Chapter 14
An Overview of Query Optimization

Figure 14.1 Typical architecture for DBMS query processing.

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

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

Figure 14.4 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.

Figure 14.6  Heuristic search of the query execution plan space.

Input: A logical plan $E_1 \bowtie \cdots \bowtie E_k$
Output: A "best" left-deep plan $(\cdots (E_i \bowtie \cdots (E_2 \bowtie E_1)))$.

Variables:
- Plans: set of all left-deep plans
- Best: best 1-plan
- $1$-plans: set of all 1-planet plans
- $t$: iteration number
- $e_t$: element

Algorithm:
1. Initialize Plans: $\{\text{best 1-plan}\}$
2. For each plan $p$ in Plans:
   a. If $p$ is a 1-plan for some $e_t$ that has not been used so far in Best:
      i. Add $p$ to Plans, where plan has the lowest cost.
   b. If $p$ is better than Best:
      i. Update Best to $p$.
3. Return Best.