

高级数据库系统及其应用

第2部分 关系数据库系统实现

第4章 数据存储和组织管理

xshxie@ustc.edu.cn

LOGO

第4章 数据存储和组织管理



4.1

物理存储介质

4.2

磁盘空间管理

4.3

文件的页组织

4.4

页表示格式

4.5

记录表示格式

4.6

DB元信息及其组织管理

4.7

DB缓冲区管理

4.1 物理存储介质



4.1.1 存储介质的层次

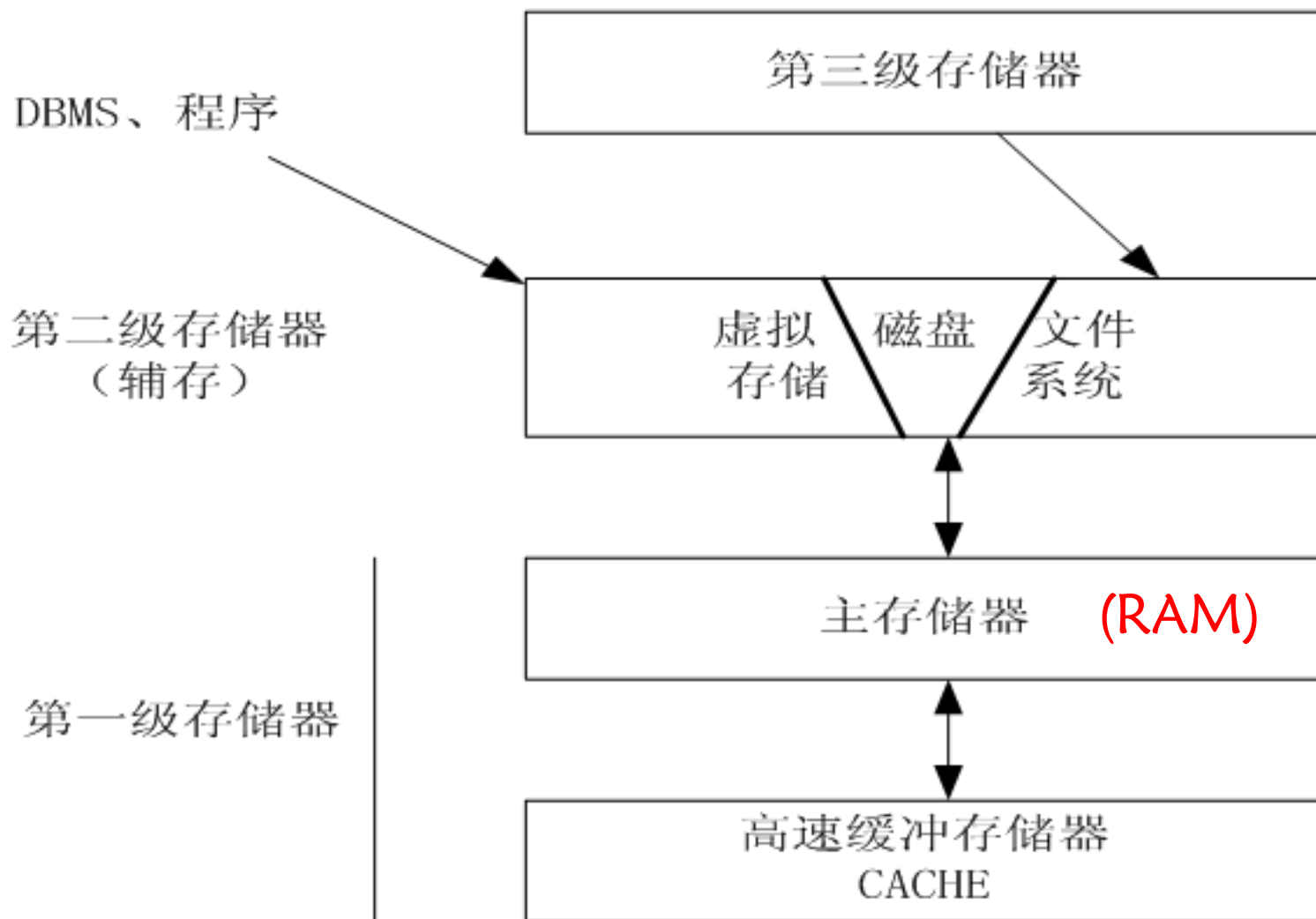
4.1.2 磁盘的物理特性

4.1.3 磁盘故障及其处理策略

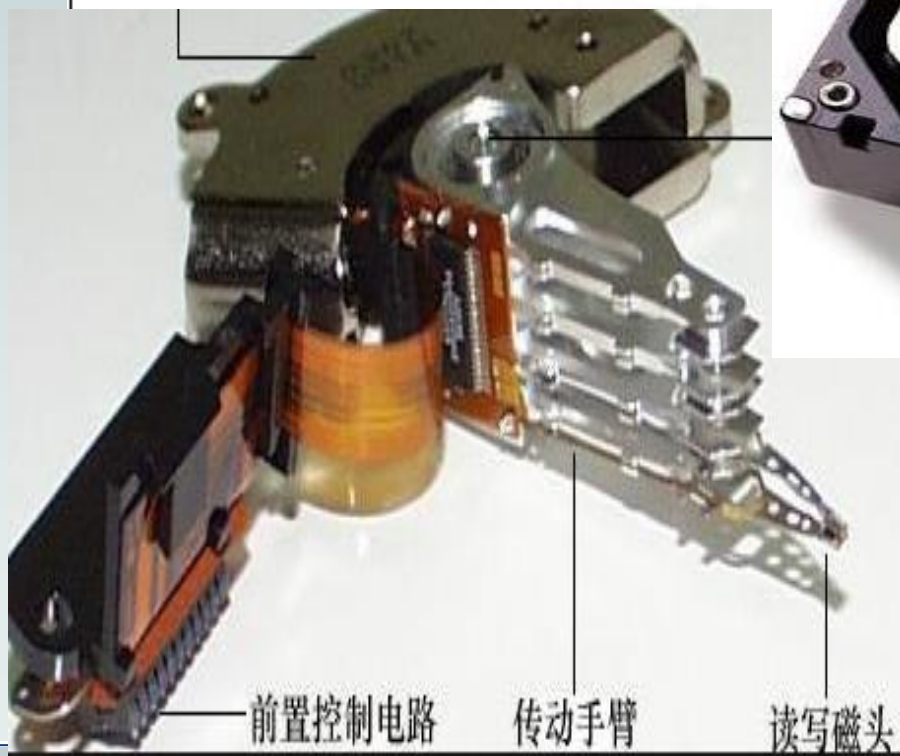
4.1.4 磁盘块存取优化

4.1.1 计算机系统的存储介质结构层次

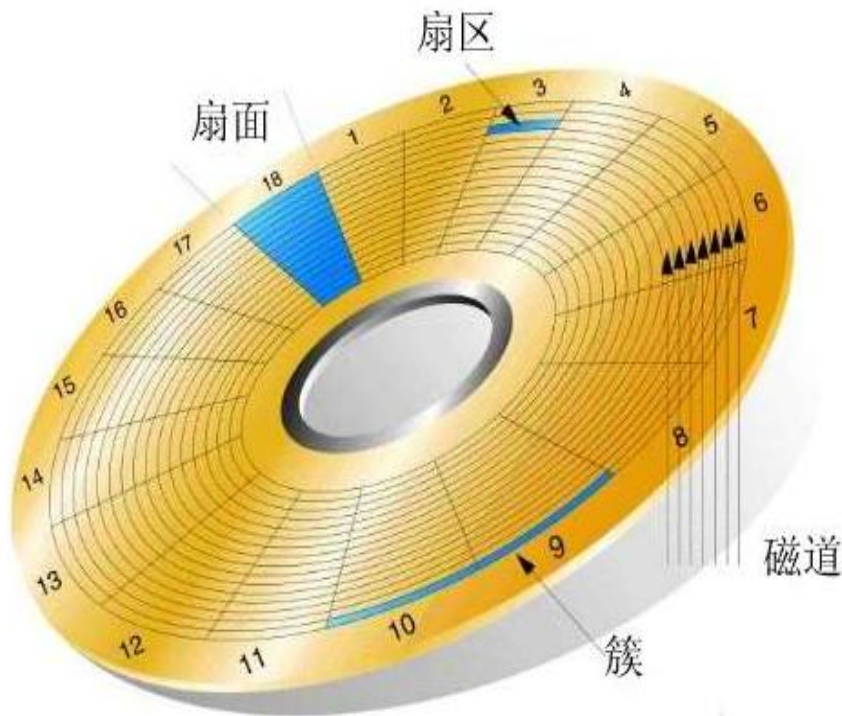
磁带、光盘、网络、……



磁盘（物理硬盘）



物理磁盘的逻辑抽象

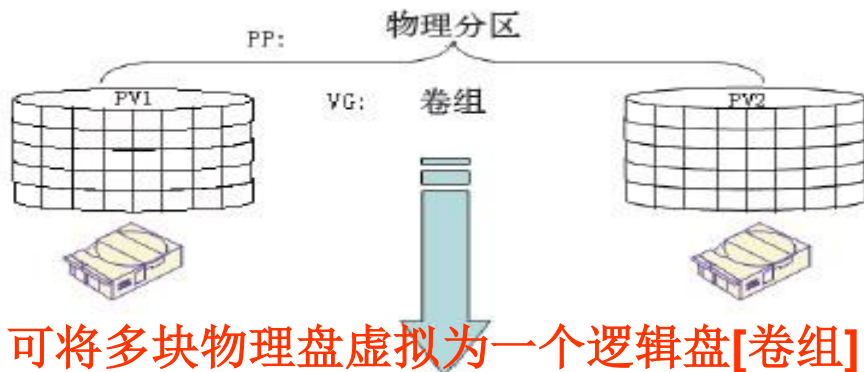


0 1 2 3 4 5 6 7 ... n

								...	
--	--	--	--	--	--	--	--	-----	--

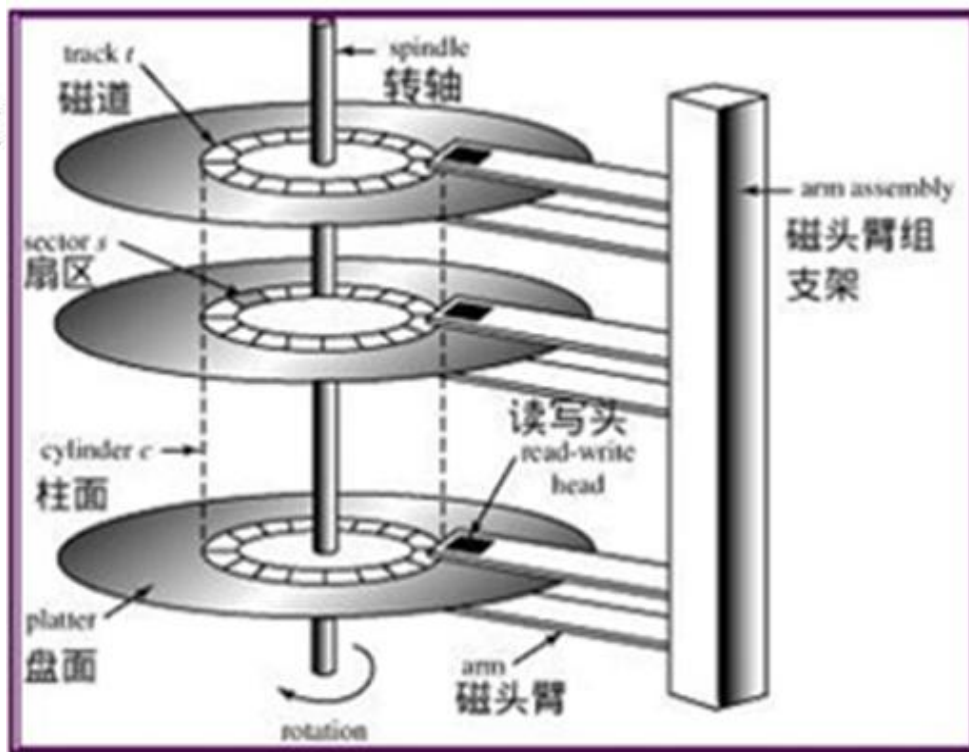
单块磁盘可进一步虚拟化抽象为：

磁盘块的序列（数组）



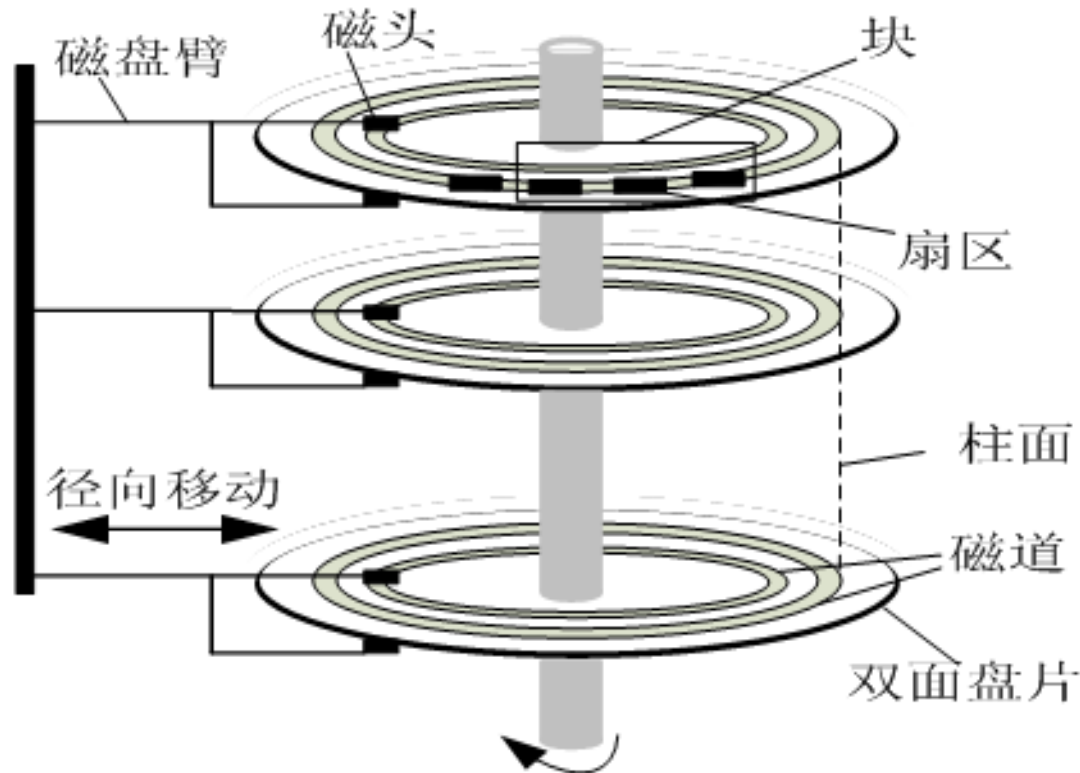
可将多块物理盘虚拟为一个逻辑盘[卷组]

A large grid representing logical disk blocks, with 10 columns and 10 rows.



4.1.2 磁盘的物理特性

(1) 磁盘结构



硬盘容量＝

盘面数 × 每盘面磁道数 × 每磁道扇区数 × 每扇区字节数

4.1.2 磁盘的物理特性



(2) 磁盘基本操作特性

- ❖ 磁盘读写的**最小单位**是扇区。但在**OS**或**DBMS**系统层次，磁盘读写的基本单位是**块(block)**。
 - 默认的磁盘块大小：4KB。
- ❖ 进行实际磁盘读写时，主存中必须有磁盘块缓冲区；在磁盘和主存之间传送一个磁盘块称为**1次I/O**操作。
- ❖ 读写一个块的时间：**寻道时间+旋转延迟时间+传输时间**。

例4.1

设有一含**3**个盘片的硬盘，共有**4**个面，转速为**4500**转/分钟，盘面有效记录区的内、外径分别为**10**、**30cm**，记录位密度为**250**位/mm，磁道密度为**8**道/mm，每个磁道有**16**扇区，每扇区**512**字节。试计算：

1) 磁盘的总磁道数 $4 \times \left(\frac{30-10}{2} \times 10 \right) \times 8 = 3200$

2) 非格式化容量和格式化容量

总磁道数*内径周长*位密度 = $3200 \times [3.14 \times 10 \times 10] \times 250 / 8 = 29.95 \text{MB}$

格式化容量 = $3200 \times 16 \times 512 = 25 \text{MB}$

3) 平均数据传输速率。

每磁道字节数 x 每秒转数 = $16 \times 512 \times 4500 / 60 = 600 \text{KB/s}$

例4.2

$\because 3840/60=64$ 转/秒，即 $(1/64)$ 秒 / 转

设有某硬盘：含有**4**个盘片，**8**个盘面；
每个盘面有**8192**个磁道，每个磁道有**256**
个扇区；每个扇区**512**个字节。试计算以下
磁盘参数：

容量 = $8 \times 8192 \times 256 \times 512 = 8 \text{ GB}$
每道块数 = $256 \times 512 / 4096 = 32$
每块：占8个扇区。

- 1) 磁盘格式化容量。
- 2) 若一个块大小为**4096**字节，求每个磁道能存的块数。
- 4) 若磁盘转速为**3840**转/分，即**1/64**秒转一周。磁头起落**1**次**1**毫秒，每移过**500**个磁道另加**1**毫秒，试计算读写一个块的平均时间。
 - 寻道时间 = $[1\text{ms} + 8192/500] / 3 = 5.8 \text{ ms}$
 - 旋转等待时间 = $[1 / 64] / 2 = 15.6\text{ms} / 2 = 7.8 \text{ ms}$
 - $\because 360 \times (7/256) \times 0.1 + 360 \times (8/256) \times 0.9 = 360 \times 7.9/256$

• 读写1个块时间 = $15.6 \text{ ms} \times (360 \times 7.9/256) / 360 \approx 0.5 \text{ ms}$

4.1.3 磁盘故障及其处理策略

一、磁盘故障分类

磁盘故障通常有以下几种方式或类型：

- 间断性故障。【时好时坏】
- 写故障。【数据写入后再读出结果不同】
- 部分介质损坏。
- 磁盘崩溃。

二、校验和技术 【引入冗余位的校验技术】

- 在磁盘扇区存储一些冗余位，以可帮助识别从扇区读出的内容是否正确。
- 最简单的校验和:是基于扇区内所有位的奇偶性。
- 通过增加奇偶位数，可降低检不出错误的概率。
 - 若用 n 个位存储校验和,则漏检错误的概率仅为 $1/2^n$

4.1.3 磁盘故障及其处理策略



一、磁盘故障分类

二、校验和技术

三、稳定存储技术

- 校验和技术能帮助检测读写故障或介质故障，但不能帮助我们纠正错误。
- 基于稳定存储(stable storage)的多副本策略，可能帮助我们一定程度上解决这个问题。

四、从崩溃的磁盘故障恢复：**RAID**技术

- 基于磁盘冗余阵列 的磁盘组织技术。
- **Redundant Array of Inexpensive Disks**

几种常用的**RAID**级简介



1. RAID0级(nonredundant striping)

2. RAID1级(mirrored disks)

- 为每一个磁盘配置一**镜像磁盘**，有效容量利用率只有50%，成本较高。
- 适合于安全性要求很高场合。

3. RAID2/3级(位级拆分，极少或基本不用)

几种常用的**RAID**级简介

4. **RAID4级**(**block-Interleaved Parity**块-奇偶交替)

- 采用**块级拆分存储技术**，能充分利用块设备工作特性，且能适应各种数据量传输的磁盘请求。
- 用3个数据磁盘,1个冗余盘的配置方案，冗余块存储各数据盘中对应块的奇偶校验数据。
- 存储利用率**75%**。

5. **RAID5级**

- 是RAID4的改进。
- 校验块是交替分布在各磁盘上。

几种常用的**RAID**级简介



★盘组合特点

- 数据盘对应列至少有两个1；冗余列只有一个1
- 阵列中所有列都是不同的；
- 每个行中，值为1列对应盘正好构成一组RAID4

	数据盘				冗余盘		
盘号	1	2	3	4	5	6	7
	1	1	1	0	1	0	0
	1	1	0	1	0	1	0
	1	0	1	1	0	0	1

- **RAID6**的故障恢复步骤
 - 设a, b两个盘坏了，一定能找到a,b列值不同的行—RAID4组合

4.1.4 磁盘块存取优化

- ❖ 在多数**OS**中，磁盘块I/O请求是由**文件系统**和**虚拟内存管理器**产生的。
- ❖ 在**DB**系统中，系统高层的页请求，通过**磁盘空间管理器**，也会产生磁盘块I/O请求。
- ❖ 由于存取磁盘比存取主存要慢好几个量级，所以，**DB**系统改善磁盘块存取性能非常重要。

4.1.4 磁盘块存取优化



一、磁头调度技术

- 先到先服务
- 电梯算法

二、采用特殊的文件组织方式

- 按连续柱面存储数据

三、采用磁盘缓冲池技术（第6章 介绍）

- **DB**数据缓冲池技术
- 磁盘预取技术
- 双缓冲技术

4.2 磁盘空间管理



4.2.1 磁盘空间管理器

4.2.2 利用OS管理磁盘空间

4.2.3 跟踪自由块

磁盘空间管理器

❖ 是**DBMS**的底层软件模块，隐藏了**DBMS**与磁盘有关的操作细节，并支持以‘页’为单位的数据管理。

- 页(**page**)的大小通常就是磁盘块(**block**)大小，读写一个页可通过一次磁盘块I/O完成。
- 提供分配、释放和读写页的有关命令操作
- 允许高层软件认为**DB**数据是一系列以页为单位的磁盘数据集合。

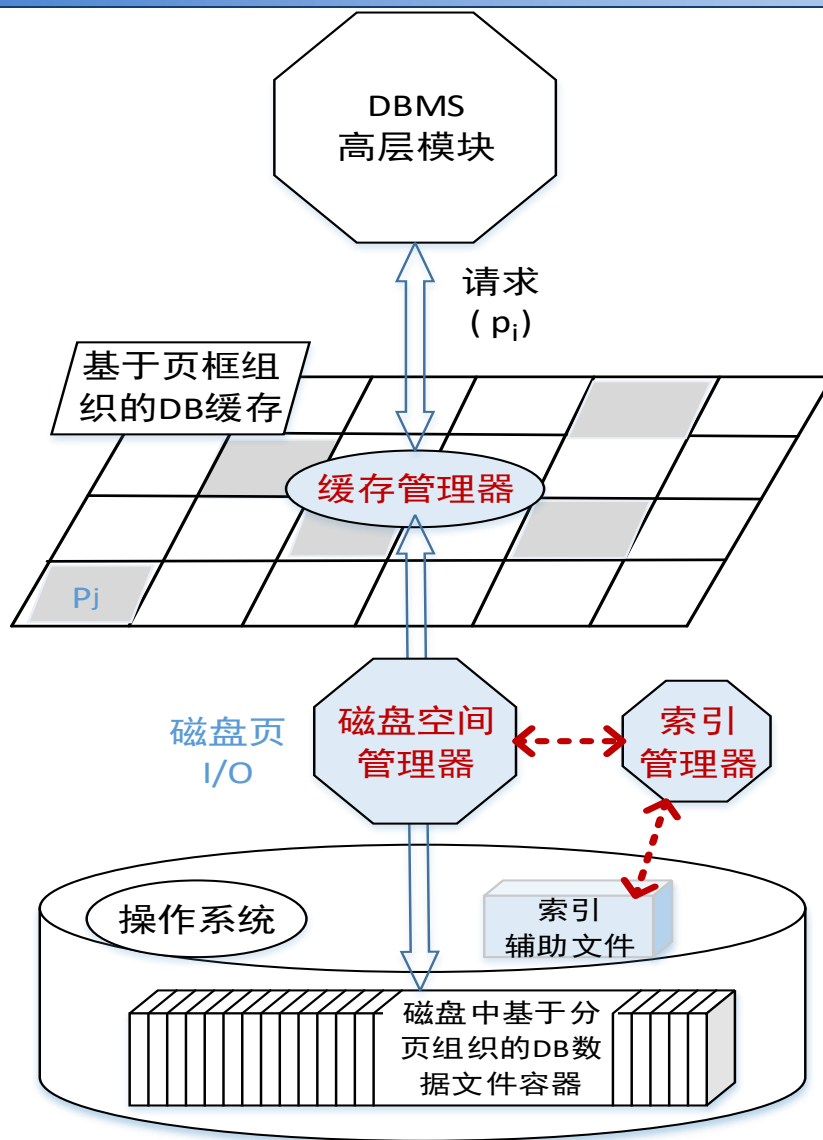
❖ **DB**中的“关系” \longleftrightarrow “关系(数据)文件”。

★ 关系文件，可视为“有名的页序列”

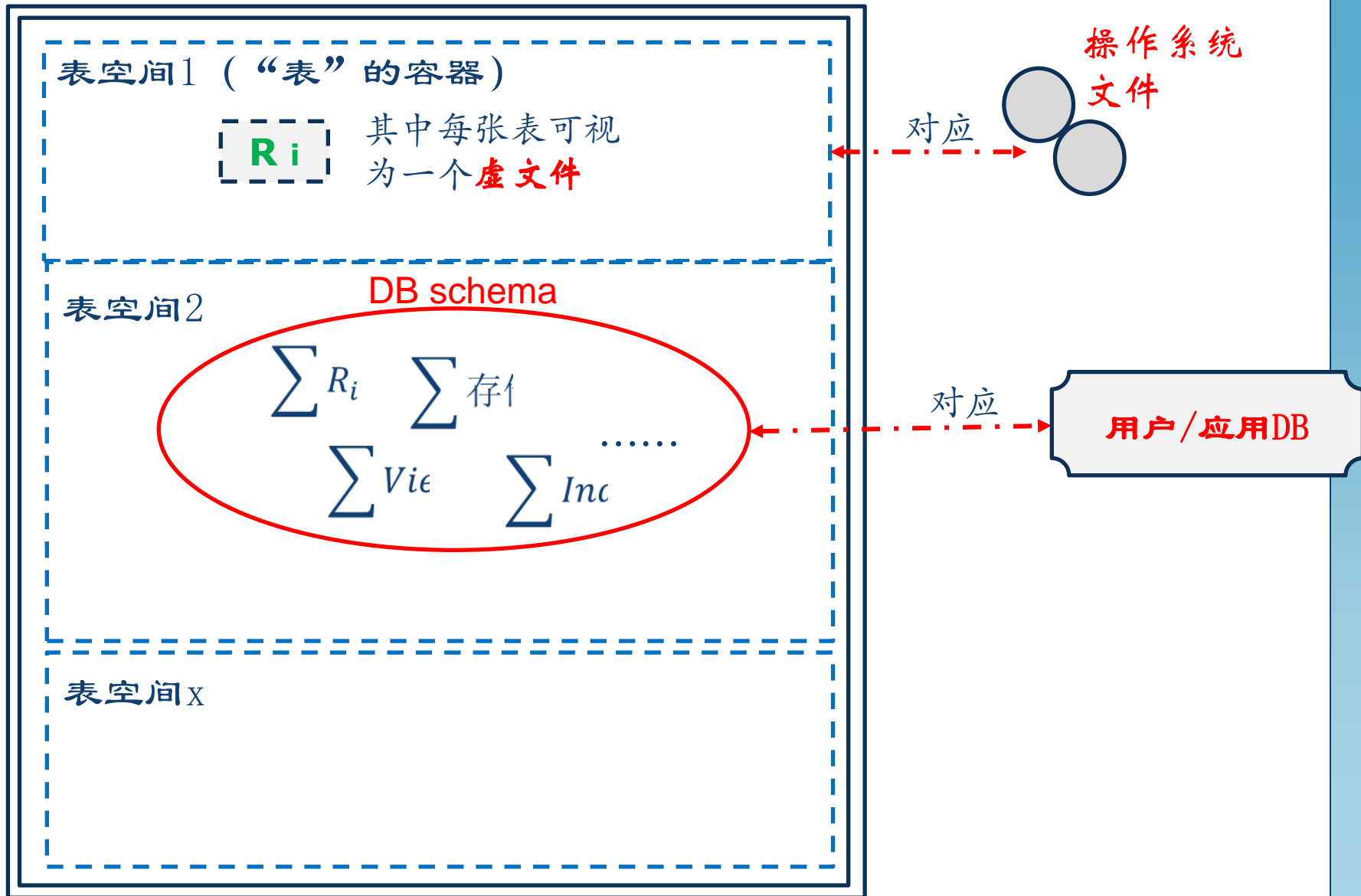
- 它既可能是实际的**OS**文件，也可能只是一个虚的**OS**文件。

对于后者，实际**DB**数据文件相当于“关系文件”的容器

DBMS底层架构图



磁盘空间组织---由若干“表空间”组成



4.3 文件的页组织



4.3.1 堆文件

4.3.2 排序文件

4.3.3 索引文件

4.3.4 散列文件

本节 内容 安排

- ◆ 一个关系文件所包含的记录集，可能存储在若干不同的页上。
- ◆ 高层**DBMS**一般将“页”视为容纳多个记录的对象。
- ◆ 本节讨论文件中---页的关联组织方式。

4.3.1 堆文件

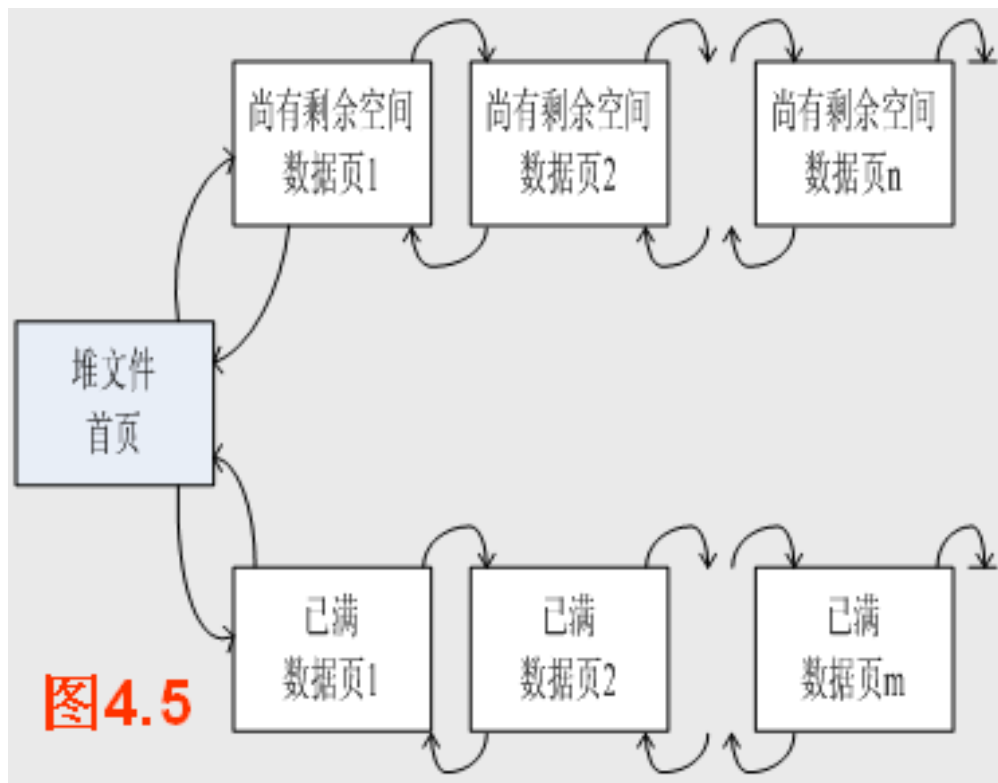
- ❖ 属无序文件，文件中页的大小相同。
- ❖ 堆文件页中的记录是无序的，查询指定的检索键值或键值取值范围，只能顺序存取。
- ❖ 堆文件管理支持
 - 创建/删除堆文件； 扫描文件；
 - 插入/删除/检索给定rid的记录。
但rid如何获得？（←索引）
 - 不支持： 定位满足指定查询条件的有关记录rids

基于双向页链表的堆文件组织

❖ 将文件页以双链表方式链接在一起。

❖ 缺点

- 变长记录情况下，可能所有页都有空闲；检索记录可能需顺序扫描多个页



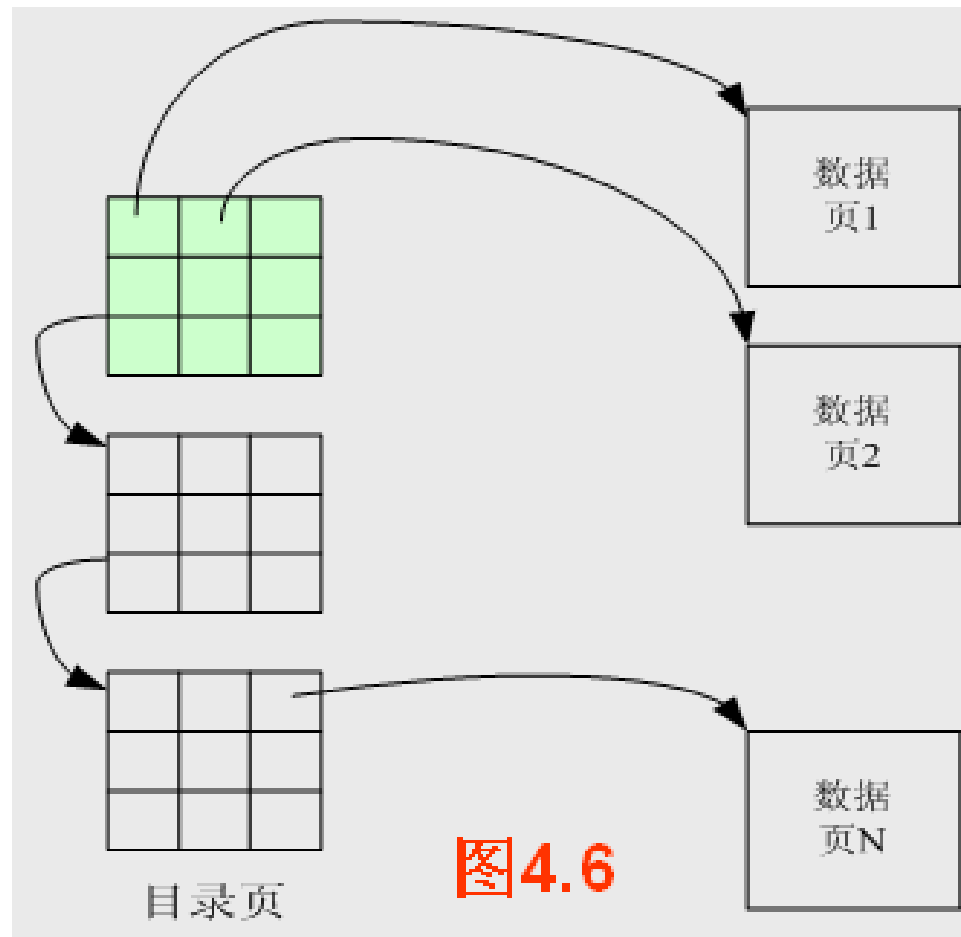
基于目录页的堆文件组织

❖ 组织结构

- 允许有多个目录页，不同的目录页通过指针链接在一起。
- 目录页中包含多个目录项，每个目录项标识一个页。


❖ 优点：

- 有利于更有效搜索足够容纳新记录的数据页。



4.3.2 排序文件

A-101	500		
B-131	800		
E-301	500		
G-515	2000		
H-201	1000		
J-305	750		
K-251	1100		



A-101	500		
B-131	800		
E-301	500		
G-515	2000		
H-201	1000		
J-305	750		
K-251	1100		

溢出页

F-555	750		

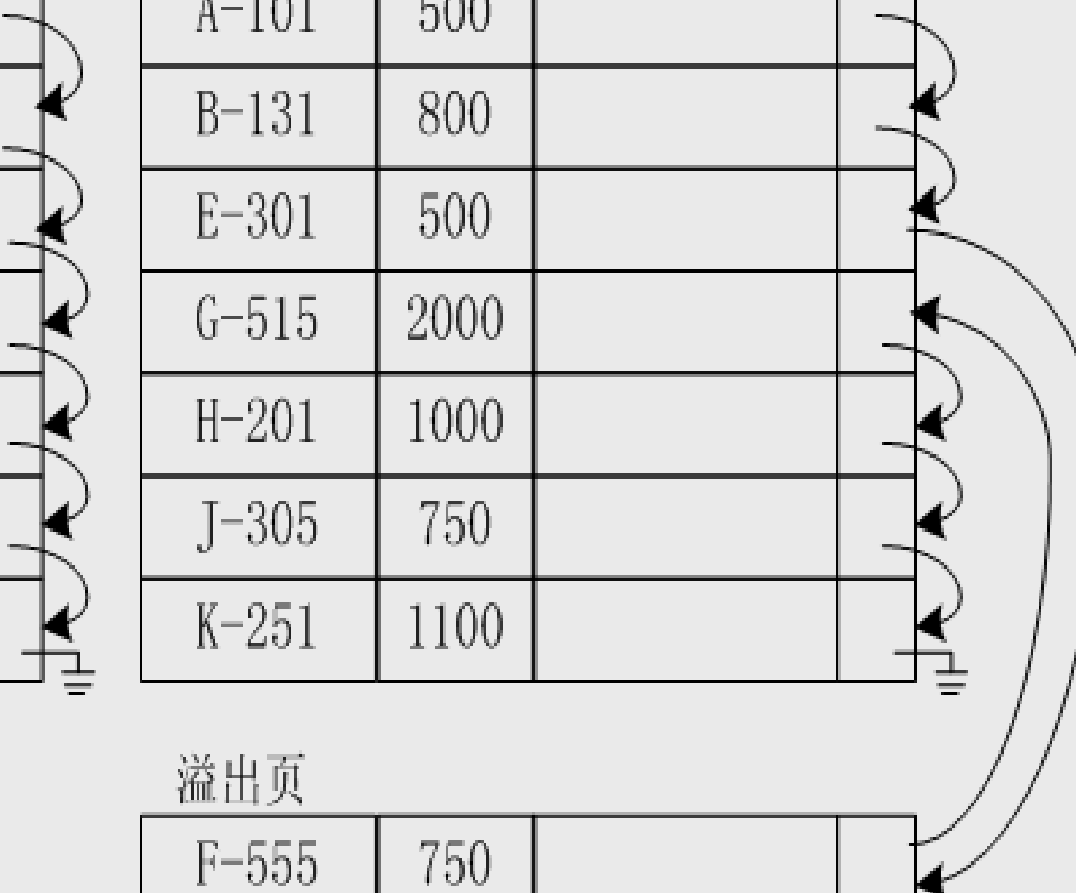


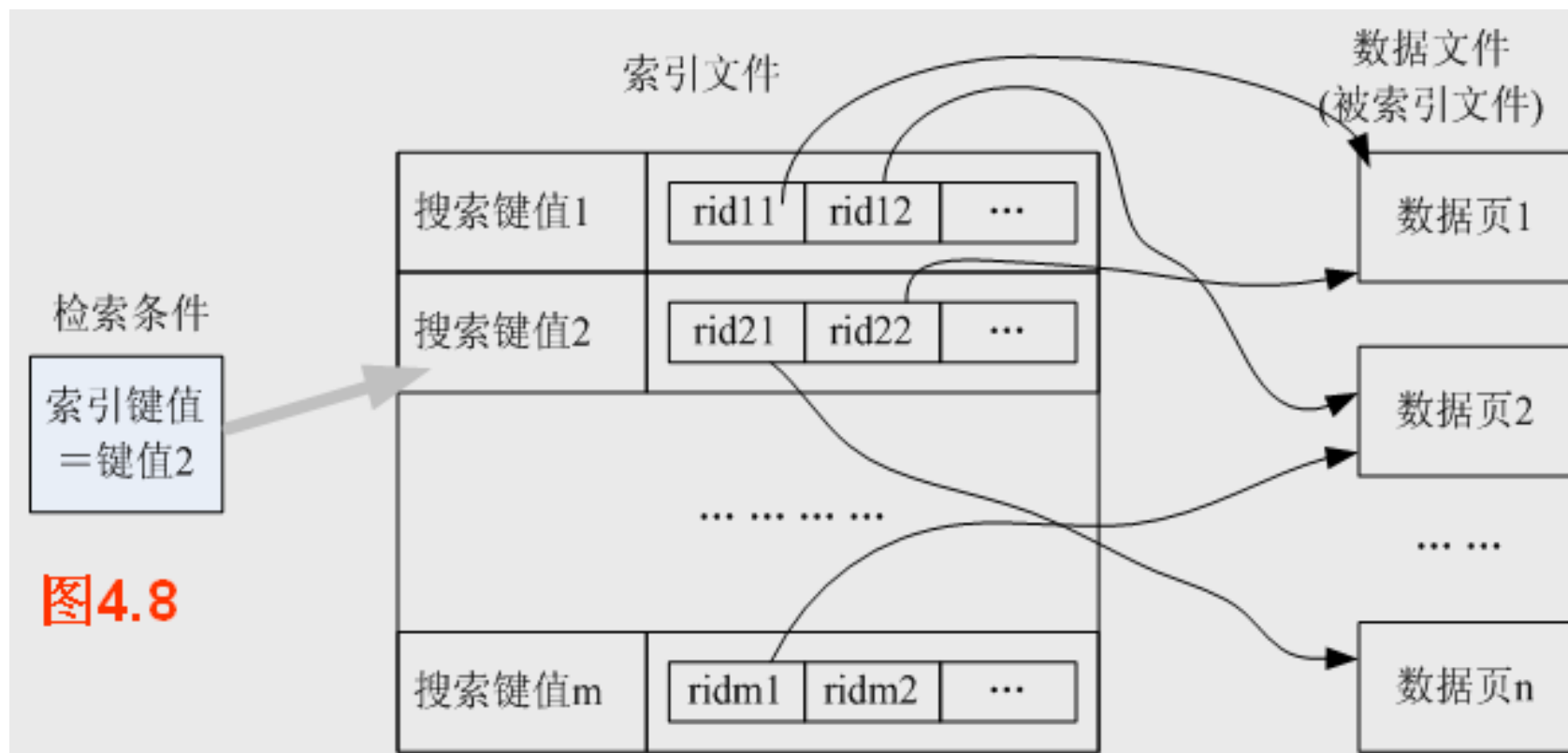
图4.7

(a) 一个简单的顺序文件示例

(b) 插入一条新记录后的指针调整变化

4.3.3 基于索引的文件组织

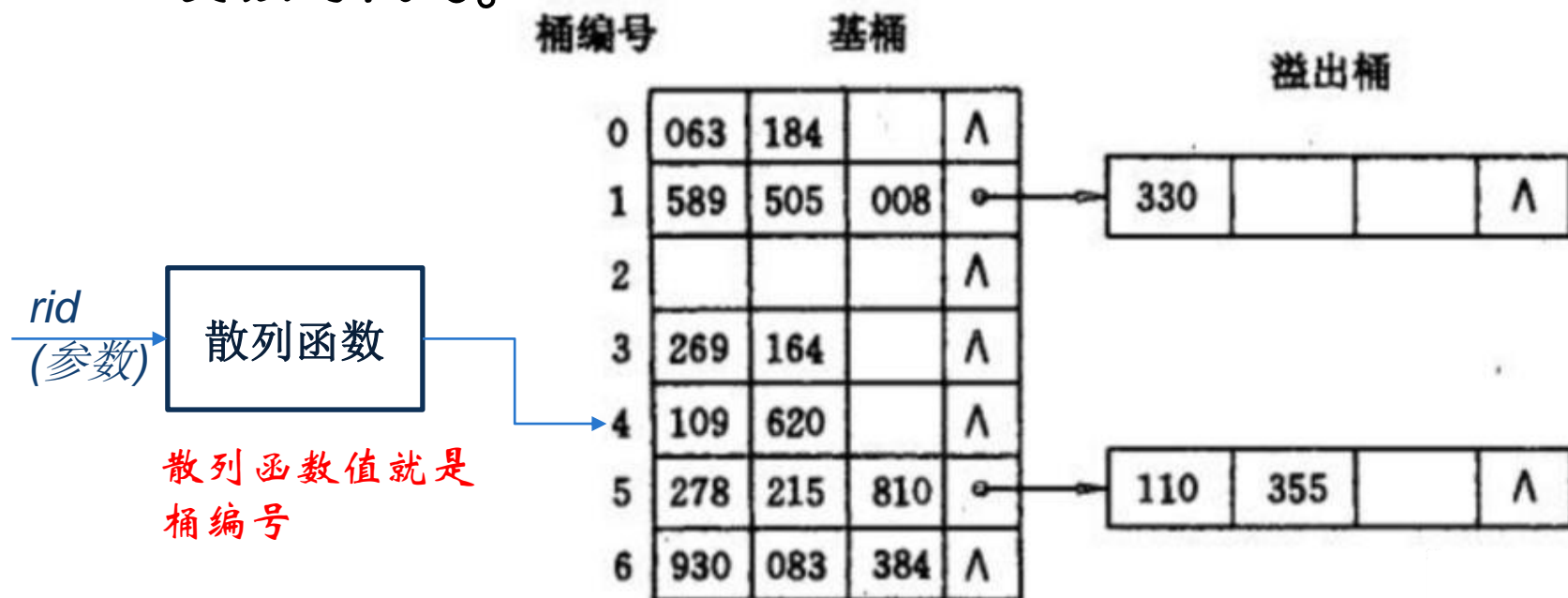
- ❖ 利用辅助索引文件来帮助定位数据记录。
 - 索引文件记录:索引项<搜索键值,rid或rid_list>



4.3.4 文件的散列组织方式

❖ 散列文件(hashed file)

- 以记录的某个属性值为参数，通过特定散列函数求得有限范围内的一个值作为记录的存储地址（即页地址或桶号）。
- 用于计算散列的属性也称为**散列键**，它与搜索键具有类似的概念。



★各种文件组织方式的特性分析

❖ 扫描(Scan)

❖ 等值搜索(Equality Search)

- 取文件中满足等值选择条件的所有记录
- 包含满足条件记录的所有页须从磁盘读到主存。

❖ 范围搜索(Ranging Search)

❖ 插入(insert)

- 必须先定位新记录应插入的页，并将该页读入主存，在主存页中完成插入修改，然后，再将该页写回磁盘。

❖ 删除(delete)

- 如采用等值或范围条件选择删除记录，则操作过程与‘插入/范围搜索’操作类似；
- 如直接给定删除记录rid，则可直接定位到记录所在页。

堆文件的操作特性分析

❖ 扫描 —— 操作代价为 **$B(D+RC)$**

❖ 等值搜索

- 假设：满足条件的记录只有一个，平均需检查一半的页
- 操作代价取 $0.5DB$

❖ 范围搜索 —— 必检查所有的页，操作代价 **$B(D+RC)$**

❖ 插入

- 取文件的最后页到主存，插入后，再写回磁盘
- 操作代价为 $2D+C$

❖ 删除

- 不考虑记录被删除后的空间合并
- 操作代价为：搜索时间 + D + C
- 若已知 rid ，可直接定位到目标页，可省去搜索时间

排序文件的操作特性分析

- ❖ 扫描 —— 操作代价为 $B(D+RC)$
- ❖ 等值搜索
 - 假设：满足条件的记录只有一个
 - 可用二分法搜索，操作代价取 $D \cdot \log_2 B + C \log_2 R$
 - 若满足条件记录有多个，则该代价还应加上读取包含所有这些记录的若干个连续页。
- ❖ 范围搜索 —— 等值搜索代价 + matches
- ❖ 插入
 - 插入后，需进行排序调整，假设需调整约一半的记录
 - 插入操作的代价 = 等值搜索代价 + $2 \cdot 0.5B(D+RC)$ 。
- ❖ 删除
 - 如果等值或范围删除条件，则代价与插入操作相同
 - 若已知rid，可直接定位到目标页，可省去搜索时间

散列文件的操作特性分析

❖ 扫描 —— 一页空间通常只保持约80%的占用率，扫描的实际操作代价取 $1.25B(D+RC)$

❖ 等值搜索

- 能非常有效支持等值选择

- 假设满足条件的记录只有一条且相应桶中没有溢出页，则等值搜索操作代价仅为 $D + 0.5RC$

❖ 范围搜索

- 需要扫描所有的页，操作代价 = $1.25B(D+RC)$

❖ 插入 —— 插入操作代价 = 等值搜索代价 + $D + C$ 。

❖ 删除

- 对等值选择删除，代价 = 搜索代价 + $D + C$

- 如果直接给定rid，则可省去搜索时间，代价 = $D + C$

各种文件组织方式的特性对比



文件类型	扫描	等值搜索	范围搜索	插入	删除
堆文件	BD	0.5BD	BD	2D	Search + D
排序文件	BD	$D * \log_2 B$	$D * \log_2 B + \text{matches}$	Search + BD	Search + BD
散列文件	1.25BD	D	1.25BD	2D	Search + D

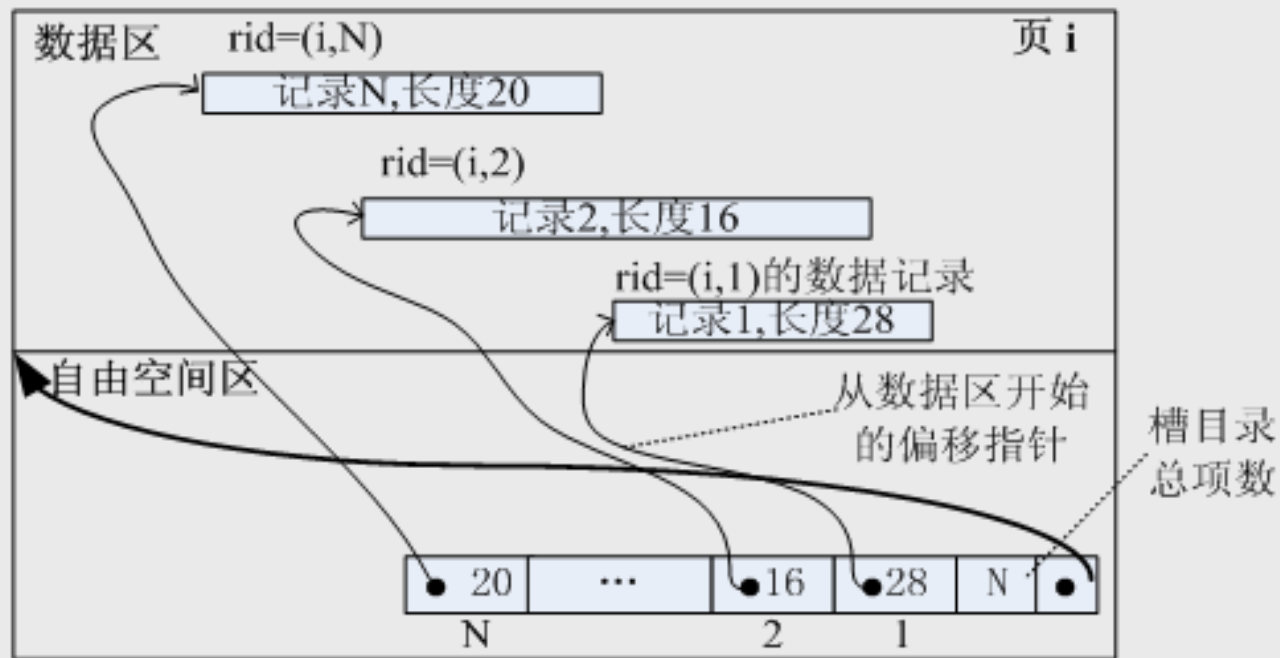
4.4 页的表示格式（即页的组织结构）

4.4.1 定长记录

4.4.2 变长记录

- ❖ 在处理与**I/O**有关主题时，通常采用页层次抽象已足够。
- ❖ 高层**DBMS**将其数据视为记录集。为提高某些特殊应用性能,系统也允许用户指定数据文件存储组织的某些选项参数
 - 这需要进一步了解页内记录的组织方式(即页格式)。
- ❖ 一般可将页视为槽的集合，每个槽可容纳一个记录。
 - 记录可通过使用rid: <page-id, slot-no>来标识定位。

基于分槽式页结构表示变长记录(图4.10)



(a) 变长记录的分槽目录页结构组织示意图



(b) 一种更规范、更常用的分槽目录页结构

4.5 记录表示格式

4.5.1 定长记录的字段表示

4.5.2 变长记录的字段表示

4.5.3 跨页记录管理技术

4.5.4 巨型字段/对象管理技术

4.5.5 指针记录管理技术——指针混写

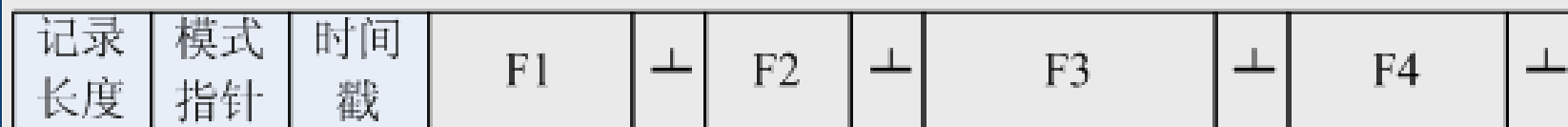
记录
首部
信息

- ◆ **DB**中记录除了存储各字段信息外，通常还有一个记录首部（记录头）。
- ◆ 记录头中存储记录层次的一般管理信息，包括记录长度、时间戳和指向记录模式描述的指针等。
- ◆ 记录是否变长主要看它是否含变长字段。
- ◆ 本节集中讨论记录中字段的表示问题。

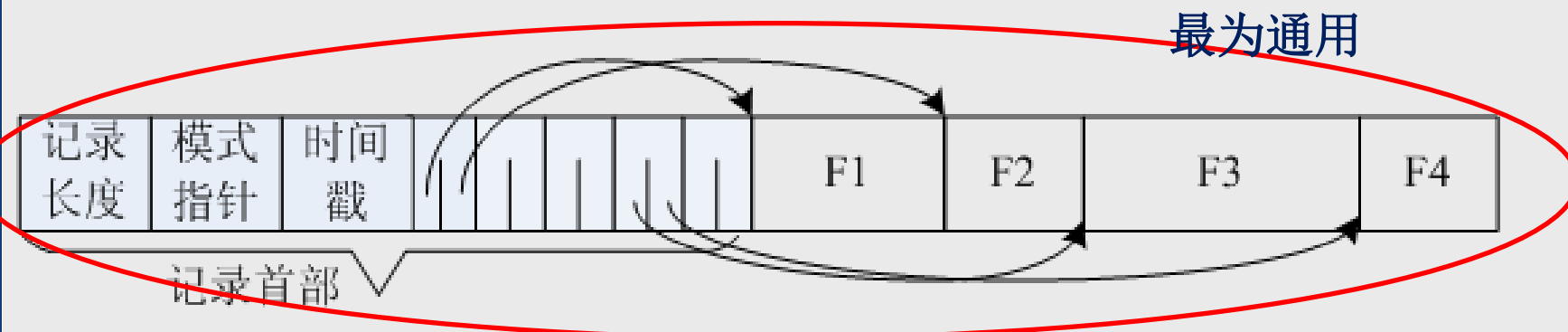
4.5 记录表示格式 (图4.11)



(a) 具有定长字段的记录组织



(b) 采用特殊字符结尾实现具有变长字段的记录组织



(c) 采用偏移数组实现具有变长字段的记录组织

4.5.3 跨页记录管理技术

❖ 跨页记录存在的原因至少有两个：

- 记录中存在大型或巨型字段；
 - 例如，一个多媒体对象可能占用几个MB的空间；一个视频序列，可能达几个GB。
 - 在RDB中，巨型字段或长字段，可使用BLOB等专门字段型来存储巨型对象。
- 出于节省存储空间的需要。虽然记录大小不超过1页，但为了利用页内零头空间，也会导致跨页记录。

❖ 跨页记录会被分割并分存到多个页中，故需要在各页中使用指针把它们链接在一起，形成单个记录的页链。

4.5.5 指针字段管理技术：指针混写（1）

- ❖ 指针或地址经常是记录的一部分，是特殊型字段。
- ❖ 当**DB**系统运行时，数据页会在主存和辅存之间移动，故指针所指向的目标页/记录，在特定时间，既可能在辅存，也可能在主存。相应地，指针或地址也就有两种形式：
 - 内存地址（物理地址）
 - 数据库地址----持久化指针。是一种在辅存中**DB**数据地址——通常是一个逻辑地址。
- ❖ **DB**系统通过在内存中“逻辑/物理地址 转换表”，管理这两类地址的有效转换。

4.5.5 指针字段管理技术：指针混写（2）

- ❖ 根据给定的指针或地址寻找目标对象的过程，称为解引用 (**dereference**)。
- ❖ 给定一个持久化指针，解引用一个对象需要额外的步骤：
 - 须通过“转换表”查找持久化指针所代表对象在内存中的实际位置。
 - 如对象不在内存，则要从磁盘读入，同时要修改转换表，并将存放该持久指针的内存单元，直接修改为目标对象的内存位置指针。
 - 下一次同一持久化指针再次被解引用时，就可以直接使用内存引用，从而可避免重复转换内存地址的过程开销。
 - 当对象被写回磁盘时，它所包含的任何被混写持久化指针必须执行反混写，
- ❖ 指针混写的时机选择
 - 自动混写；按需混写；不混写；程序控制

4.6 DB元信息及其存储管理

❖ 在**RDB**系统，除了关系，还需要维护关于整个**DB**的元描述数据，如关系的模式等。这类元信息称为数据字典(**data dictionary**)或系统目录(**system catalog**)。系统需存储的元信息类型有：

- 关系的模式（关系名、每个属性名字/类型/长度）。
- 在**DB**上定义的视图名字和视图定义。
- 完整性约束。
- 授权名、认证密码等关于用户帐户的信息。
- 当前关系实例的统计/描述数据。如每个关系中的元组总数，或各字段取值的统计直方图信息等描述信息。
-

❖ 实际上，所有这些信息组成了一个微型数据库

4.7 缓冲区管理



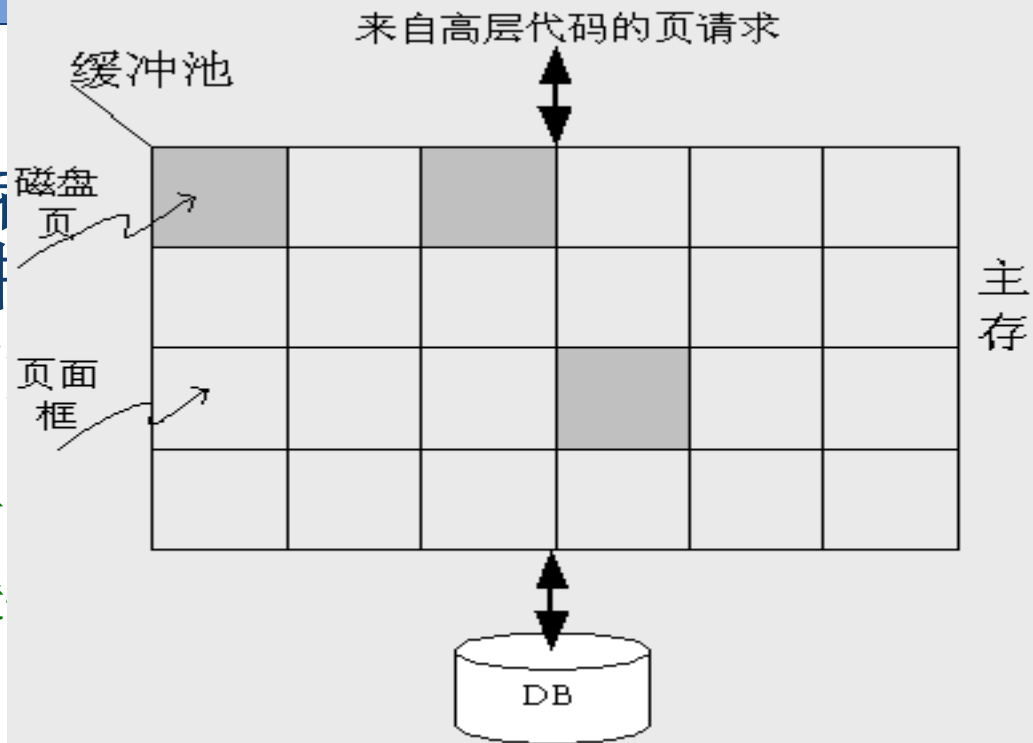
4.7.1 DB缓冲池与缓冲区管理器

4.7.2 缓冲区置换策略

4.7.3 DBMS与OS的缓冲区管理对比

❖ DB缓冲池

- DBMS系统一般都有专门的、称为**DB缓冲池**的：
 - 该主存区被按页框(frame)。
 - 为叙述方便，有‘缓存’、‘缓谓**DB缓冲池**。



❖缓冲区管理器

- 指DBMS中专门负责管理DB缓冲池的软件模块。

缓冲区置换策略

- 当新页请求发生且没有空闲缓冲页时，决定替换缓冲区哪些页的策略。

缓冲区管理器响应高层页请求的基本过程

- ❖ 检查缓冲池中是否存在该页，如不在，则进一步执行以下一些操作。
- ❖ 基于置换策略，选择一个可被置换的**frame**，将其的**pin_count**计数加1。
- ❖ 如果该**frame**中原先页被修改过(**dirty=1**)，则将原先页写回磁盘。
- ❖ 从磁盘读入新请求页到该**frame**中，同时置**dirty=0**。
- ❖ 返回包含请求页的**frame**地址给请求者。

几种常用的缓冲区置换策略简介



1. LRU(least recently used)

- 它选择最近最久未使用的页予以淘汰。

2. 时钟置换策略

3. FIFO (first in first out)

- FIFO总是选择最早进入主存的页作为下一个置换页，不能反映页面的使用情况，通常效果较差。

4. MRU (most recently used)

- MRU策略则刚好与LRU相反，选择最近刚被用过的页予以淘汰。