LECTURE 5

HEURISTIC OPTIMIZATION OF QUERY TREES (1/2)

- Many different relational algebra expressions (hence different query trees) can be **equivalent** i.e. they can correspond to same query
- Query parser will generate a standard initial query tree to correspond to SQL query w/o doing any optimization
- **The job of the heuristic query optimizer is to transform this initial query tree into a final query tree that is efficient to execute**

HEURISTIC OPTIMIZATION OF QUERY TREES (2/2)

- Optimizer must include rules for equivalence among relational algebra expressions that can be applied to the initial tree
- **The heuristic query optimization rules then utilize these equivalence expressions to transform initial tree into final optimized query tree**

GENERAL TRANSFORMATION RULES FOR RELATIONAL ALGEBRA OPERATIONS (1/3)

- If two relations have same set of attributes in a different order but two relations represent same information, the relations are considered to be **equivalent**

- Rules are listed below:-
  1. ***Cascade of σ*** :- A conjunctive selection condition can be broken up into a cascade of individual σ operations

  2. ***Commutativity of σ*** :- The σ operation is commutative

  3. ***Cascade of π*** :- In cascade sequence of π operations, all but the last one can be ignored

GENERAL TRANSFORMATION RULES FOR RELATIONAL ALGEBRA OPERATIONS (2/3)

3. ***Commuting σ with π*** :- If selection condition involves only those attributes in projection list, two operations can be commuted

5. ***Commutativity of X and JOIN*** :- the JOIN is commutative as is also the X operation. Though the order of the attributes are not the same resulting from the two joins or two CARTESIAN products, but meaning is same because order of attributes is not important

6. ***Commuting σ with JOIN (or X)*** :- If all attributes in selection condition involve only the attributes of one of the relations being joined, the two operations can be commuted.

7. ***Commuting π with JOIN (or X) :-*** If join condition involves only those attributes in projection list, then two operations can be commuted.

GENERAL TRANSFORMATION RULES FOR RELATIONAL ALGEBRA OPERATIONS (3/3)

8. ***Commutativity of set operations*** :- Union and Intersection operations are commutative but set difference is not

9. ***Associativity of JOIN, X, U, Intersection*** :- The four operations are individually associative

10. ***Commuting σ with set operations*** :- the σ operation commutes with union, intersection and set difference

11. ***π operation commutes with U***

12. ***Converting a (σ, X) sequence into JOIN***

OUTLINE OF HEURISTIC ALGEBRAIC OPTIMIZATION ALGORITHM (1/3)

- Utilize rules before to transform initial query tree into optimized tree that is efficient to execute

Step 1:

Using Rule 1, break up any SELECT operations with conjunctive conditions into a cascade of SELECT operations. This permits a greater degree of *freedom in moving SELECT operations down different branches* of the tree

Step 2:

Using rules 2, 4, 6 and 10 concerning commutativity of SELECT with other operations, *move each SELECT* operation as far down the query tree *as is permitted* by the attributes involved in select condition

OUTLINE OF HEURISTIC ALGEBRAIC OPTIMIZATION ALGORITHM (2/3)

Step 3

Using rules 5 and 9 concerning commutativity and associativity of binary operations, rearrange leaf nodes of tree using following criteria.

- ***First***, position the leaf node relations with the most restrictive SELECT operations so they are executed first in query tree representation
- ***Second***, make sure that ordering of leaf nodes does not produce CARTESIAN PRODUCT operations

Step 4:

Using rule 12 *combine* a *CARTESIAN PRODUCT* operation *with* a subsequent *SELECT* operation in the tree into a *JOIN operation*, if condition represents a join condition

OUTLINE OF HEURISTIC ALGEBRAIC OPTIMIZATION ALGORITHM (3/3)

Step 5:

Using rules 3, 4, 7 and 11 concerning cascading of PROJECT and commuting of PROJECT with other operations, break down and *move lists of projection attributes down the trees* as far as possible by creating new PROJECT operations as needed. Only those *attributes needed in query result* and in subsequent operations in query tree should be *kept after each PROJECT* operation

Step 6:

Identify subtrees that represent groups of operations that can be executed by a single algorithm