

## 数据库系统 | Database Systems

第六课:散列索引

Lecture 6: Hash Index

Lecturer: Harbour

Date: 2021.12.13



Scan CC WeChat to Join the Community添加CC好友,接受进群邀请

Welcome to follow the GitHub repo

欢迎关注我们的代码仓库

https://github.com/cnosdatabase/cnosdb





## 00 计划教学内容



#### 必讲

DB/DBMS

关系模型与关系代数

数据库存储

#### 散列索引(哈希)

B+树

查询处理

并发控制

#### 如果有兴趣

SQL

查询优化

恢复系统

分布式OLTP/OLAP

#### 本节内容



- 哈希表
- 哈希函数
- 静态哈希策略
- 动态哈希策略

#### 哈希表



- 哈希表实现了将键映射到值的无序关联数组。
- 使用哈希函数为给定键计算数组的偏移量,从该偏移量可以可以找到值。

#### 哈希表



• 空间复杂度: O(n)

• 操作复杂度:

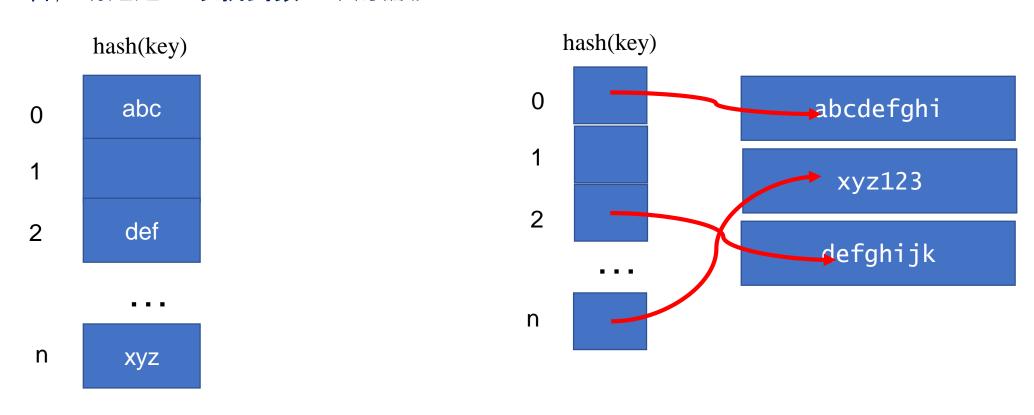
• 平均: 0(1)

• 最差: O(n)

#### 哈希表



- 静态哈希表
  - **原理**:分配一个巨大的数组,该数组为您需要存储的每个元素分配一个插槽。要查找条目,请通过%n以找到数组中的偏移量。





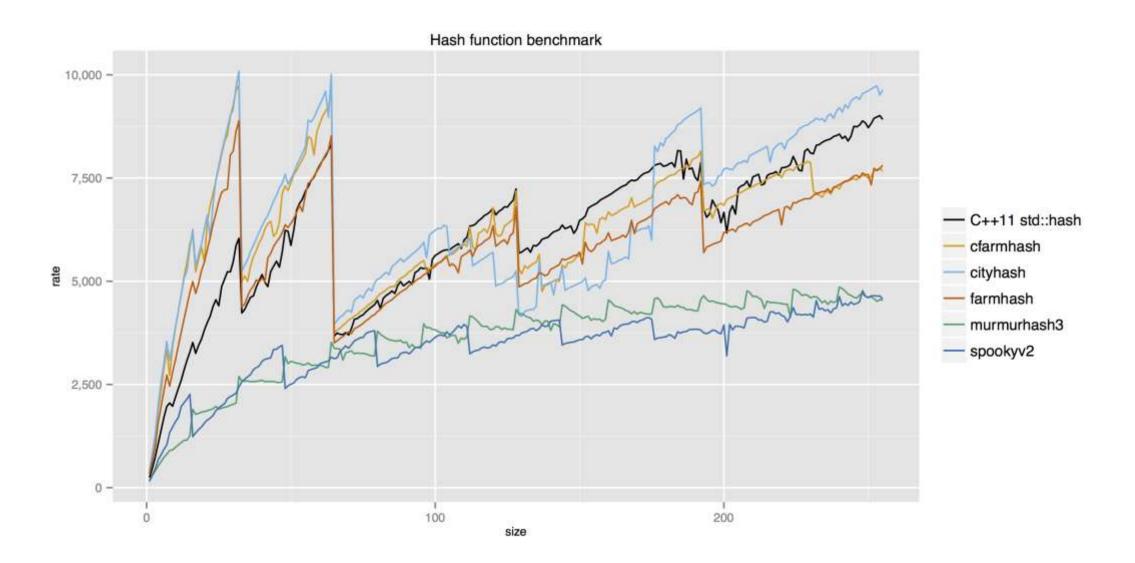
- 什么是哈希函数
  - 对于任何输入键(key), **返回**该键的整数表示。
- 哈希函数的设计
  - 如何将较大的密钥空间映射到较小的域中。
  - 在速度与冲突率之间做更佳的权衡。



#### • 各种哈希函数

- CRC-64 (1975) 在网络中用于错误检测
- MurmurHash (2008) 设计用于快速通用哈希函数
- Google CityHash (2011) 设计为了较短的key(<64 bytes).
- Facebook XXHash (2012) 来自zstd压缩的创建者
- Google FarmHash (2014) 较新版本的CityHash具有更低的冲突率





#### 静态哈希策略

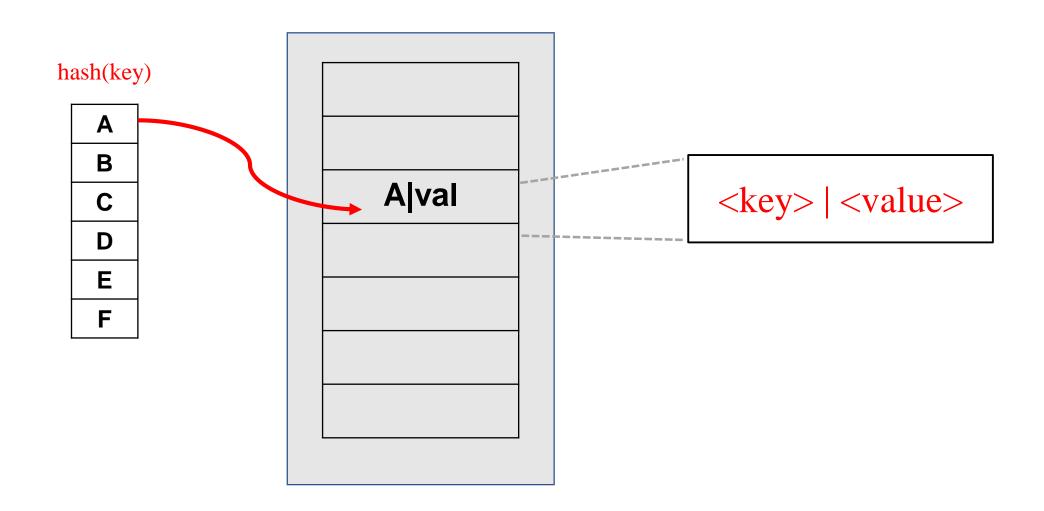


- 哈希策略的设计思路
  - 如何解决哈希冲突
  - 分配大型哈希表与查找/插入键的附加指令见权衡。
- 静态哈希策略
  - Linear Probing 线性探测
  - Robin Hood 罗宾汉
  - Cuckoo 布谷鸟

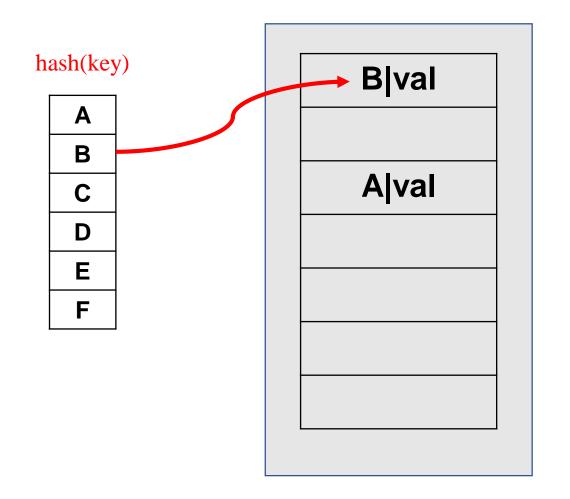


- 单个巨大的插槽表。
- 通过线性搜索表中的下一个空闲槽来解决冲突。
  - 要确定元素是否存在, hash到索引中的某个位置并扫描它。
  - 必须将key存储在索引中才能知道何时停止扫描。
  - 插入和删除是查找的概括。

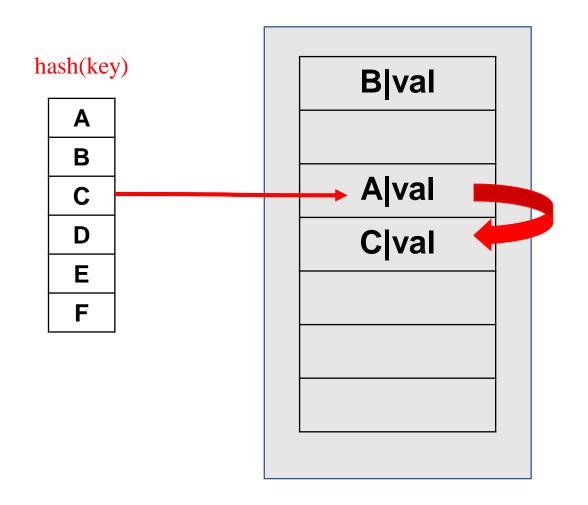




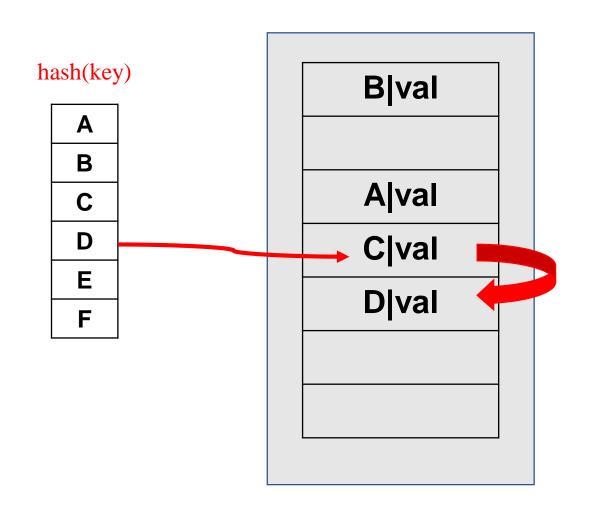




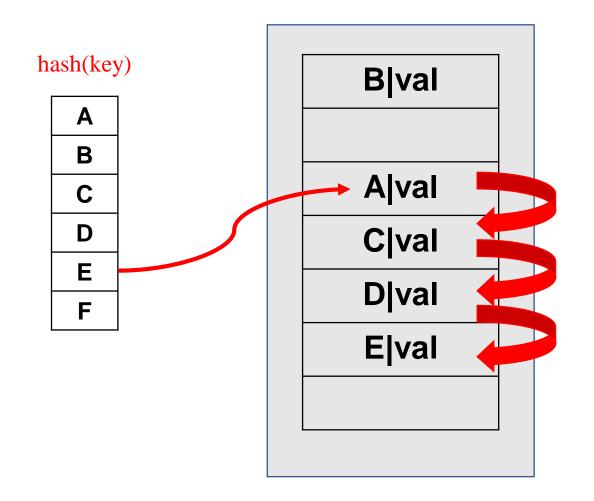




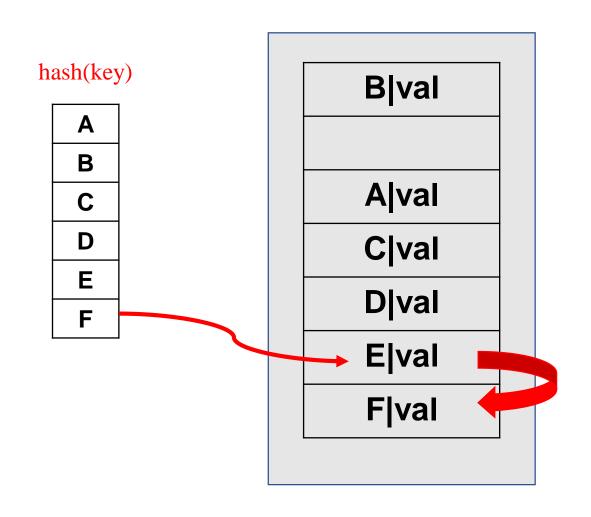




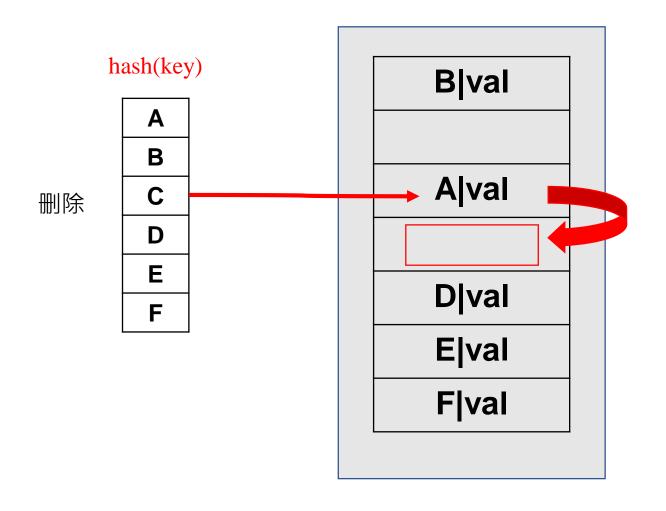




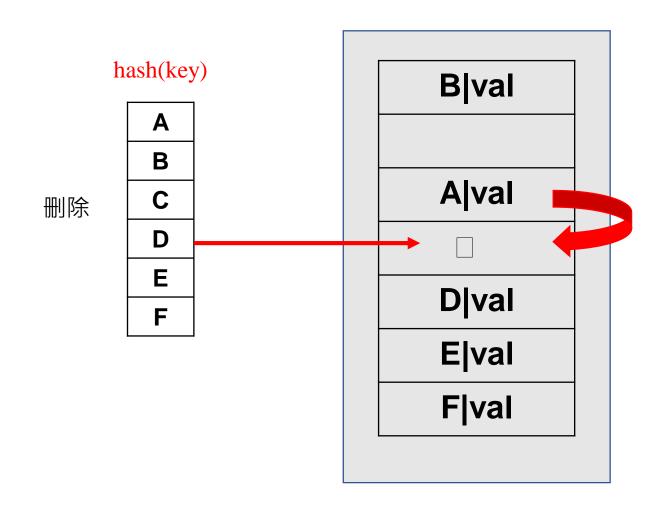






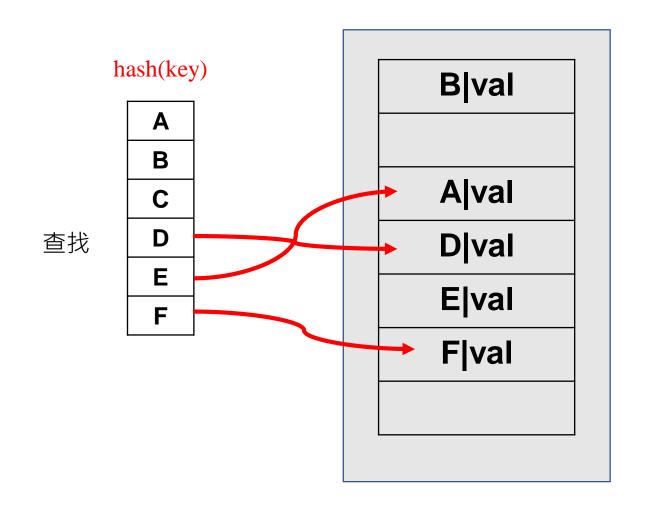






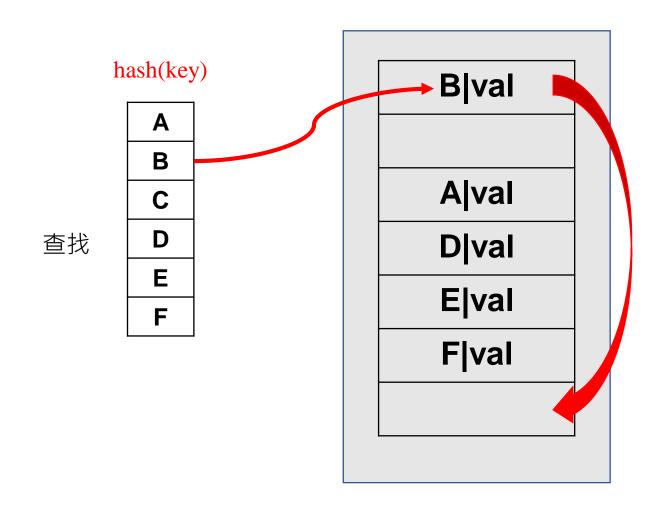
1 设置墓碑





- 1 设置墓碑
- 2 移动





- 1 设置墓碑
- 2 移动



- 如果是非唯一性索引的情况呢?
  - 选择1:单独的链表
  - →将每个键的值存储在单独的存储区中。
  - 选择2: 冗余密钥
  - →**将重复的**键条目存储在哈希表中。

#### 动态哈希策略



• **先前的静**态哈希,哈希表要求**DBMS知道其要存**储的元素数量,否则,如果需要增加/缩小大小,它将重建表,动态哈希表可根据需要调整大小。

- 动态哈希策略
  - Chained Hashing 链式哈希
  - Extendible Hashing 可扩展哈希
  - Linear Hashing 线性哈希

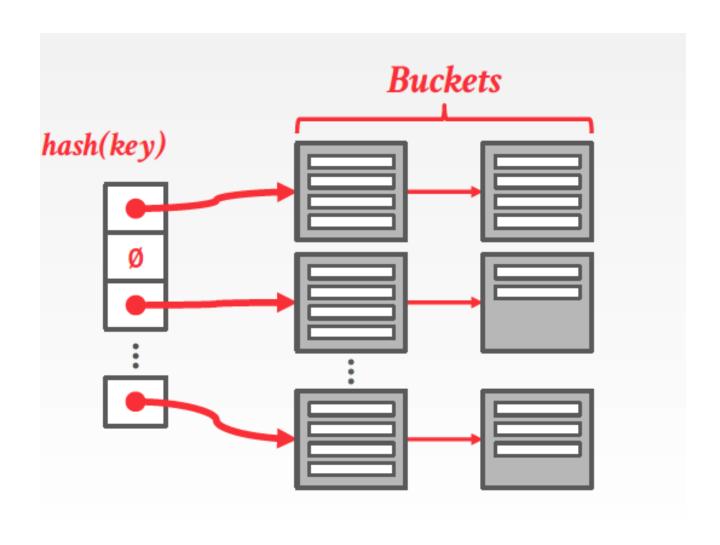
#### 动态哈希策略-链式哈希



- 维护哈希表中每个插槽的存储桶的链表
- 通过将具有相同哈希键的所有元素放入同一存储桶中来解决冲突
  - 要确定是否存在某个元素,请哈希到其存储桶并对其进行扫描
  - 插入和删除是查找的概括

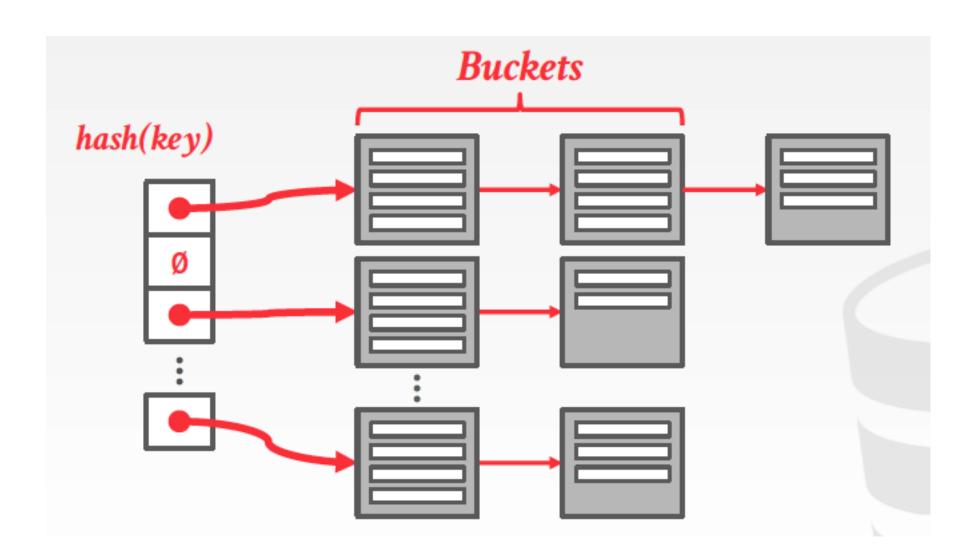
### 动态哈希策略-链式哈希





### 动态哈希策略-链式哈希







# Q&A



Scan CC WeChat to Join the Community 添加CC好友,接受进群邀请

Welcome to follow the GitHub repo 欢迎关注我们的代码仓库 https://github.com/cnosdatabase/cnosdb



