

Chapter 15. Query Processing

query] → parser → translator → 简化表达式 exp. $E_1 \rightarrow E_2 = T_1(E_1) \rightarrow T_2(E_2)$
 query c - data. output → evaluation engine
 $t_1 \text{ transfers} + t_2 \text{ seek}, \text{page}$

Select: ① linear worst: $b/t+t_s$
 ② $\min(b/t+t_s)$
 ③ $b/t \cdot t_b \cdot t_s$
 ④ $b/t \cdot t_b \cdot t_s + b/t$
 ⑤ $b/t \cdot t_b \cdot t_s + b/t \cdot t_b \cdot t_s$
 ⑥ $b/t \cdot t_b \cdot t_s + b/t \cdot t_b \cdot t_s + b/t \cdot t_b \cdot t_s$
 ⑦ $b/t \cdot t_b \cdot t_s + b/t \cdot t_b \cdot t_s + b/t \cdot t_b \cdot t_s + b/t \cdot t_b \cdot t_s$
 ⑧ $b/t \cdot t_b \cdot t_s + b/t \cdot t_b \cdot t_s$
 ⑨ $b/t \cdot t_b \cdot t_s + b/t \cdot t_b \cdot t_s$
 ⑩ $b/t \cdot t_b \cdot t_s + b/t \cdot t_b \cdot t_s$
 ⑪ $b/t \cdot t_b \cdot t_s + b/t \cdot t_b \cdot t_s$

DB+ 权衡：① 直接上等位目 ② B+ 树帮助，B+树辅助。③ B+树帮助，B+树辅助，同时对树的插入、删除操作进行平衡。④ B+树帮助，B+树辅助，同时对树的插入、删除操作进行平衡，同时对树的插入、删除操作进行平衡。

DB+，B+树：① 通过增加 b/t，减少 t_b 和 t_s，从而减少磁盘访问次数，提高查询效率。
 DB+，B+树：② 通过增加 b/t，减少 t_b 和 t_s，从而减少磁盘访问次数，提高查询效率。

DB+，B+树：③ 通过增加 b/t，减少 t_b 和 t_s，从而减少磁盘访问次数，提高查询效率。

DB+，B+树：④ 通过增加 b/t，减少 t_b 和 t_s，从而减少磁盘访问次数，提高查询效率。

DB+，B+树：⑤ 通过增加 b/t，减少 t_b 和 t_s，从而减少磁盘访问次数，提高查询效率。

DB+，B+树：⑥ 通过增加 b/t，减少 t_b 和 t_s，从而减少磁盘访问次数，提高查询效率。

DB+，B+树：⑦ 通过增加 b/t，减少 t_b 和 t_s，从而减少磁盘访问次数，提高查询效率。

DB+，B+树：⑧ 通过增加 b/t，减少 t_b 和 t_s，从而减少磁盘访问次数，提高查询效率。

DB+，B+树：⑨ 通过增加 b/t，减少 t_b 和 t_s，从而减少磁盘访问次数，提高查询效率。

DB+，B+树：⑩ 通过增加 b/t，减少 t_b 和 t_s，从而减少磁盘访问次数，提高查询效率。

DB+，B+树：⑪ 通过增加 b/t，减少 t_b 和 t_s，从而减少磁盘访问次数，提高查询效率。

DB+，B+树：⑫ 通过增加 b/t，减少 t_b 和 t_s，从而减少磁盘访问次数，提高查询效率。

DB+，B+树：⑬ 通过增加 b/t，减少 t_b 和 t_s，从而减少磁盘访问次数，提高查询效率。

Chapter 16. Recovery System

log record: $< T_i \text{ starts}, < T_i \text{ commit}, < T_i \text{ abort} >$
 log record: $< T_i \cdot X, V_1, V_2 > \rightarrow \text{physical}$

这样把写到 db 前，要先把因果关系的稳定 storage 保留在 history 中，再写入 db。这样 repeat history。根据 history 读取，输出历史 log，再数据。< T_i, V>

log record buffer: $R \leftarrow < T_i \text{ commit}, < T_i \text{ ends} >$ 进入 stable storage，事务才结束。

log record buffer: $R \leftarrow < T_i \text{ abort}, < T_i \text{ ends} >$ 进入 stable storage，事务才结束。

log record buffer: $R \leftarrow < T_i \text{ check point open}, < T_i \text{ ends} >$ 进入 stable storage，事务才结束。

log record buffer: $R \leftarrow < T_i \text{ check point close}, < T_i \text{ ends} >$ 进入 stable storage，事务才结束。

log record buffer: $R \leftarrow < T_i \text{ check point end}, < T_i \text{ ends} >$ 进入 stable storage，事务才结束。

log record buffer: $R \leftarrow < T_i \text{ force policy}, < T_i \text{ ends} >$ 进入 stable storage，事务才结束。

log record buffer: $R \leftarrow < T_i \text{ fuzzy check pointing}, < T_i \text{ ends} >$ 进入 stable storage，事务才结束。

log record buffer: $R \leftarrow < T_i \text{ logical unit update}, < T_i \text{ ends} >$ 进入 stable storage，事务才结束。

log record buffer: $R \leftarrow < T_i \text{ operation begin}, < T_i \text{ ends} >$ 进入 stable storage，事务才结束。

log record buffer: $R \leftarrow < T_i \text{ operation end}, < T_i \text{ ends} >$ 进入 stable storage，事务才结束。

log record buffer: $R \leftarrow < T_i \text{ operation abort}, < T_i \text{ ends} >$ 进入 stable storage，事务才结束。

log record buffer: $R \leftarrow < T_i \text{ begin}, < T_i \text{ ends} >$ 进入 stable storage，事务才结束。

log record buffer: $R \leftarrow < T_i \text{ end}, < T_i \text{ ends} >$ 进入 stable storage，事务才结束。

log record buffer: $R \leftarrow < T_i \text{ dump}, < T_i \text{ ends} >$ 进入 stable storage，事务才结束。

log record buffer: $R \leftarrow < T_i \text{ page fault}, < T_i \text{ ends} >$ 进入 stable storage，事务才结束。

Chapter 17. Transactions

ACID: 原子性 atomic, 一致性 consistency, 隔离性 isolation, 持久性 durability.

事务：① 深度 Join Tree: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

事务：② 左右 Deep Join Trees: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

事务：③ 左右深度嵌套 Join Tree: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

事务：④ 左右深度嵌套 Join Tree: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

事务：⑤ 左右深度嵌套 Join Tree: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

事务：⑥ 左右深度嵌套 Join Tree: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

事务：⑦ 左右深度嵌套 Join Tree: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

事务：⑧ 左右深度嵌套 Join Tree: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

事务：⑨ 左右深度嵌套 Join Tree: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

事务：⑩ 左右深度嵌套 Join Tree: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

事务：⑪ 左右深度嵌套 Join Tree: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

事务：⑫ 左右深度嵌套 Join Tree: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

事务：⑬ 左右深度嵌套 Join Tree: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

事务：⑭ 左右深度嵌套 Join Tree: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

事务：⑮ 左右深度嵌套 Join Tree: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

事务：⑯ 左右深度嵌套 Join Tree: $T: br/m/n \rightarrow br/m/n \rightarrow br/m/n \rightarrow br/m/n$

Chapter 18. Concurrency Control

2-PL: 脱机两阶段提交两阶段提交

Chapter 19. Pipelines

$M > \frac{b}{m} + 1$, 不用缓冲区

① 只有 X 锁 ② 第一个锁住，随后所有锁住的锁都释放

② unlock 不释放锁

