Query Optimization

Objectives of Query optimization

- Minimization of response time of query (time taken to produce the results to user's query).
- Maximize system throughput (the number of requests that are processed in a given amount of time).
- Reduce the amount of memory and storage required for processing.
- Increase parallelism.

Query Parsing and Translation

Initially, the SQL query is scanned. Then it is parsed to look for syntactical errors and correctness of data types. If the query passes this step, the query is decomposed into smaller query blocks. Each block is then translated to equivalent relational algebra expression.

Steps for Query Optimization

Query optimization involves three steps

1. query tree generation
2. plan generation
3. query plan code generation.

Step 1 − Query Tree Generation

A query tree is a tree data structure representing a relational algebra expression. The tables of the query are represented as leaf nodes. The relational algebra operations are represented as the internal nodes. The root represents the query as a whole.

During execution, an internal node is executed whenever its operand tables are available. The node is then replaced by the result table. This process continues for all internal nodes until the root node is executed and replaced by the result table.

For example, let us consider the following schemas −

EMPLOYEE

EmpID	EName	Salary	DeptNo	DateOfJoining

DEPARTMENT

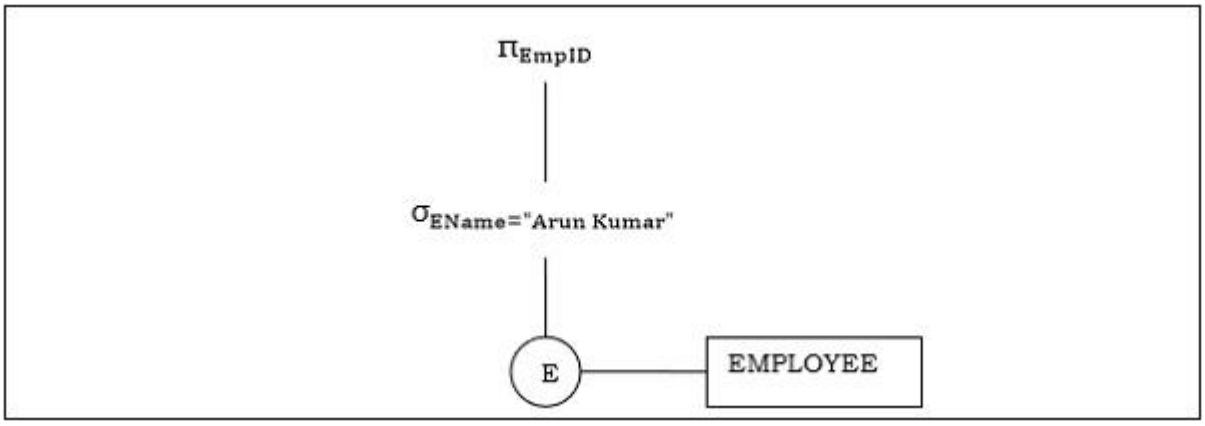
DNo	DName	Location

Example 1

Let us consider the query as the following.

pi_{EmpID} (sigma_{EName = "ArunKumar"} {(EMPLOYEE)})

The corresponding query tree will be –

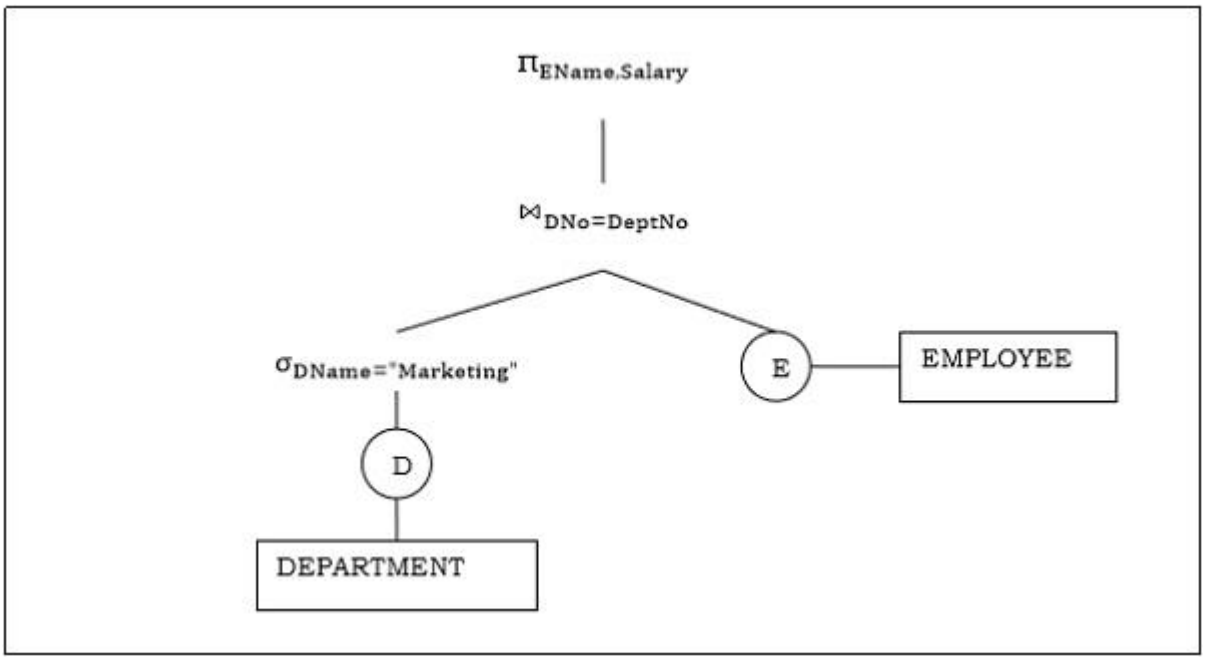


Example 2

Let us consider another query involving a join.

pi_{EName, Salary} (sigma_{DName = l "Marketing"} {(DEPARTMENT)}) {DNo=DeptNo}{(EMPLOYEE)}

Following is the query tree for the above query.



Step 2 – Query Plan Generation

After the query tree is generated, a query plan is made. A query plan is an extended query tree that includes access paths for all operations in the query tree. Access paths specify how the relational operations in the tree should be performed.

Besides, a query plan also states how the intermediate tables should be passed from one operator to the next, how temporary tables should be used and how operations should be pipelined/combined.

Step 3– Code Generation

Code generation is the final step in query optimization. It is the executable form of the query, whose form depends upon the type of the underlying operating system. Once the query code is generated, the Execution Manager runs it and produces the results.

Some of the rules for Query optimization are –

- Perform select and project operations before join operations. This is done by moving the select and project operations down the query tree. This reduces the number of tuples available for join.

- Perform the most restrictive select/project operations at first before the other operations.

- Avoid cross-product operation since they result in very large-sized intermediate tables.