

数据库系统

created by wth

储存和索引

Chapter 10

- ☐ 文件组织，记录怎么表示，定长记录怎么放
- ☐ 不定长记录怎么放

slotted page: 开头是记录属性的数量，前面是record pointers（指向record entry 而不是record ?），后面是records。

record: 前面是定长内容和不定长内容的指针，后面是不定长内容，null bitmap在中间。如果全零的话只有bitmap（全为1）了。

- ☐ 记录删除、插入怎么处理。Free Lists

Chapter 11

- ☐ hash不做要求，主要考索引，B+树，一般3-4叉

n叉指的是一个节点最多有n个指针，所以是n-1个值

非叶节点有 $\lceil n/2 \rceil \sim n$ 个儿子

叶节点有 $\lceil (n-1)/2 \rceil \sim n-1$ 个值（相当于是非叶节点去掉了指针）

- ☐ B+树构建，注意是批量构建还是单个插入。

排序后从底向上建树: 叶子节点取多少视具体情况而定，一般可以尽量满。

- ☐ B+树，插入，删除，merge，split

- ☐ 估计fan out，即最多几叉

bs: block_size ps: pointer_size rs: value_size

n个指针，n-1个值

$(bs - ps) / (rs + ps) + 1$

- ☐ ★估计B+树的高度，最大高度，最小高度，直接估算log 4也差不多；

先算出叶子节点再取log $\lceil n/2 \rceil$ 和log n，直接取log也行

- ☐ ★估计B+树的size，类似。

查询处理和优化

Chapter 12

- ☐ ★估算查询代价, number of seeks, blocks read, blocks written.
transfer 时间和 seek 时间分开
- ☐ 给查询语句转化为关系代数表达式, 哪种是比较优化的, 几个规则
 - 1 选择操作尽量先做。2 投影尽量早做 3 连接 (这里可能是错的, 看一下ppt)

Statistical information about relations. Examples: number of tuples, number of distinct values for an attribute

Statistics estimation for intermediate resultsto compute cost of complex expressions

Cost formulae for algorithms, computed using statistics
- ☐ 外部排序要掌握, 书本的估算有偏差, 看一下勘误表
- ☐ select, sorting, join这几个都可以考, join最重要, sorting第二
- ☐ join的方法, 评价效率主要看外存, 所有join都可能考
- ☐ nested-loop join

transfer: $nr * bs + br$ seek: $nr + br$? 为什么br每次要重新seek, 可能是因为写了数据
- ☐ block nested-loop join 2层外存, 2层内存。

$[br/(m-2)] * bs + br$, $2 * [br / (m-2)]$. []表示下取整; 剩下的两个块一个用来读inner, 一个用了output, 并行
- ☐ indexed nested-loop join, Nr是记录个数 (外关系, 较小的时候用这个比较有效率), c是B+树层数。

Tt: transfer time Ts: seek time c: B+ tree seek time + Tt + Ts

$br * (Tt + Ts) + Nr * c$ 。
- ☐ merge join, 怎么分配内存, 不应该相同。 $\min(br/x + bs/y) \quad x + y = M$

transfer: $br + bs$ seek: $br/x + bs/y$, xy具体取值求个导算算
- ☐ hash join, partition必须能装到内存里, 递归的partition, 看一下书, 勘误表

transfer: $3(br + bs) + 4 * nh$

seek: $2(br / bb + bs / bb)$
- ☐ 要会算有多少块、一块多少内存之类的

并发控制和恢复

- ☐ 会画前趋图, 判断可串行(无环), 判断可恢复 (访问脏数据要求脏数据的事物commit在前, 这里看一下啊, 不太懂), cascadeless (无级联回滚, 不读脏数据)

先用数据的先提交

casceadeless:读之前提交
- ☐ ★ 并发控制, 恢复, 重点, 14 15 16, 可串行调度, 可恢复调度, 两阶段并发控制, 两阶段封锁协议, 充分条件, 严格两阶段封锁协议,

- ☐ ★ 重要结论：遵守两阶段封锁协议是可串行的充分条件，不是必要条件。要会举反例，可以按log point串行。
- ☐ 检测死锁，意向锁，IS IX相容性之类的，有没有环
- ☐ 银行balance冲突很高，个人balance冲突比较小，为了增加concurrency减小公共锁锁定时间
- ☐ 多粒度锁，相容性
- ☐ ★ 基于日志的回复，repeating history, checkpoint, logical undo, ARIS算法，必考。中文422页，英文755页例子，熟读
- ☐ dirty page table, transaction data table
- ☐ Aris算法的checkpoint不会引起脏页面写
- ☐ 3个过程，analysis, redo, undo，看书别看ppt
- ☐ Log Sequence Number