Query Processing and Optimization

Introduction

- In this chapter we shall discuss the techniques used by a DBMS to process, optimize and execute high-level queries.
- The techniques used to split complex queries into multiple simple operations and methods of implementing these low-level operations.
- The query optimization techniques are used to chose an efficient execution plan that will minimize the runtime as well as many other types of resources such as number of disk I/O, CPU time and so on.

Query Processing

- Query Processing is a procedure of transforming a high-level query (such as SQL) into a correct and efficient execution plan expressed in low-level language.
- When a database system receives a query for update or retrieval of information, it goes through a series of compilation steps, called execution plan.
- Query processing goes through various phases:
- first phase is called syntax checking phase, the system parses the query and checks that it follows the syntax rules or not.
- It then matches the objects in the query syntax with the view tables and columns listed in the system table.

- In second phase the SQL query is translated in to an algebraic expression using various rules.
- So that the process of transforming a high-level SQL query into a relational algebraic form is called Query Decomposition.
- The relational algebraic expression now passes to the query optimizer.
- In third phase optimization is performed by substituting equivalent expression depends on the factors such that the existence of certain database structures, whether or not a given file is stored, the presence of different indexes & so on.
- Query optimization module work in tandem with the join manager module to improve the order in which joins are performed.

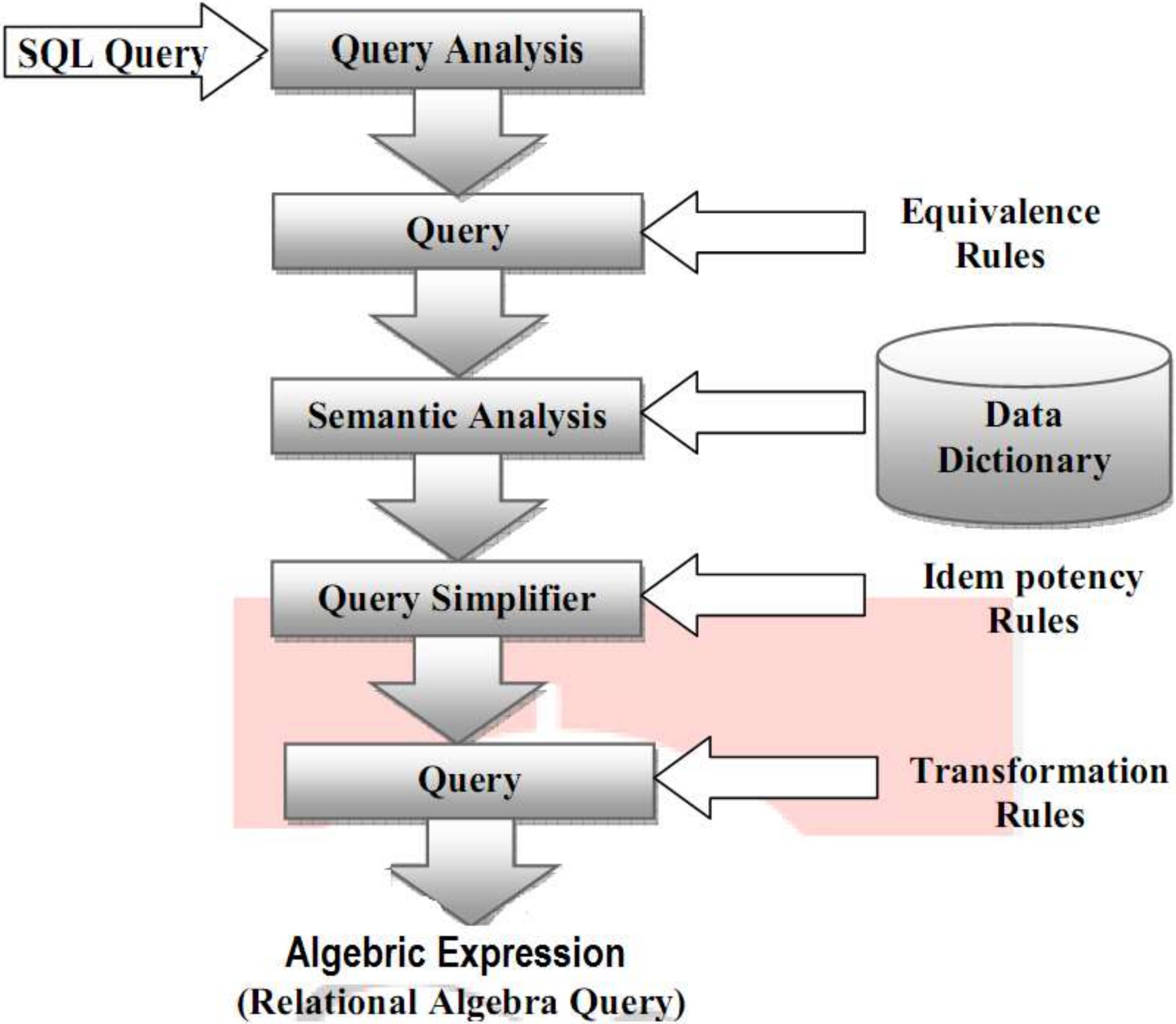
- At this stage the cost model and several other estimation formulas are used to rewrite the query.
- The modified query is written to utilize system resources so as to bring the optimal performance.
- The query optimizer then generates an action plan also called a execution plan.
- This action plans are converted into a query codes that are finally executed by a run time database processor.
- The run time database processor estimate the cost of each action plan and chose the optimal one for the execution.

Query Analyzer

- The syntax analyzer takes the query from the users, parses it into tokens and analyses the tokens and their order to make sure they follow the rules of the language grammar.
- Is an error is found in the query submitted by the user, it is rejected and an error code together with an explanation of why the query was rejected is return to the user.

Query Decomposition

- In query decomposition the query processing aims are to transfer the high-level query into a relational algebra query and to check whether that query is syntactically and semantically correct.
- Thus the query decomposition is start with a high-level query and transform into query graph of low-level operations, which satisfy the query.
- The SQL query is decomposed into query blocks (low-level operations), which form the basic unit.
- Hence nested queries within a query are identified as separate query blocks.
- The query decomposer goes through five stages of processing for decomposition into low-level operation and translation into algebraic expressions.


SQL Query
Query Analysis
Query
Equivalence Rules
Semantic Analysis
Data Dictionary
Query Simplifier
Idem potency Rules
Query
Transformation Rules
Algebric Expression
(Relational Algebra Query)

Query Analysis

- During the query analysis phase, the query is syntactically analyzed using the programming language compiler (parser).
- A syntactically legal query is then validated, using the system catalog, to ensure that all data objects (relations and attributes) referred to by the query are defined in the database.
- The type specification of the query qualifiers and result is also checked at this stage.

- Example:

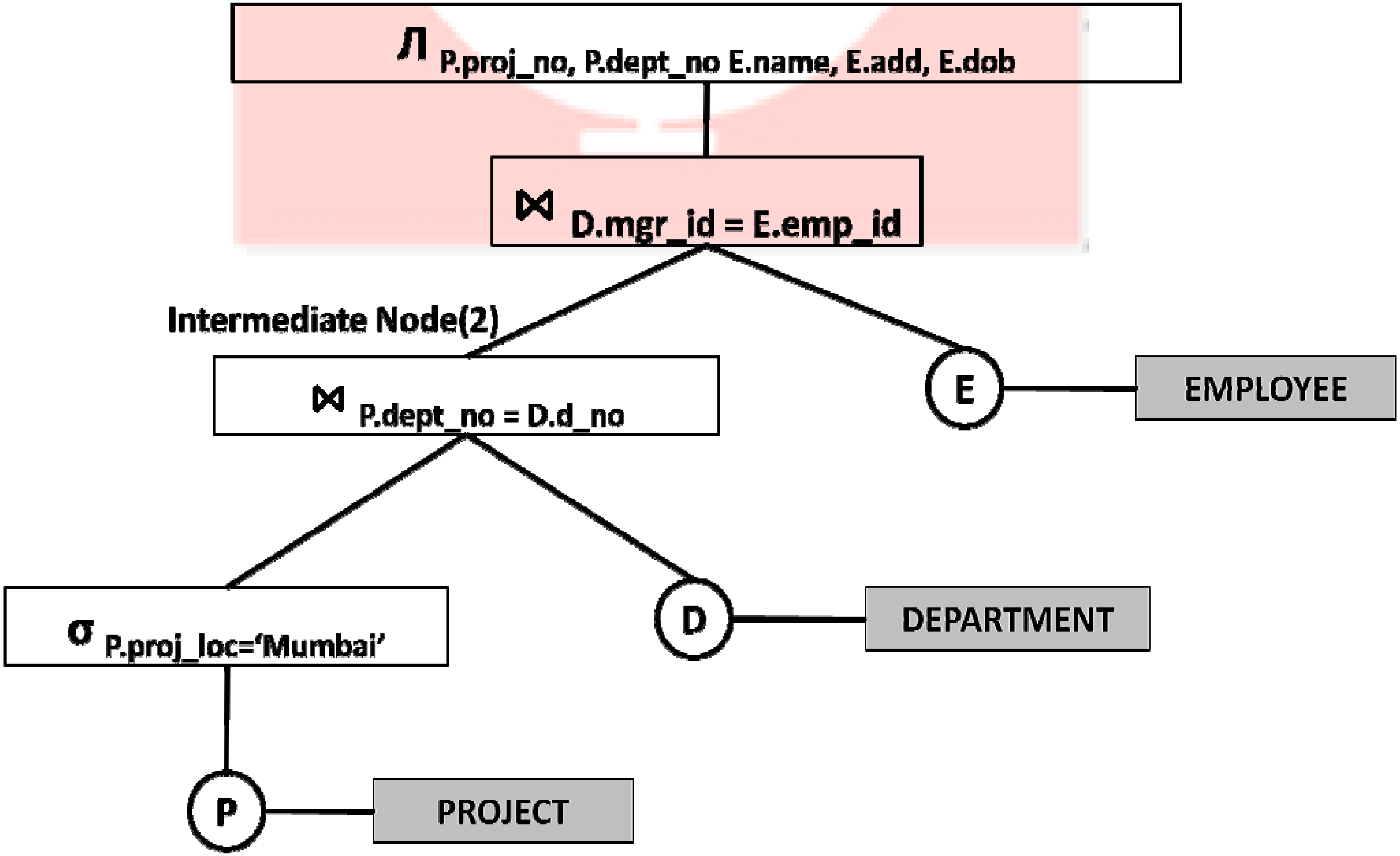
  SELECT emp_nm FROM EMPLOYEE WHERE emp_desg>100

  This query will be rejected because the comparison “>100” is incompatible with the data type of emp_desg which is a variable character string.

- At the end of query analysis phase, the high-level query (SQL) is transformed into some internal representation that is more suitable for processing.
- This internal representation is typically a kind of query tree.

- A Query Tree is a tree data structure that corresponds expression.
- A Query Tree is also called a relational algebra tree.
  - **Leaf node** of the tree, representing the base input relations of the query.
  - **Internal nodes** result of applying an operation in the algebra.
  - **Root** of the tree representing a result of the query.

- **SELECT** (P.proj_no, P.dept_no, E.name, E.add, E.dob)
- **FROM** PROJECT **P**, DEPARTMENT **D**, EMPLOYEE **E**
- **WHERE** P.dept_no = D.d_no AND D.mgr_id = E.emp_id AND P.proj_loc = ‘Mumbai’ ;



$\pi_{\text{P.proj\_no, P.dept\_no E.name, E.add, E.dob}}$

$\bowtie_{\text{D.mgr\_id = E.emp\_id}}$

Intermediate Node(2)

$\bowtie_{\text{P.dept\_no = D.d\_no}}$

E

EMPLOYEE

$\sigma_{\text{P.proj\_loc='Mumbai'}}$

D

DEPARTMENT

P

PROJECT

- The three relations PROJECT, DEPARTMENT, EMPLOYEE are represent as a leaf nodes P, D and E, while the relational algebra operations of the represented by internal tree nodes.
- Same SQL query can have man different relational algebra expressions and hence many different query trees.
- The query parser typically generates a standard initial (canonical) query tree.

Query Normalization

- The primary phase of the normalization is to avoid redundancy.
- The normalization phase converts the query into a normalized form that can be more easily manipulated.
- In the normalization phase, a set of equivalency rules are applied so that the projection and selection operations included on the query are simplified to avoid redundancy.
- The projection operation corresponds to the SELECT clause of SQL query and the selection operation correspond to the predicate found in WHERE clause.
- The equivalency transformation rules that are applied.

Semantic Analyzer

- The objective of this phase of query processing is to reduce the number of predicates.
- The semantic analyzer rejects the normalized queries that are incorrectly formulated.
- A query is incorrectly formulated if components do not contribute to the generation of result.
- This happens in case of missing join specification.
- A query is contradictory if its predicate cannot satisfy by any tuple in the relation.
- The semantic analyzer examine the relational calculus query (SQL) to make sure it contains only data objects that is table, columns, views, indexes that are defined in the database catalog.
- It makes sure that each object in the query is referenced correctly according to its data type.
- In case of missing join specifications the components do not contribute to the generation of the results, and thus, a query may be incorrect formulated.