9 查询处理

- 考试要点
  - Concept of query optimization 查询优化概念
  - Query cost of selection operation 查询操作的花销
  - Number of I/Os for nested-loop join, block nested-loop join, indexed nested-loop join 嵌套循环，块嵌套循环，加索引的嵌套循环（课后题）
  - Basic steps of External merge sort 外部归并排序 基本步骤
    - 建立多个排好序的归并段
    - 对归并段进行归并
- 查询花销
  - 受很多因素影响：磁盘访问、CPU 和网络通信的时间
  - 用响应时间来估计，忽略CPU和网络通信时间
  - 计算查询cost
    - 简单起见，只用磁盘存取代价来度量查询执行计划的代价：传送磁盘块数+搜索磁盘次数

      Disk cost can be estimated as:
      - 寻道次数 * 平均寻道成本
      - 读取的块数 * 平均块读取成本
      - 写入的块数 * 平均块写入成本
      - 为简单起见，我们仅使用从**磁盘传输的块数**和**寻道数**作为成本度量
      - t_T – time to transfer one block
        - 简单起见，假设读取和写入的时间一样
      - t_S – time for one seek
      - Cost for b block transfers + S 次 seeks
        $b * t_T + S * t_S$
- 选择运算
  - 使用文件扫描和索引的选择

	算　法	开　销	原　因
A1	线性搜索	$t_s + b_r * t_T$	一次初始搜索加上 b_r 个块传输，b_r 表示在文件中的块数量
A1	线性搜索，码属性等值比较	平均情形 $t_s + (b_r/2) * t_T$	因为最多一条记录满足条件，所以只要找到所需的记录，扫描就可以终止。在最坏的情形下，仍需要 b_r 个块传输
A2	B⁺树主索引，码属性等值比较	$(h_i + 1) * (t_T + t_s)$	（其中 h_i 表示索引的高度）。索引查找穿越树的高度，再加上一次 I/O 来取记录；每个这样的 I/O 操作需要一次搜索和一次块传输
A3	B⁺树主索引，非码属性等值比较	$h_i * (t_T + t_s) + b * t_T$	树的每层一次搜索，第一个块一次搜索。b 是包含具有指定搜索码记录的块数。假定这些块是顺序存储（因为是主索引）的叶子块并且不需要额外搜索
A4	B⁺树辅助索引，码属性等值比较	$(h_i + 1) * (t_T + t_s)$	这种情形和主索引相似
A4	B⁺树辅助索引，非码属性等值比较	$(h_i + n) * (t_T + t_s)$	（其中 n 是所取记录数。）索引查找的代价和 A3 相似，但是每条记录可能在不同的块上，这需要每条记录一次搜索。如果 n 值比较大，代价可能会非常高
A5	B⁺树主索引，比较	$h_i * (t_T + t_s) + b * t_T$	和 A3，非码属性等值比较情形一样
A6	B⁺树辅助索引，比较	$(h_i + n) * (t_T + t_s)$	和 A4，非码属性等值比较情形一样

图 12-3　选择算法代价估计

- 连接运算
  - 嵌套循环连接——两个for，外层关系为r，内层关系为s，b_r表示guanxir中元组的磁盘块数，n_r表示r的元组数
    - 最坏情况下：缓冲区只能容纳每个关系的一个数据块
      - 传输次数：$n_r * b_s + b_s$
      - 搜索磁盘次数：$n_r + b_r$
    - 最好情况下
      - 传输次数：$b_r + b_s$
      - 磁盘搜索：2
  - 块嵌套循环连接：先分块，再对块中的元组进行配对
    - 最坏情况下
      - 传输次数：$b_r * b_s + b_r$
      - 磁盘搜索：$2b_r$
    - 最好的情况：内存可以容纳内层关系
      - 传输次数：$b_r + b_s$
      - 磁盘搜索：2
  - 索引嵌套循环链接：内层循环的连接属性上有索引，用索引查找来替代文件扫描
    - 最坏情况下：只能容纳r的一块和索引的一块
      - $b_r(t_r + t_s) + n_r * c$，c是那些对单次选择操作的估计值

以上内容整理于 幕布文档