ROHIT KUMAR SINGH (UE20CS093)

Sol:1 1. PROJECT:

Cost Calculation: Cost_Project(n, m) = m * n

Explanation: The cost of the PROJECT operation is proportional to the number of tuples (n) in the input relation multiplied by the number of attributes (m) to the project. Each tuple needs to be processed to retain only the specified attributes

2. UNION:

- Cost Function: Cost_Union(n1, n2)
- Cost Calculation: Cost_Union(n1, n2) = n1 + n2
- Explanation: The cost of the UNION operation is the sum of the tuples in both input relations. This assumes that duplicate elimination is not costly; otherwise, additional operations may be involved.

3. INTERSECTION

- Cost Function: Cost_Intersection(n_1, n_2)
- Cost Calculation: min(n1, n2)

Explanation: The cost of the INTERSECTION operation is the minimum of the tuples in both input relations since we only need to keep the common tuples between them.

4. SET DIFFERENCE

- Cost Function: Cost_set Difference(n1, n2)

Cost Calculation: Cost_set Difference(n1, n2) = n1

- Explanation: The cost of the SET DIFFERENCE operation is equivalent to the no. of tuples in the first relation since we retain all tuples

Sol: 4) Query Rewrite: Oracle has a feature called Query Rewrite that can automatically rewrite certain types of queries to improve performance. For Ex. it can rewrite subqueries as joins as materialized views to optimize query execution.

Query Hints: Oracle provides query hints that allow database administrators or developers to influence the optimizer's choice for specific queries

CARTESIAN PRODUCT

- Cost Function: Cost_Cartesian Product (n1, n2)
- Cost Calculation: Cost_Cartesian Product (n1, n2) = W*n1*n2
- Explanation: The cost of the CARTESIAN PRODUCT operation in the product of the no. of tuples in both relation.

Sol:2 Representation of a relational Algebra Expression:

Leaf Nodes
Internal Nodes
Edges

The rules for transformation of query trees during query optimization.

1. selection Pushdown
2. Projection Pushdown
3. Join Reorder
4. Join Decomposition
5. Union "
6. Set operation Pushdown

Sol:3 Due to commutative and associative rules of the given operation. All permutation are valid so

Diff. join orders = n! = 10!

left deep Join tree is a binary tree in which right child of each non-leaf node is always a base relation.

left deep trees are also = n! = 10!