Chapter 7

Query Processing

Database Management System (DBMS)
4th Semester , BEIT & BCE

Compiled by :

Ankit Bhattarai, Assistant Professor
ankitbhattarai@cosmoscollege.edu.np

Cosmos College of Management & Technology

Syllabus

Full marks: 100

Internal: 50
Final: 50

Internal Marks:

Theory: 30
Practical: 20

Unit	Contents	Hours	Remarks
1.	Introduction	4	
2.	Data Models	4	
3.	Relational Model	4	
4.	Relational Database Query Languages	8	Important
5.	Database Constraints and Relational Database Design	8	Important
6.	Security	3	
7.	Query Processing	3	
8.	File Organization & Indexing	4	
9.	Crash Recovery	3	
10.	Transaction Processing & Concurrency Control	4	
11.	Advanced Database Concepts	3	

Chapter 7
Main Topics

(3 hours)

- Steps in Query Processing
- Query Cost Estimation
- Equivalence/ Transformation Rules
- Operator Tree
- Query Optimization

Query Processing

- The main goal of creating a database is to store the related data at one place, access and manipulate them as and when it is required by the user.

- Accessing and manipulating the data should be done efficiently i.e. it should be accessed easily and quickly.

- It refers to the activities involved in retrieving data from the database as

➢ SQL query translation into low level language implementing relational algebra

➢ Query Execution

Query Processing: Steps

- The phases/ steps of query processing includes

1. Parsing & translation
2. Optimization
3. Evaluation

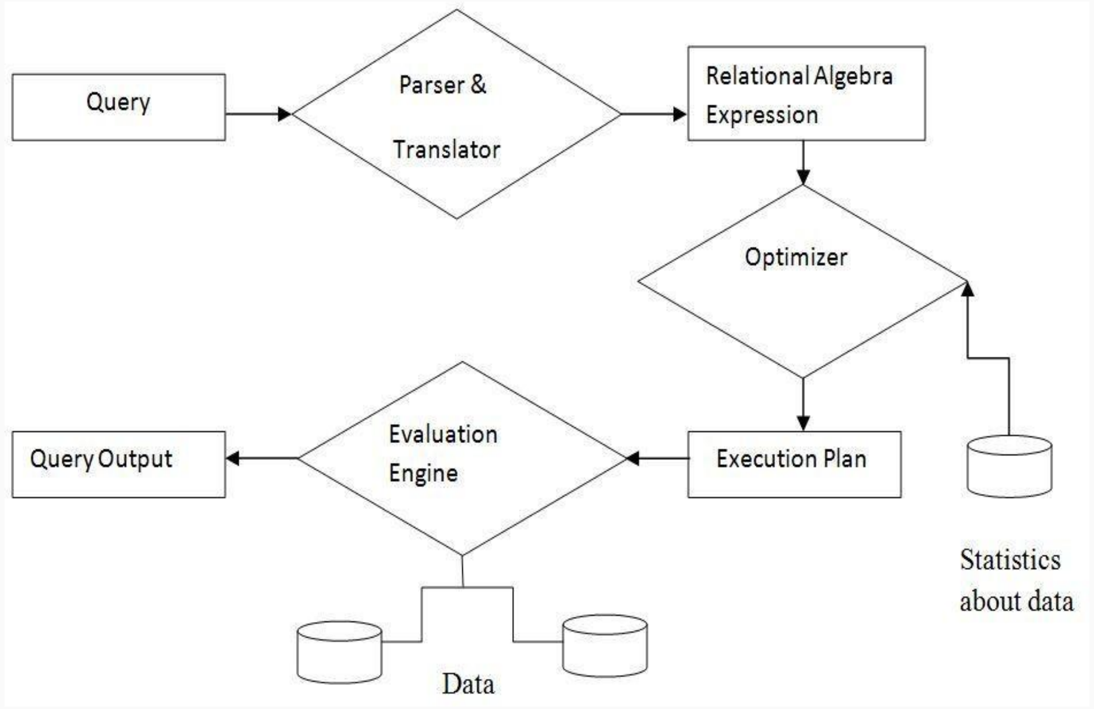


Figure I. Steps in query Processing

Query Processing: Steps

1. **Parsing & translation**

- To translate a given query into its internal form
- Similar to the work performed by the parser of a compiler.
- Internal form: The parser check the syntax of the user's query, verifies that the relation is formulated according to the syntax rules of the query language.

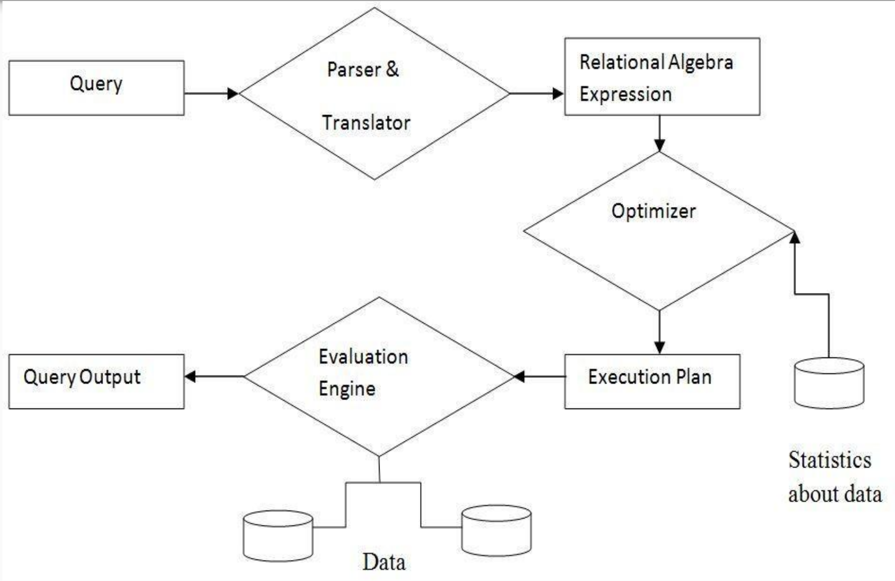


Figure I. Steps in query Processing

Query Processing: Parsing & Translation

- It is then translated into relational algebra.
- Translation involves conversion of high level query to low level instruction in relational algebra.
- **Example** : Select book_title, price From Book Where price > 400

This query can be translated into either of the following relational-algebra expressions.

$$\pi_{\text{book\_title, price}} (\sigma_{\text{price} > 400} (\text{Book}))$$

$$\sigma_{\text{price} > 400} (\pi_{\text{book\_title, price}} (\text{Book}))$$

Query Processing: Steps

2. Optimization

- It is a process in which multiple query execution plan for satisfying a query are examined and most efficient query plan is satisfied for execution.
- Optimizer uses the statistical data stored as part of data dictionary.
- The statistical data are information about the size of the table, the length of records, the indexes created on the table, etc.

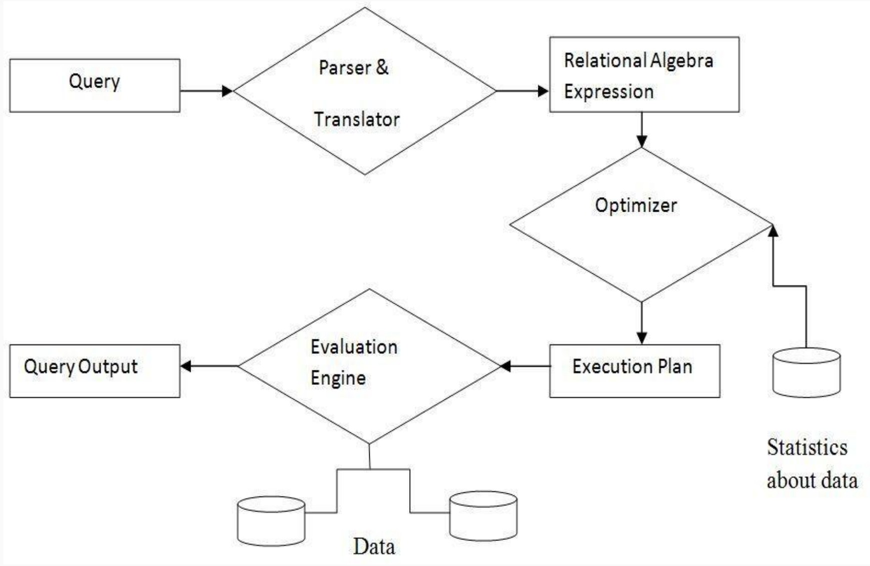


Figure I. Steps in query Processing

Query Processing: Optimization

- A relational algebra expression may have many equivalent expressions.

  $\pi_{\text{book\_title, price}} (\sigma_{\text{price} > 400} (\text{Book}))$ Is equivalent to

  $\sigma_{\text{price} > 400} (\pi_{\text{book\_title, price}} (\text{Book}))$

- The process of choosing a suitable one with lowest cost is known as query optimization.
- Cost is estimated using the statistical information from database catalog. The different statistical information is number of tuples in each relation, size of tuples etc.
- At last, choose the one with the cheapest possible evaluation plan.

Query Processing: Steps

3. Evaluation

- The query evaluation engine takes execution plan, executes that plan and return the result.
- The query output is the set of tuples (results) as requested by the query.

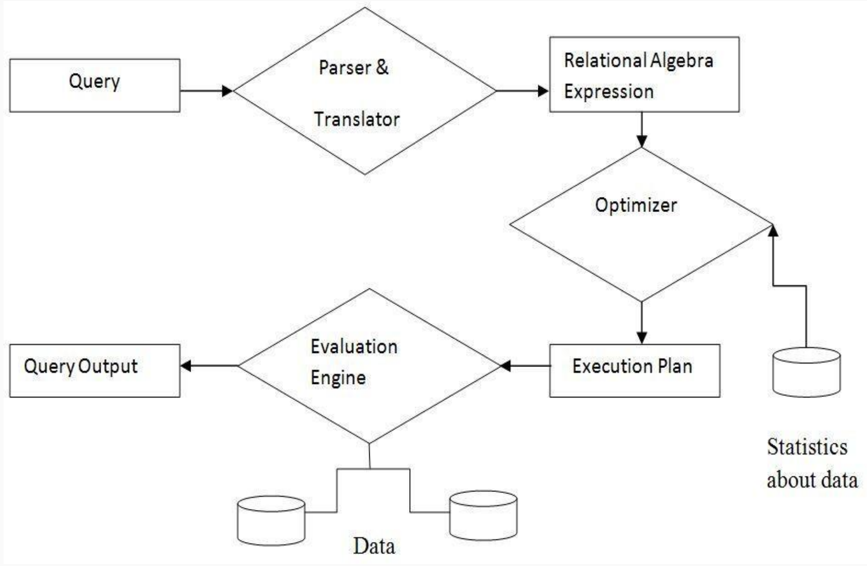


Figure I. Steps in query Processing

Query Cost Estimation

- Each query is translated into a number of semantically equivalent plans.
- So, there are several alternatives now the question is which one is the most efficient execution plan to be selected for execution.
- Cost is generally measured as total time elapsed for answering query. To convert high level query to desired query we need some measurements.
- Basic measure for query cost are:

✓ disk access

✓ CPU cycle

✓ Transit time in network.

Query Cost Estimation

- Here CPU cost is difficult to calculate.
- For **disk access**, how many disk access required to convert the high level query to desired query.

- For **CPU cycle**, how many CPU cycle are consumed to evaluate desired query. CPU speed increases at faster rate than disk speed. Due to which CPU cost is relatively lower than disk cost.

- **Transit time in network**: It is primarily considered with parallel/distributed system.

Equivalence/ Transformation Rules

- A query can be expressed in several different ways with different cost of evaluations.
- Here, two relational algebra expressions are said to be equivalent if, on every legal database instance, the two expressions generate the same set of tuples.
- **Equivalence Rules**

➢ It says that expressions of two forms are equivalent.

➢ By using the equivalence rule which is concerned with basic relational algebra operator, we can formulate any equivalent expressions for a single query.

Equivalence/ Transformation Rules

1. Conjunctive selection operations can be deconstructed into a sequence of individual selections. It is referred to as cascade of σ.

$$\sigma_{\theta_1 \wedge \theta_2}(E) = \sigma_{\theta_1}(\sigma_{\theta_2}(E))$$

2. Selection operations are commutative. $\sigma_{\theta_1}(\sigma_{\theta_2}(E)) = \sigma_{\theta_2}(\sigma_{\theta_1}(E))$

3. Only the final operations in a sequence of projection operations are needed; the others can be omitted. This transformation can be referred to as a cascade of π.

$$\Pi_{L_1}(\Pi_{L_2}(\ldots(\Pi_{L_n}(E))\ldots)) = \Pi_{L_1}(E)$$

Equivalence/ Transformation Rules

4. Selections can be combined with Cartesian product and theta join.

$$\sigma_\theta(E_1 \times E_2) = E_1 \bowtie_\theta E_2$$

$$\sigma_{\theta_1}(E_1 \bowtie_{\sigma_{\theta_2}} E_2) = E_1 \bowtie_{\theta_1 \wedge \theta_2} E_2$$

5. Theta join operations are commutative:

$$E_1 \bowtie_\theta E_2 = E_2 \bowtie_\theta E_1$$

6. Natural join operations are associative.

$$(E_1 \bowtie E_2) \bowtie E_3 = E_1 \bowtie (E_2 \bowtie E_3)$$

Theta joins are associative in the following manner.

$$(E_1 \bowtie_{\theta_1} E_2) \bowtie_{\theta_2 \wedge \theta_3} E_3 = E_1 \bowtie_{\theta_1 \wedge \theta_3} (E_2 \bowtie_{\theta_2} E_3)$$

where θ_2 involves attributes from E_2 and E_3 only.

Equivalence/ Transformation Rules

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

i. It distributes when all the attributes in selection condition Θ_0 involve only the attributes of one of the expressions (say E_1) being joined.

$$\sigma_{\theta_0}(E_1 \bowtie_\theta E_2) = (\sigma_{\theta_0}(E_1)) \bowtie_\theta E_2$$

ii. It distributes when selection condition 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/ Transformation Rules

8. The projection operation distributes over the theta join operation under the following condition :

i. Let L_1 and L_2 be set of attributes of E_1 and E_2 respectively. Suppose that the join condition Θ involves only attributes in $L_1 \cup L_2$. Then

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

ii. 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 the 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 E_2 that are involved in join condition Θ but are not in $L_1 \cup L_2$. Then ,

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

Equivalence/ Transformation Rules

9. The set operations union and intersection are commutative.

Set difference is not commutative.

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

10. Set union and intersections are associative.

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

11. The selection operation distributes over the union, intersection and set difference operation.

$$\sigma_P(E_1 - E_2) = \sigma_P(E_1) - E_2 = \sigma_P(E_1) - \sigma_P(E_2)$$

12. The projection operation distributes over union operation

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

Operator Tree

- An operator tree represents relational algebra query graphically.
- It is a tree in which leaf node is a relation stored in database and non leaf node is a intermediate relation produce by a relational algebra operator.
- The sequence of operations is directed from leaves to the root, which represents the answer to the query.

$$\begin{tikzcd}
 & \sigma_\theta & & & & \bowtie & \\
 & \bowtie \arrow[u] & {} \arrow[r, leftrightarrow, "\text{if } \theta \text{ only has attributes from } E_1"'] & {} & \sigma_\theta \arrow[ur] & & E_2 \arrow[ul] \\
E_1 \arrow[ur] & & E_2 \arrow[ul] & & E_1 \arrow[u] & &
\end{tikzcd}$$

$\boxed{\text{Rule no 7(i)}}$

Operator Tree

Q. Represent the following relational algebra expression using operator tree.

(i) $\Pi_{customer\_name}(\sigma_{branch\_city='ktm' \wedge bln>1000}(branch \bowtie account \bowtie depositor))$

Solution:

$$\begin{tikzcd}
 & \Pi_{customer\_name} & & \\
 & \sigma_{branch\_city='ktm' \wedge bln>1000} \arrow[u] & & \\
 & \bowtie \arrow[u] & & \\
branch \arrow[ur] & & \bowtie \arrow[ul] & \\
 & account \arrow[ur] & & depositor. \arrow[ul]
\end{tikzcd}$$

Operator Tree

Q. Represent the following relational algebra expression using operator tree.

(ii) $\Pi_{student\_name}(\sigma_{course\_name='DBMS'}(Student \bowtie Registration \bowtie Course))$

Solution:

$$\begin{tikzcd}
& \Pi_{student\_name} & & \\
& \sigma_{course\_name='DBMS'} \arrow[u] & & \\
& \bowtie \arrow[u] & & \\
Student \arrow[ru] & & \bowtie \arrow[lu] & \\
& Registration \arrow[ru] & & Course \arrow[lu]
\end{tikzcd}$$

Query Optimization

- The process of selecting the most efficient query execution plan among the many strategies possible for processing a query.
- The selected plan minimizes the cost function.
- **Steps of Optimization:**

1. Create an initial operator (expression) tree.
2. Move select operation down the tree for the easiest possible execution.
3. Applying more restrictive select operation first.
4. Replace Cartesian product by join.
5. Creating new projection whenever needed.
6. Adjusting rest of the tree accordingly.

Query Optimization

- **Example:** The following query retrieves the customer name from branch city Kathmandu whose balance is greater then 5000

Select customer_name from Branch, Account, Depositor where city= ‘ktm’ and bln >5000;

- Some of the **evaluation plans** are:

1. Join relation Branch and Account, join the result with Depositor and then do the restriction.
2. Join the relation Branch and Account, do the restrictions and then join the result with Depositor.
3. Do the restriction, join the relations Branch and Account, and join the result with Depositor.

Query Optimization

Select customer_name from Branch, Account, Depositor where city= ‘ktm’ and bln >5000;

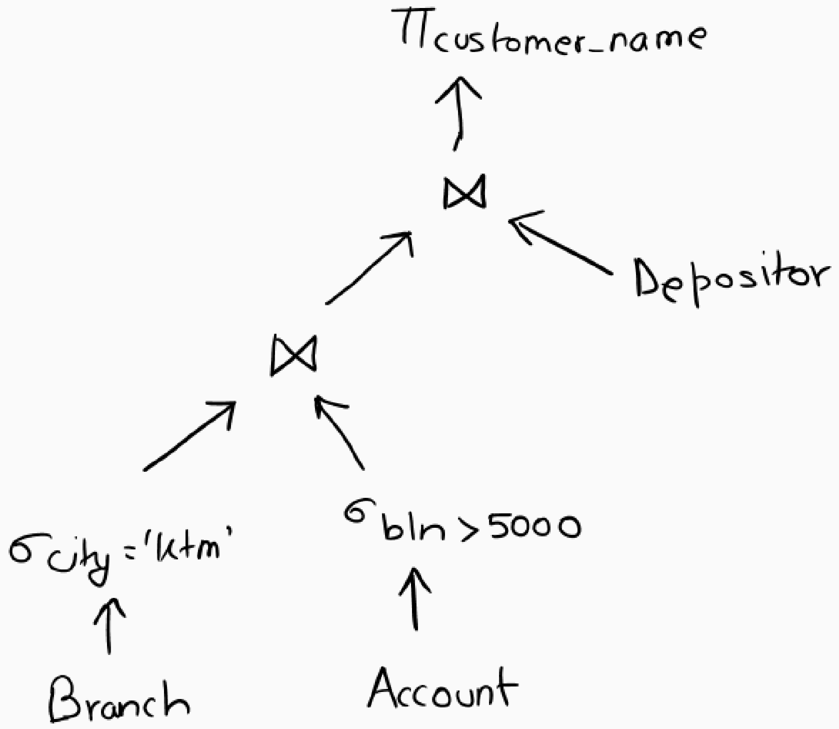
RA: $\Pi_{customer\_name}(\sigma_{city='ktm' \wedge bln>5000}(Branch \bowtie Account \bowtie Depositor))$

Operator tree :

$$\begin{tikzcd}
 & \Pi_{customer\_name} & & \\
 & \sigma_{city='ktm' \wedge bln>5000} \arrow[u] & & \\
 & \bowtie \arrow[u] & & \\
Branch \arrow[ur] & & \bowtie \arrow[ul] & \\
 & Account \arrow[ur] & & Depositor \arrow[ul]
\end{tikzcd}$$

Query Optimization

→ The final operator tree after multiple transformations are;



Questions Collection

1. Define query processing. Write down the steps of query processing
2. What is equivalence rules ? List down the different equivalence rules.
3. Define query cost estimation. Explain with example the steps of optimization.
4. Write short notes on:

i. Operator Tree

ii. Equivalence rules

iii. Query Cost Estimation

THANK YOU

Any Queries ?