Parul® University

Unit 7- Query Processing & Optimization

Subject Code: 303105203

Prof. S.W.Thakare
Assistant Professor,
Computer science & Engineering

Parul®
University

DIGITAL LEARNING CONTENT

CHAPTER-7

Query Processing & Optimization

Parul®
University

DIGITAL LEARNING CONTENT

Topics

- Query processing
- Steps in query processing
- Measures of query cost
- Selection operation
- Evaluation of expressions
- Query optimization
- Transformation of relational expressions
- Cost base optimization approach

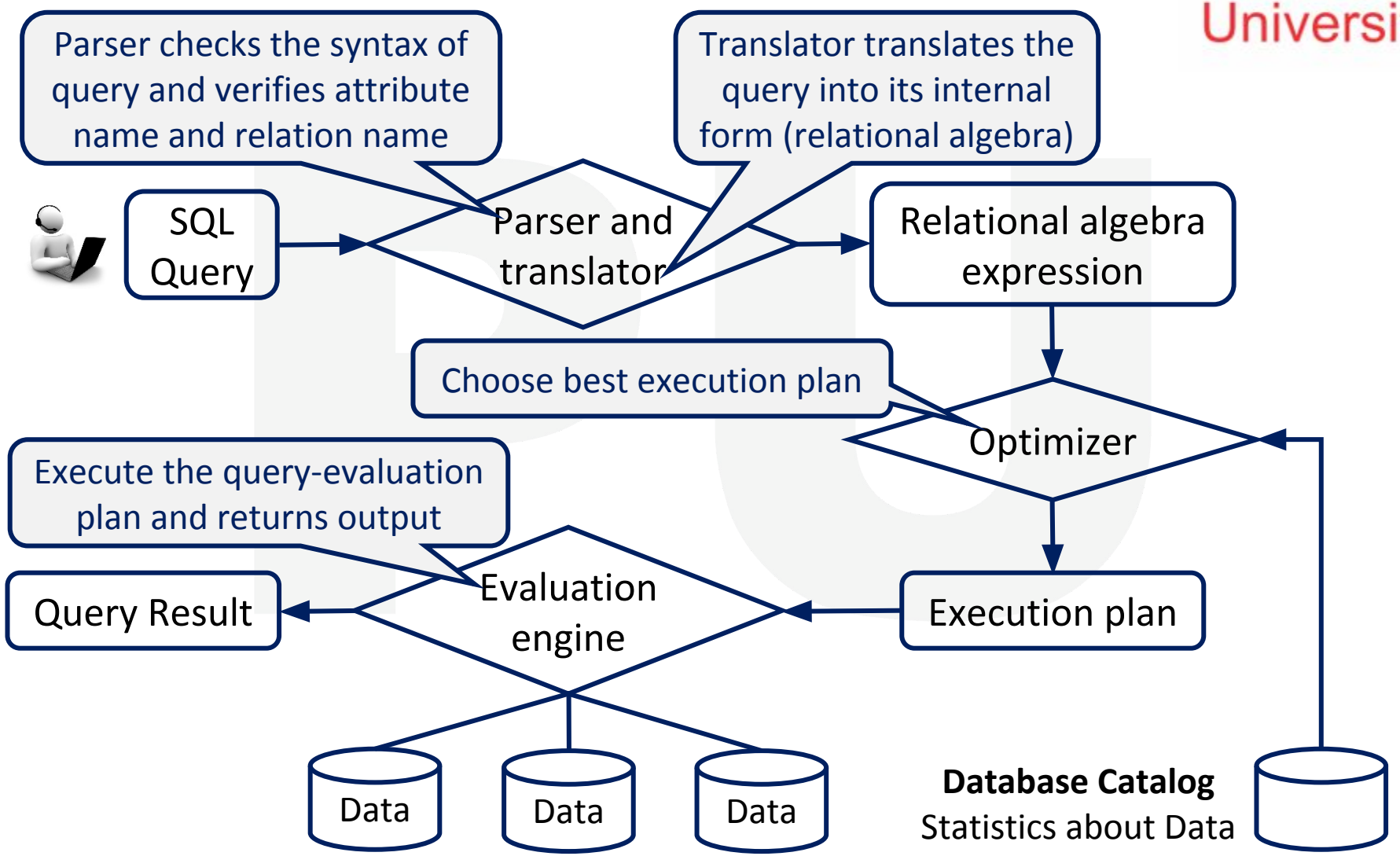
Query Processing

- **Query Processing** is process to convert high level queries to low level so machine can understand and perform the action that are requested by the user.

- It is used to extract data from database and to fetch data it takes three steps:

1. Parsing and Translation
2. Optimization
3. Evaluation

Steps in Query Processing



Step in Query Processing

1. **Parsing and Translation:**

   - SQL is suitable for humans.
   - Relational Algebra is suitable for system.
   - First step in query processing is to convert SQL to Relational Algebra Expression.

- **Parsing(Parser):**
  - Check Syntax
  - Check Schema Elements
- **Translation(Translator)**
  - Parse Tree ⟶ Relational Algebra

Step in Query Processing

2. Optimization(Optimizer):

- Selects the best Query Evaluation Plan to evaluate the query.
- Generate Query Evaluation Plan for all possible option
  - **Query Evaluation plan** = Query Tree + Algorithms

Step in Query Processing

3. Evaluation Engine:

- Evaluates the Query Plan (selected by Optimizer) and fetches the data from database.

Measures of Query cost

- Total time taken by statement/query to execute and to fetch data from database is Query Cost.

- Some factors are:

  - **Communication cost:**
    - Applicable to distributed/parallel system.

  - **CPU Cycles:**
    - Difficult to calculate
    - CPU speed improves at much faster rate as compared to Disk speed

Measures of Query cost

- **Disk Access:**
  - Dominates the total time to execute a query

- **Disk Access Cost:**
  - No. of seeks
  - No. of blocks Read
  - No. of Blocks Write

Note: Generally cost of writing is greater than cost of reading.

Selection operation

- **File Scan:** Search algorithm that used to locate and retrieve data that satisfy a selection condition in a file.
- **Symbol for Selection operator:** σ (Sigma)
- **Syntax:** $\sigma_{condition}$ (Relation)
- Searching algorithm for Selection Operation:

1. Linear Search(A1)
2. Binary Search(A2)

Selection operation

1. **Linear Search(A1):**
   - This algorithm will search and scan all blocks available and tests all records/data to determine whether or not they satisfy the selection condition.
     - **Cost(A1)** = B_R (**worst case**)

       where B_R denotes number of blocks
   - If the condition is on a **Key(primary) attribute**, then system can stop searching if desired record found.
     - **Cost(A1)** = $B_R/2$ (**best case**)
   - If the condition is on **non (primary) key attribute**, then multiple blocks may contain desired records, then the price of scanning such blocks have to be added to the estimate value.
   - This is slower than Binary Search.

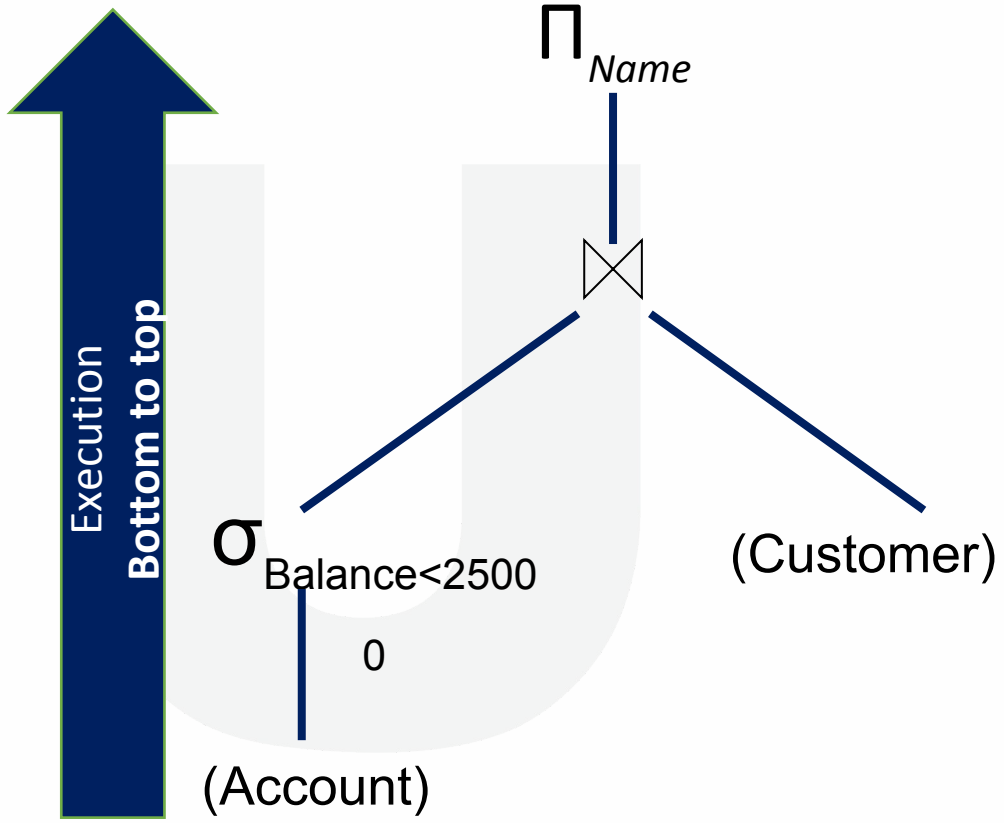
Selection operation

2. **Binary Search(A2):**
   - File (relation) ordered based on attribute A (primary index).
     - **Cost(A2)** = $\log_2(B_R)$
   - This is faster than Linear Search.

Evaluation of expressions

- Query(Expression) may contain multiple operations and due to that solving query (Expression) will be difficult.
- To evaluate such type of Query we have to solve one by one in proper order.
- There are two methods to evaluate multiple operations expression:

1. Materialization
2. Pipelining



Materialization

- **Materialization** starts the bottom of the expression and performs a single operation at a time.
- **Materialized**(store in temporary relation) each intermediate result of all operations performed and use this result as input to evaluate next-level operations.
- The **cost of materialization** can be quite high as overall cost can be compute as:

  Overall Cost = Sum of Costs of individual operations + Cost of writing intermediate results to the disk
- **Disadvantages of Materializations are:**
  - Due to intermediate results, it creates lots of temporary relations.
  - It performs many Input/Output operations.

Pipelining

- **In Pipelining**, the output of one operation is passed as input to another operation. i.e. it forms a queue.
- As the output of one operation is passed to the next operation in the **Pipelines, the number of intermediate temporary relations will be reduced.**
- Performing operations in Pipeline **eliminates the cost of writing and reading temporary relations.**

- It can be executed in two ways:
  - Demand Driven(Lazy Evaluation)
  - Producer Driven(Eager Pipelining)

Pipelining(Cont..)

- **Demand Driven(Lazy Evaluation):**

  System repeatedly requests for tuples from operation at the top of pipeline.

- **Producer Driven(Eager Pipelining):**

  Operations do not wait for request to produce tuples, but generate the tuples eagerly.

Query Optimization

- **Query optimization** is the process of choosing the best evaluation plan having lowest cost from the available multiple plans.
- **For Example,**

Π_{Name} ($\sigma_{Balance<25000}$ (Account) ⋈ (Customer))

Efficient plan

2 records

4 records

Π_{Name} ($\sigma_{Balance<25000}$ (Account ⋈ Customer))

4 records

4 records

Customer

CID	ANO	Name
CSE1	A1	Jay
CSE2	A2	Abhi
CSE3	A3	Parth
CSE4	A4	Pratik

Account

ANO	Balance
A1	30000
A2	10000
A3	20000
A4	40000

Query Optimization Approaches

- **Cost Based Optimization (Exhaustive Search Optimization):**
  - In this, it initially generates all possible plans and then select the best plan from it.
  - Its provides the best solution.

- **Heuristic Based Optimization:**
  - These technique is less expensive.
  - To decide optimized query execution plan there are some heuristic rules:

1. To reduce the number of tuples, **Perform selection as early as possible.**
2. To reduce the number of attributes, **Perform projection as early as possible.**
3. Perform most restrictive selection and join operations (i.e. with smallest result size) before other similar operations.

Transformation of relational expressions

- Two relational algebra expressions are said to be **equivalent** if the two expressions generate the same set of tuples on every legal database instance.

- An **equivalence rule** says that expressions of two forms are equivalent.
  - Can replace expression of first by second, or vice versa.

Transformation of relational expressions(Conti..)

- **For Example,**

Customer

CID	ANO	Name
CSE1	A1	Jay
CSE2	A2	Abhi
CSE3	A3	Parth
CSE4	A4	Pratik

Account

ANO	Balance
A1	30000
A2	10000
A3	20000
A4	40000

Π_{Name} ($\sigma_{Balance<25000}$ (Account ⋈ Customer))

=

Π_{Name} ($\sigma_{Balance<25000}$ (Account ⋈ Customer))

Customer

Name
Meet
Jay

Equivalence Rules

1. **Conjunctive(Combined) selection operations can be deconstructed** into sequence of individual selections. This is known as **Cascade of σ.**

Customer

CID	ANO	Name	Balance
CS1	1	Jay	30000
CS2	2	Abhi	10000
CS3	3	Parth	20000
CS4	4	Pratik	40000

$\sigma_{ANO<3 \wedge Balance<20000}(Customer)$

$\equiv$

$\sigma_{ANO<3}(\sigma_{Balance<20000}(Customer))$

OUTPUT

Output

CID	ANO	Name	Balance
CSE2	2	Abhi	10000

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

Equivalence Rules

2. **Selection operations are commutative**

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

3. If **many projection used in expression** then only **the last in a sequence of projection operations is required. So Omit all other projection operation.**

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

4. **Selection operation can be combined with Cartesian products and theta joins.**

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

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

Equivalence Rules

5. **Theta-join operations (and natural joins) are commutative.**

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

6. **Natural join operations are associative**

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

Equivalence Rules

7. **The selection operation distributes over the theta join operation under the following two conditions**:

(a) When all the attributes in θ_0 involve only the attributes of one of the expressions (E_1) being joined.

$$\sigma_{\theta 0}(E_1 \bowtie_{\theta} E_2) = (\sigma_{\theta 0}(E_1)) \bowtie_{\theta} E_2$$

(b) When θ_1 involves only the attributes of E_1 and θ_2 involves only the attributes of E_2.

$$\sigma_{\theta 1} \wedge_{\theta 2} (E_1 \bowtie_{\theta} E_2) = (\sigma_{\theta 1}(E_1)) \bowtie_{\theta} (\sigma_{\theta 2}(E_2))$$

Equivalence Rules

8. The projection operation distributes over the theta join operation as follows:

(a) if θ involves only attributes from $L_1 \cup L_2$:

$$\prod_{L_1 \cup L_2}(E_1 \bowtie_\theta E_2) = (\prod_{L_1}(E_1)) \bowtie_\theta (\prod_{L_2}(E_2))$$

(b) Consider a join $E_1 \bowtie_\theta E_2$.

- Let L_1 and L_2 be sets of attributes from E_1 and E_2, respectively.
- Let L_3 be attributes of E_1 that are involved in join condition θ, but are not in $L_1 \cup L_2$, and
- Let L_4 be attributes of E_2 that are involved in join condition θ, but are not in $L_1 \cup L_2$.

$$\prod_{L_1 \cup L_2}(E_1 \bowtie_\theta E_2) = \prod_{L_1 \cup L_2}(\prod_{L_1 \cup L_3}(E_1) \bowtie_\theta (\prod_{L_2 \cup L_4}(E_2)))$$

Equivalence Rules

9. The set operations union and intersection are commutative

$$E_1 \cup E_2 = E_2 \cup E_1$$
$$E_1 \cap E_2 = E_2 \cap E_1$$

Note: set difference is not commutative

10. Set union and intersection are associative.

$$(E_1 \cup E_2) \cup E_3 = E_1 \cup (E_2 \cup E_3)$$
$$(E_1 \cap E_2) \cap E_3 = E_1 \cap (E_2 \cap E_3)$$

Equivalence Rules

11. The selection operation distributes over ∪, ∩ and –.

$\sigma_\theta (E_1 - E_2) = \sigma_\theta (E_1) - \sigma_\theta(E_2)$

and similarly for ∪ and ∩ in place of –

Also: $\sigma_\theta (E_1 - E_2) = \sigma_\theta(E_1) - E_2$

and similarly for ∩ in place of –, but not for ∪

12. The projection operation distributes over union

$\Pi_L(E_1 \cup E_2) = (\Pi_L(E_1)) \cup (\Pi_L(E_2))$

Cost Based Optimization Approach

Query optimization is the process of choosing the most efficient or the most favourable type of executing an SQL statement.

It is an art of science for applying rules to rewrite the tree of operators that is invoked in a query and to produce an optimal plan.

A plan is said to be optimal if it returns the answer in the least time or by using the least space.

Features of the cost-based optimization-

- The cost-based optimization is based on the cost of the query that to be optimized.
- The query can use a lot of paths based on the value of indexes, available sorting methods, constraints, etc.
- The aim of query optimization is to choose the most efficient path of implementing the query at the possible lowest minimum cost in the form of an algorithm.

Cost Based Optimization Approach

- The aim of query optimization is to choose the most efficient path of implementing the query at the possible lowest minimum cost in the form of an algorithm.
- The cost of executing the algorithm needs to be provided by the query Optimizer so that the most suitable query can be selected for an operation.

The cost of an algorithm also depends upon the cardinality of the input

Cost Based Optimization Approach

- **Cost Estimation:**
  To estimate the cost of different available execution plans or the execution strategies the query tree is viewed and studied as a data structure that contains a series of basic operation which are linked in order to perform the query.
- The cost of the operations that are present in the query depends on the way in which the operation is selected such that, the proportion of select operation that forms the output.
- It is also important to know the expected cardinality of an operation output.
- The cardinality of the output is very important because it forms the input to the next operation.

Cost Based Optimization Approach

The cost of optimization of the query depends upon the following-

1.Cardinality-

Cardinality is known to be the number of rows that are returned by performing the operations specified by the query execution plan. The estimates of the cardinality must be correct as it highly affects all the possibilities of the execution plan.

2.Selectivity-

Selectivity refers to the number of rows that are selected. The selectivity of any row from the table or any table from the database almost depends upon the condition. The satisfaction of the condition takes us to the selectivity of that specific row.

3.Cost-

Cost refers to the amount of money spent on the system to optimize the system. The measure of cost fully depends upon the work done or the number of resources used.

Cost Based Optimization Approach

Cost Components Of Query Execution:

The following are the cost components of the execution of a query-

1.Access cost to secondary storage-

This can be the cost of searching, reading, or writing data blocks that originally found on the secondary storage, especially on the disk. The cost of searching for records in a file also depends upon the type of access structure that file has.

2.Memory usage cost-

The cost of memory usage can be calculated simply by using the number of memory buffers that are needed for the execution of the query.

3.Storage cost-

The storage cost is the cost of storing any intermediate files(files that are the result of processing the input but are not exactly the result) that are generated by the execution strategy for the query.

Cost Based Optimization Approach

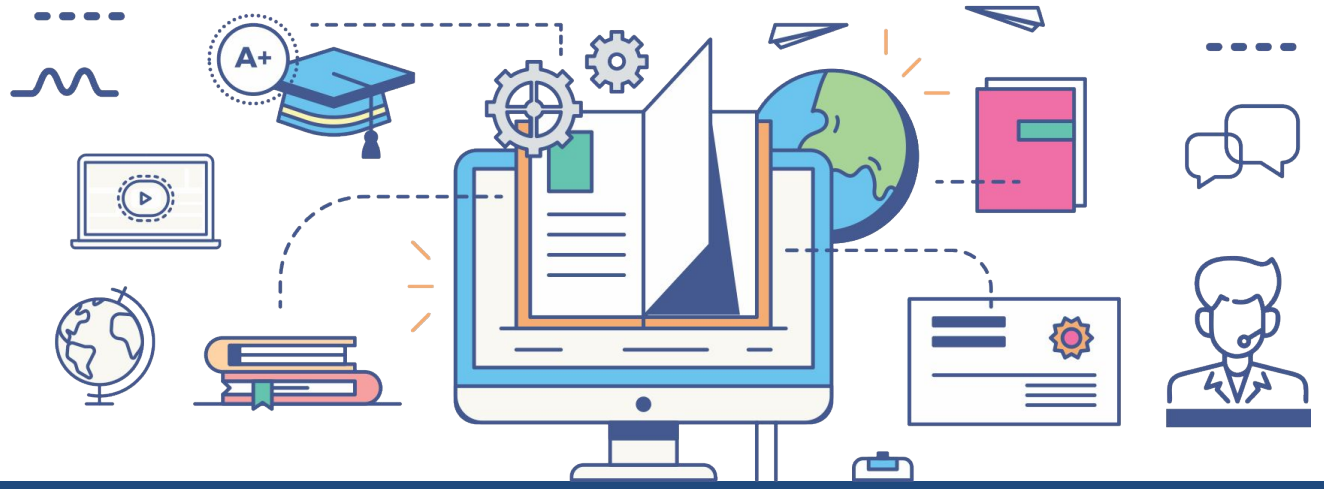
4.Computational cost-

This is the cost of performing the memory operations that are available on the record within the data buffers. Operations like searching for records, merging records, or sorting records. This can also be called the CPU cost.

5.Communication cost-

This is the cost that is associated with sending or communicating the query and its results from one place to another. It also includes the cost of transferring the table and results to the various sites during the process of query evaluation.

DIGITAL LEARNING CONTENT



Parul® University

www.paruluniversity.ac.in