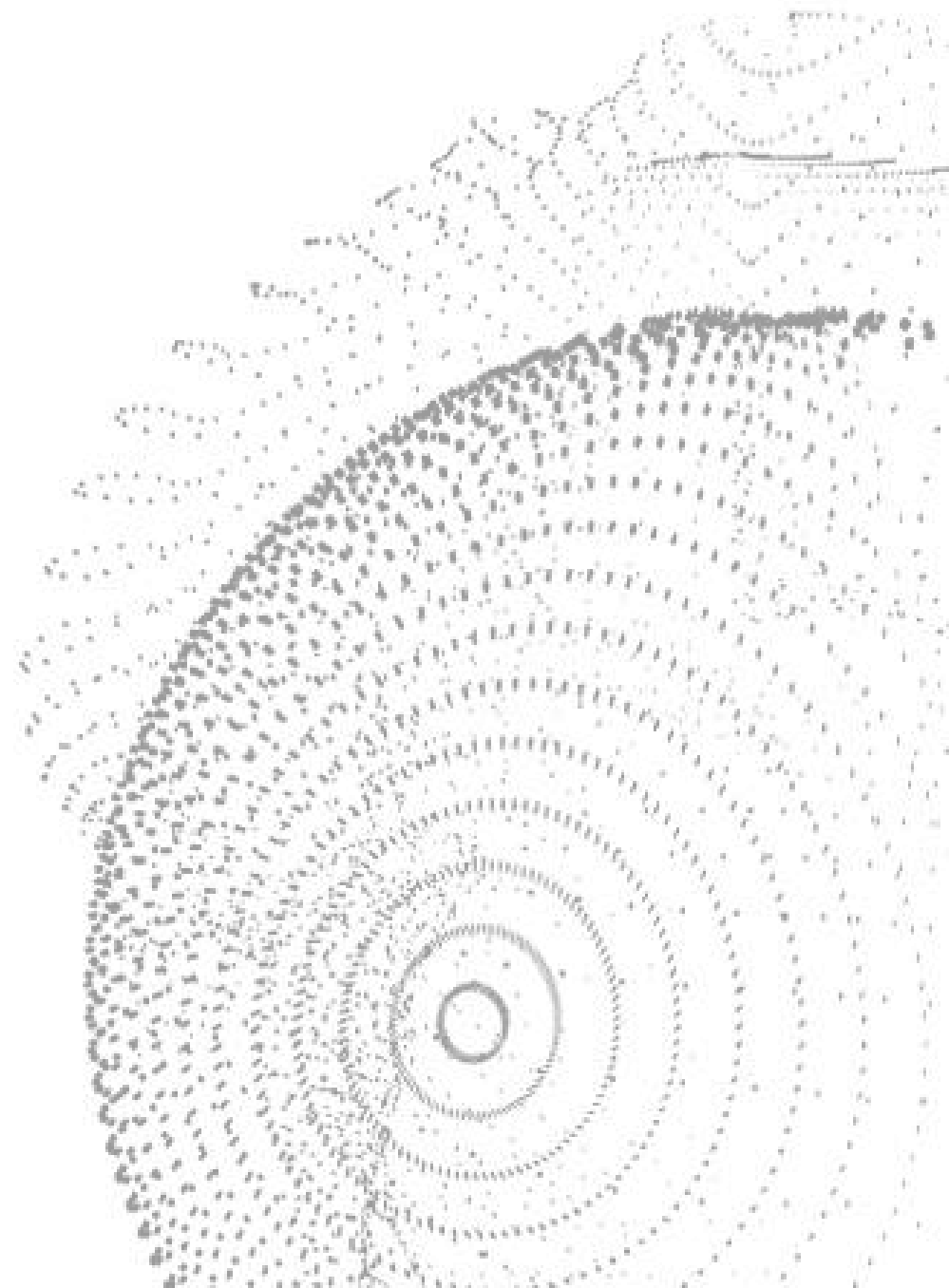


Chapter 06

数据库的物理组织

Physical organization of the database



第6章 目录

-01

概论

-02

数据库的物理存储介质

-03

索引技术与散列技术

-04

数据库与文件

- **研究目的**
 - 提高存储效率，加快存取速度

- **研究内容**
 - 磁盘的物理结构
 - 文件的组织与设计

- **主要技术**
 - 索引技术
 - 散列技术

第6章 目录

-01

概论

-02

数据库的物理存储介质

-03

索引技术与散列技术

-04

数据库与文件

- **三级存储器结构**

- 第一级：**主存储器** (main memory)

- 包括：高速缓冲存储器 (cache)、主存储器 (memory)
-

- 第二级：**磁盘存储器** (secondary storage)

- 也称为：二级存储器或次级存储器
-

- 第三级：**辅助存储器** (tertiary storage)

- 包括：磁带存储器、自动光盘机
 - 是一种辅助存储设备，也称三级存储器

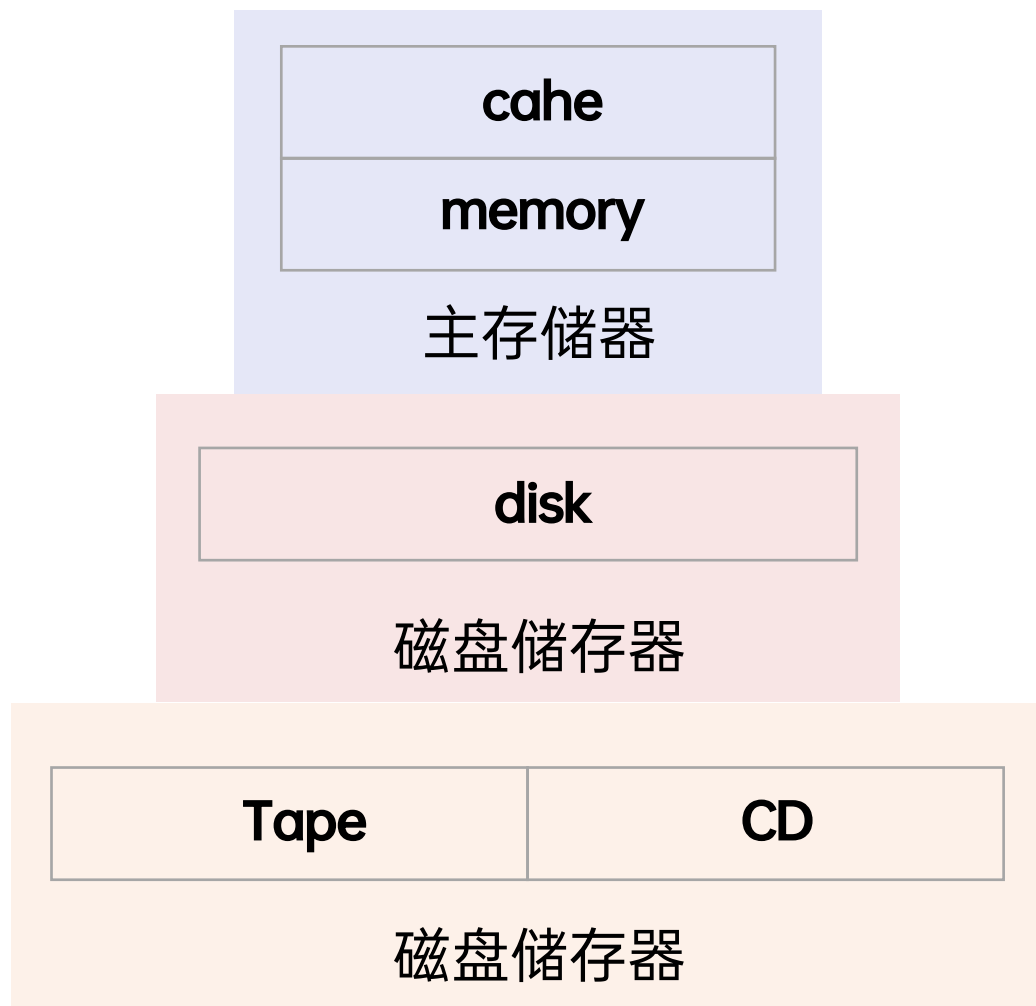
6.2

数据库的物理存储介质

	存储容量	访问速度	访问类型	存储单位
第一级 存储器	100Mb~10GB	10^{-8} 秒~ 10^{-7} 秒	随机	字节
第二级 存储器	10GB~ 20^3 GB	10毫秒~30毫秒	随机	磁盘块
第三级 存储器	10^6 GB	几秒钟~几分钟	随机	磁盘块

6.2

数据库的物理存储介质



小

快

高

大

慢

低

存储容量

访问速度

制造成本

第6章 目录

-01

概论

-02

数据库的物理存储介质

-03

索引技术与散列技术

-04

数据库与文件

6.3

索引技术与散列技术

- 在数据库中数据查找的速度是至关重要的，但是当数据库中数据量增大时，数据查找速度就会受到影响，当数据量极大时，查找速度将会受到严重影响，为解决此问题需引入“索引技术”与“散列技术”

- 内容

1. 顺序文件

2. 索引文件

3. B/B+树文件

4. HASH文件

- 在顺序文件上的索引技术
 - 稠密索引、稀疏索引、多级索引
- 非顺序文件中的索引技术
- 多维索引

6.3.1

索引技术与散列技术 - 顺序文件

- 按照某个属性(项)的取值进行排序而构成的数据文件被称为**顺序文件**

例（假设每个磁盘块可以存放2条记录）

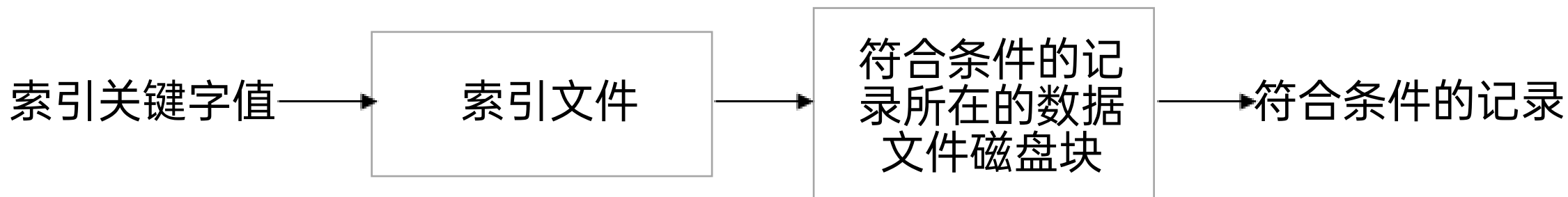
1	记录1	第1个磁盘块
2	记录2	
3	记录3	第2个磁盘块
4	记录4	
5	记录5	第3个磁盘块
6	记录6	
...

主关键字值

6.3.2

索引技术与散列技术 -索引文件

- **建立索引文件的目的：**以记录的特征值（通常是一个或多个字段的值）为输入，能够快速找出具有该特征的记录
- **利用索引文件进行记录查找的过程：**首先在索引文件中按照特征字段的值进行查找，找出具有该特征的记录的记录指针（即记录在数据文件中的存储地址），从而可以在数据文件中进行直接定位，读出所需要的记录



利用索引文件访问数据文件中的记录的示意图

6.3.2.1 索引文件-在顺序文件上的索引

- 稠密索引

- 稀疏索引

- 多级索引
 - 在讨论上述的三种索引文件时，我们假设其索引属性的值具有唯一性。否则我们可以使用：

- 具有重复键值的索引

6.3.2.2

稠密索引

- **索引文件**

- 存放记录的关键字值以及指向记录本身的指针(记录的存储地址), 并且按照关键字值的顺序进行排序
 - 一个关键字值和一个记录指针构成的键-指针对我们称为一个索引项:

关键字K

记录指针P

- **稠密索引**

- 如果数据文件中的每条记录在索引文件中都存在一个相对应的索引项, 则该索引被成为稠密索引
- 例如:

6.3.2.3

稀疏索引

- 如果数据文件是顺序文件，我们可以在索引文件中只为数据文件的每个磁盘块设一个索引项，记录该磁盘块中第一条数据记录的关键字值及该磁盘块的首地址
- 这样建立起来的索引文件被称为稀疏索引

6.3.2.4

多级索引

- **索引文件本身也可能占据多个存储块，为了能够快速找到这些索引块在磁盘中的存储位置，需要引入新的索引结构，即在索引文件上再建立索引，从而构成了多级索引**
 - 由于索引文件本身是顺序文件，因此索引文件上的索引组织方法采用的是稀疏索引
- **我们将直接建立在数据文件上的索引（例2、例3中建立的索引）称为第一级索引，根据第一级索引文件建立的索引称为第二级索引，依此类推，从而可以建立一个多级索引结构**
 - 在多级索引组织结构中，第一级索引可以是稠密索引，也可以是稀疏索引
 - 从第二级索引开始建立的都是稀疏索引

6.3

索引技术与散列技术

- 到此为止，我们介绍了基于顺序文件的稠密索引、稀疏索引和多级索引，并且索引属性是关系的关键字，具有唯一性。以此为基础，我们可以构造出索引属性不唯一以及非顺序文件上的索引。
- **在顺序数据文件中建立具有重复键值的索引**
 - 在数据文件中索引关键字的值是不唯一的，即一个索引关键字值对应着若干条记录。
 - 具有重复键值的索引文件可以采用以下的组织方法：
 - 稠密索引
 - 稀疏索引

- 具有重复键值的索引文件组织

- 具有重复键值的稠密索引

- 为数据文件中的 **每一个索引关键字值 K 定义一个索引项**，其中的记录指针指向索引关键字值等于 K 的第一条记录
 - 在索引文件中索引关键字的值是唯一的
 - 利用该索引文件查找键值为 K 的记录时，所找到的记录指针（如果找到具有键值为 K 的索引项）指向数据文件中第一条键值为 K 的记录
 - 由于 **数据文件是顺序文件**，因此，我们可以在数据文件中从找到的第一条记录开始向后扫描读取每一个键值为 K 的记录，直到出现一条键值不为 K 的记录，查找结束

- **非顺序文件中的索引技术**

- 在数据库系统中所使用的数据文件通常都是无序的（堆文件组织），因此迫切需要利用上述的索引技术来建立非顺序文件上的索引，以提高记录查找的速度
- 如果索引关键字值在非顺序数据文件中具有唯一性，则可以按照下述步骤建立其上的索引文件：
 - 首先为非顺序数据文件建立第一级的稠密索引
 - 然后再根据需要建立该稠密索引上的多级稀疏索引

6.3.2

索引技术与散列技术

- 如果索引关键字的值不唯一，则可以通过在第一级的稠密索引和数据文件之间加一个记录指针桶(bucket)。此时索引项(K_i, P_i)中的记录指针 P_i 不再是指向数据文件中的记录，而是指向一个记录指针桶，在桶中存放着索引关键字值为 K_i 的记录的记录指针
 - 在这里，记录指针桶的大小 $BSize$ 是预先确定的
 - 当在数据文件中插入第一条索引关键字值为 K_i 的记录 R_i 时，在索引文件中将生成一个新的索引项(K_i, P_i)，同时系统将自动申请一个大小为 $BSize$ 的记录指针桶 B_i ，将指针 P_i 指向该记录指针桶 B_i ，同时将当前记录 R_i 的记录指针保存在记录指针桶 B_i 中
 - 当新的索引关键字值为 K_i 的记录被加入到数据文件中时，只需要将新记录的记录指针加入到记录指针桶 B_i 中。如果记录指针桶 B_i 已满，系统将申请一个新的记录指针桶，并链接到原来的记录指针桶的后面

- **多维索引**

- 在上述介绍的索引文件中，我们并没有关心索引关键字的构成，它可以由关系中的单个属性上的值组成，也可以是多个属性值的一个组合
- 根据多个属性值的组合来建立的索引文件被称为多维索引

6.3

索引技术与散列技术

- 索引文件本身是一个顺序文件，利用索引文件或多级索引可以大大提高数据文件的访问效率
- 索引顺序文件的不足
 - 记录查找算法的效率不高 ($\log_2 N$)
 - 在数据文件是非顺序文件，或索引关键字值的变化比较频繁的情况下，索引顺序文件自身的维护非常复杂，对索引项的插入、修改和删除操作会导致索引项在索引文件中的大量移动
 - 在上述情况下，如果通过引入链接磁盘块的方法来减少索引项的移动，又会减低存储空间的利用率，并最终影响到系统的性能
 - 因此需要引入适合于索引文件的存储组织的文件结构，这就是在数据库系统中广泛使用的多级索引技术：**B/B+树索引**

6.3.3

B/B⁺树索引文件

- B/B⁺树是一种多级索引组织方法，是适合于组织存放在外存的大型磁盘文件的一种树状索引结构。其中用得比较多的是B⁺树。下面的讨论如不特别声明都是指B⁺树索引文件
- B/B⁺树的结点划分
 - 叶结点：B/B⁺树的最下一级索引是树的叶结点
 - 内部结点：B/B⁺树中的其它结点(非叶结点)，其中：
 - 根结点：B/B⁺树的最上一级索引是树的根结点
- 在B/ B⁺树中，由叶结点所构成的最下面的一级索引通常采用稠密索引，而其它层次上的索引则采用稀疏索引

6.3.4

散列技术

- 基本原理

- 是一种‘利用散列函数 $h(K)$ 建立起数据文件中指定项值 K 与磁盘物理块间的映射关系’的索引技术
- 这样只要给出指定项的值 K_i 立即可通过 $h(K_i)$ 得到保存其记录的物理磁盘块地址



6.3.4

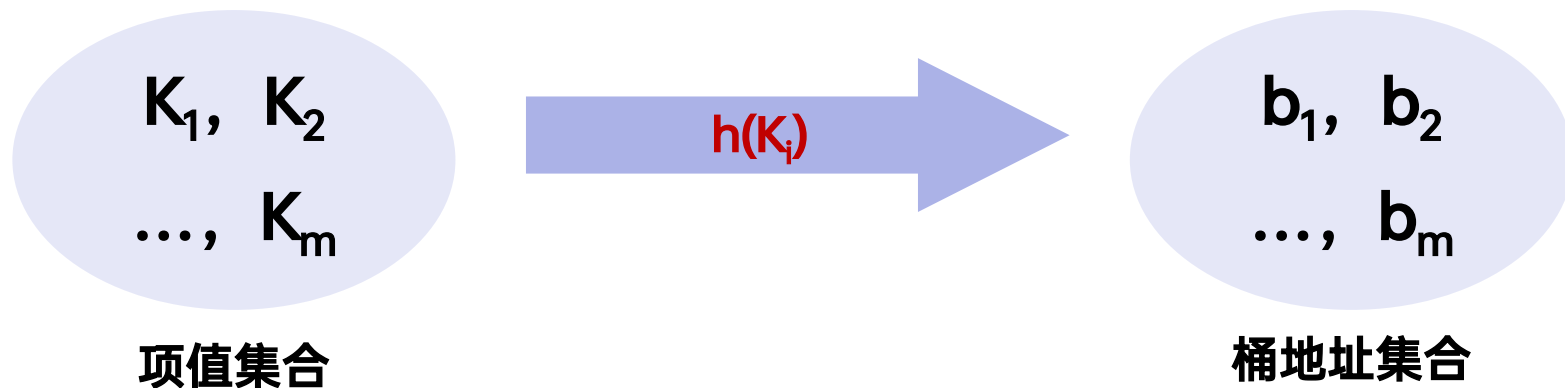
散列技术

- 散列技术的实现方法

- 建立数据文件的指定项 K 以及该项值的集合 $\{K_1, K_2 \dots, K_n\}$
- 建立磁盘物理存储单位桶 (Bucket) 以及桶地址的集合集合 $\{b_1, b_2 \dots, b_n\}$
 - 一个桶可以存放多条记录 (或记录指针)
 - 一个桶可以是一个磁盘块, 也可以是比磁盘块还大的物理空间
- 设计散列函数 $h(K_i)$
 - 以建立数据文件中指定项的值与桶 (桶地址) 之间的对应关系, 即一个 K_i 通过 $h(K_i)$ 必能找到惟一的一个桶地址
 - 设计良好的散列函数将使得 n 个指定项值被平均分配到 m 个桶中去

6.3.4

散列技术-散列技术的实现方法



- 在散列技术中，桶的空间大小是固定的，即一个桶中可以存放的记录（指针）数是固定的
- 但是在实际应用中，记录在指定项值上的分布往往是不均衡的，从而使得在某些桶中存在空间浪费现象，而另外一些桶则存在空间溢出问题
- 当一个桶的空间溢出时，需要通过链接的方法申请“溢出桶”与其相连，以达到扩大桶空间的目的

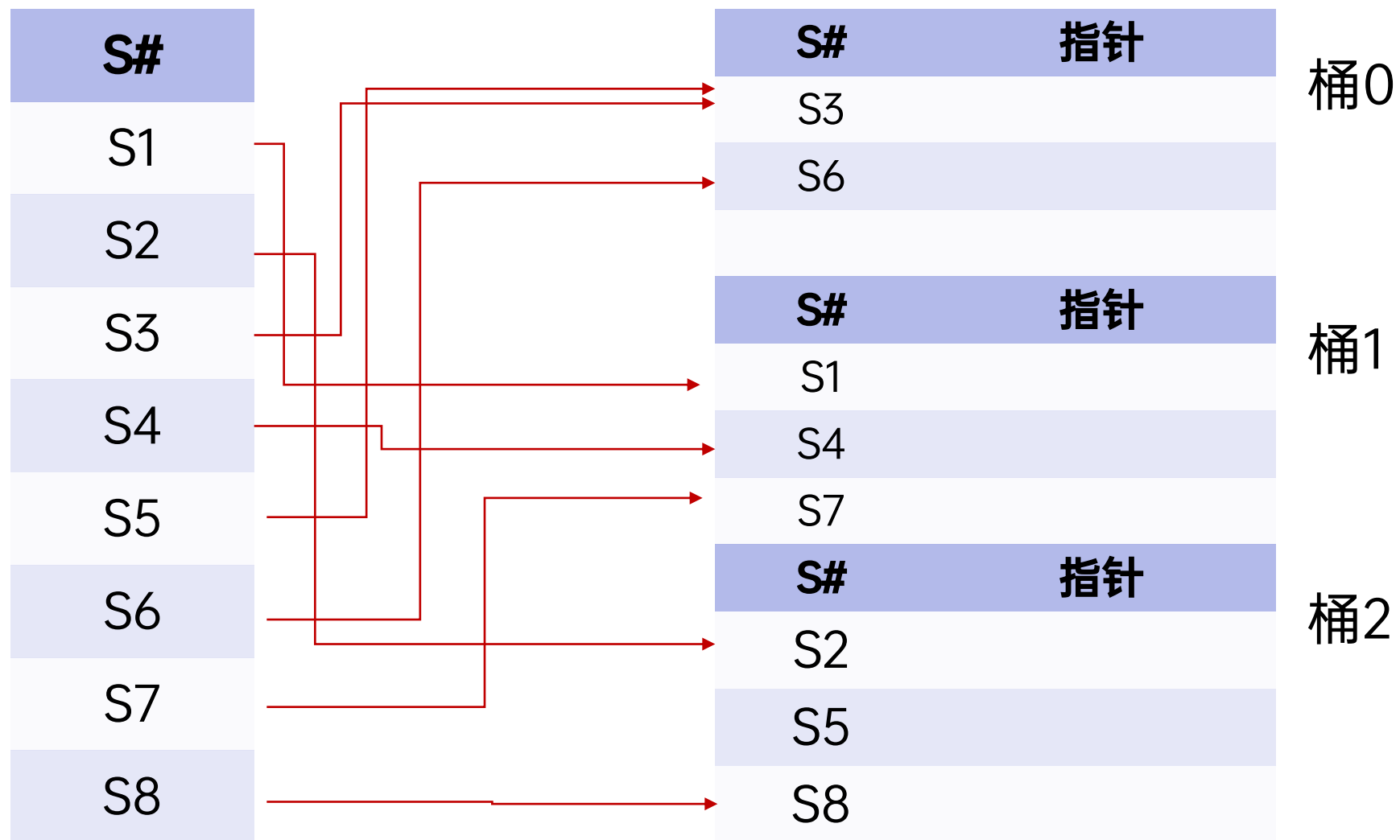
- 散列索引

- 将散列技术与索引技术相结合形成的“散列索引”技术
- 散列索引的构造方法
 - 为数据文件中的每个索引项值建立一个索引记录
 - 将这些索引记录组织成散列结构
- 例如：为学生关系建立一个基于学号（S#）的散列索引，其中初始化桶地址的空间为{0, 1, 2}，每个桶可以存放三条索引记录，其散列函数是：

$$h(K_i) = i \bmod 3$$

6.3.4

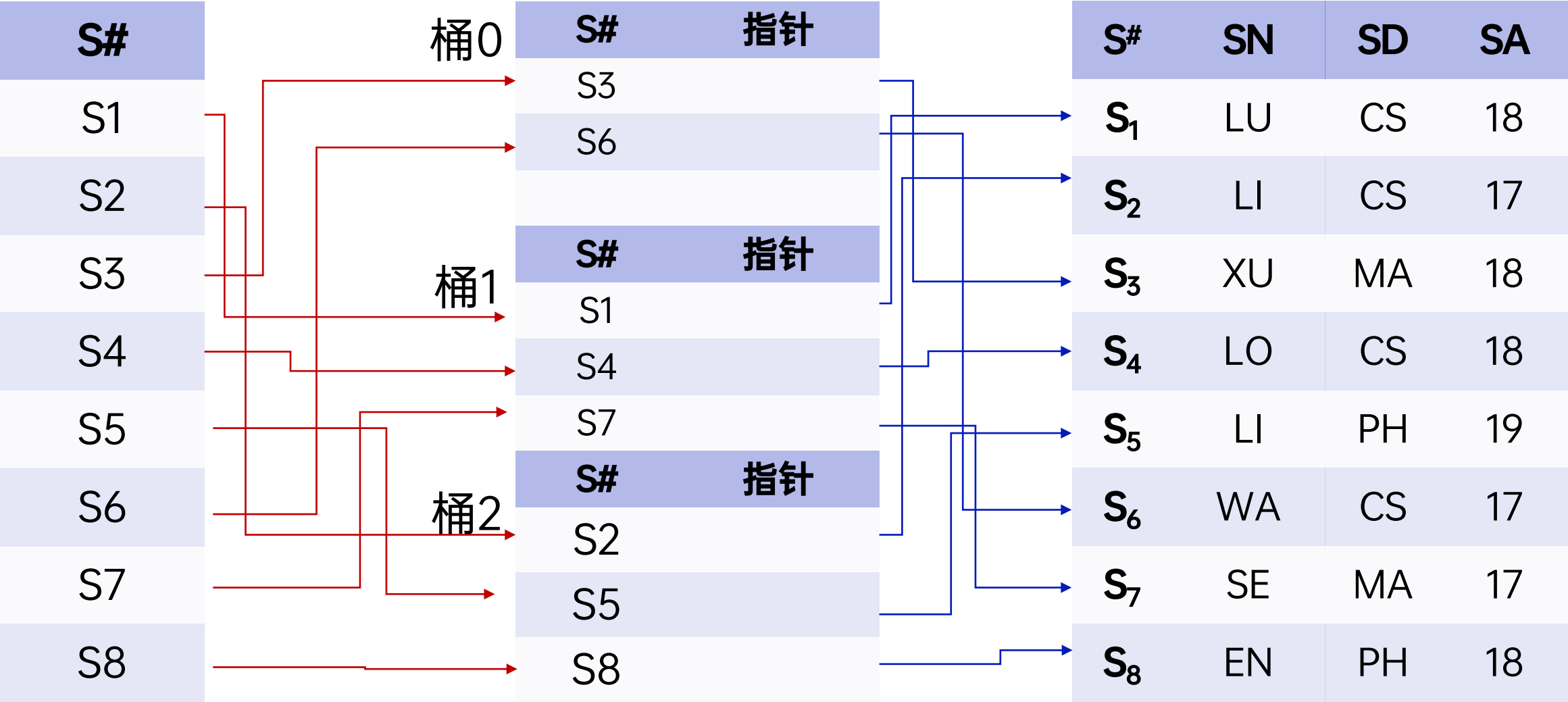
散列技术



散列示意图

6.3.4

散列技术



散列示意图

6.3.4

可扩展的动态散列

- **静态散列一般通过增加溢出页来处理溢出问题**
 - 如不希望增加溢出页，也可修改散列函数，将桶的数目扩大（如扩大一倍），然后重组数据文件
 - 但这种重组的代价可能很大：
 - 需要读入 n 个页，并写回 $2n$ 个页
- **克服这个问题的一个方法**
 - 引入一个仅存储桶指针的目录数组，用翻倍目录数组来取代翻倍数据桶数目
 - 由于每个目录项只含有一个桶指针，目录所需存储页一般很少，这样翻倍的代价就很小
 - 每次只分裂有溢出的桶

- **位图索引**

- 是一种主要针对多键查询设计的特殊类型索引，尽管每个位图索引都是建立在单键码之上
-
- 为了使用位图索引，关系或数据文件中的记录必须进行顺序编号 $1, 2, \dots, n$ ，使得给定一个编号 i ，能方便检索到第 i 个数据记录

6.3.5

位图索引

- 位图索引的结构

- 考虑一个共有 n 个记录的关系 R 和它的一个属性 A ，假设 R 的所有记录在 A 上只有 m 个的不同取值， v_1, v_2, \dots, v_m
- R 在 A 上的位图索引
 - 是一组位图向量
 - $R.A$ 的每个取值 v_j ($j=1..m$)，分别对应一个位图向量 bit_v_j
 - 共有 m 个这样的位图向量
 - bit_v_j 是一宽度为 n 的二进制数，该数的第 i 位值
 - $bit_v_j[i] = 1$ ，如果第 i 条记录的 $R.A$ 取值 = v_j ；
 - $bit_v_j[i] = 0$ ，如果第 i 条记录的 $R.A$ 取值 $\neq v_j$ ；

6.3.5

位图索引示例

Record number	name	gender	address	Income - level
0	John	m	Perryridge	L1
1	Diana	f	Brooklyn	L2
2	Mary	f	Jonestown	L3
3	Peter	m	Brooklyn	L4
4	Kathy	f	Perryridge	L5

gender的位图

m	10010
f	01101

Income-level的位图

L1	10000
L2	01000
L3	00100
L4	00010
L5	00001

6.3.5

位图索引

- 多属性检索应用

- 条件匹配查询

$$\sigma_{\text{gender}=f \wedge \text{income-level} = L2} (r)$$

- 范围查询

$$\sigma_{\text{gender}=f \wedge \text{income-level} > L2} (r)$$

第6章 目录

-01

概论

-02

数据库的物理存储介质

-03

索引技术与散列技术

-04

数据库与文件

- **数据库中数据分类**
-
- **数据库存储空间组织**

6.4.1

数据库中数据分类

1. 数据主体及辅助数据

- 数据主体：用户数据自身
- 辅助数据：数据主体的控制信息
 - 如数据的长度、数据项的分隔符等，通常它们与数据主体是结合在一起存储的

2. 数据字典

- 有关数据的描述信息
- 数据字典的数据量较小，但使用频率高，需要有较高的访问速度

3. 数据间联系信息

- 数据主体内部存在着数据间的联系，需要用一定的“数据”来表示
- 这方面信息也是一类数据，在不同的数据库系统中有不同的实现方法
 - 在层次及网状数据库中，采用‘指引元’或‘邻接关系’来表示数据间的联系信息
 - 在关系数据库中，用‘外关键字’来表示数据间的联系，这类数据也被包含在数据主体中

4. 数据存取路径信息

- 在层次及网状数据库中，不需要附加的数据来表示访问路径，访问方式较单一
- 在关系数据库中，一般通过索引来定义数据存取路径（如：主关键字上的索引），用户也可以通过创建新的索引文件来提供新的访问路径
 - 所有索引文件都需要占用新的存储空间

5. 其它信息：日志信息、用户信息、审计信息

- 特殊的数据需要有特殊的处理办法，包括：
 - 日志信息
 - 用于记录对数据库作“更新”操作的有关信息，以对付数据库遭受破坏时恢复之用
 - 用户信息
 - 有关数据库用户登录信息以及相应的用户安全性信息
 - 审计信息
 - 用于跟踪用户是否正确使用数据库的审计信息

- **系统区**
 - 数据字典
 - 日志
 - 用户信息
 - 审计信息
- **数据区**
 - 数据主体及其辅助数据
 - 数据之间的联系信息
 - 数据的访问路径

6.4.2

数据库存储空间组织

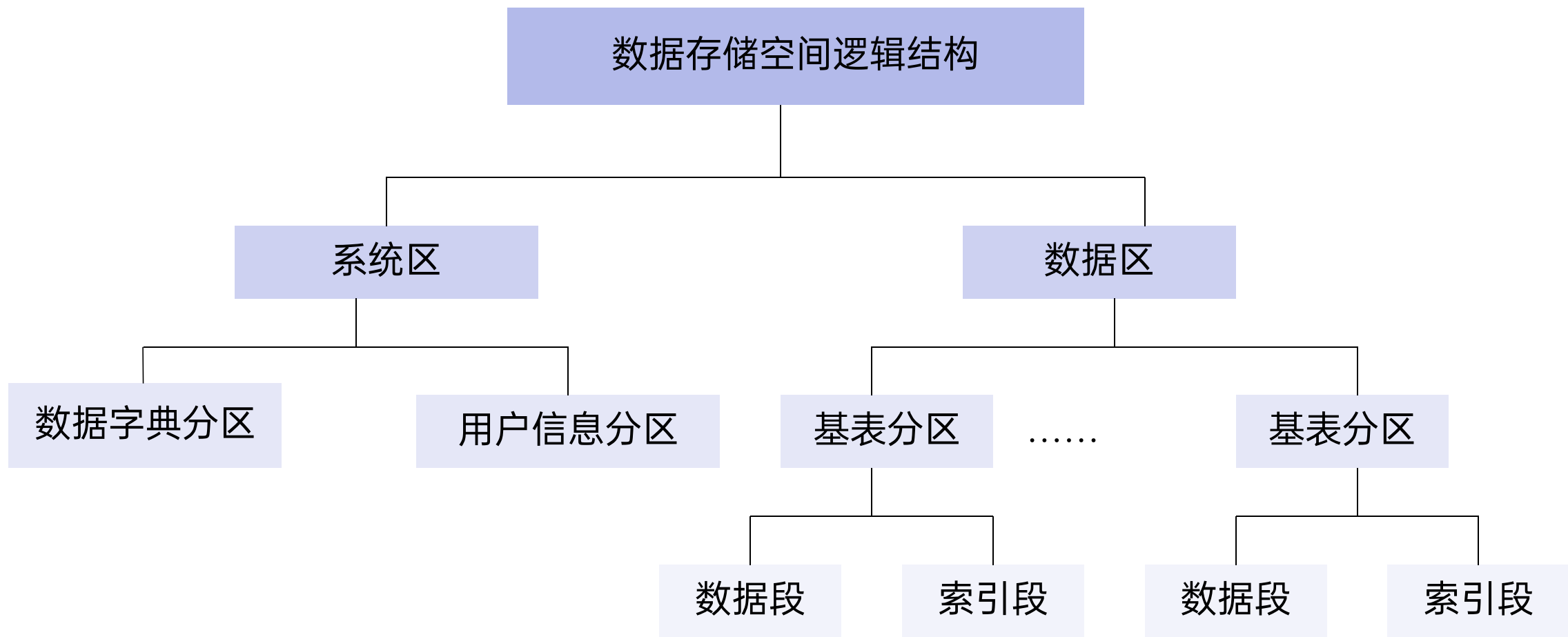


图6.22 数据存储空间逻辑结构图

第一次课后作业