LECTURE 7

QUERY PROCESSING

Muhammad Hamiz Mohd Radzi
Faiqah Hafidzah Halim

Contents

Objectives

- At the end of this lesson, you should be able to:

  - *Define Query Processing, Query Optimization and Query Decomposition.*
  - *Compare different strategies of Query Processing.*
  - *Explain phases of Query Processing.*
  - *Explain advantages and disadvantages of dynamic and static optimization.*
  - *Describe the stages involve in Query Decomposition.*

Introduction

- When the relational model was first launched commercially, one of the major criticisms often cited was inadequate performance of queries.

- Since then, a significant amount of research has been devoted to developing highly efficient algorithms for processing queries.

- There are many ways in which a complex query can be performed, and one of the aims of query processing is to determine which one is the most cost effective.

- It is programmer's responsibility to select the most appropriate execution strategy.

- SQL (declarative language) user specifies what data is required rather than how it is to be retrieved. (to make SQL more universally usable)

- Giving DBMS the responsibility for selecting best strategy and more control over system performance.

Overview of Query Processing

- **QP** : Activities involved in **parsing, validating, optimizing and executing** a query.

- **Aim of QP**: To **transform** a query written in a **high level language** (SQL) into a **correct and efficient execution strategy** expressed in a **low-level language** (implementing Relational Algebra) and to **execute** the strategy to retrieve required data.

- An important aspect of query processing is **query optimization**.

Query Optimization (QO)

- QO: The activity of **choosing an efficient execution strategy** for processing a query.

- Generally, we try to reduce the **total execution time of the query**, which is the sum of the execution times of all individual operations that make up the query.

- However, **resource usage** may also be viewed as the **response time of the query**, in which case we concentrate on maximizing the number of parallel operations.

- Since the problem is computationally hard to control with a large number of relations, the strategy adopted is generally reduced to finding a near optimum solution.

- The first technique of QO uses heuristic rules that order the operations in a query.

- The other technique compares different strategies based on their relative costs and selects the one that minimizes resource usage.

- Both methods of query optimization depend on database statistics to evaluate properly the different options that are available.

- The accuracy and currency of these statistics have a significant bearing on the efficiency of the execution strategy chosen.
- The statistics cover information about relations, attributes, and indexes.
- There a lot of methods in order to keep the statistics which is update regularly (problematic), periodic update or user specified.

Find all Managers who work at a London branch.

We can write this query in SQL as:

```
SELECT *
FROM Staff s, Branch b
WHERE s.branchNo = b.branchNo AND
        (s.position = 'Manager' AND b.city = 'London');
```

Three equivalent relational algebra queries corresponding to this SQL statement are:

(1) $\sigma_{(position='Manager') \wedge (city='London') \wedge (Staff.branchNo=Branch.branchNo)}(Staff \times Branch)$

(2) $\sigma_{(position='Manager') \wedge (city='London')}(Staff \bowtie_{Staff.branchNo=Branch.branchNo} Branch)$

(3) $(\sigma_{position='Manager'}(Staff)) \bowtie_{Staff.branchNo=Branch.branchNo} (\sigma_{city='London'}(Branch))$

Calculation Result

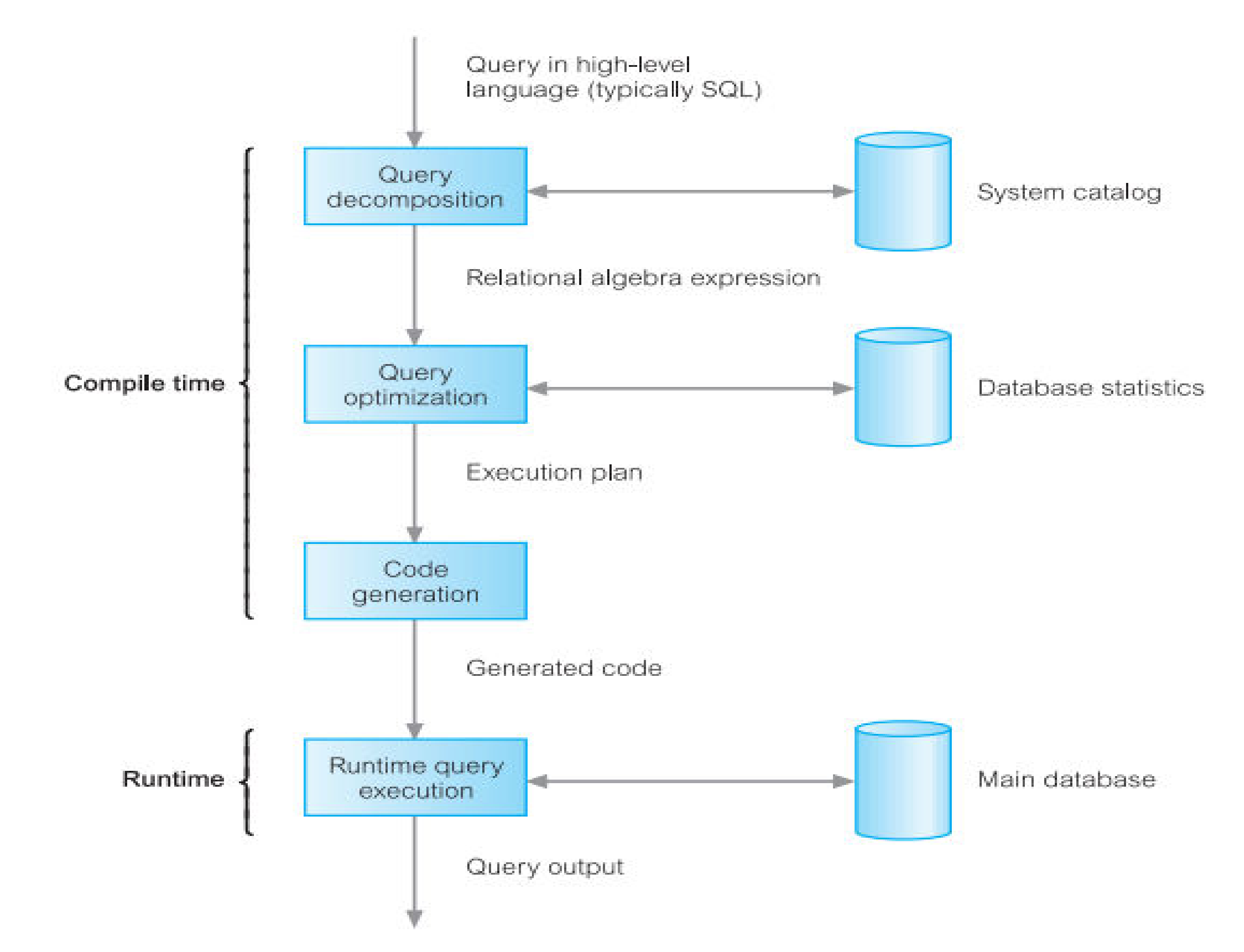
$$(1000 + 50) + 2*(1000 * 50) = 101\,050 \text{ disk accesses}$$

- 1)

- $$2*1000 + (1000 + 50) = 3050 \text{ disk accesses}$$

- $$1000 + 2*50 + 5 + (50 + 5) = 1160 \text{ disk accesses}$$

Phases of Query Processing



Dynamic VS Static Processing

- There are two choices for when the first three phases of query processing can be carried out.

1. Dynamically carry out decomposition and optimization every time the query is run.

2. Just do it once (statically) for query to parsed, validated, and optimized

	ADVANTAGES	DISADVANTAGES
DYNAMIC	**All information required to select an optimum strategy is up to date.**	**Performance of the query is affected because the query has to be parsed, validated and optimized before it can be executed.** **May be necessary to reduce the number of execution strategies to be analyzed to achieve an acceptable overhead, which may have the effect of selecting a less optimum strategy.**
STATIC	**Runtime overhead is removed and more time available to evaluate a larger number of execution strategies (increasing the chances to find optimum strategy)** **For queries that are executed many times, taking some additional time to find a more optimum plan may prove to be highly beneficial.**	**Execution strategy is chosen to be optimal when the query is compiled may no longer be optimal when the query is running.**

Query Decomposition

- Query decomposition is the first phase of query processing.
- The aims of query decomposition are to transform a high-level query into a relational algebra query, and to check that the query is syntactically and semantically correct.
- The typical stages of query decomposition are

1. Analysis

2. Normalization

3. Semantic Analysis

4. Simplification

5. Query restructuring

Analysis

- The query is **lexically and syntactically analyzed** using the techniques of programming language compilers.

- Verifies that the **relations and attributes** specified in the query **are defined in the system catalog**.

- Verifies that any **operations applied** to database objects **are appropriate** for the object type.

- On completion of the analysis, the **high-level query** has been **transformed** into some **internal representation** (query tree) that is more suitable for processing.

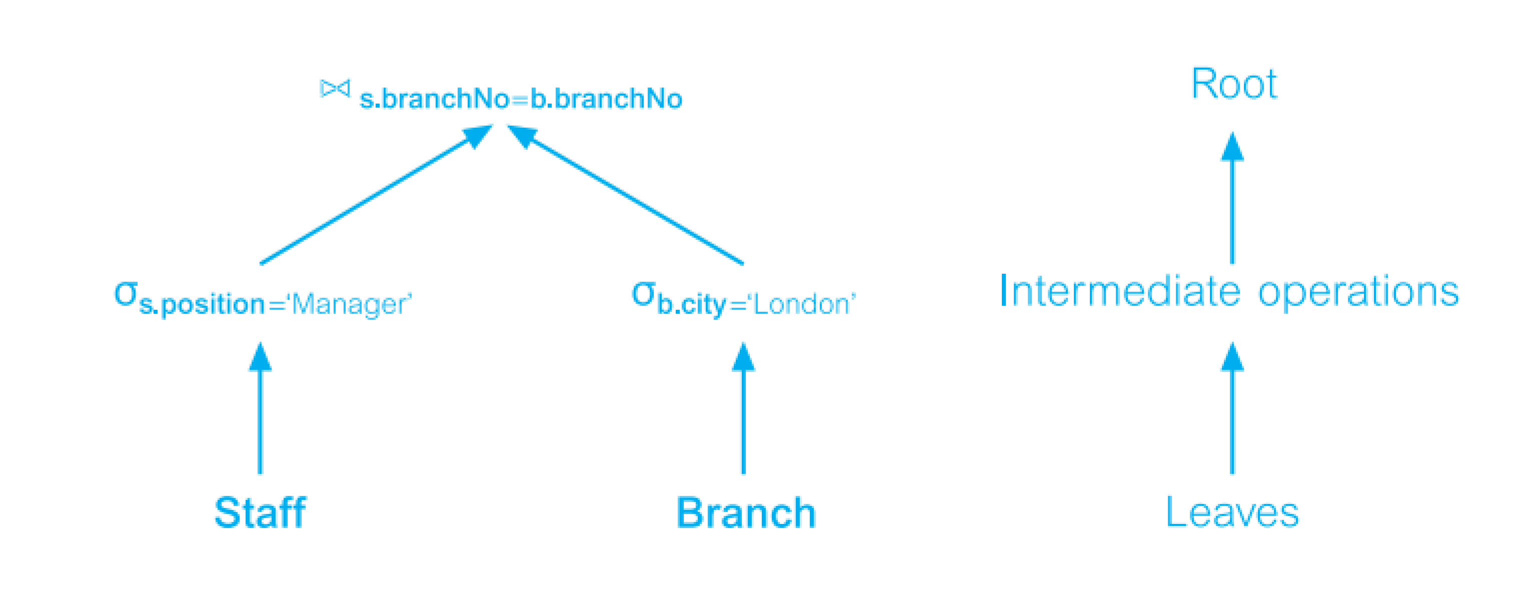
Example

SELECT staffNumber
FROM Staff
WHERE position > 10;

This query would be rejected on two grounds:

(1) In the select list, the attribute staffNumber is not defined for the Staff relation (should be staffNo).

(2) In the WHERE clause, the comparison '>10' is incompatible with the data type position, which is a variable character string.



Normalization

- To transform the query into normalized form to facilitate further processing.

- Transformation rules are:
  - *Conjunctive NF: A sequence of conjuncts that are* ***connected with the ∧(AND) operator****. Each conjunct contains one or more terms connected by the ∨(OR) operator.*

  - *Disjunctive NF: A sequence of disjuncts that are connected with* ***the ∨(OR) operator****. Each disjunct contains one or more terms connected by the ∧(AND) operator.*

– Conjunctive normal form

$$(p_{11} \vee p_{12} \vee \cdots \vee p_{1n}) \wedge \cdots \wedge (p_{m1} \vee p_{m2} \vee \cdots \vee p_{mn})$$

– Disjunctive normal form

$$(p_{11} \wedge p_{12} \wedge \cdots \wedge p_{1n}) \vee \cdots \vee (p_{m1} \wedge p_{m2} \wedge \cdots \wedge p_{mn})$$

- Example: Find an employees who have been working for project P1 for 12 or 24 months.

- SQL :

```
SELECT e.empName
FROM emp e, assignment a
WHERE e.empNo = s.empNo
AND a.projNo = ‘P1’
AND duration = 12 OR duration = 24;
```

- Conjunctive:

(e.empNo = a.empNo) ^ (a.projNo = ‘P1’) ^ (duration =12 v duration = 24)

- Disjunctive:

(e.empNo = a.empNo ^ a.projNo = ‘P1’ ^ duration =12) V

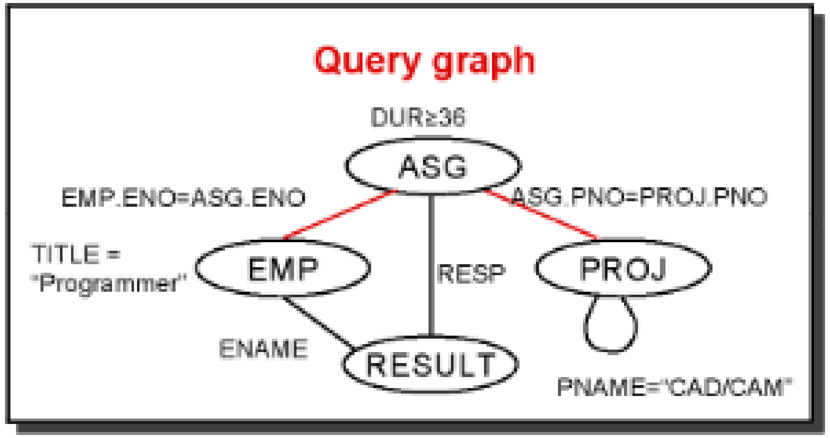
(e.empNo = a.empNo ^ a.projNo = ‘P1’ ^ duration =24)

Semantic Analysis

- Identify and reject type incorrect or semantically incorrect queries.
- Type incorrect: attributes and relations name not in global schema and data type of attribute is not match.
- Semantically incorrect: the relations are not joined in the query and join graphs.

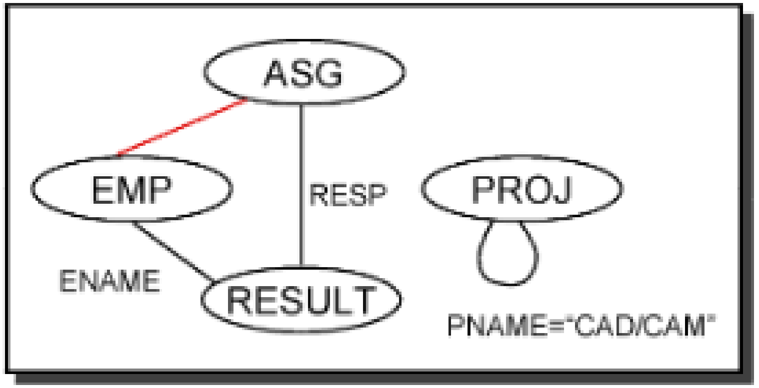
- **Example:** Consider a query:

```
SELECT  ENAME,RESP
FROM    EMP, ASG, PROJ
WHERE   EMP.ENO = ASG.ENO
AND     ASG.PNO = PROJ.PNO
AND     PNAME = "CAD/CAM"
AND     DUR ≥ 36
AND     TITLE = "Programmer"
```



- **Example:** Consider the following query and its query graph:

```
SELECT  ENAME,RESP
FROM    EMP, ASG, PROJ
WHERE   EMP.ENO = ASG.ENO
AND     PNAME = "CAD/CAM"
AND     DUR ≥ 36
AND     TITLE = "Programmer"
```



Simplification

- To detect **redundant qualifications**, **eliminate common sub-expressions** , and **transform** the query to a **semantically equivalent** but more easily and efficiently computed form.

- Access restrictions, view definitions, and integrity constraints are considered at this stage.

Query Restructuring

- In the final stage of query decomposition, the query is restructured to provide a more efficient implementation.

Reference

- *Database Systems: A Practical Approach to Design, Implementation, and Management,* Thomas Connolly and Carolyn Begg, 5th Edition, 2010, Pearson.