

12.2

$$\Pi_{T.branch \text{ name}}((\Pi_{branch \text{ name}, assets}(\rho_T(branch)))$$

$$\bowtie_{T.assets > S.assets} (\Pi_{assets}(\sigma_{(branch \text{ city}='Brooklyn')}(\rho_S(branch)))))$$

此表达式对可能的最小数据量执行 **theta** 连接。它通过将连接的右侧操作数限制为仅布鲁克林中的那些分支来实现, 并且还消除了两个操作数中不需要的属性。

12.3 [b]

Block nested-loop join:

If r_1 is the outer relation, we need $\lceil \frac{800}{M-1} \rceil * 1500 + 800$ disk accesses,
if r_2 is the outer relation we need $\lceil \frac{1500}{M-1} \rceil * 800 + 1500$ disk accesses.

13.4

由于连接的关联和交换属性, 无论我们怎么连接它们, 由 r_1 , r_2 和 r_3 的连接产生的关系都是相同的。将 r_1 与 r_2 连接将产生至多 1000 个元组的关系, 因为 C 是 r_2 的关键。同样, 将该结果与 r_3 连接将产生至多 1000 个元组的关系, 因为 E 是 r_3 的关键。因此, 最终关系最多只有 1000 个元组。

•计算此连接的有效方式是为关系 r_2 创建属性 C 的索引, 为 r_3 创建 E 的索引。然后对于 r_1 中的每个元组, 我们执行以下操作:

- a . 使用 r_2 的索引查找最多匹配 r_1 的 C 值的一个元组。
- b . 使用 E 上创建的索引在 r_3 中查找最多一个元组, 该元组与 r_2 中 E 的唯一值匹配。

13.15

使用 (dept name, building) 上的索引, 我们找到第一个元组 (构建 "Watson" 和 dept 名称 "Music")。然后, 只要构建小于 "Watson", 我们就会按照指针检索连续的元组。从检索到的元组中, 拒绝不满足条件的那些 (预算 < 55000)。