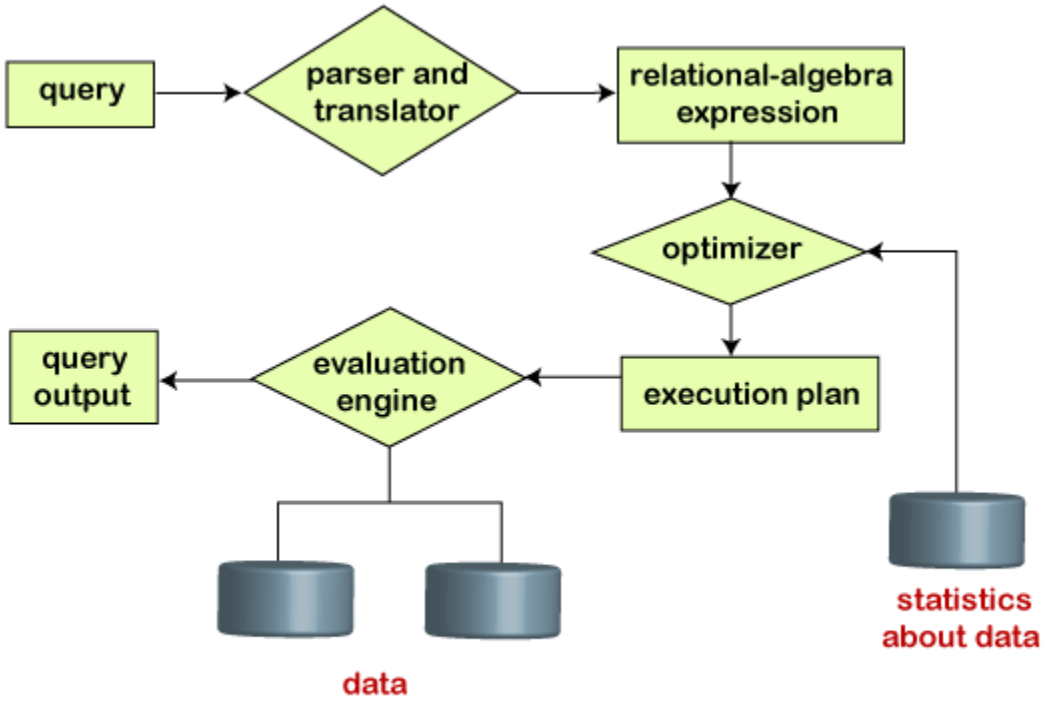
Describe or explain query processing.

Query processing

- It is a process of transforming a high-level query into low level expression (Relational Algebra).
- In query processing, it takes various steps for fetching the data from the database. The steps involved are:
  - ✓ Parsing and Translation
  - ✓ Optimization
  - ✓ Evaluation



Steps in query processing

- Parser checks the syntax of query and verifies attribute name and relation name and determines whether it is formulated according to the syntax rules of the query language.
- After that, Translator translates the query into its internal form (relational algebra).
- Optimization is a process in which multiple query execution plan for satisfying a query are examined and most efficient query plan is satisfied for execution.
- Database catalog stores the execution plans and then optimizer passes the lowest cost plan for execution to evaluation engine.
- Finally, evaluation engine runs the query and display the required result.

Explain the measures of query cost. OR Explain the measures of finding out the cost of a query in query processing.

Measures of query cost

- The cost of query evaluation can be measured in terms of a number of different resources including disk access, CPU time to execute a query and in a distributed or parallel database system the cost of communication.
- The response time for a query evaluation plan i.e the time required to execute the plan (assuming no other activity is going on) on the computer would account for all these activities.
- In large database system, however disk accesses are usually the most important cost, since disk access are slow compared to in memory operation.
- Moreover, CPU speeds have been improving much faster than have a disk speed.
- Therefore, it is likely that the time spent in disk activity will continue to dominate the total time to execute a query.
- Estimating the CPU time is relatively hard, compared to estimating disk access cost.
- Therefore, disk access cost a reasonable measure of the cost of a query evaluation plan.
- We use the number of block transfers from disk as a measure of actual cost.
- To simplify our computation, we assume that all transfer of blocks has same cost.
- To get more precise numbers we need to distinguish between sequential I/O where blocks read are contiguous on disk and random I/O where blocks are non-contiguous and an extra seek cost must be paid for each disk I/O operations.
- We also need to distinguish between read and write of blocks since it takes more time to write a block on disk than to read a block from disk.

Discuss different search algorithm for selection operation. OR Describe linear search and binary search algorithm for selection operation.

- There are two scan algorithms to implement the selection operation:
    1. Linear search
    2. Binary search

Linear search

- In a linear search, the systems scan each file block and tests all records to see whether they satisfy the selection condition.

- The cost of linear search in terms of number of I/O operations is b_r where b_r is the number of blocks in file(relation).
- Cost of linear search (worst case) = b_r
- For a selection on a key attribute, the system can terminate the scan if the requires record is found, without looking at the other records of the relation.
- Selection on key attribute has an average cost of $b_r/2$.
- This algorithm can be applied to any file(relation) regardless of the ordering of records in the file(relation) or selection condition.
- It may be a slower algorithm than any another algorithm.

Binary search

- If the file is ordered on attribute and the selection condition is an equality comparison on the attribute, we can use a binary search to locate the records that satisfy the condition.
- The number of blocks that need to be examined to find a block containing the required record is $\log_2(b_r)$.
- If the selection is on non-key attribute more than one block may contain required records and the cost of reading the extra blocks has to be added to the cost estimate.
- This algorithm is faster than linear search algorithm.

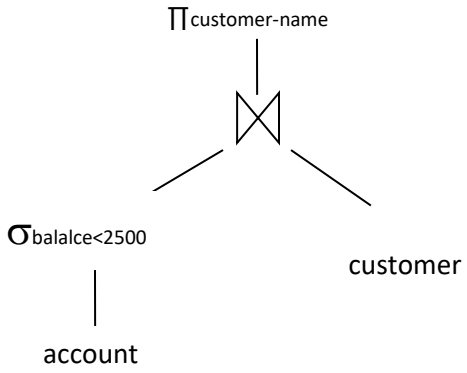
Discuss various steps involved in query evaluation. OR Describe query evaluation process. OR Explain evaluation expression process in query optimization.

- There are two methods for the evaluation of expression
    1. Materialization
    2. Pipelining

Materialization

- In this method we start from bottom of the tree and each expression is evaluated one by one in bottom to top order. The result of each expression is materialized (stored) in temporary relation (table) for later use.

$$\Pi_{\text{customer-name}} (\sigma_{\text{balalce<2500}} (\text{account}) \bowtie \text{customer})$$



- In our example, there is only one such operation, selection operation on account.
- The inputs to the lowest level operation are relations in the database.
- We execute these operations and we store the results in temporary relations.
- We can use these temporary relations to execute the operation at the next level up in the tree, where the inputs now are either temporary relations or relations stored in the database.
- In our example the inputs to join are the customer relation and the temporary relation created by the selection on account.
- The join can now be evaluated, creating another temporary relation.
- By repeating the process, we will finally evaluate the operation at the root of the tree, giving the final result of the expression.
- In our example, we get the final result by executing the projection operation at the root of the tree, using as input the temporary relation created by the join. Evaluation just described is called materialized evaluation, since the results of each intermediate operation are created and then are used for evaluation of the next level operations.
- The cost of a materialized evaluation is not simply the sum of the costs of the operations involved. To compute the cost of evaluating an expression is to add the cost of all the operation as well as the cost of writing intermediate results to disk.
- The disadvantage of this method is that it will create temporary relation (table) and that relation is stored on disk which consumes space on disk.
- It evaluates one operation at a time, starting at the lowest level.

Pipelining

- We can reduce the number of temporary files that are produced by combining several relations operations into pipeline operations, in which the results of one operation are passed along to the next operation in the pipeline. Combining operations into a pipeline eliminates the cost reading and writing temporary relations.
- In this method several expressions are evaluated simultaneously in pipeline by using the result of one operation passed to next without storing it in a temporary relation.

$$\Pi_{\text{customer-name}} (\sigma_{\text{balalce<2500}} (\text{account}) \bowtie \text{customer})$$

- First it will compute records having balance less than 2500 and then pass this result directly to project without storing that result in any temporary relation (table). And then by using this result it will compute the projections on customer-name.
- It is much cheaper than materialization because in this method no need to store a temporary relation to disk.

Explain the method of query optimization. OR Describe query optimization process.

Query optimization

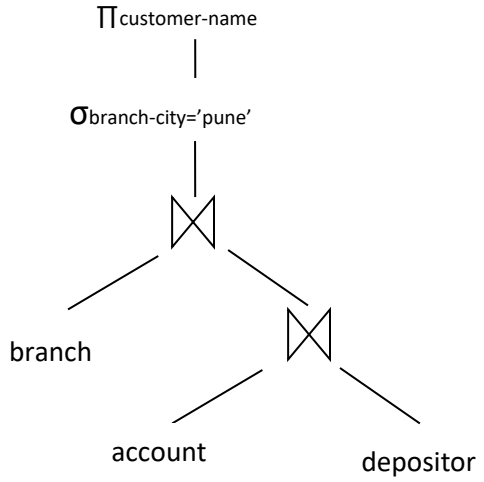
- It is a process of selecting the most efficient query evaluation plan from the available possible plans for processing a given query.
- There are two phases of query optimization
    1. Optimization which occurs at the relational algebra level. In this phase the system will find an expression that is equivalent to the given expression but more efficient to execute.
    2. Selecting a detailed strategy for processing the query such as choosing algorithm and specific indices, etc.
- For example, consider the relational algebra expression for the query
- "Find the name of all customers who have account at any branch located in Pune"

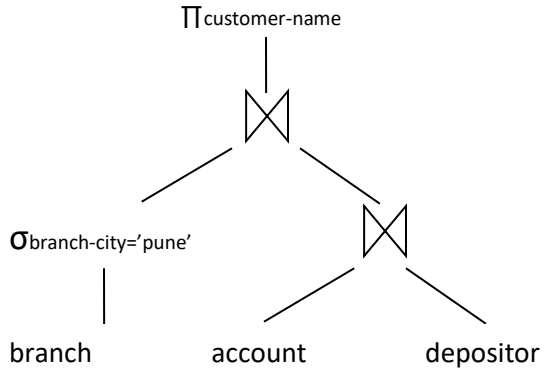
$$\Pi_{customer\_name}\ (\ \sigma_{branch\_city="pune"}\ (branch \bowtie (account \bowtie depositor)\ )\ )$$

- The above query may be written as below

$$\Pi_{customer\_name}\ (\ \sigma_{branch\_city="pune"}\ (branch) \bowtie (account \bowtie depositor)\ )$$

- In the second algebra expression the size of intermediate result is smaller than first because it will only contain the records of pune branch city. Final result of both the expression is same.





- To choose from the different query evaluation plan, the optimizer has to estimate the cost of each evaluation plan.
- Optimizer use statically information about the relation such as relation size and index depth to make a good estimate of the cost of a plan.

Explain transformation of relational expression to equivalent relational expression.

- Two relational algebra expressions are said to be equivalent (same) if on every legal database operation, the two expressions give the same set of tuples (records). Sequence of records may be different but no of records must be same.

Equivalence rules

- This rule says that expressions of two forms are same.
- We can replace an expression of first form by an expression of the second form.
- The optimizer uses equivalence rule to transform expression into other logically same expression.
- We use

  θ1, θ2, θ3 and so on to denote condition
  L1, L2, L3 and so on to denote list of attributes (columns)
  E1, E2, E3 and so on to denote relational algebra expression.

Rules 1

- Combined selection operation can be divided into sequence of individual selections. This transformation is called cascade of σ.

  $\sigma_{\theta 1 \wedge \theta 2}(E) = \sigma_{\theta 1}(\sigma_{\theta 2}(E))$

Rules 2

- Selection operations are commutative.

  $\sigma_{\theta 1}(\sigma_{\theta 2}(E)) = \sigma_{\theta 2}(\sigma_{\theta 1}(E))$

Rules 3

- If more than one projection operation is used in expression then only the outer projection operation is required. So skip all the other inner projection operation.

  $\Pi_{L1}(\Pi_{L2}(\ldots(\Pi_{Ln}(E))\ldots)) = \Pi_{L1}(E)$

Rules 4

- Selection operation can be joined with Cartesian product and theta join.

  $\sigma_{\theta}(E1 \bowtie E2) = E1 \bowtie_{\theta} E2$

  $\sigma_{\theta 1}(E1 \bowtie_{\theta 2} E2) = E1 \bowtie_{\theta 1 \wedge \theta 2} E2$

Rules 5

- Theta operations are commutative.

  $E1 \bowtie_{\theta 2} E2 = E2 \bowtie_{\theta 2} E1$

Rules 6

- Natural join operations are associative.

$(E1 \bowtie E2) \bowtie E3 = E1 \bowtie (E2 \bowtie E3)$

- Theta join operations are associative.

$(E1 \bowtie_{\theta 1} E2) \bowtie_{\theta 2 \wedge \theta 3} E3 = E1 \bowtie_{\theta 1 \wedge \theta 3} (E2 \bowtie_{\theta 2} E3)$

Rules 7

- The selection operation distribute over theta join operation under the following condition
  - ✓ It distribute when all the attributes in the selection condition θ0 involves only the attributes of the one of the expression (says E1) being joined.

    $\sigma_{\theta 0} (E1 \bowtie E2) = (\sigma_{\theta 0} (E1)) \bowtie_{\theta} E2$
  - ✓ It distributes when the selection condition θ1involves only the attributes of E1 and θ2 involves only the attributes of E2.
  - ✓ $\sigma_{\theta 1 \wedge \theta 2} (E1 \bowtie_{\theta} E2) = (\sigma_{\theta 1}(E1) \bowtie_{\theta} (\sigma_{\theta 2} (E2)))$