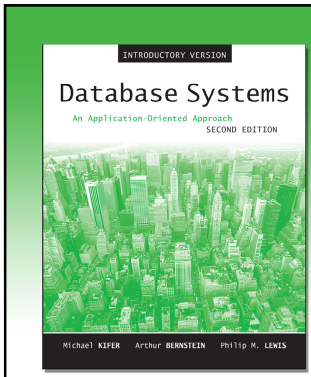


Chapter 14

An Overview of Query Optimization

PEARSON Addison Wesley

Figure 14.1 Typical architecture for DBMS query processing.

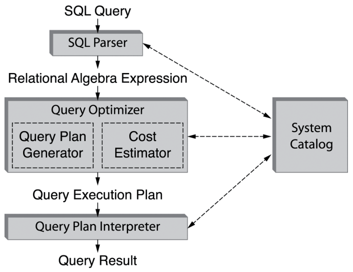


Figure 14.2 Query trees for relational expressions (14.3) through (14.6).

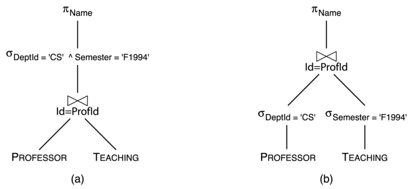


Figure 14.2 Query trees for relational expressions (14.3) through (14.6). (cont'd)

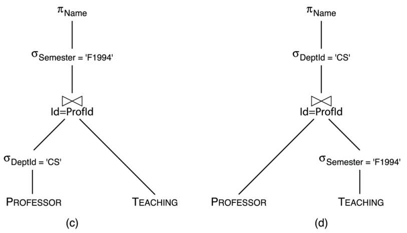


Figure 14.3 Query execution plans for relational expressions (14.3) through (14.6).

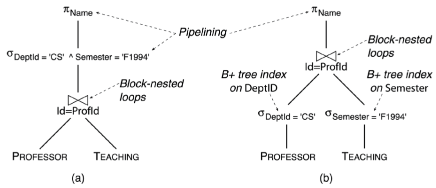


Figure 11.4 Query execution plans for relational expressions (14.3) through (14.6). (cont'd)

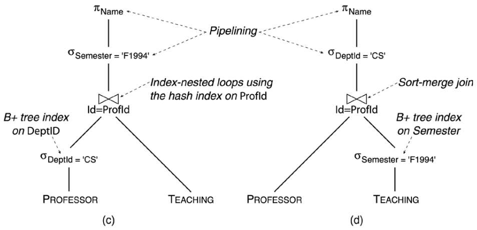


Figure 14.4 Transforming a query tree into a logical query execution plan.

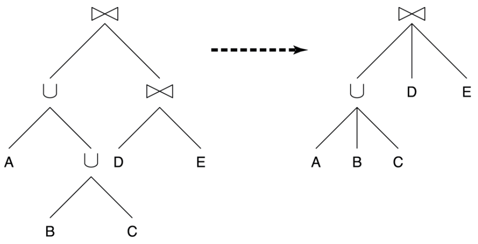
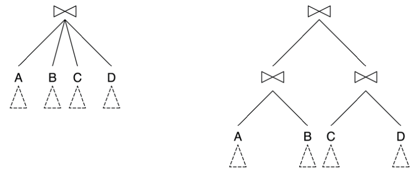


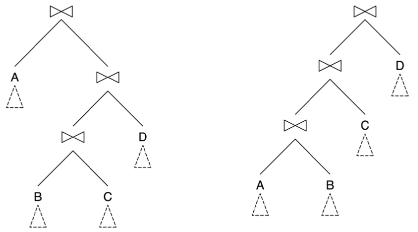
Figure 14.5 Logical plan and three equivalent query trees.



(a) Logical Query Plan

(b) An Equivalent Query Tree

Figure 14.5 Logical plan and three equivalent query trees. (cont'd)



(c) Another Equivalent Query Tree (d) Yet Another Equivalent Tree: *Left-Deep Query Tree*

Figure 14.6 Heuristic search of the query execution plan space.

Input: *A logical plan* $E_1 \bowtie \cdots \bowtie E_N$
Output: *A "good" left-deep plan* $(\ldots((E_{i_1} \bowtie E_{i_2}) \bowtie E_{i_3}) \bowtie \ldots) \bowtie E_{i_N}$

```
1-Plans := all 1-relation plans
Best := all 1-relation plans with lowest cost
for ($i := 1$; $i < N$; $i$++) do
    // Below, $\overset{\text{meth}}{\bowtie}$ denotes join marked with an implementation method, meth*
    Plans := { best $\overset{\text{meth}}{\bowtie}$ 1-plan | best $\in$ Best;  1-plan $\in$ 1-Plans, where
                                  1-plan is a plan for some $E_j$ that has not
                                  been used so far in best }
          Best := { plan | plan $\in$ Plans, where plan has the lowest cost }
end
return Best;
```