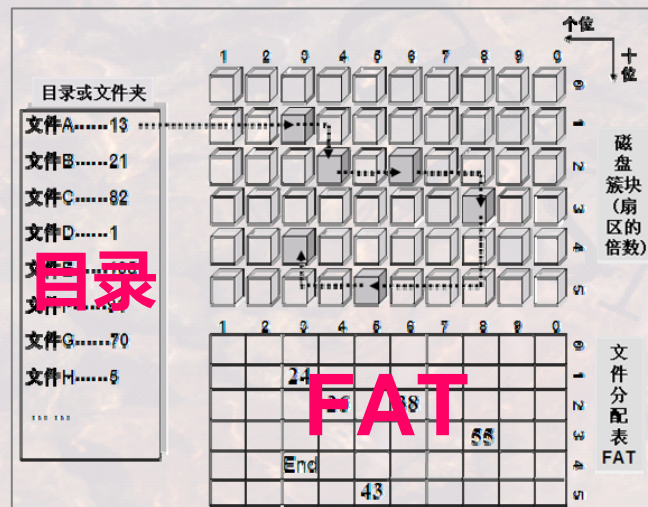
- 节省一些存储空间
- 磁盘块间有关联需串行





### □数据库-表所占磁盘块的分配方法

- ✓ 连续分配: 数据块被分配到连续的磁盘块上(会存在扩展困难问题)
- ✓ 链接分配: 数据块中包含指向下一数据块的指针(访问速度问题)
- ✓ 按簇分配: 按簇分配, 簇是若干连续的磁盘块, 簇之间靠指针连接;  
簇有时也称片段Segment或盘区extent
- ✓ 索引分配: 索引块中存放指向实际数据块的指针





# 数据库之文件组织方法

战德臣

哈尔滨工业大学 教授·博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent  
**C**omputing for **E**nterprises & **S**ervices,  
**H**arbin **I**nstitute of **T**echnology



# 数据库之文件组织方法

## (1)数据组织与存取方法



### □ 数据组织要考虑更新(增、删、改)和检索需求

- ✓ 更新将涉及数据存储空间的扩展与回收问题
- ✓ 检索将涉及扫描整个数据库的问题、大批量处理数据问题
- ✓ 不同的需求要求不同的数据组织方法和存取方法

✓ **文件组织**(File Organization)指的是数据组织成记录、块和访问结构的方式，包括把记录和块存储在磁盘上的方式，以及记录和块之间相互联系的方法

- ✓ **存取方法**(Access Method)指的是对文件所采取的存取操作方法
- ✓ 一种文件组织可以采取多种存取方法进行访问



# 数据库之文件组织方法

## (2)无序文件组织

### □ 文件组织方法之一：无序记录文件(堆文件heap或pile file)

- ✓ 特点：记录可存储于任意有空间的位置，磁盘上存储的记录是无序的。更新效率高，但检索效率可能低
- ✓ 方法1：新记录总插入到文件尾部；删除记录时，可以直接删除该记录所在位置的内容，也可以在该记录前标记“删除标记”
- ✓ 方法2：在前者基础上，新增记录可以利用那些标记为“删除标记”的记录空间





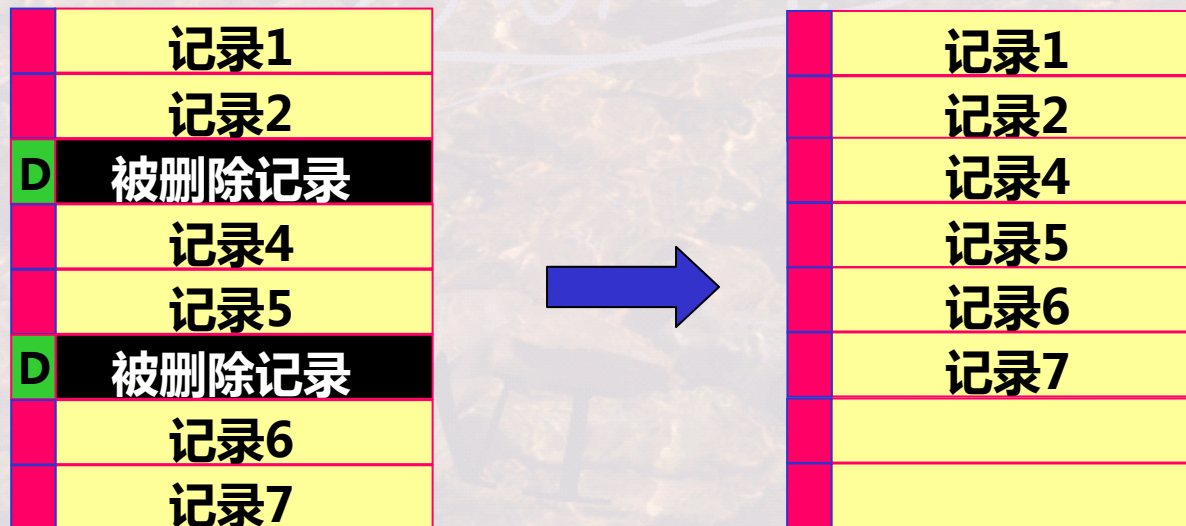
## 数据库之文件组织方法

### (2)无序文件组织

#### □ 文件组织方法之一：无序记录文件(堆文件heap或pile file)(续)

✓ 频繁删增记录时会造成空间浪费，所以需要周期性重新组织数据库

✓ **数据库重组(Reorganization)**是通过移走被删除的记录使有效记录连续存放，从而回收那些由删除记录而产生的未利用空间。





# 数据库之文件组织方法

## (3)有序文件组织



### □ 文件组织方法之二：有序记录文件(排序文件Sequential)

- ✓ 特点：记录按某属性或属性组值的顺序插入，磁盘上存储的记录是有序的。检索效率可能高。
- ✓ 用于存储排序的属性通常称为**排序字段**(Ordering field)，通常，排序字段使用关系中的主码，所以又称**排序码**(Ordering key)
- ✓ 当按排序字段进行检索时，速度得到很大提高；但当按非排序字段检索时，速度可能不会提高很多

	(S#:001的)记录
	(S#: 003的)记录
	(S#: 005的)记录
	(S#: 006的)记录
	(S#: 007的)记录
	(S#: 009的)记录
	(S#: 010的)记录
尾部	(S#: 012的)记录

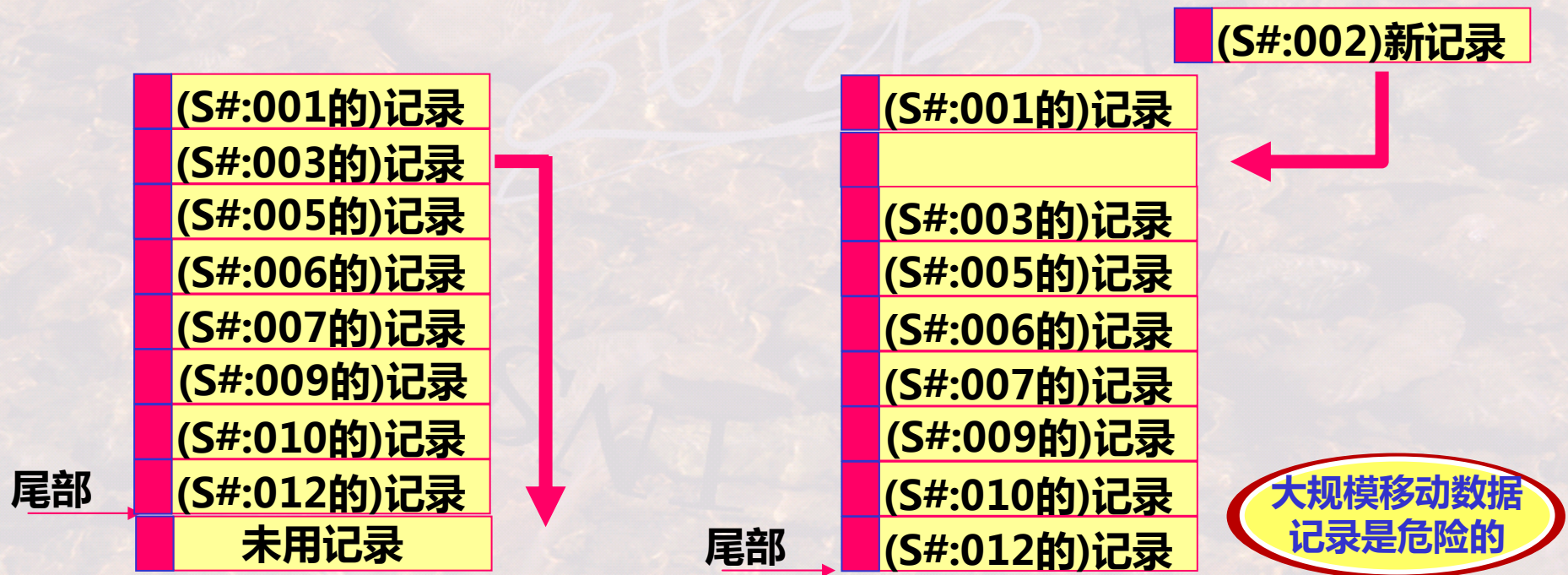


## 数据库之文件组织方法

### (3)有序文件组织

#### □ 文件组织方法之二：有序记录文件(排序文件Sequential)(续)

- ✓ 有序记录文件的更新效率可能很低
- ✓ 因为：在更新时要移动其他记录，为插入记录留出空间





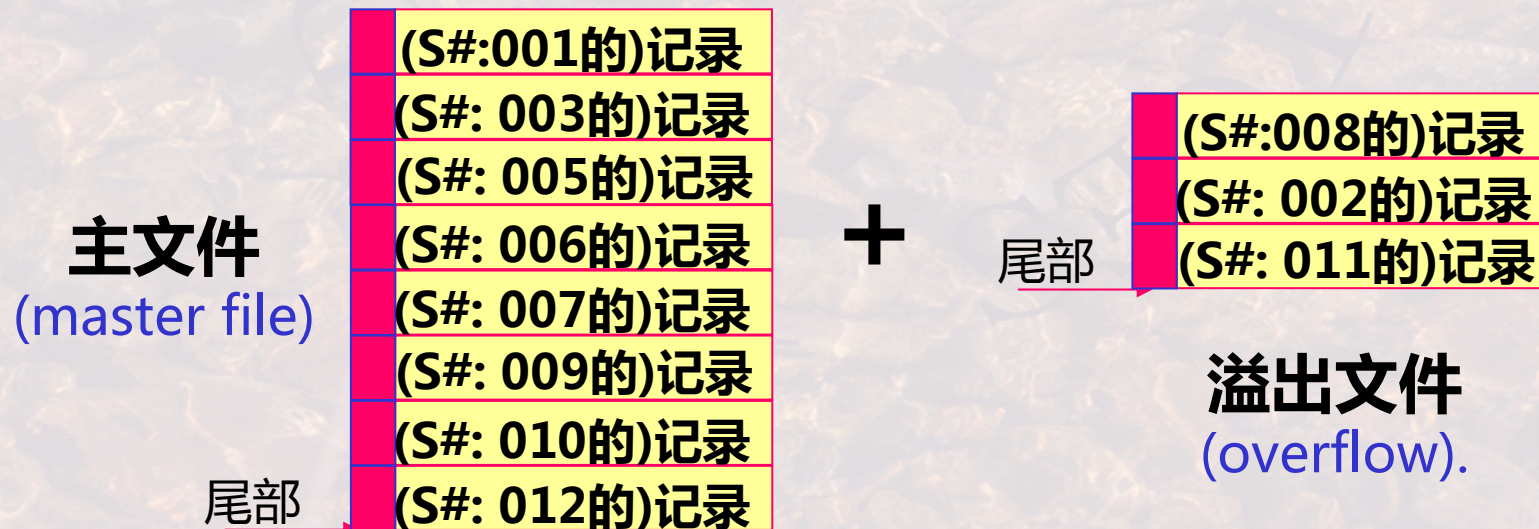
# 数据库之文件组织方法

## (3)有序文件组织



### □ 文件组织方法之二：有序记录文件(排序文件Sequential) (续)

- ✓ 改进措施是可为将来有可能插入的元组预留空间(这可能造成空间浪费), 或者再使用一个临时的无序文件(被称为溢出文件)保留新增的记录。
- ✓ 当采取溢出文件措施时, 检索操作既要操作主文件, 又要操作溢出文件。
- ✓ 所以需要周期性重新组织数据库
- ✓ **数据库重组**是将溢出文件合并到主文件中, 并恢复主文件中的记录顺序。



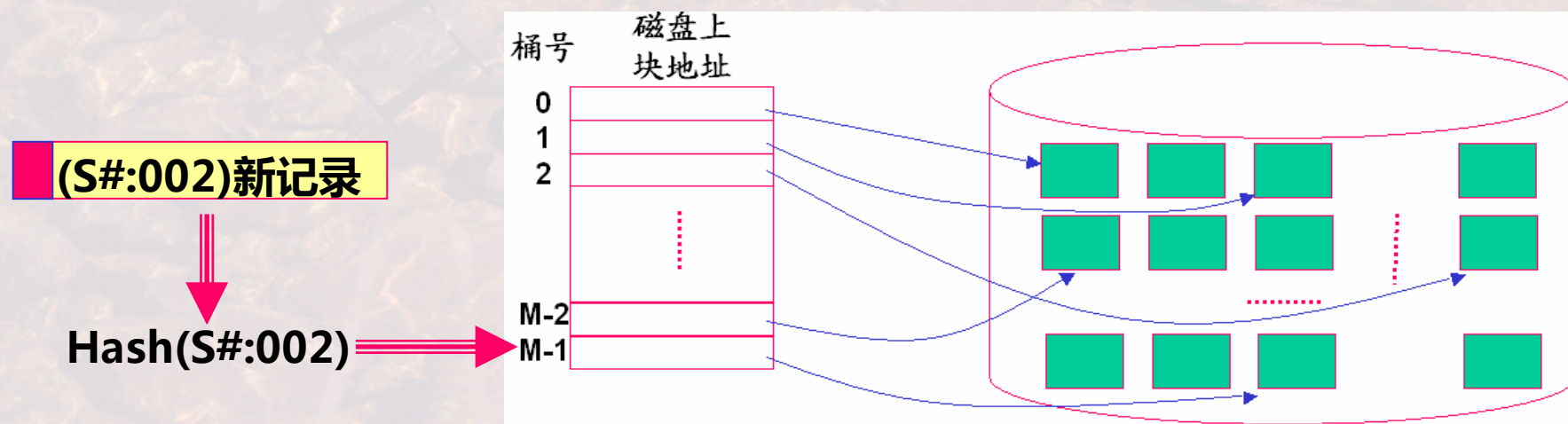


# 数据库之文件组织方法

## (4)散列文件组织

### □ 文件组织方法之三：散列文件(Hash file)

- ✓ 特点：可以把记录按某属性或属性组的值, 依据一个散列函数来计算其应存放的位置：桶号(Bucket, 块号或簇号等)。检索效率和更新效率都有一定程度的提高
- ✓ 用于进行散列函数计算的属性通常称为散列字段(Hash field)，散列字段通常也采用关系中的主码，所以又称散列码(hash key)





# 数据库之文件组织方法

## (4)散列文件组织

### □ 文件组织方法之三：散列文件(Hash file)(续)

✓ 不同记录可能被hash成同一桶号，此时需在桶内顺序检索出某一记录

bucket 0

--	--	--

bucket 1

--	--	--

bucket 2

--	--	--

bucket 3

A-217	Brighton	750
A-305	Round Hill	350

bucket 4

A-222	Redwood	700

bucket 5

A-102	Perryridge	400
A-201	Perryridge	900
A-218	Perryridge	700

bucket 6

--	--	--

bucket 7

A-215	Mianus	700

bucket 8

A-101	Downtown	500
A-110	Downtown	600

bucket 9

--	--	--

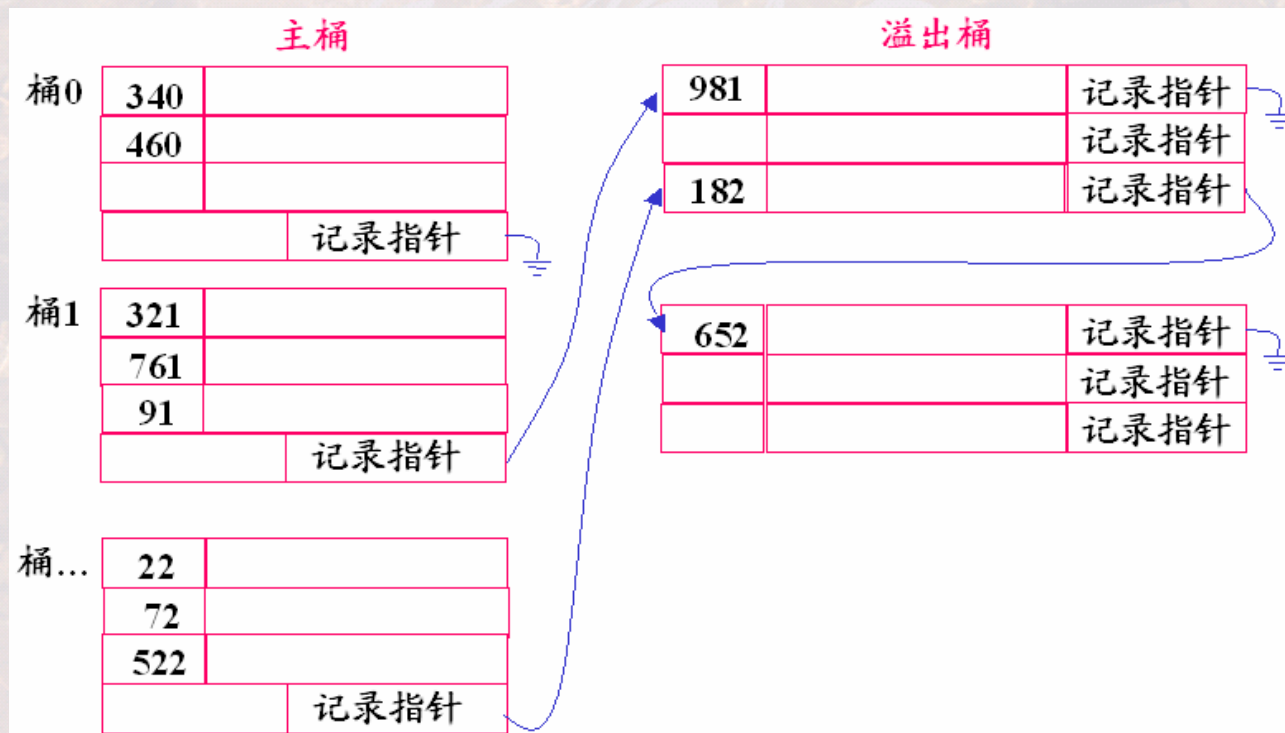


# 数据库之文件组织方法

## (4)散列文件组织

### □ 文件组织方法之三：散列文件(Hash file)(续)

- ✓ 链接法处理溢出
- ✓ 散列还有许多问题及许多的处理技巧, 如散列桶的数目以及桶的大小, 动态散列技术等等(这里不再叙述)





### □ 文件组织方法之四：聚簇文件(Clustering file)

- ✓聚簇：将具有相同或相似属性值的记录存放于连续的磁盘簇块中
- ✓多表聚簇：将若干个相互关联的Table存储于一个文件中——这可提高多表情况下的查询速度(有很多问题及相关的处理技巧，不再详细叙述)





增删改时如何快速“存”

检索查询时如何快速“取”





# Oracle DB物理存储简介

战德臣

哈尔滨工业大学 教授·博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent  
**C**omputing for **E**nterprises & **S**ervices,  
**H**arbin **I**nstitute of **T**echnology



- 10/8/13

数据库(database)	数据库(如SCT)																			
表空间(tablespace)	tspace1												SYSTEM							
文件(datafile)	fname1						fname2						fname3							
表(table)	student	dept	sc	course	teacher	Index1														
段(segment)	DATA	DATA	DATA	DATA	DATA	INDEX														
盘区(extent)																				
基本数据块(data block)																				



database  
tablespace  
data file  
table

数据库(如SCT)															
tspace1										SYSTEM					
fname1				fname2				fname3							
student		dept		sc		course		teacher		Index1					
DATA		DATA		DATA		DATA		DATA		INDEX					



- 物理存储层由段(segment)、盘区(extent)和数据块(data block)构成
- 数据块是最小的IO存储单位，又称为Oracle块、页(相当于扇区)
- 盘区是特定数量的连续数据块(相当于簇)。Oracle中的盘区是可以动态变化的，随不同数据库存储需求而调整
- 段是一组分配了特定数据结构的盘区，又分为数据段、索引段和临时段等
- 一个表的数据可以存放在一个段内，也可以存放在多个段内。一个段可以存放一个表的数据，也可以存放多个表的数据(如聚簇文件)

segment  
extent  
data block

数据库(如SCT)																	
tspace1												SYSTEM					
fname1				fname2				fname3									
student	dept	sc	course	teacher	Index1												
DATA	DATA	DATA	DATA	DATA	INDEX												



- Oracle物理数据库相关的定义语句
- 定义表空间时，需要考虑物理存储性能。注意参数的使用，参数的含义参见Oracle手册。

```
CREATE TABLESPACE tblspname
    DATAFILE 'filename'
        [SIZE n [K|M]]
        [REUSE]
        [AUTOEXTEND OFF | AUTOEXTEND ON
            [NEXT n [K|M] [MAXSIZE {UNLIMITED | n [K|M]}]
            {, 'filename' ...}]
        [ONLINE | OFFLINE]
        [DEFAULT STORAGE ([ INITIAL n] [NEXT n] [MINEXTENTS n]
            [MAXEXTENTS {n | UNLIMITED}] [PCTINCREASE n])]
        [MINIMUM EXTENT n [K|M]] ;
```



### ➤简要理解

一个表空间  
可以由多个  
文件构成

表空间的文  
件是否可自  
动扩展

```
CREATE TABLESPACE tblspname
DATAFILE 'filename'
[SIZE n [K|M]]
[REUSE]
[AUTOEXTEND OFF | AUTOEXTEND ON
[NEXT n [K|M] [MAXSIZE {UNLIMITED | n [K|M]}]
{, 'filename' ...}]
[ONLINE | OFFLINE]
[DEFAULT STORAGE ([ INITIAL n] [NEXT n] [MINEXTENTS n]
[ MAXEXTENTS {n | UNLIMITED}] [PCTINCREASE n])]
[MINIMUM EXTENT n [K|M]] ;
```

数据库(如SCT)														
tspace1										SYSTEM				
fname1					fname2					fname3				
student	dept	sc	course	teacher										
				Index1										
DATA	DATA	DATA	DATA	DATA	INDEX									

可设置表空间的初始  
容量、最小盘区数、  
每次扩展容量等

盘区的大小  
是可以动态  
调整的

可设置构成  
表空间的最  
小盘区



➤定义表(table)时，需要考虑物理存储性能。注意参数的使用，参数的含义参见Oracle手册。

```
CREATE TABLE tablename
    ( {colname datatype [default { constant | NULL}]
      {, colname datatype etc. . . .}
    [ORGANIZATION HEAP | ORGANIZATION INDEX ]
    [TABLESPACE tblspname]
    [STORAGE ([initial n [K|M]]
              [next n [K|M]] [minextents n]
              [maxextents n] [pctincrease n] ) ]
    [PCTFREE n]
    [PCTUSED n]
```

SQL的CREATE TABLE的三种功能：

(1)定义模式；(2)定义物理存储结构；(3)定义完整性约束



```
CREATE TABLE tablename  
  ( {colname datatype [default { constant | NULL}]  
    {, colname datatype etc. . . .}  
  [ORGANIZATION HEAP | ORGANIZATION INDEX]  
  [TABLESPACE tblspname]  
  [STORAGE ([initial n [K|M]]  
            [next n [K|M]] [minextents n]  
            [maxextents n] [pctincrease n] )]  
  [PCTFREE n]  
  [PCTUSED n]
```

Initial	# 表示分配给表（段）的初始盘区的大小
next	# 指定第二个盘区的大小
pctincrease	# 指定第三个及后续分配的盘区的增长百分比
minextents	# 指定创建表时，至少要分配多少个盘区给这个表（段）
maxextents	# 指定可以给这个表（段）的盘区的最大数量

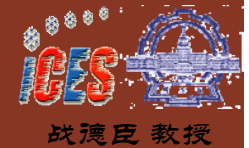
Pctused	# 制定块中数据使用空间的最低百分比。
pctfree	# 为存储块中行的更新预留空闲空间的最小百分比。

例如，假定在 Create table 语句中指定了 pctfree 为 20，则说明在该表的数据段内每个数据块的 20%被作为可利用的空闲空间，用于更新已在数据块内存在的数据行。其余 80%是用于插入新的数据行，直到达到 80%为止。显然，pctfree 值越小，则为现存行更新所预留的空间越少。因此，如果 pctfree 设置得太高，则在全表扫描期间增加 I/O，浪费磁盘空间；如果 pctfree 设置得太低，则会导致行迁移。



# Oracle DB物理存储简介

## (3)小结



数据库(database)	数据库(如SCT)																	
表空间(tablespace)	tspace1												SYSTEM					
文件(datafile)	fname1						fname2						fname3					
表(table)	student	dept	sc	course	teacher	Index1												
段(segment)	DATA	DATA	DATA	DATA	DATA	INDEX												
盘区(extent)																		
基本数据块(data block)																		

**CREATE DATABASE sct**

将多个表空间关联在一起

**CREATE TABLESPACE tspace1...**

用多个操作系统文件占位，将多个表的存取统一在一个空间中  
直接对数据块-盘区-Segment进行操作。

**CREATE TABLE tablename ...**

表的存取，直接映射对数据块-盘区的操作



# 回顾本讲学了什么？

战德臣

哈尔滨工业大学 教授·博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent  
**C**omputing for **E**nterprises & **S**ervices,  
**H**arbin **I**nstitute of **T**echnology



# 回顾本讲学习了什么？

## 数据库物理存储

