

8 index

- 考点
 - B+树构建, 查询, 插入, 删除-----会画图
 - 主索引和辅助索引概念
 - 稠密和稀疏索引概念
- 基本概念
 - 顺序索引中, 按搜索键的值排序来存储
 - 主索引 (聚集索引): 在顺序排序的文件中, 搜索键指定文件顺序的索引 (类比于在新华字典中的拼音排序)。搜索键通常但不一定是主键
 - 辅助索引 (非聚集索引): 搜索键明确的顺序与文件的顺序不同。必须是稠密索引
 - 如果搜索码不是候选码, 附加间接指针层
 - 索引顺序文件: 在搜索码上有聚集索引
 - 缺点: 创建溢出块, 性能下降, 定期重组
 - 稀疏索引: 只为搜索码的某些值建立索引项。当关系按搜索码排列顺序存储时才可以使用, 也即只有索引时聚集索引时才可以使用稀疏索引。
 - 更少的空间, 更少的插入和删除维护开销
 - 比稠密索引慢
 - 稠密索引: 文件中的每个搜索码值都有一个索引项
 - 索引操作
 - 删除
 - 稠密索引: 删掉搜索码
 - 稀疏索引: 若要删掉搜索码指的, 替换, 没有替换就都删
 - 插入
 - 稠密索引: 搜索, 没有, 插入, 维护
 - 稀疏索引: 插入, 若有新的块创建新的搜索码
- B+树索引文件——平衡的
 - !!! 内部节点之间连接处没有重复的搜索码, 叶子节点和它的父亲连接处有
 - 优点: 插入删除时局部变化不用重组自身
 - 满足以下性质的有根树
 - 根到叶的每条路径长度相同
 - 每个不是根或叶的节点都有 $n/2$ (向上取整) 到 n 个指针
 - 一个叶子节点有 $(n-1)/2$ (向上取整) 到 n 个值
 - 特殊情况
 - 如果根不是叶子, 则它有至少两个指针

- 如果根是叶子，它有0- (n-1) 个值
- k个搜索码，树高 $\log_{n/2} K$
- 插入
 - 分裂，前n/2 (向上取整) 分出
 - 叶子节点和内部节点不同
- 删除
 - 删叶子节点，太空，和兄弟合并
 - 可以合并
 - 合并之后删掉空了的节点，父节点值下移
 - 兄弟节点没位置-----重新分配
 - 删内部节点
 - 内部节点为空但有一个指针
 - 兄弟节点没有位置---重新分配
 - 兄弟最边儿上的指针移给空内部节点，从**父结点中**找值将两个指针分开，放到该内部节点中，兄弟节点里的值放到父节点中

以上内容整理于 [幕布文档](#)