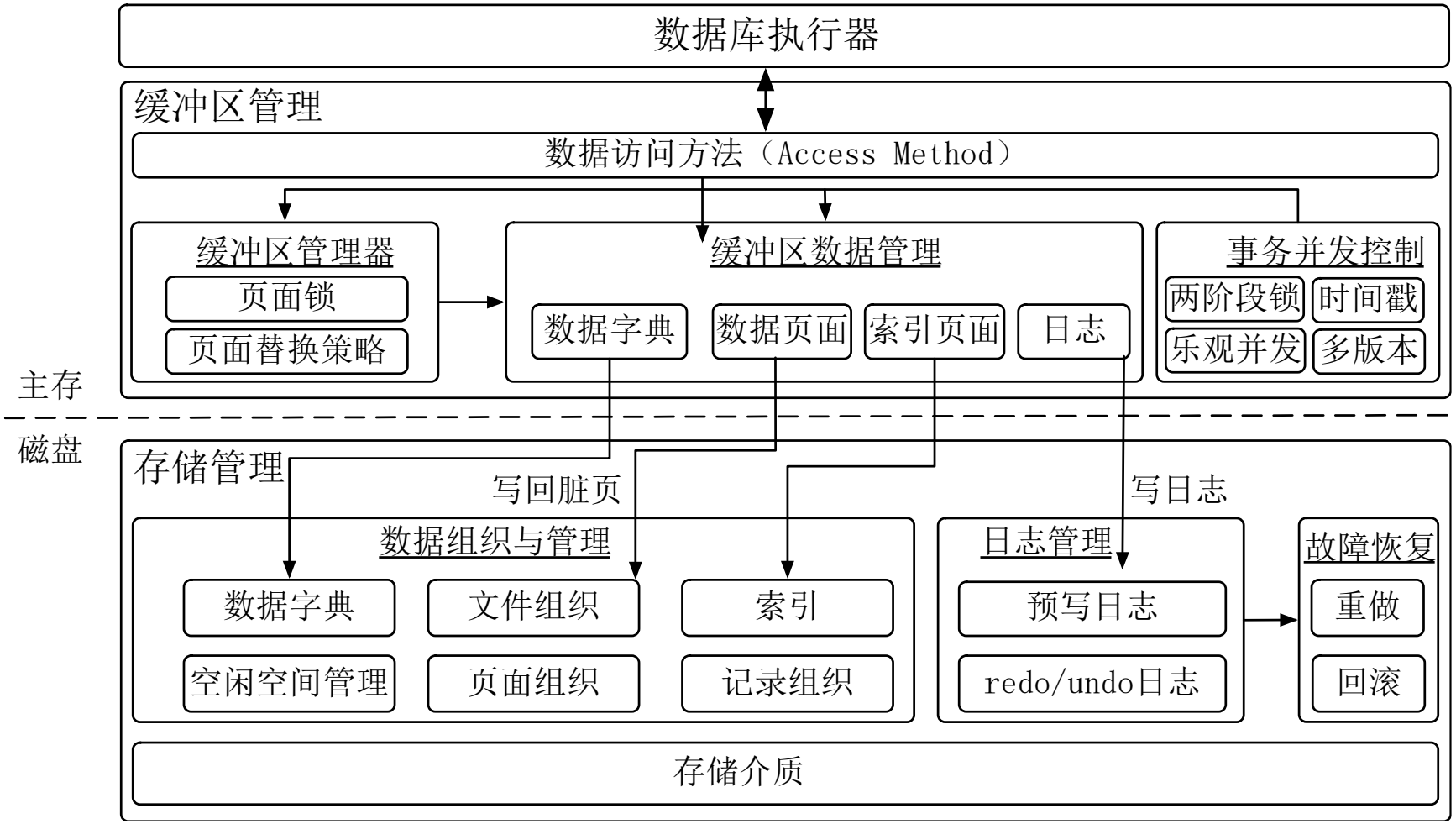
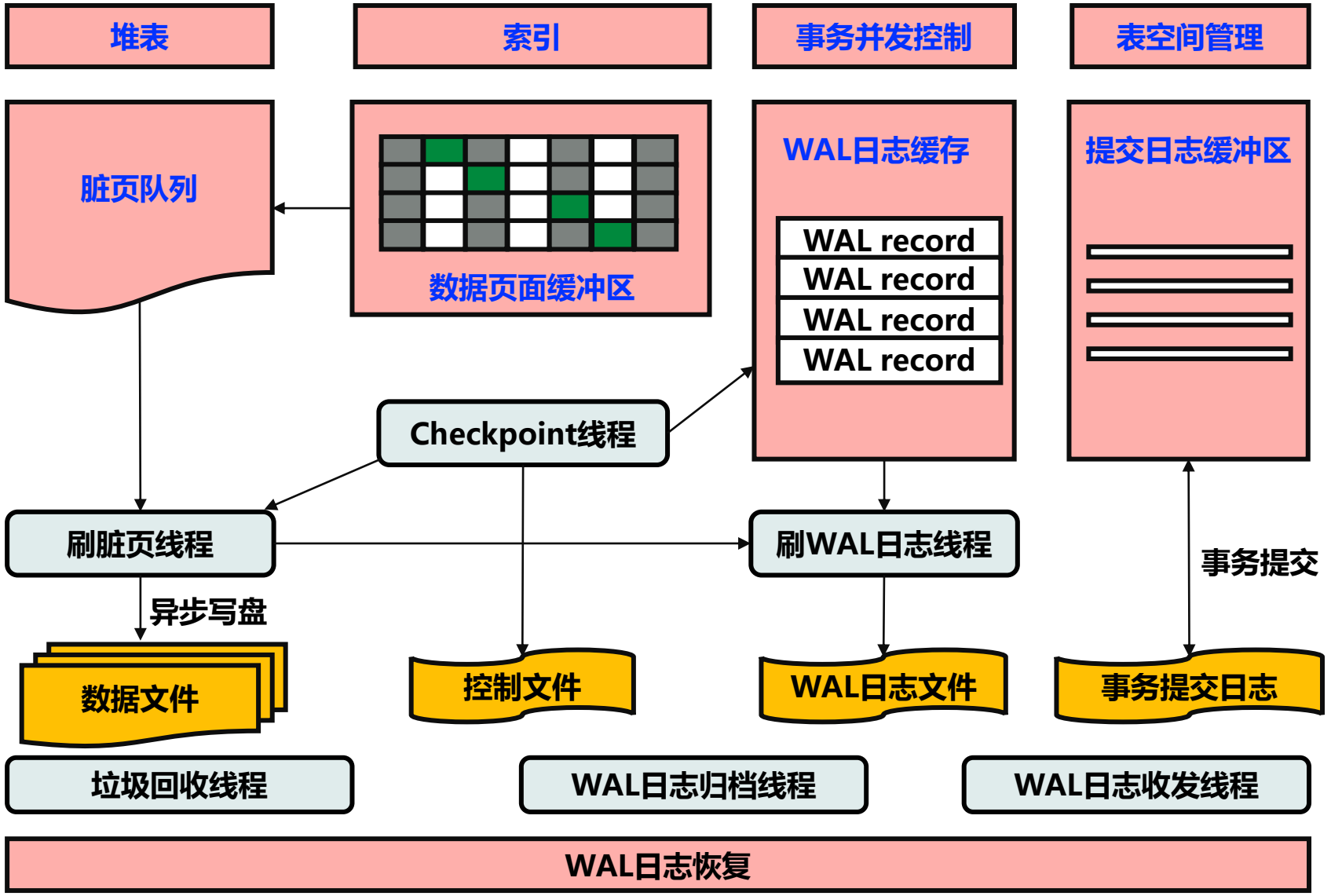
数据库存储

SQL引擎
执行器
存储引擎 **文件组织 页面组织 元数据 日志 索引 缓冲区 故障恢复 并发控制**
存储介质

存储概览


数据库执行器
缓冲区管理
数据访问方法（Access Method）
缓冲区管理器
页面锁
页面替换策略
缓冲区数据管理
数据字典
数据页面
索引页面
日志
事务并发控制
两阶段锁
时间戳
乐观并发
多版本
主存
磁盘
存储管理
写回脏页
写日志
数据组织与管理
数据字典
文件组织
索引
空闲空间管理
页面组织
记录组织
日志管理
预写日志
redo/undo日志
故障恢复
重做
回滚
存储介质

存储概览


堆表
索引
事务并发控制
表空间管理
脏页队列
数据页面缓冲区
WAL日志缓存
WAL record
WAL record
WAL record
WAL record
提交日志缓冲区
Checkpoint线程
刷脏页线程
刷WAL日志线程
事务提交
异步写盘
数据文件
控制文件
WAL日志文件
事务提交日志
垃圾回收线程
WAL日志归档线程
WAL日志收发线程
WAL日志恢复

数据组织

记录

2021310001	张三	男	计算机系	3.9
Char	Varchar	Char	Varchar	Decimal

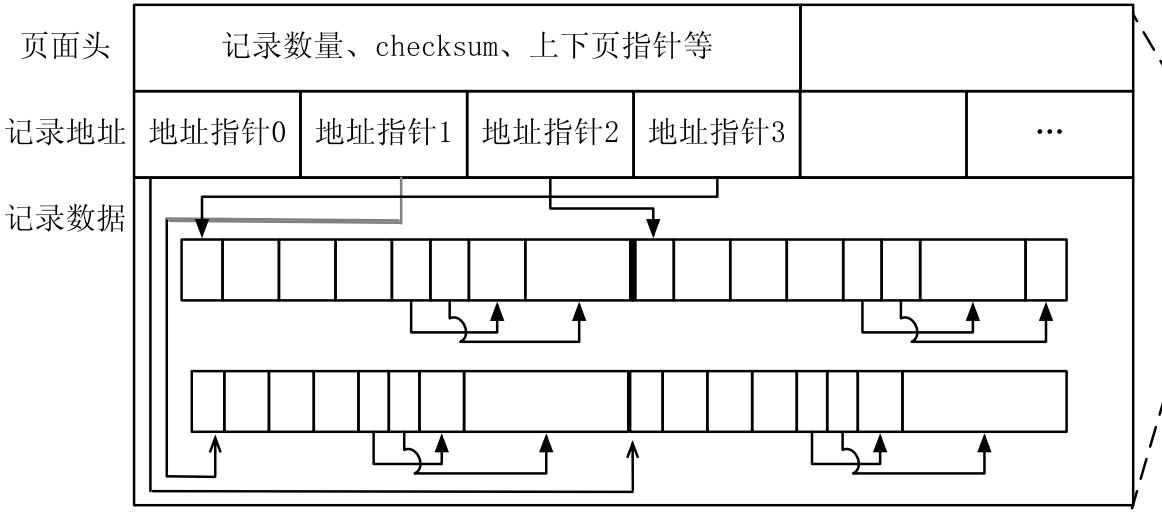
记录的字节表示

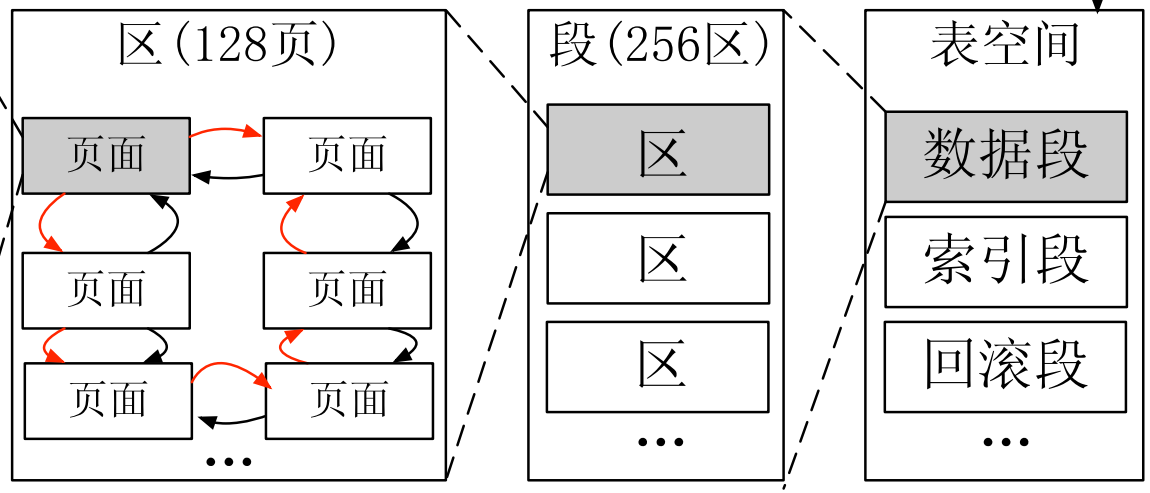
	学号	性别	绩点	姓名	院系	空值向量(00000)$_2$		
记录头	2021310001	男	3.9	37, 6	43, 12	0	张三	计算机系
字节号 0	10	20	24	28	32	36 37	43	54

数据表

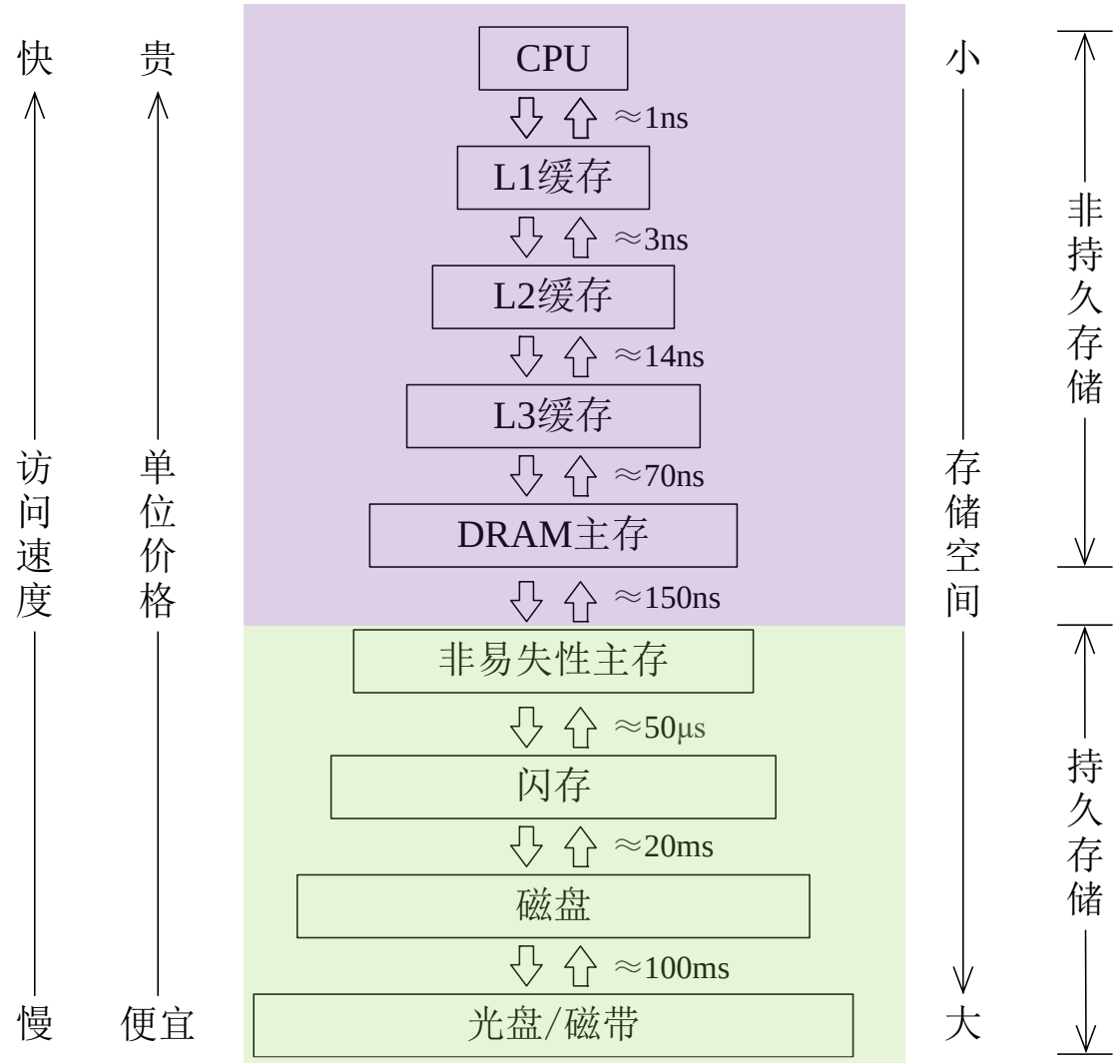
	学号	姓名	性别	院系	绩点
记录0	2021310001	张三	男	计算机系	3.9
记录1	2021310002	张四	女	计算机系	3.7
记录2	2021310003	张五	男	计算机系	4.0
记录3	2021310004	张六	男	法学系	3.7
记录4	2021310006	张七	女	物理系	3.4
记录5	2021310009	张八	女	物理系	3.6
记录6	2021310013	张九	女	法学系	3.7

分槽页面 (8K)





存储介质


快
贵
访问速度
单位价格
慢
便宜
CPU
≈1ns
L1缓存
≈3ns
L2缓存
≈14ns
L3缓存
≈70ns
DRAM主存
≈150ns
非易失性主存
≈50μs
闪存
≈20ms
磁盘
≈100ms
光盘/磁带
小
存储空间
大
非持久存储
持久存储

存储介质介绍

高速缓冲存储器

存取速度最快的存储介质，而单位价格却比较昂贵

DRAM主存(内存)

主要用于存放程序和其处理的数据。一般为4GB～32GB（2022年）

磁盘

容量大、数据持久性好，存储长期数据。一般为256GB～10TB（2022年）

闪存

又叫固态硬盘，个人计算机闪存大小一般为128GB～1TB（2022年）

非易失性主存

速度能DRAM相媲美，字节级寻址，在断电时不会丢失数据

存储介质	访问速度	存储空间	单位价格	数据持久性
高速缓存	快	小	高	易失
DRAM	较快	中等	中等	易失
非易失性主存	中等	中等	较高	非易失
闪存	较慢	较大	较低	非易失
磁盘	慢	大	低	非易失

磁盘结构

盘片

覆盖磁性物质，通过改变磁性物质的磁场方向来存储二进制的0和1，高速旋转
每分钟5400转至7200转

磁头

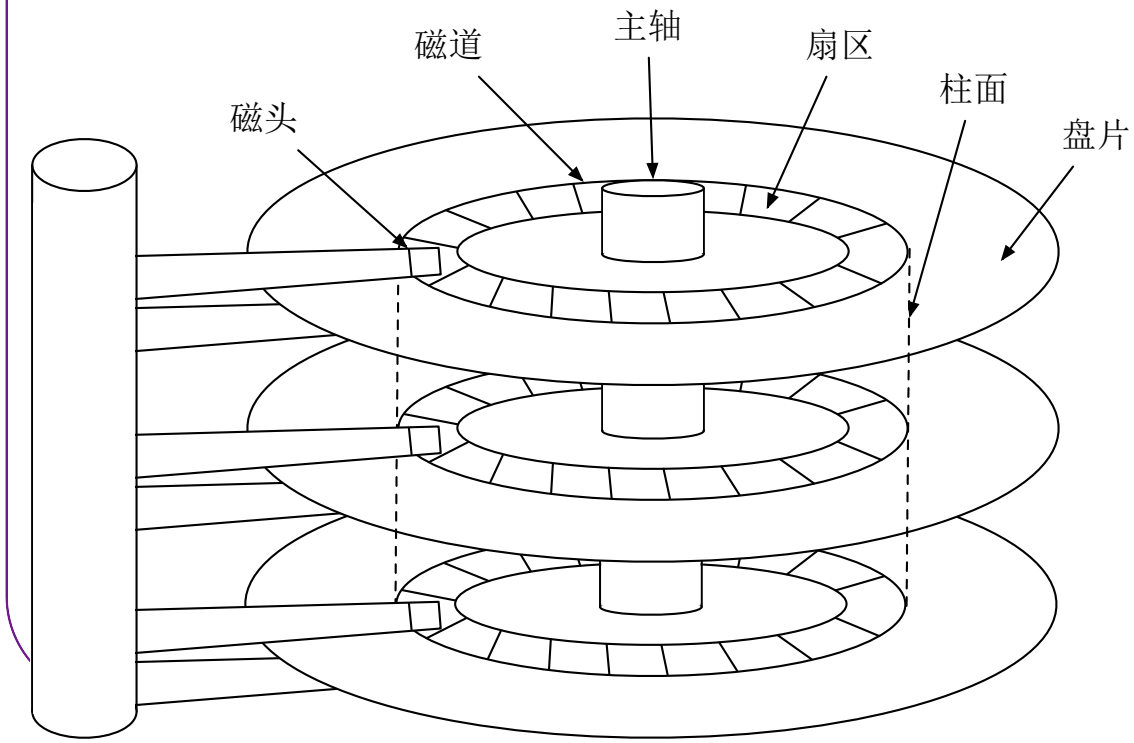
改变磁性物质状态并且读写数据

磁道

盘片上的圆环区域

柱面

纵向对齐的多个磁道



磁盘性能与优化

访问时间

系统发出读写指令后到磁盘开始返回数据的时间

数据传输率

每秒钟磁盘读写的数据量

磁盘访问优化技术

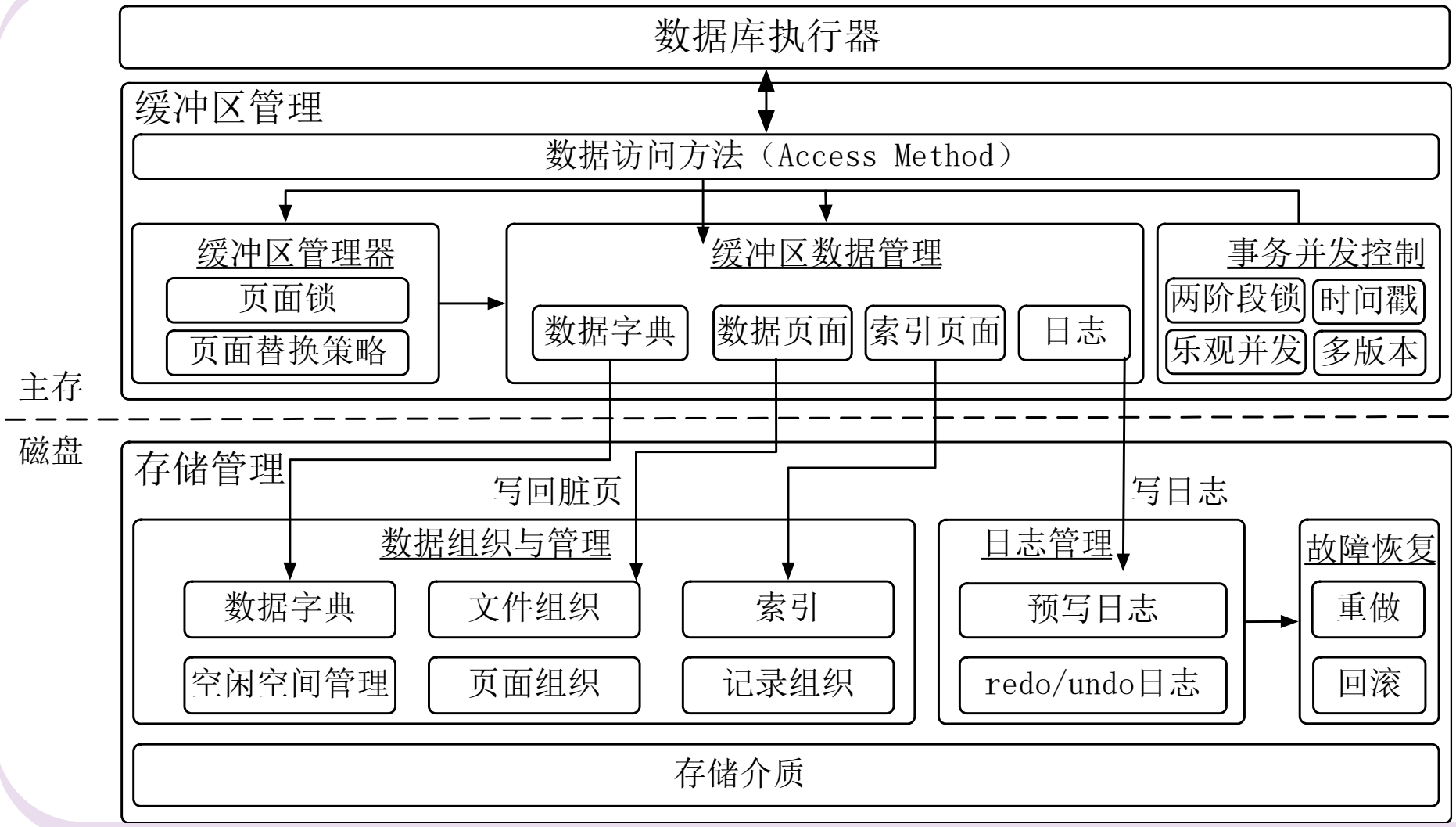
缓冲

预读

调度

文件组织

存储结构


数据库执行器
缓冲区管理
数据访问方法（Access Method）
缓冲区管理器
页面锁
页面替换策略
缓冲区数据管理
数据字典
数据页面
索引页面
日志
事务并发控制
两阶段锁
时间戳
乐观并发
多版本
主存
磁盘
存储管理
写回脏页
写日志
数据组织与管理
数据字典
文件组织
索引
空闲空间管理
页面组织
记录组织
日志管理
预写日志
redo/undo日志
故障恢复
重做
回滚
存储介质

存储结构

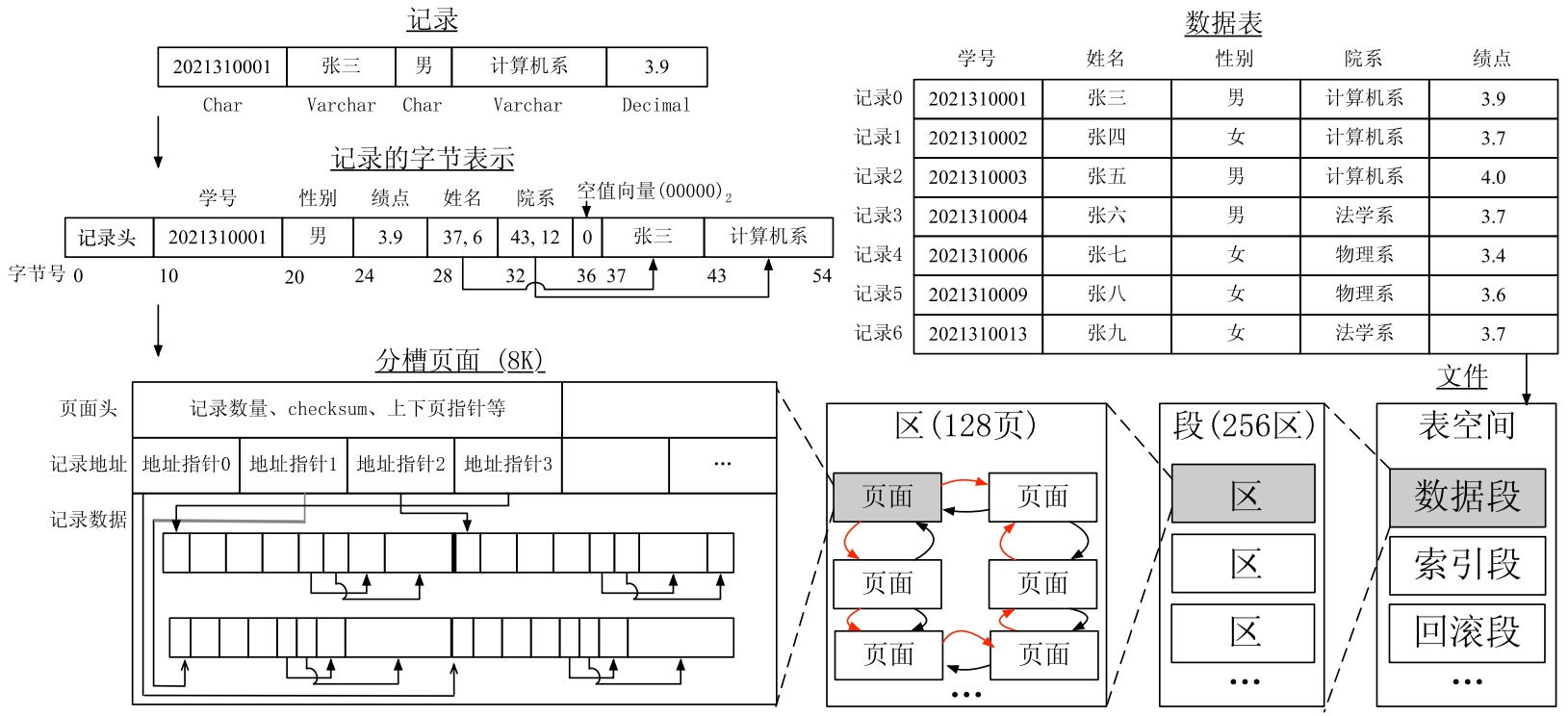
数据块

存储介质上的数据最小存储单位

页面

统一大小来管理数据

一般为8KB、16KB

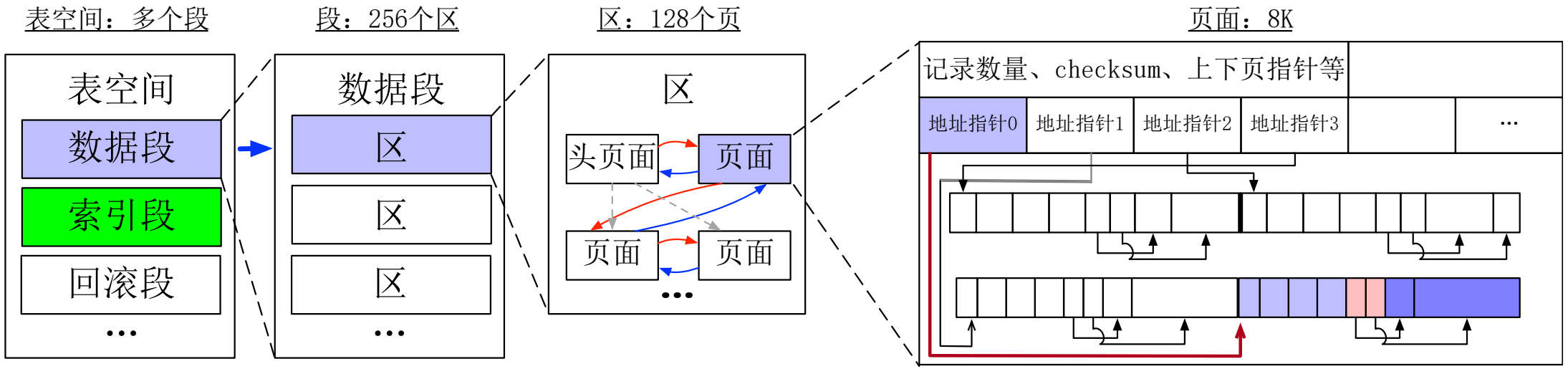


数据表空间管理

数据表空间由段（segment）、区（extent）、页（page）组成。

数据段、索引段、回滚段分别存储

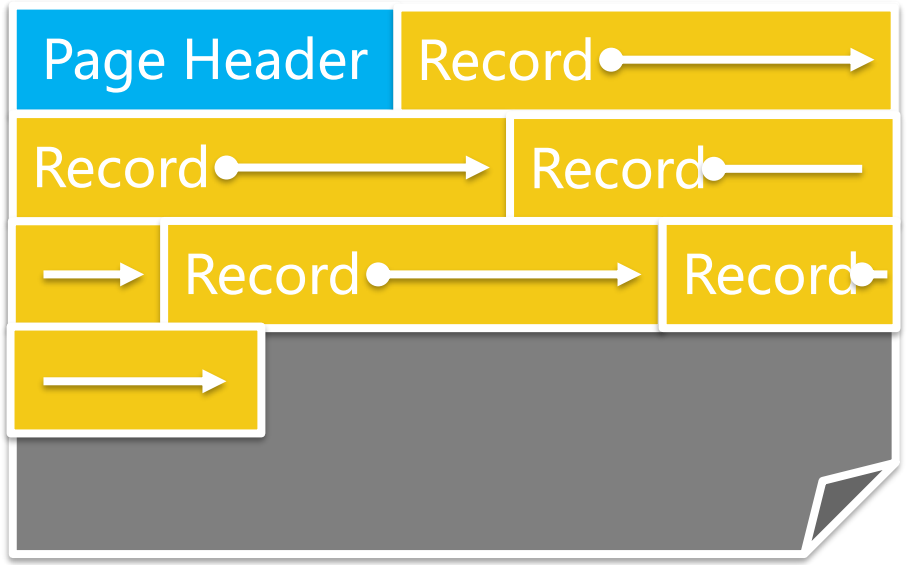
一次申请一个区（多个连续页），让相邻页的物理位置也相邻，方便实现顺序I/O。



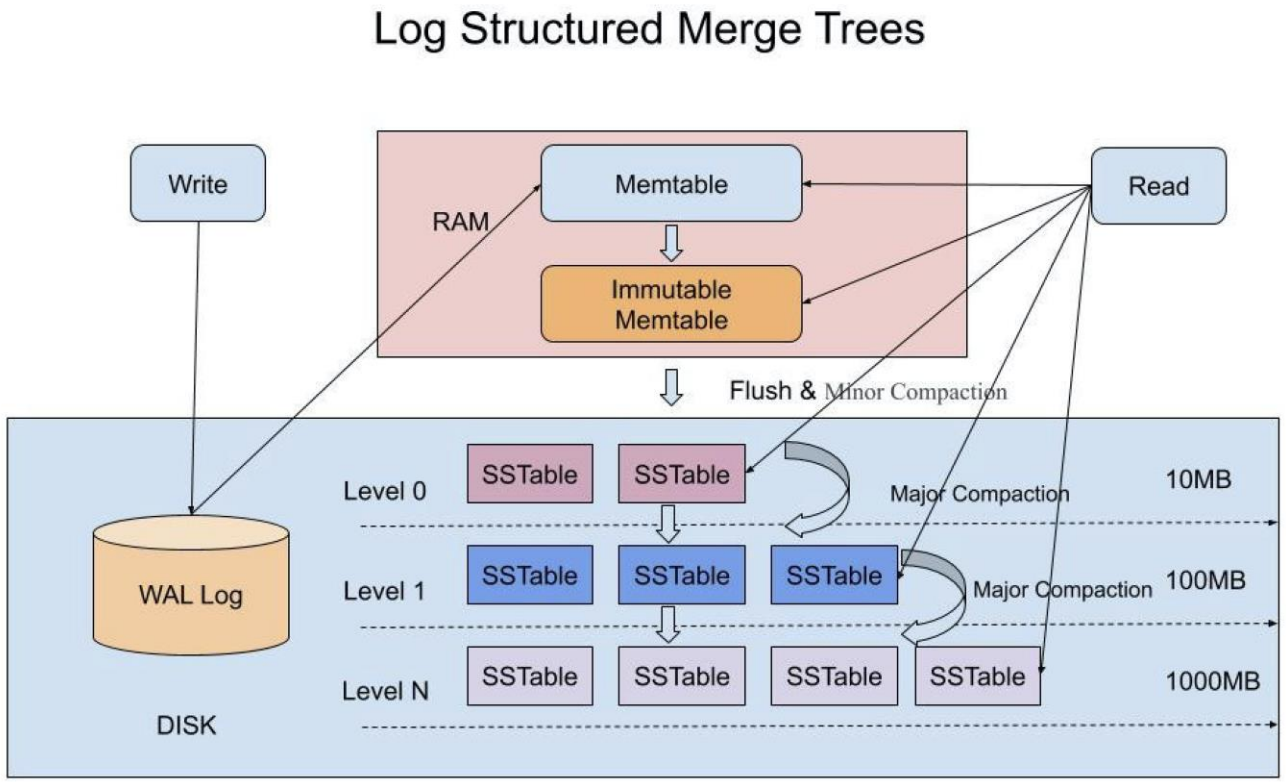
页面组织

元组管理

页面头	记录数量、checksum、上下页指针等				记录删除状态位向量 $(11011\cdots)_2$
记录地址	地址指针0	地址指针1	地址指针2	地址指针3	地址指针4	…
记录数据	记录0	记录1	记录2	记录3	记录4
	记录5	记录6	记录7	记录8	记录9
	记录10	记录11	记录12	记录13	记录14
	…				



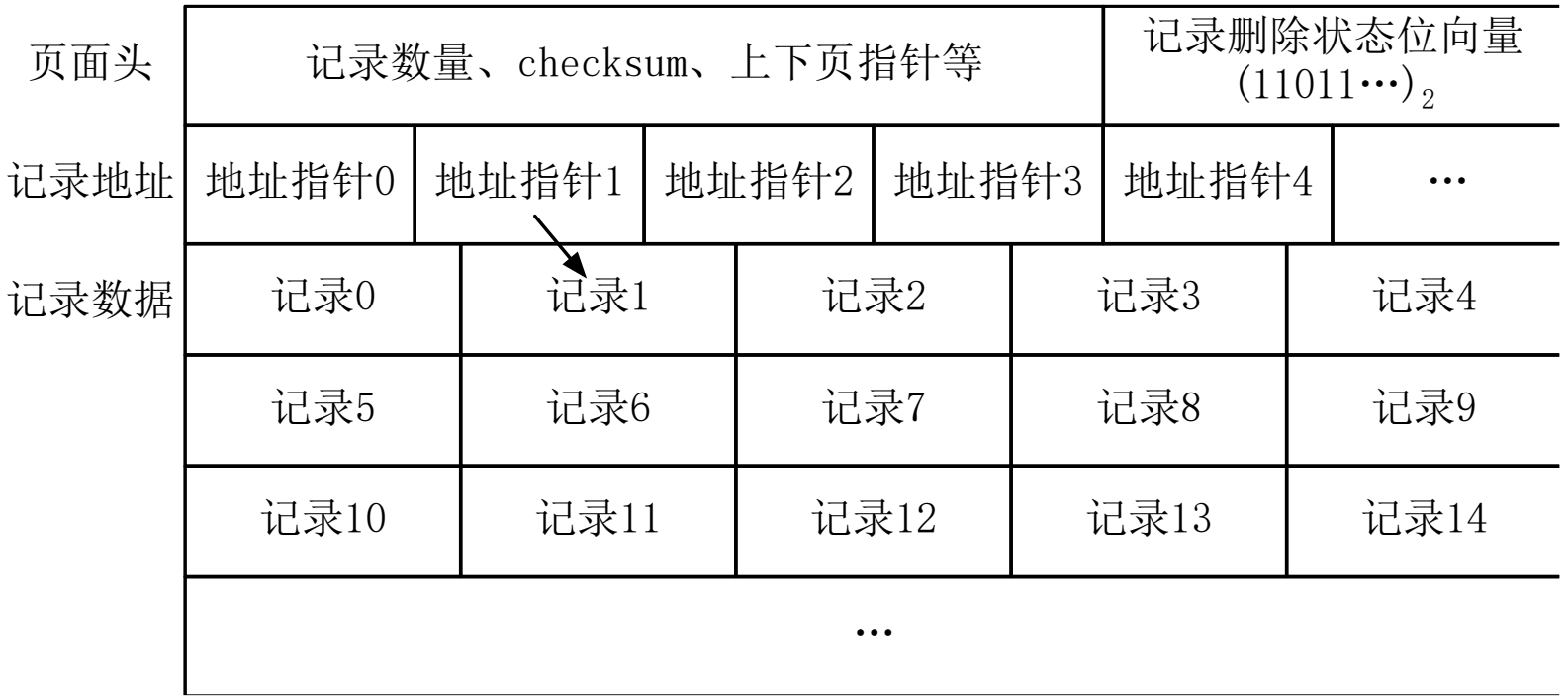
追加日志管理



页面组织

页面大小为8KB或者16KB，一般为2的整数次幂

一个页面往往包含了多条记录



如果页面可以容纳新纪录，添加到页面最后

记录添加

	学号	姓名	性别	院系	绩点
记录0	2021310001	张三	男	计算机系	3.9
记录1	2021310002	张四	女	计算机系	3.7
记录2	2021310003	张五	男	计算机系	4.0
记录3	2021310004	张六	男	法学系	3.7
记录4	2021310006	张七	女	物理系	3.4
记录5	2021310009	张八	女	物理系	3.6

	学号	姓名	性别	院系	绩点
记录0	2021310001	张三	男	计算机系	3.9
记录1	2021310002	张四	女	计算机系	3.7
记录2	2021310003	张五	男	计算机系	4.0
记录3	2021310004	张六	男	法学系	3.7
记录4	2021310006	张七	女	物理系	3.4
记录5	2021310009	张八	女	物理系	3.6
记录6	2021310013	张九	女	法学系	3.7

删除记录，并依次将后续记录向前移动

记录删除

	学号	姓名	性别	院系	绩点
记录0	2021310001	张三	男	计算机系	3.9
记录1	2021310002	张四	女	计算机系	3.7
记录2	2021310003	张五	男	计算机系	4.0
记录3	2021310004	张六	男	法学系	3.7
记录4	2021310006	张七	女	物理系	3.4
记录5	2021310009	张八	女	物理系	3.6
记录6	2021310013	张九	女	法学系	3.7

	学号	姓名	性别	院系	绩点
记录0	2021310001	张三	男	计算机系	3.9
记录1	2021310002	张四	女	计算机系	3.7
记录3	2021310004	张六	男	法学系	3.7
记录4	2021310006	张七	女	物理系	3.4
记录5	2021310009	张八	女	物理系	3.6
记录6	2021310013	张九	女	法学系	3.7

将最后一条记录向前移动

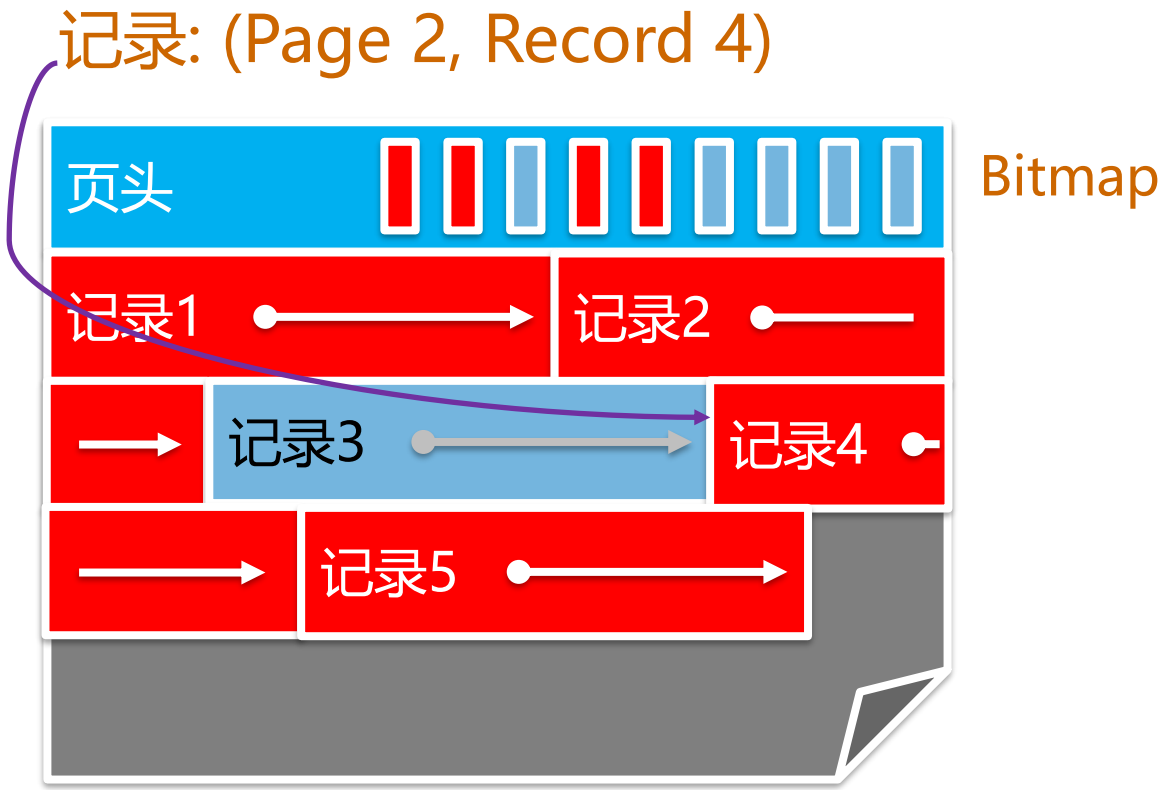
记录删除

	学号	姓名	性别	院系	绩点
记录0	2021310001	张三	男	计算机系	3.9
记录1	2021310002	张四	女	计算机系	3.7
记录2	2021310003	张五	男	计算机系	4.0
记录3	2021310004	张六	男	法学系	3.7
记录4	2021310006	张七	女	物理系	3.4
记录5	2021310009	张八	女	物理系	3.6
记录6	2021310013	张九	女	法学系	3.7

	学号	姓名	性别	院系	绩点
记录0	2021310001	张三	男	计算机系	3.9
记录1	2021310002	张四	女	计算机系	3.7
记录6	2021310013	张九	女	法学系	3.7
记录3	2021310004	张六	男	法学系	3.7
记录4	2021310006	张七	女	物理系	3.4
记录5	2021310009	张八	女	物理系	3.6

定长记录（bitmap）

- Bitmap位图：标记记录是否存在
- **Insert插入：**找到第一个空槽
- **Delete删除：**清楚槽标志位
- Bitmap数目根据页面大小和记录长度计算



文件组织

数据文件中页面之间的组织方式

目标：高效的数据页面访问（插入、删除、查找）

主要的文件组织方法

堆表

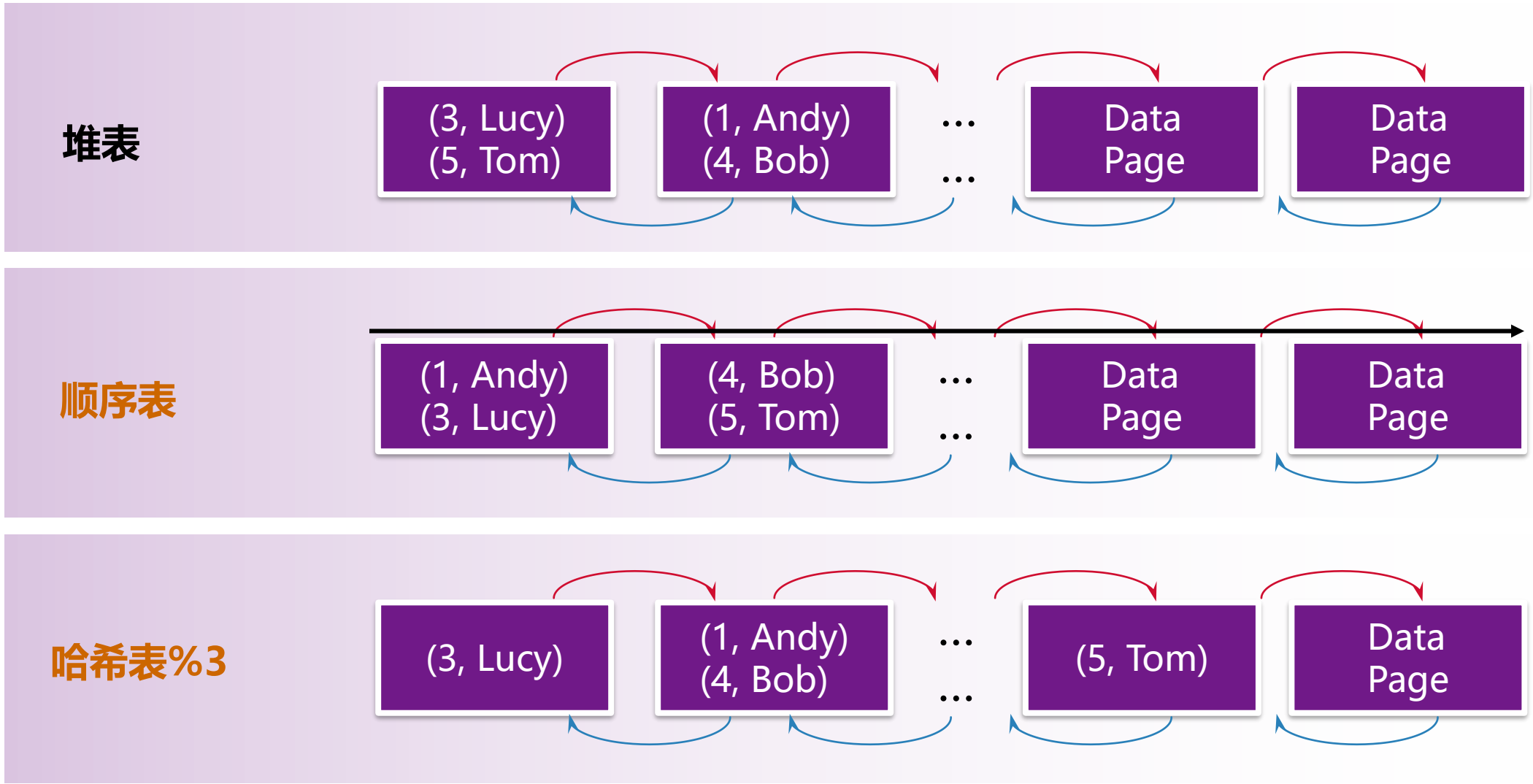
顺序表

哈希表

B+树

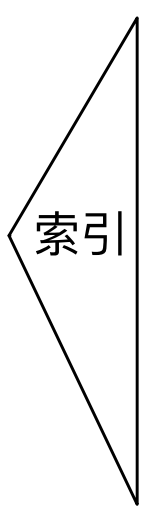
多表聚簇

文件组织


堆表
(3, Lucy)
(5, Tom)
(1, Andy)
(4, Bob)
Data Page
Data Page
顺序表
(1, Andy)
(3, Lucy)
(4, Bob)
(5, Tom)
Data Page
Data Page
哈希表%3
(3, Lucy)
(1, Andy)
(4, Bob)
(5, Tom)
Data Page

堆表

记录的顺序没有限制，将记录简单排列在文件中



	页面0
	页面1
	页面2
	页面3

	学号	姓名	性别	院系	绩点
记录0	2021310001	张三	男	计算机系	3.9
记录1	2021310002	张四	女	计算机系	3.7
记录2	2021310003	张五	男	计算机系	4.0
记录3	2021310004	张六	男	法学系	3.7
记录4	2021310006	张七	女	物理系	3.4
记录5	2021310009	张八	女	物理系	3.6
记录6	2021310013	张九	女	法学系	3.7

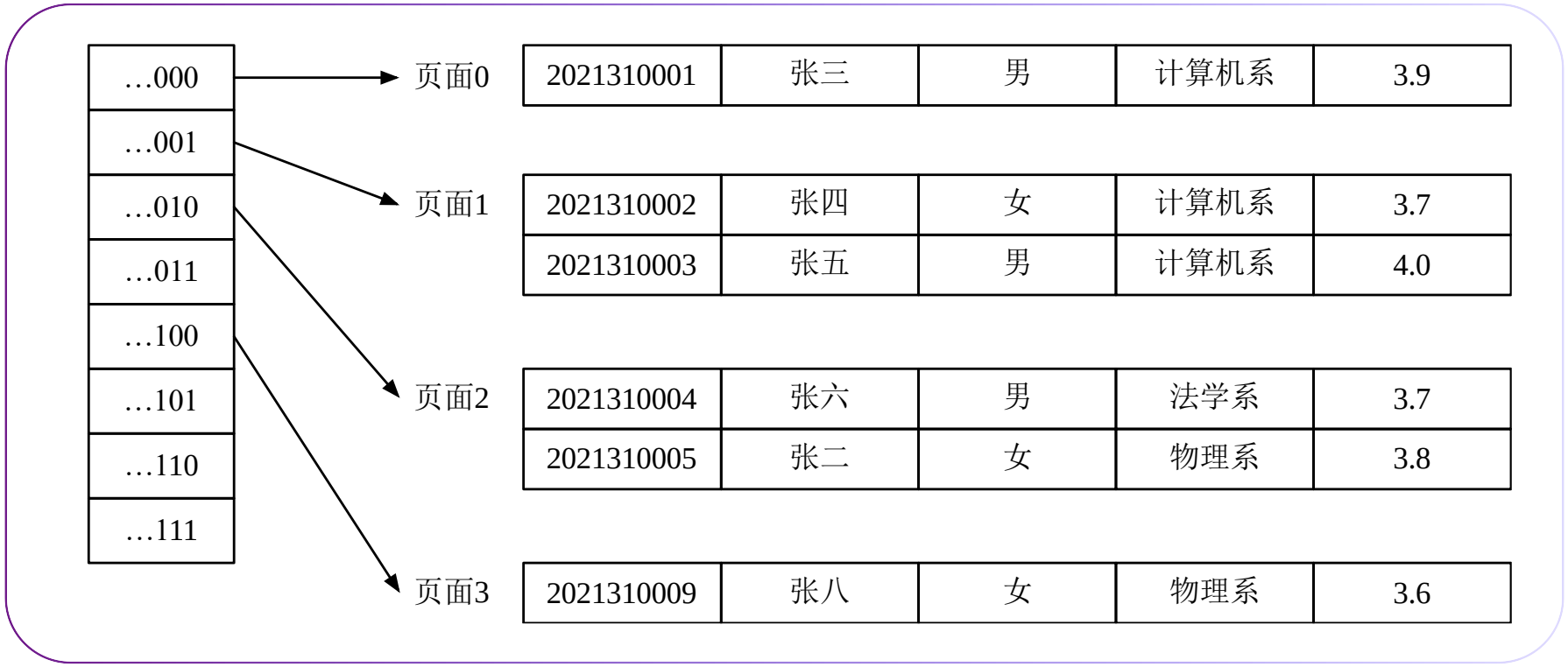
顺序表

记录按某个或某些字段的大小顺序存储

页面0	学号	姓名	性别	院系	绩点
记录0	2021310001	张三	男	计算机系	3.9
记录1	2021310002	张四	女	计算机系	3.7
记录2	2021310003	张五	男	计算机系	4.0
记录3	2021310004	张六	男	法学系	3.7
记录4	2021310006	张七	女	物理系	3.4
记录5	2021310009	张八	女	物理系	3.6
记录6	2021310013	张九	女	法学系	3.7

哈希表

使用哈希表将记录存储到不同的页面或者不同的页面集合



B^+树

在实际的文件系统中，是使用B_树的一种变体，称为m阶B^+树。 它与B_树的主要不同是叶子结点中存储记录。在B^+树中，所有的非叶子结点可以看成是索引，而其中的关键字是作为“分界关键字”，用来界定某一关键字的记录所在的子树。一棵m阶B_树，或者是空树，或者是满足以下性质的m叉树：

(1) 根结点或者是叶子，或者至少有两棵子树，至多有m棵子树；

(2) 除根结点外，所有非终端结点至少有$\lceil m/2 \rceil$棵子树，至多有m棵子树；

(3) 所有叶子结点都在树的同一层上；

(4) 每个结点应包含如下信息：

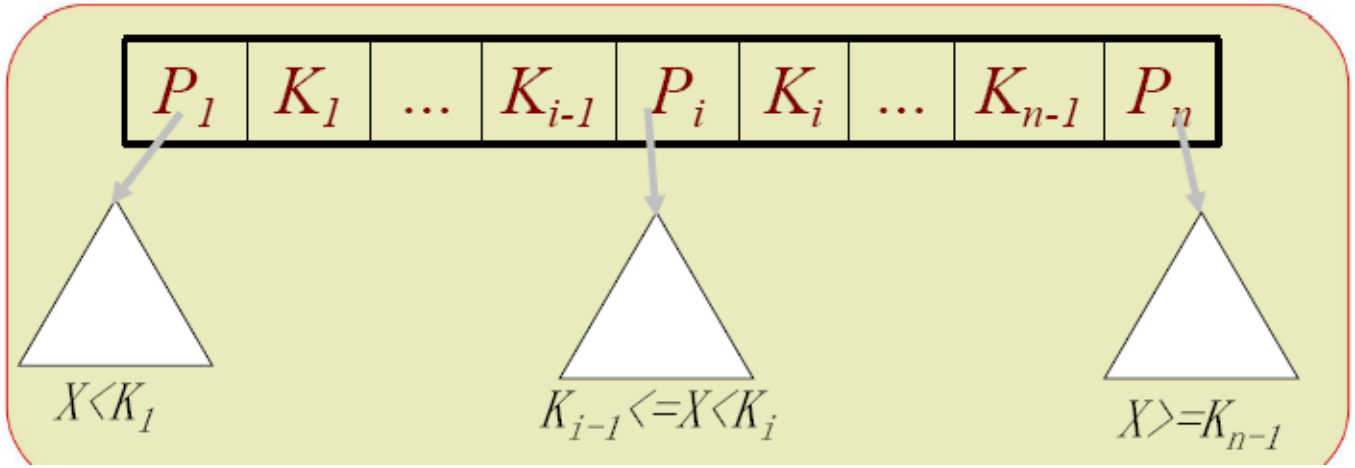
$(n，A_0，K_1，A_1，K_2，A_2，\cdots，K_n，A_n)$

其中$K_i(1\le i\le n)$是关键字，且$K_i<K_{i+1}$ $(1\le i\le n-1)$；$A_i(i=0，1，\cdots，n)$为指向孩子结点的指针，且A_{i-1}所指向的子树中所有结点的关键字都小于K_i，A_i所指向的子树中所有结点的关键字都大于K_i；n是结点中关键字的个数，且$\lfloor m/2\rfloor-1\le n\le m-1$，n+1为子树的棵数。

当然，在实际应用中每个结点中还应包含n个指向每个关键字的记录指针

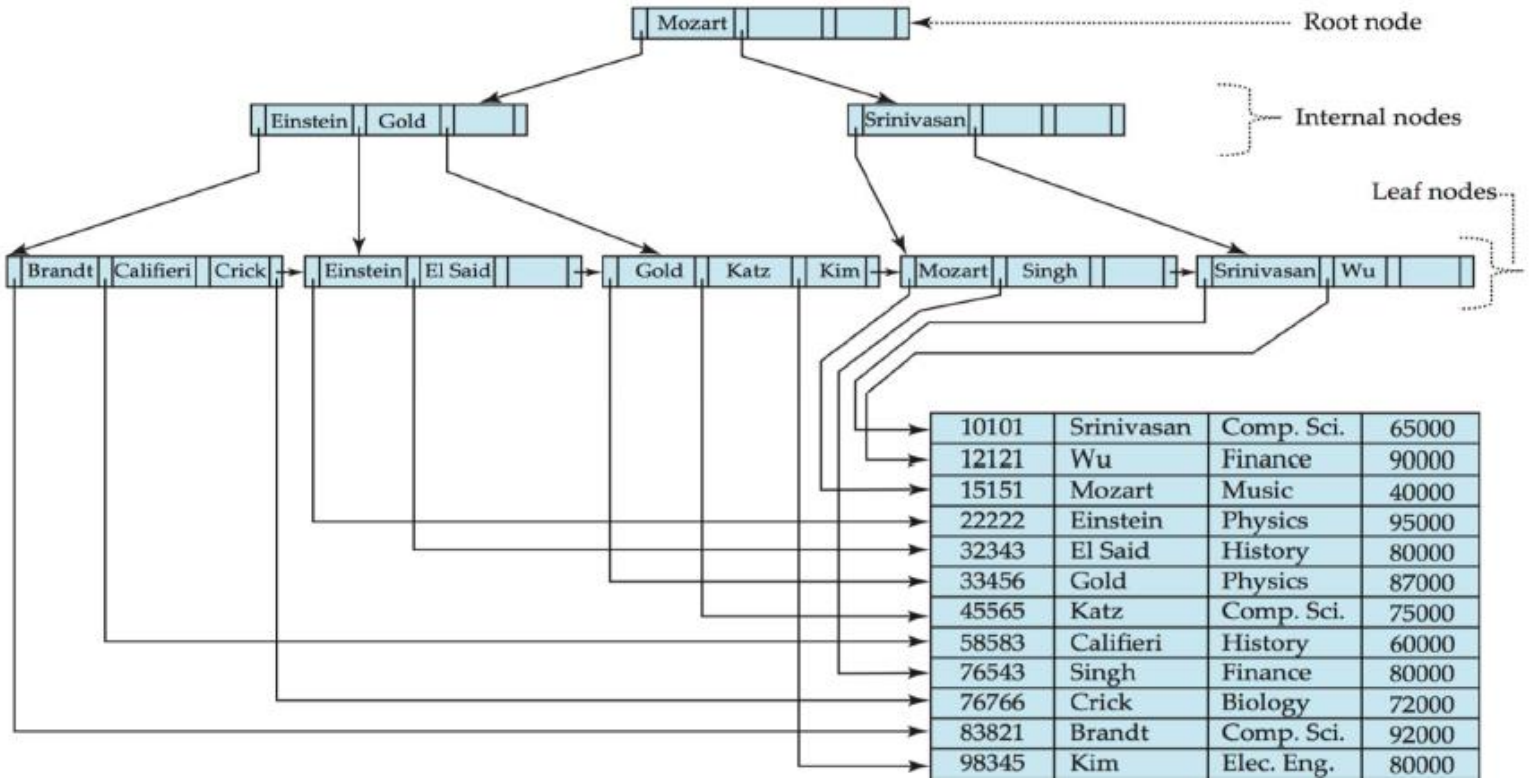
B+-树索引结构

- B+-树索引是一种多级索引，采用平衡树的结构
  - 数据插入和删除的情况下仍能保持执行效率
- 典型的B+-树结点结构如下，它最多包含n-1个搜索码值$K_1,K_2,\cdots,K_{n-1}$，以及n个指针$P_1,P_2,\cdots,P_n$。每个结点中的搜索码值排序存放
  - 数据如果i<j，那么$K_i<K_j$



B+-树索引结构

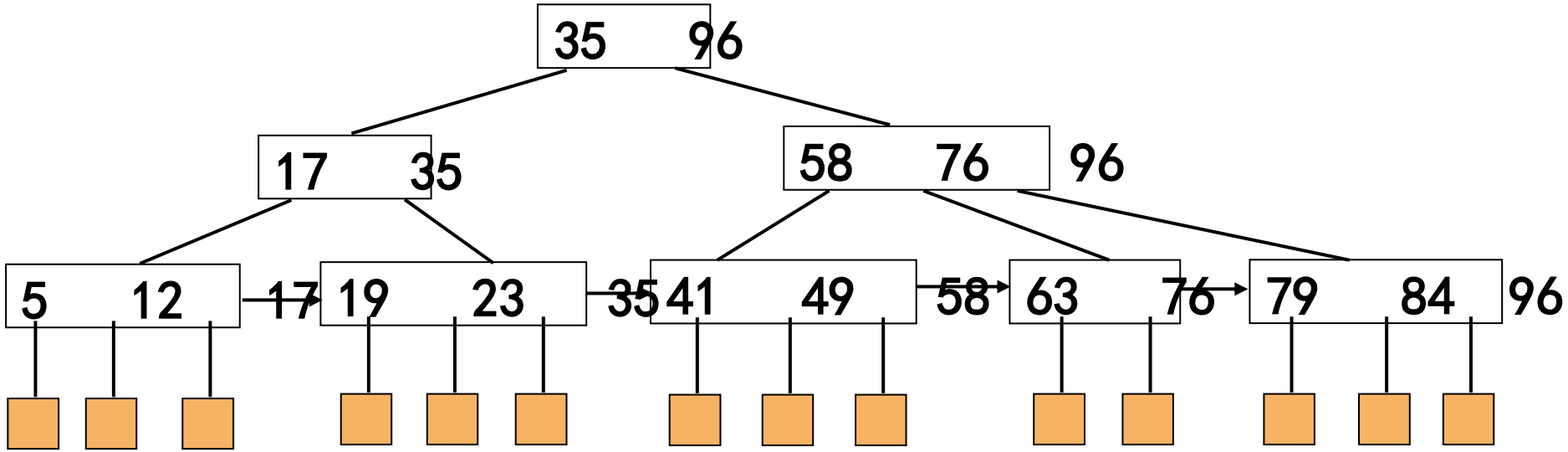
- B+-树都是平衡的，即从根节点到叶节点的每条路径长度都相同(例：n=4)



10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	80000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	60000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

下图是一棵3阶B+树。

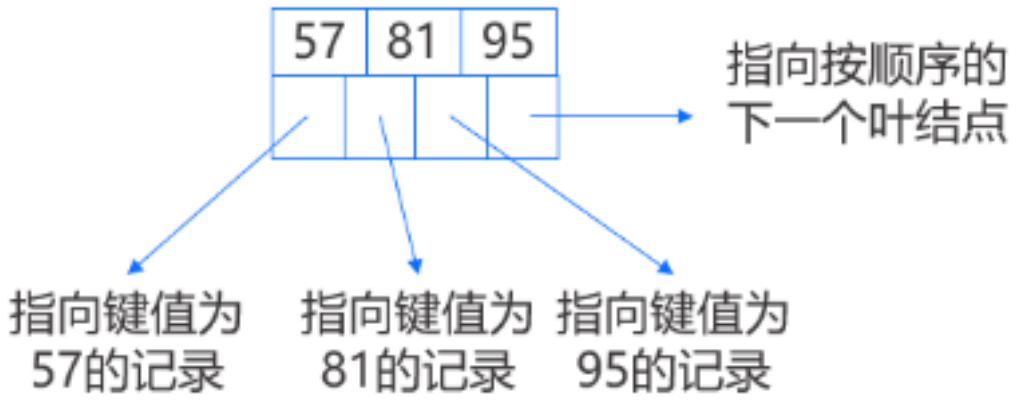
由于B+树的叶子结点和非叶子结点结构上的显著区别，因此需要一个标志域加以区分，结点结构定义如下：



一棵3阶B+树

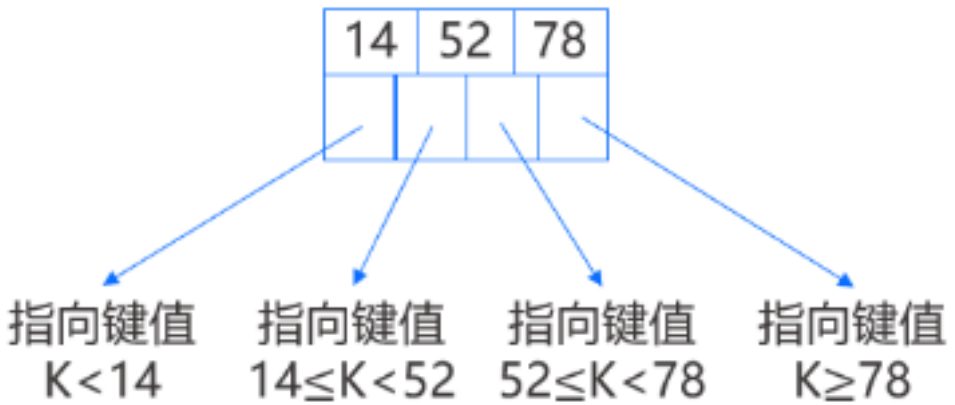
B+树

典型叶结点



至少$\lceil (n+1)/2 \rceil$ =2个键值对

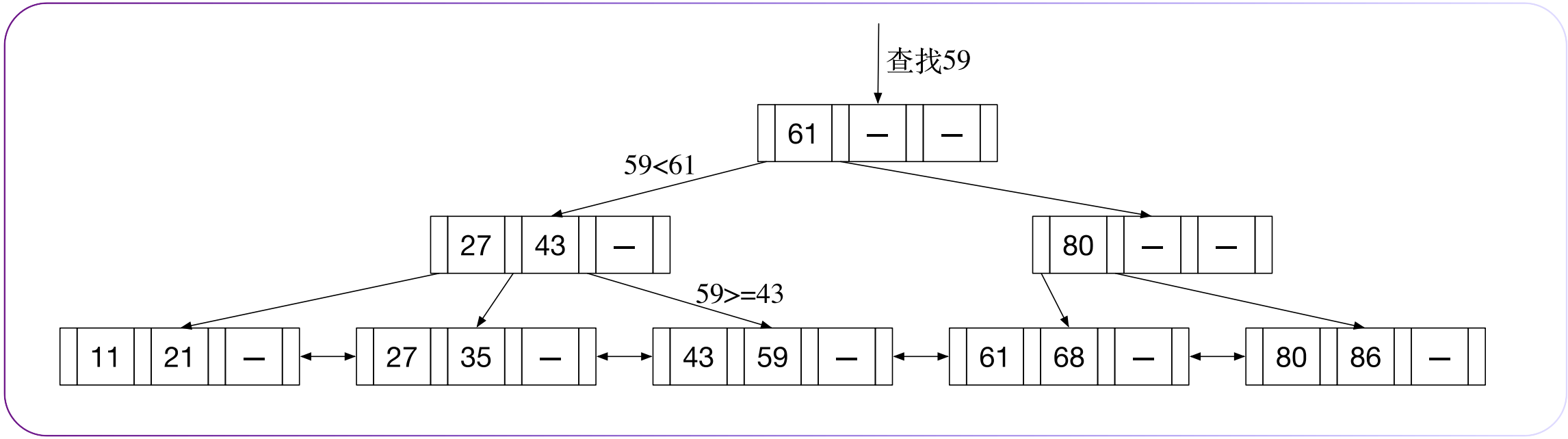
典型内部结点



至少$\lceil (n+1)/2 \rceil$ =2个指针和1个键

B+树

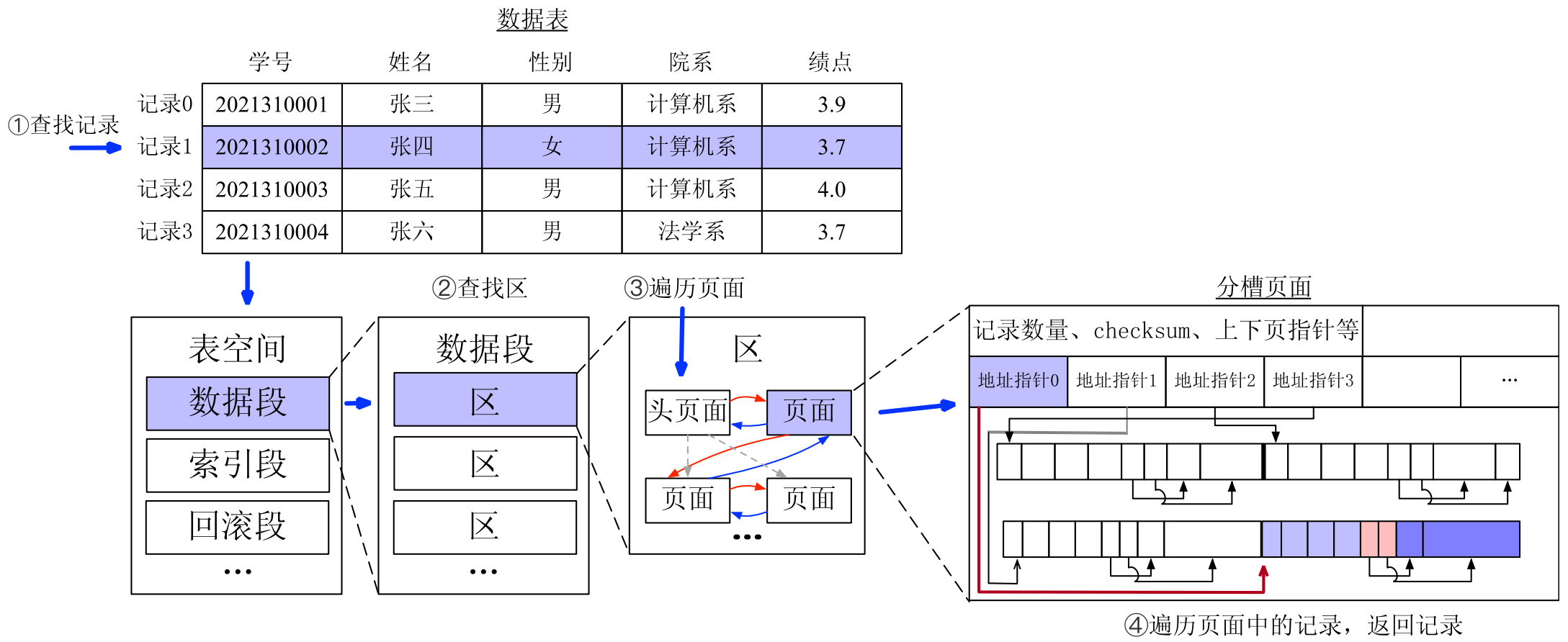
平衡多叉树



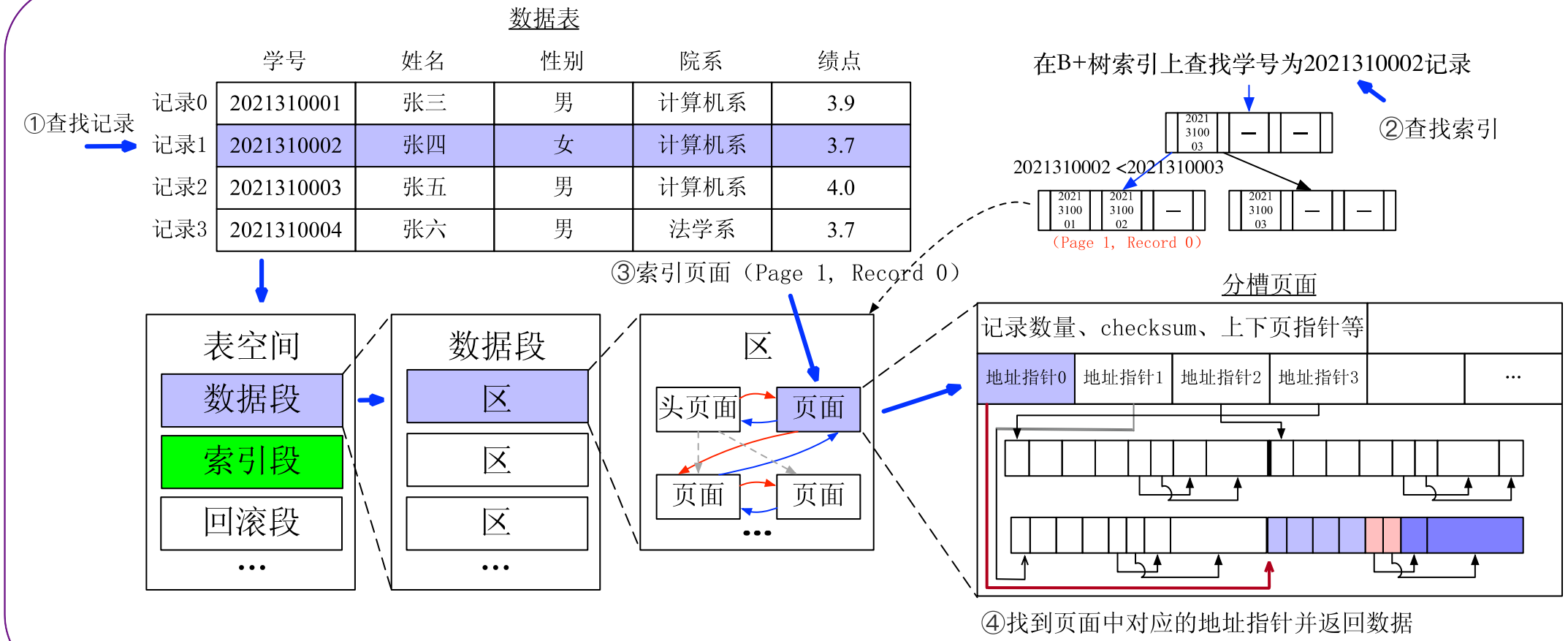
多个相关的表中的数据存储在同一个文件/页面中

院系记录0	计算机系	东主楼			
学生记录0	2021310001	张三	男	计算机系	3.9
学生记录1	2021310002	张四	女	计算机系	3.7
学生记录2	2021310003	张五	男	计算机系	4.0
院系记录1	物理系	西主楼			
学生记录3	2021310006	张七	女	物理系	3.4
学生记录4	2021310009	张八	女	物理系	3.6

查询记录 – 无索引

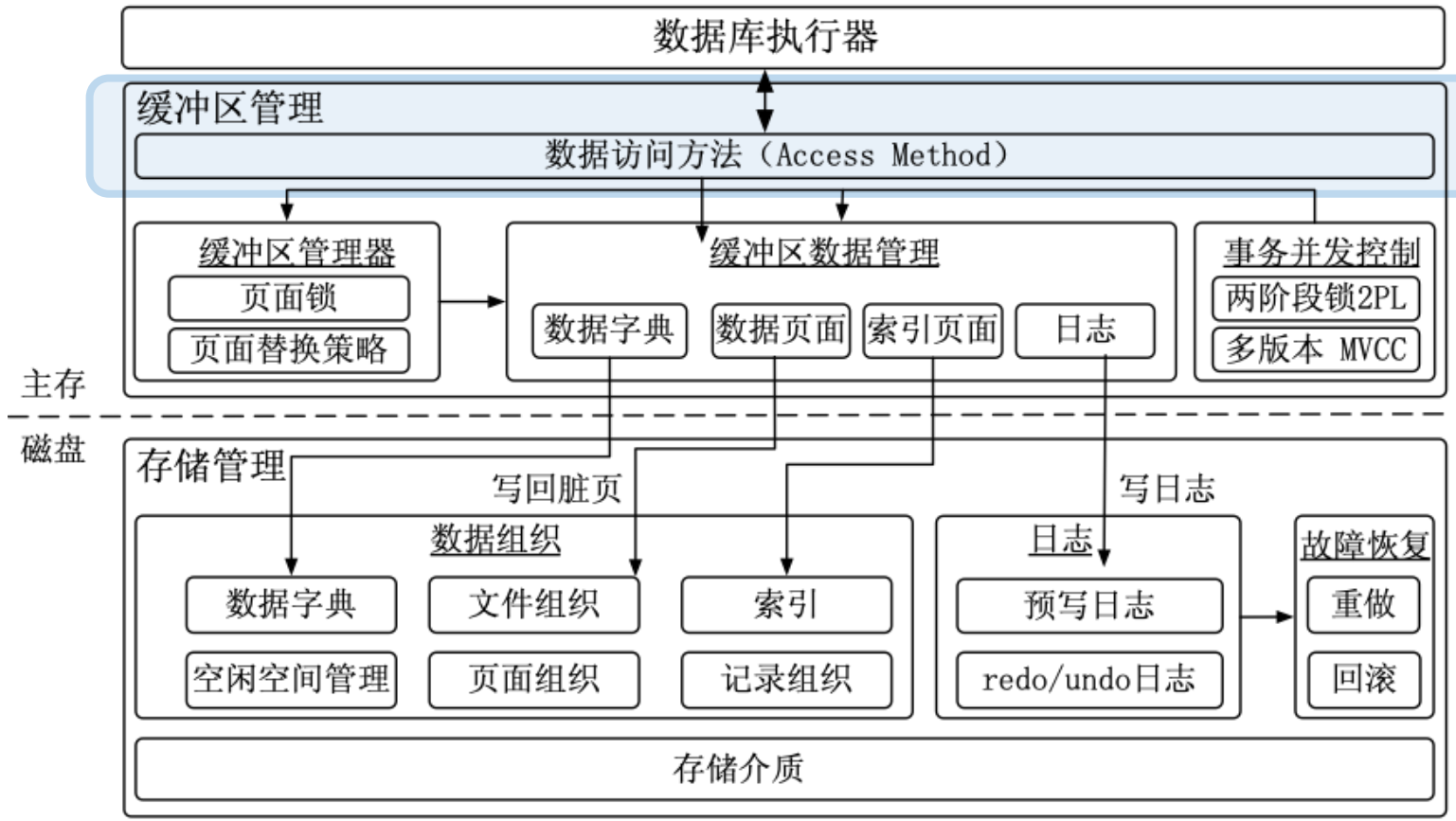

数据表
学号
姓名
性别
院系
绩点
①查找记录
记录0
2021310001
张三
男
计算机系
3.9
记录1
2021310002
张四
女
计算机系
3.7
记录2
2021310003
张五
男
计算机系
4.0
记录3
2021310004
张六
男
法学系
3.7
②查找区
③遍历页面
分槽页面
表空间
数据段
索引段
回滚段
…
数据段
区
区
区
…
区
头页面
页面
页面
页面
…
记录数量、checksum、上下页指针等
地址指针0
地址指针1
地址指针2
地址指针3
…
④遍历页面中的记录，返回记录

查询记录 – 有索引


数据表
学号
姓名
性别
院系
绩点
①查找记录
记录0
2021310001
张三
男
计算机系
3.9
记录1
2021310002
张四
女
计算机系
3.7
记录2
2021310003
张五
男
计算机系
4.0
记录3
2021310004
张六
男
法学系
3.7
在B+树索引上查找学号为2021310002记录
②查找索引
2021310002 <2021310003
(Page 1, Record 0)
③索引页面（Page 1, Record 0）
表空间
数据段
索引段
回滚段
…
数据段
区
区
区
…
区
头页面
页面
页面
页面
…
分槽页面
记录数量、checksum、上下页指针等
地址指针0
地址指针1
地址指针2
地址指针3
…
④找到页面中对应的地址指针并返回数据

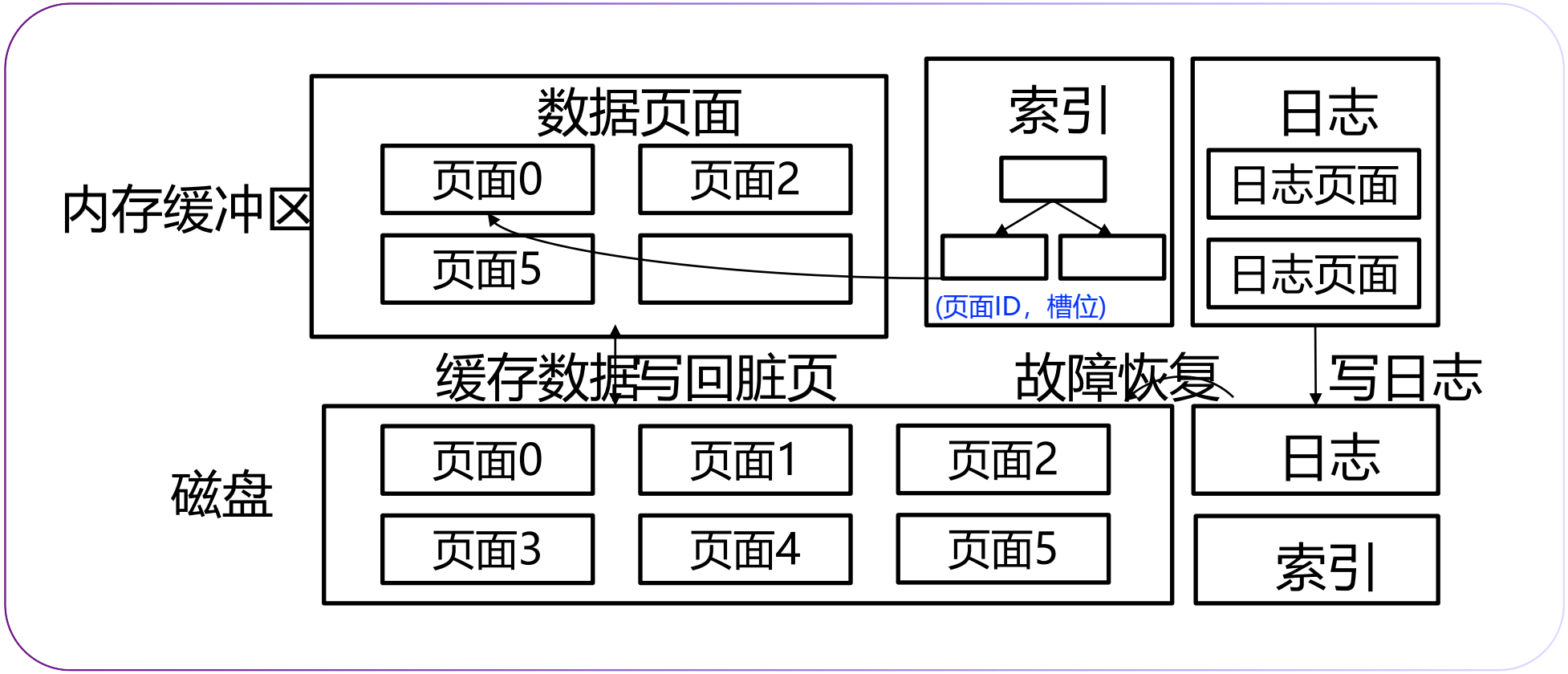
缓冲区

将存储于磁盘的数据同时存储于主存，减少磁盘访问



缓冲区组成

缓冲区数据主要包含缓存的磁盘数据页面、索引、日志等。日志可用于故障恢复。



预写日志与故障恢复

为什么使用日志：

如果缓冲区中的脏页面在写回到磁盘过程中发生故障，则无法保证写入的数据的正确性。

页面随机写　日志的顺序写

数据库使用预写日志 （write-ahead log，WAL）来备份数据的修改以防丢失

写入数据前先写入日志，日志包含即将写入的页面编号和写入的内容，也可以包含原先页面中的内容作为备份以供恢复。

在日志写入到磁盘后再进行事务的提交，保证事务的持久性。

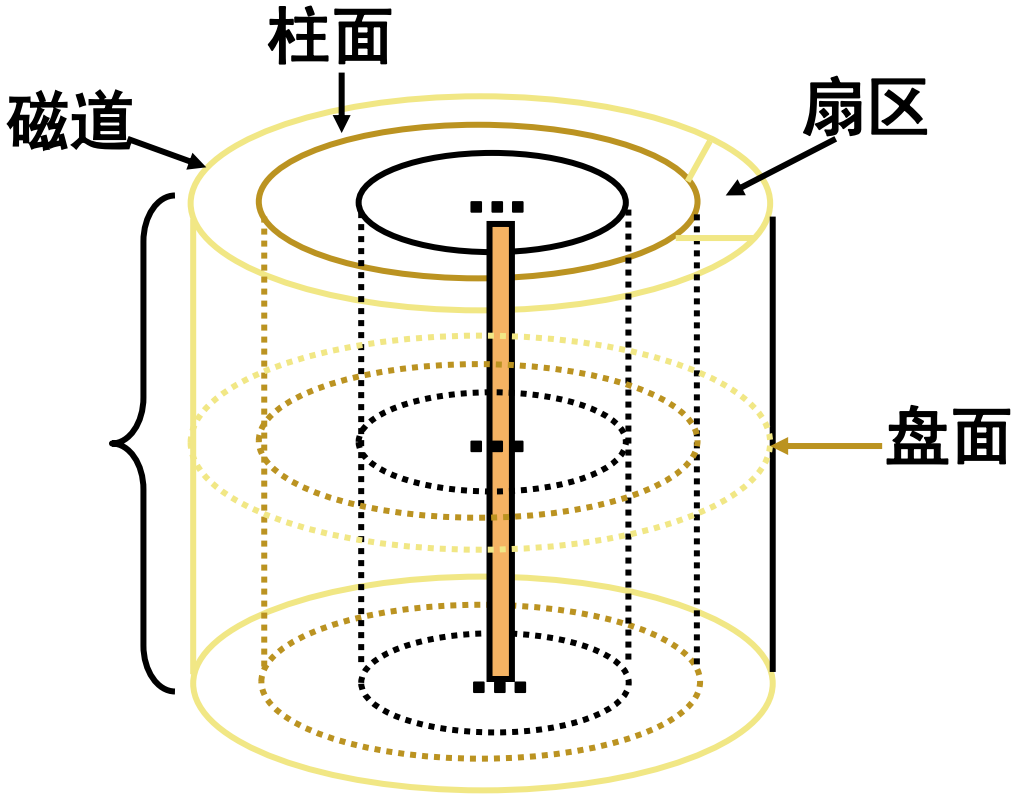
故障并重启时，扫描日志来判断每个操作的成功与失败，并决定撤销或者重做这些操作。

ISAM文件

ISAM(Indexed Sequential Access Method，顺序索引存取方法)，是专为磁盘存取设计的一种文件组织方式，采用静态索引结构，是一种三级索引结构的顺序文件。下图是一个磁盘组的结构图。

ISAM文件由基本文件、磁道索引、柱面索引和主索引组成。

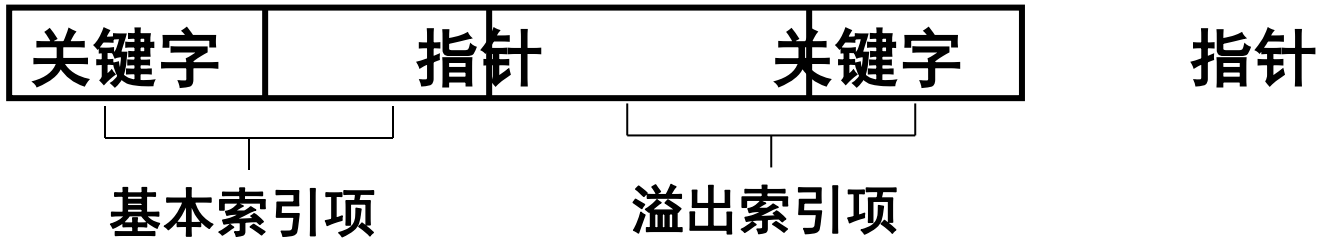
基本文件按关键字的值顺序存放，首先集中存放在同一柱面上，然后再顺序存放在相邻柱面上；对于同一柱面，则按盘面的次序顺序存放。



一个磁盘组结构形式

在每个柱面上，还开辟了一个溢出区，存放从该柱面的磁道上溢出的记录。同一磁道上溢出的记录通常由指针相链接。

ISAM文件为每个磁道建立一个索引项，相同柱面的磁道索引项组成一个索引表，称为磁道索引，由基本索引项和溢出索引项组成，其结构是：



◆ 基本索引项：关键字域存放该磁道上的最大关键字；指针域存放该磁道的首地址。

◆ 溢出索引项：是为插入记录设置的。关键字域存放该磁道上溢出的记录的最大关键字；指针域存放溢出记录链表的头指针。

在磁道索引的基础上，又为文件所占用的柱面建立一个柱面索引，其结构是：

关键字	指针

关键字域存放该柱面上的最大关键字；指针域指向该柱面的第1个磁道索引项。

当柱面索引很大时，柱面索引本身占用很多磁道，又可为柱面索引建立一个主索引。

1 ISAM文件的检索

根据关键字查找时，首先从主索引中查找记录所在的柱面索引块的位置；再从柱面索引块中查找磁道索引块的位置；然后再从磁道索引块中查找出该记录所在的磁道位置；最后从磁道中顺序查找要检索的记录。

2 记录的插入

首先根据待插入记录的关键字查找到相应位置；然后将该磁道中插入位置及以后的记录后移一个位置(若溢出，将该磁道中最后一个记录存入同一柱面的溢出区，并修改磁道索引)；最后将记录插入到相应位置。

3 记录的删除

只需找到要删除的记录，对其做删除标记，不移动记录。当经过多次插入和删除操作后，基本区有大量被删除的记录，而溢出区也可能有大量记录，则周期性地整理ISAM文件，形成一个新的ISAM文件。

4 ISAM文件的特点

- 优点：节省存储空间，查找速度快；
- 缺点：处理删除记录复杂，多次删除后存储空间的利用率和存取效率降低，需定期整理ISAM文件。

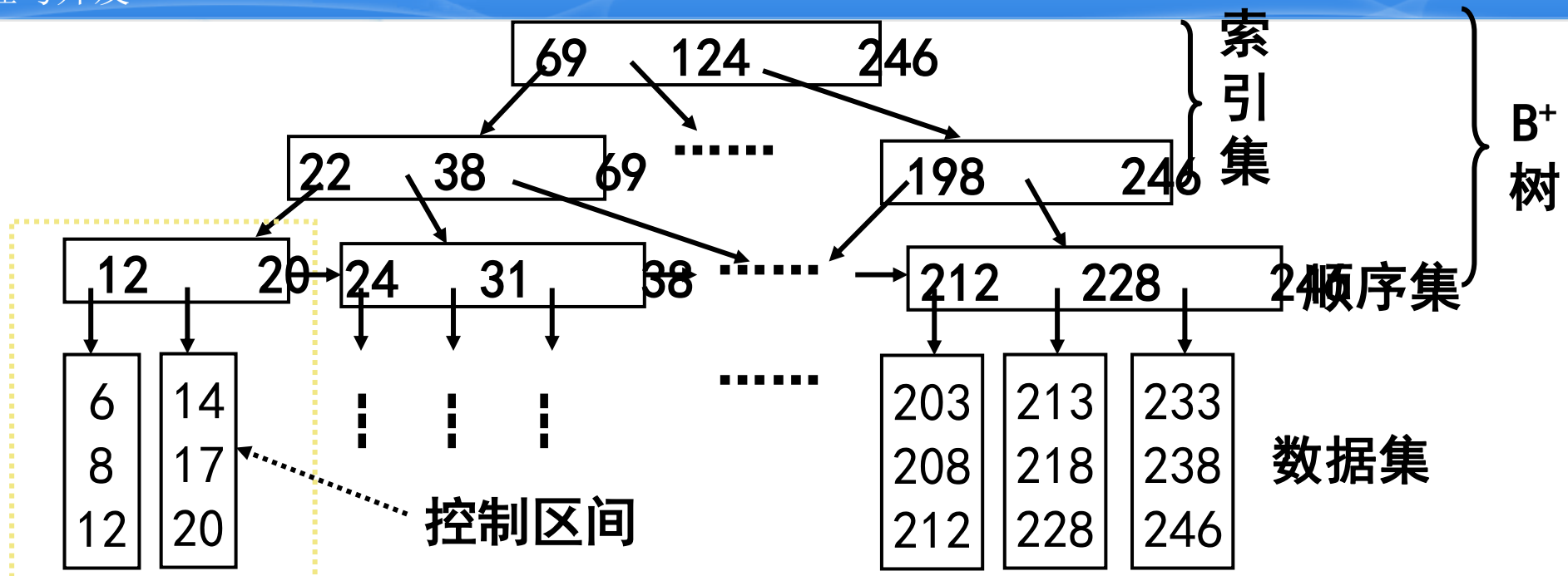
VSAM文件

VSAM(Virtual Storage Access Method，虚拟存取方法)，也是一种索引顺序文件组织方式，利用OS的虚拟存储器功能，采用的是基于B^+树的动态索引结构。

文件的存取不是以柱面、磁道等物理空间为存取单位，而是以逻辑空间——控制区间(Control Interval)和控制区域(Control Range)为存取单位。

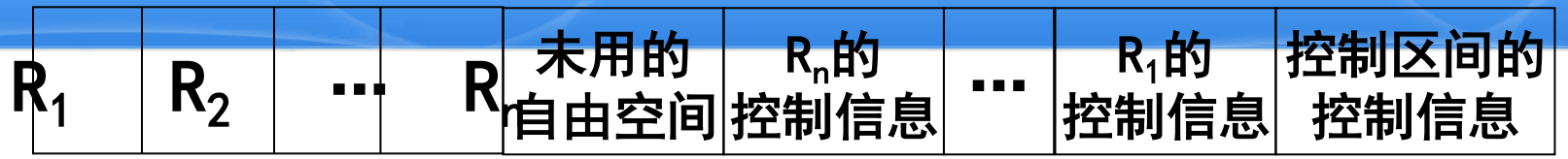
一个VSAM文件由索引集、顺序集和数据集组成，如图所示。

文件的记录都存放在数据集中，数据集又分成多个控制区间；VSAM进行I/O操作的基本单位是控制区间，由一组连续的存储单元组成，同一文件的控制区间大小相同；



VSAM文件结构示意图

每个控制区间存放一个或多个逻辑记录，记录是按关键字值顺序存放在控制区间的前端，尾端存放记录的控制信息和控制区间的控制信息，如图所示。



控制区间的结构

顺序集是由B^+树索引结构的叶子结点组成。每个结点存放若干个相邻控制区间的索引项，每个索引项存放一个控制区间中记录的最大关键字值和指向该控制区间的指针。顺序集中的每个结点及与它所对应的全部控制区间组成一个控制区域。

顺序集中的结点之间按顺序链接成一个链表，每个结点又在其上层建立索引，并逐层向上按B^+树的形式建立多级索引。则顺序集中的每一个结点就是B^+树的叶子结点；在顺序集之上的索引部分称为索引集。

在VSAM文件上既可以按B^+树的方式实现记录的查找，又可以利用顺序集索引实现记录顺序查找。

VSAM文件中没有溢出区，解决方法是留出空间：

- 每个控制区间中留出空间；
- 每个控制区域留出空的控制空间，并在顺序集的索引中指出。

1 记录的插入

首先根据待插入记录的关键字查找到相应的位置：

- 若该控制区间有可用空间：将关键字大于待插入记录的关键字的记录全部后移一个位置，在空出的位置存放待插入记录；
- 若控制区间没有可用空间：利用同一控制区域的一个空白控制空间进行区间分裂，将近一半记录移到新的控制区间中，并修改顺序集中相应的索引，插入新的记录；

◆ 若控制区域中没有空白控制空间：则开辟一个新的控制区域，进行控制区间域分裂和相应的顺序集中的结点分裂。也可按B^+树的分裂方法进行。

2 记录的删除

先找到要删除的记录，然后将同一控制区间中比删除记录关键字大的所有记录逐个前移，覆盖要删除的记录。当一个控制区间的记录全部删除后，需修改顺序集中相应的索引项。

3 VSAM文件的特点

(1) 优点

◆ 能动态地分配和释放空间；

- 能保持较高的查询效率，无论是查询原有的还是后插入的记录，都有相同的查询速度；
- 能保持较高的存储利用率(平均75%)；
- 永远不需定期整理文件或对文件进行再组织。

(2) 缺点

- 为保证具有较好的索引结构，在插入或删除时索引结构本身也在变化；
- 控制信息和索引占用空间较多，因此，VSAM文件通常比较庞大。

基于B^+树的VSAM文件通常被作为大型索引顺序文件的标准。