Concept 6 of 6: Query Processing

BBT 3104: Advanced Database Systems

Week 12 to 13 of 13

July 2020

By Allan O. Omondi – aomondi [at] strathmore.edu
Faculty of Information Technology

CC BY-NC-SA 4.0

Recommended citation:

Omondi, A. O. (2020). *BBT 3104 – Advanced Database Systems – Concept 6 of 6: Query Processing* [PDF Lecture Slides]. Retrieved from Strathmore University eLearning website https://elearning.strathmore.edu/pluginfile.php/227974/mod_resource/content/1/6.a.Concept6of6-Query_Processing.pdf

Strathmore UNIVERSITY

Adapted from Pixabay (2015)

Concept 6 of 6: Query Processing

BBT 3104: Advanced Database Systems
Week 12 to 13 of 13
July 2020

Allan O. Omondi
aomondi [at] strathmore.edu
Copyright © 2020

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Introduction to Query Processing

Strathmore UNIVERSITY

- SQL is a non-procedural (declarative) programming language
- This means that it specifies what is to be done rather than how to do it
- A DBA, therefore, uses SQL as a tool to express what is desired in terms of what is known
  - For example: if it is known that the database has a relation that stores client's data, then a DBA can specify that he/she wants a report on the list of clients that are being served by a specific sales representative
  - In this case, the DBA does not need to specify the procedure that the DBMS should use to retrieve the data required for the report
- This relieves DBAs from the responsibility of determining what constitutes a good procedure of how to retrieve the data

Introduction to Query Processing

- The procedure of how to retrieve the data is referred to as an execution strategy
  - Analogy: An army commander can have an objective "to win a war". This defines what the army commander wants.
  - A strategy must be executed to win the war. The execution of this strategy defines how the army commander wants to win the war.

Introduction to Query Processing

Strathmore UNIVERSITY

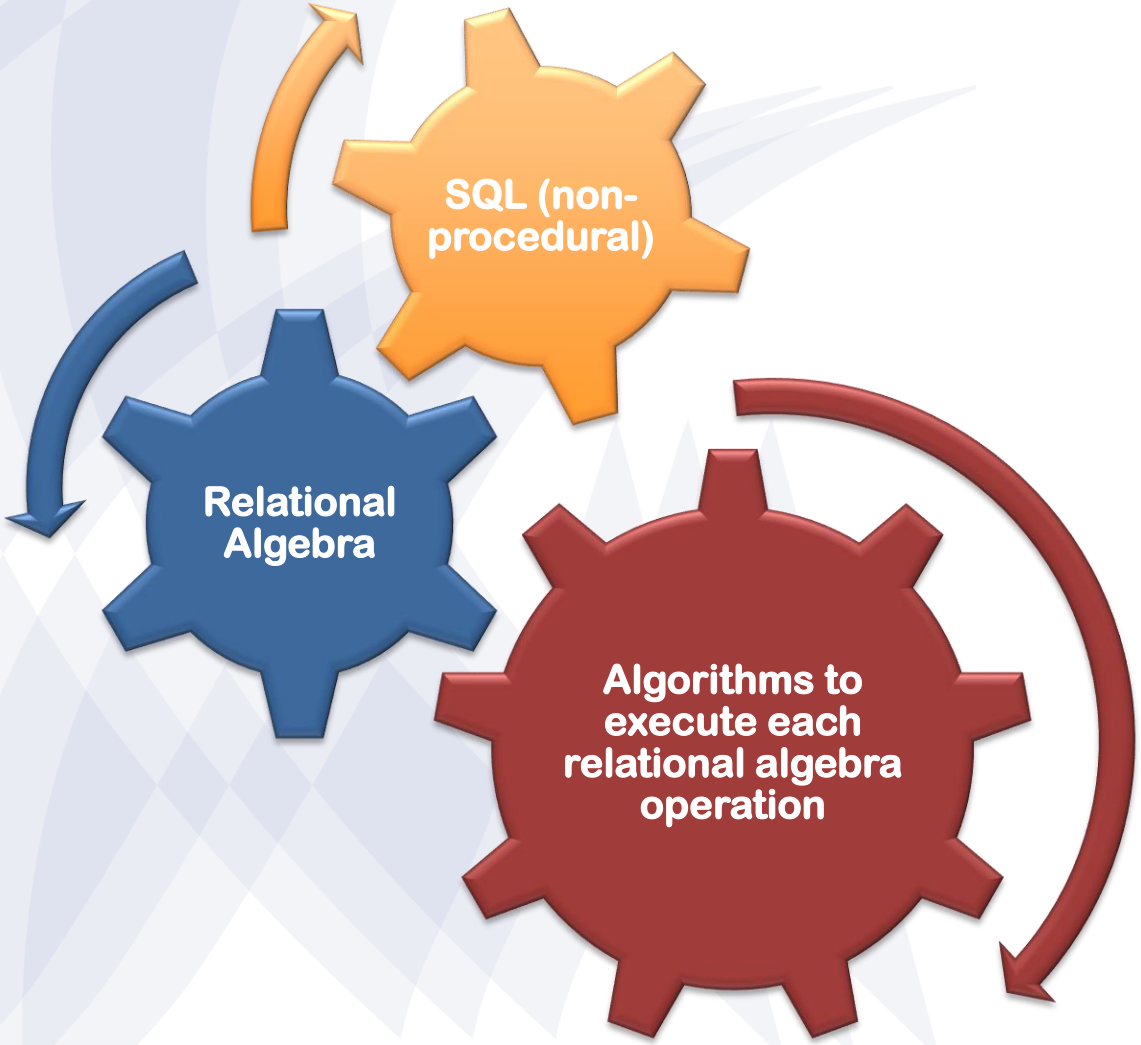
- Giving the DBMS the responsibility for selecting the best execution strategy prevents DBAs from choosing strategies that are known to be inefficient
- This gives the DBMS more control over the performance of the database system
- Reminder (from concept 3 of 6 on data storage): The performance of a database system can be measured by the time it takes to write data into the database (transaction throughput) and the time it takes the database system to respond to a request to retrieve data (response-time latency)

Introduction to Query Processing

Strathmore UNIVERSITY

- Query processing aims to transform a query expressed in a high-level, non-procedural language, e.g. SQL, into an efficient execution strategy expressed in a low-level language, e.g. relational algebra, and to execute the strategy to retrieve the required data

Introduction to Query Processing



✔ SQL Structured Query Language: Current
✗ SEQUEL Structured English Query Language: 1976
✗ SQUARE Specifying Queries As Relational Expressions: 1975

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Class Discussion on High-Level to Low-Level Query Transformation

Strathmore UNIVERSITY

- Transform the following query expressed using a high-level, non-procedural language (SQL) into its respective low-level form expressed using relational algebra

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

Class Discussion on High-Level to Low-Level Query Transformation

Strathmore UNIVERSITY

- Each of the 3 options below represents the low-level relational algebra form of the SQL query

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

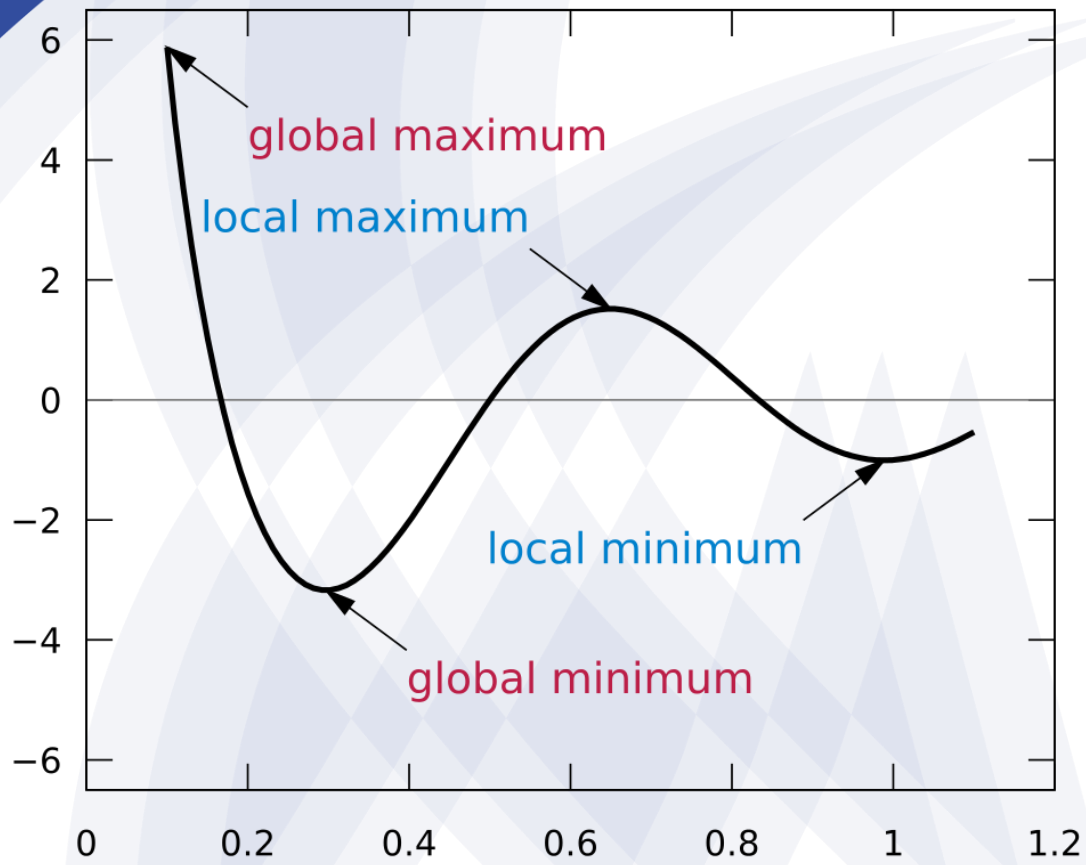
2. $\sigma_{(position='Manager') \wedge (city='London')} (Staff \bowtie_{Staff.branchNo=Branch.branchNo} Branch)$

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

Introduction to Query Processing

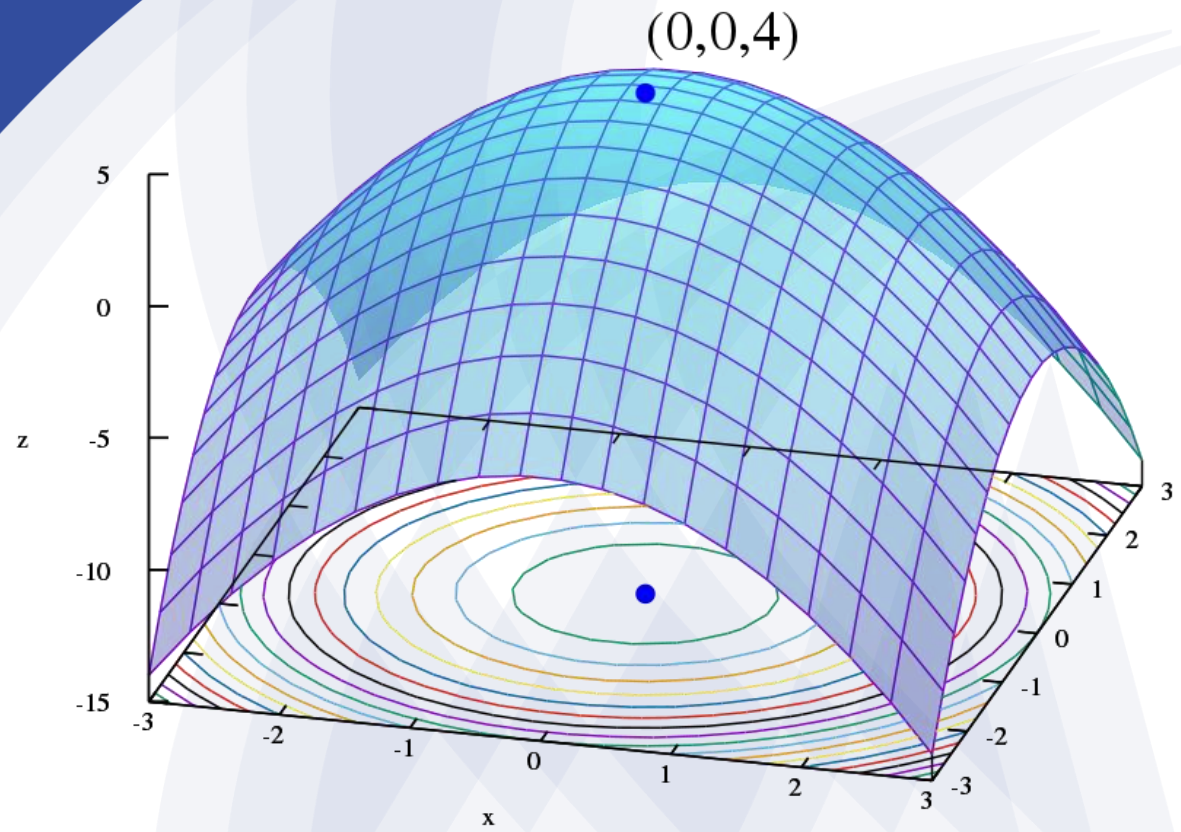
- Which of the 3 options is the most efficient?
- "Efficiency", in this case, is expressed in terms of the least amount of time required to execute the relational algebra operation

Introduction to Query Processing



- This can be expressed as a mathematical optimization problem that involves the selection of the best combination of element from some set of available alternatives
- Choosing the best combination of elements can be based on the one that minimizes a reward or the one that maximizes a reward

Introduction to Query Processing



- This can be expressed as a mathematical optimization problem that involves the selection of the best combination of element from some set of available alternatives
- Choosing the best combination of elements can be based on the one that minimizes a reward or the one that maximizes a reward

Introduction to Query Processing

Strathmore UNIVERSITY

- In this case, the DBMS is expected to select the combination of relational algebra operations that minimizes the total execution time
- Minimizing the total execution time positively affects the performance of the database system

Introduction to Query Processing

Strathmore UNIVERSITY

- The development of algorithms/formulae to solve mathematical optimization problems has been of interest in mathematics for centuries
- It is common to encounter mathematical optimization problems in computers science and engineering disciplines

Introduction to Query Processing

- The total execution time in database systems is mainly affected by the time it takes for data to be read from or written to storage (storage in this case refers to disk (HDD/SSD) and not RAM)
- We referred to this as the access time
- Reminder:

$$Access\ Time = Seek\ Time + Rotational\ Latency + Head\ Switch\ Time$$

- Disk access (access time), therefore, tends to be the dominant cost in query processing for a centralized DBMS

Introduction to Query Processing

Strathmore UNIVERSITY

- Assume that there are:
  - 1,000 tuples in Staff
  - 50 tuples in Branch
  - 50 managers (one for each branch)
  - 5 branches in London
- For simplicity, assume that the tuples are accessed one at a time (although in reality, access to tuples is based on blocks each of which contains several tuples)

Introduction to Query Processing

Strathmore UNIVERSITY

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

- The first query calculates the Cartesian product of Staff and Branch, which requires (1000 + 50) disk accesses to read the relations, and creates a relation with (1000 x 50) tuples
- We then have to read each of these tuples again to test them against the selection predicate at a cost of another (1000 x 50) disk accesses, giving a total cost of:

$$(1,000 + 50) + (1,000 \times 50) + (1,000 \times 50) = 101,050 \; disk \; accesses$$

Introduction to Query Processing

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

- The second query joins Staff and Branch on the branch number branchNo, which again requires (1,000 + 50) disk accesses to read each of the relations
- We know that the join of the two relations has 1,000 tuples: one for each member of staff (a member of staff can only work at one branch)
- The Selection operation requires 1,000 disk accesses to read the result of the join, giving a total cost of:

$$(1,000 + 50) + 1,000 + 1,000 = 3,050 \; disk \; accesses$$

Introduction to Query Processing

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

- The third relational algebra expression first reads each Staff tuple to determine the Manager tuples, which requires 1000 disk accesses and produces a relation with 50 tuples
- The second selection operation reads each Branch tuple to determine the London branches, which requires 50 disk accesses and produces a relation with 5 tuples
- The final operation is the join of the reduced Staff and Branch relations, which requires (50 + 5) disk accesses, giving a total cost of:

$$(1,000 + 50) + (50 + 5) + (50 + 5) = 1,160\ disk\ accesses$$

Introduction to Query Processing

Strathmore UNIVERSITY

- At this point, it makes sense to choose the 3rd option because it has the least execution time

Option	Cost of Execution (predominantly disk access)
1	101,050
2	3,050
3	1,160

- Query optimization involves choosing an efficient execution strategy for processing a query
- It, therefore, forms a significant part of query processing

Introduction to Query Processing

Strathmore UNIVERSITY

- Unfortunately, computing the cost of each option takes time
- DBMSs therefore have pre-defined rules that enable it to make a faster decision on which option to select
- These rules are referred to as heuristic rules

Introduction to Query Processing

Strathmore UNIVERSITY

- A heuristic is any approach to problem solving, learning, or discovery that employs a practical method not guaranteed to be optimal or perfect, but sufficient for the immediate goals
- Heuristic rules are commonly applied in cases where the problem is computationally intractable, and it is therefore reduced to finding a "near-optimum" solution (a "good-enough" solution)
- Heuristics can be therefore be considered to be mental shortcuts that ease the cognitive load of making a decision

Introduction to Query Processing

- There are 2 main approaches for query optimization:
  - Heuristic Approach: Uses heuristic rules to order relational algebra operations
  - Cost-Based Approach: Compares different execution strategies based on their relative cost (predominantly disk access time) and selects the one that has the least cost
- Most DBMS combine both of the 2 approaches when performing query optimization

Introduction to Query Processing

Strathmore UNIVERSITY

- Example of a heuristic rule (which essentially avoids the need to prove it by computing the cost of the operations):
  - the choice to use a join as opposed to a Cartesian product is based on heuristic rules that state that joins are better than Cartesian products and that joining smaller relations is better than joining larger ones

Introduction to Query Processing

- The aim of query processing is, therefore, to transform a query written in a high-level language, typically SQL, into a correct and efficient execution strategy expressed in a low-level language like relational algebra, and to execute the strategy to retrieve the required data

Presentation Outline

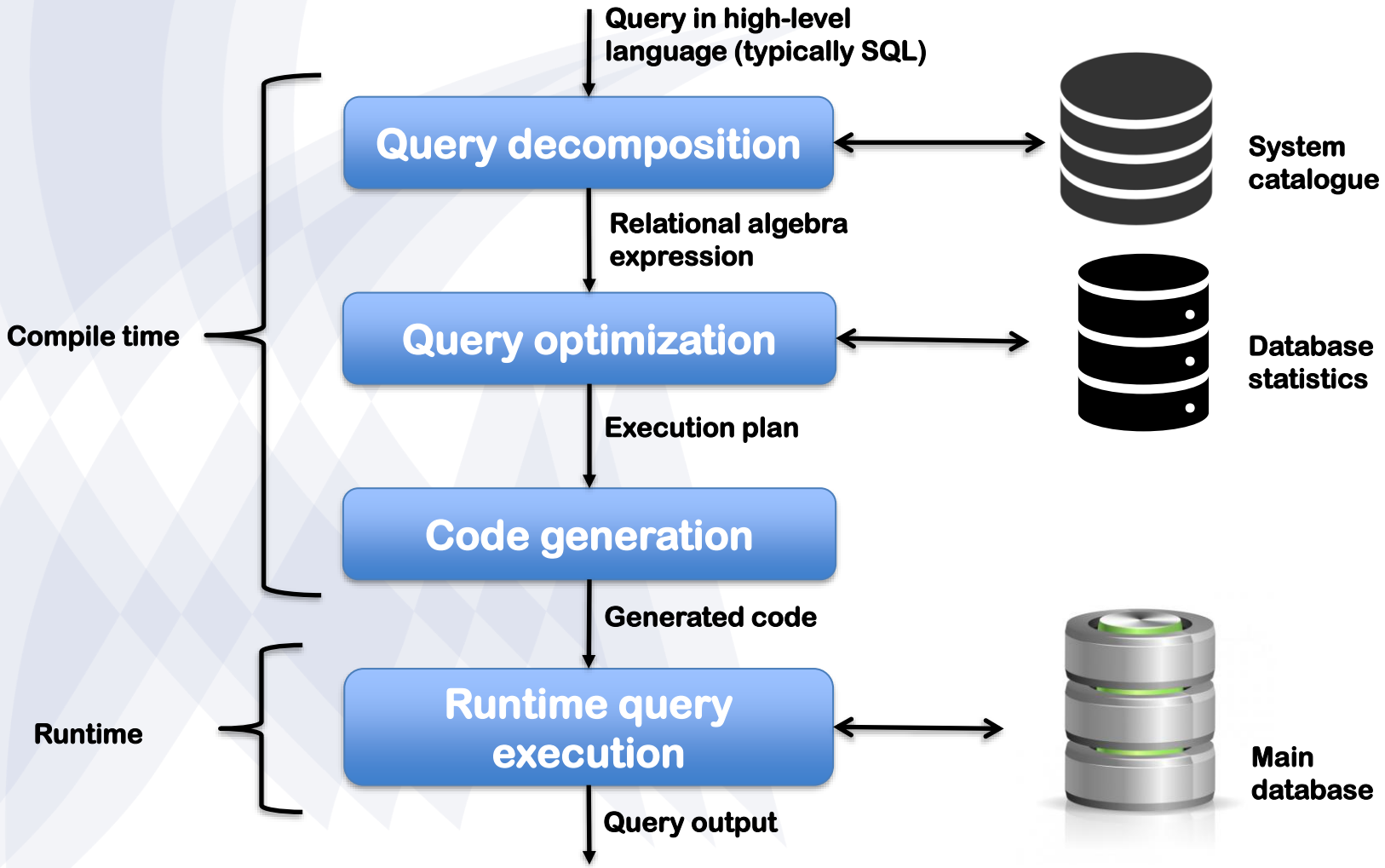
Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

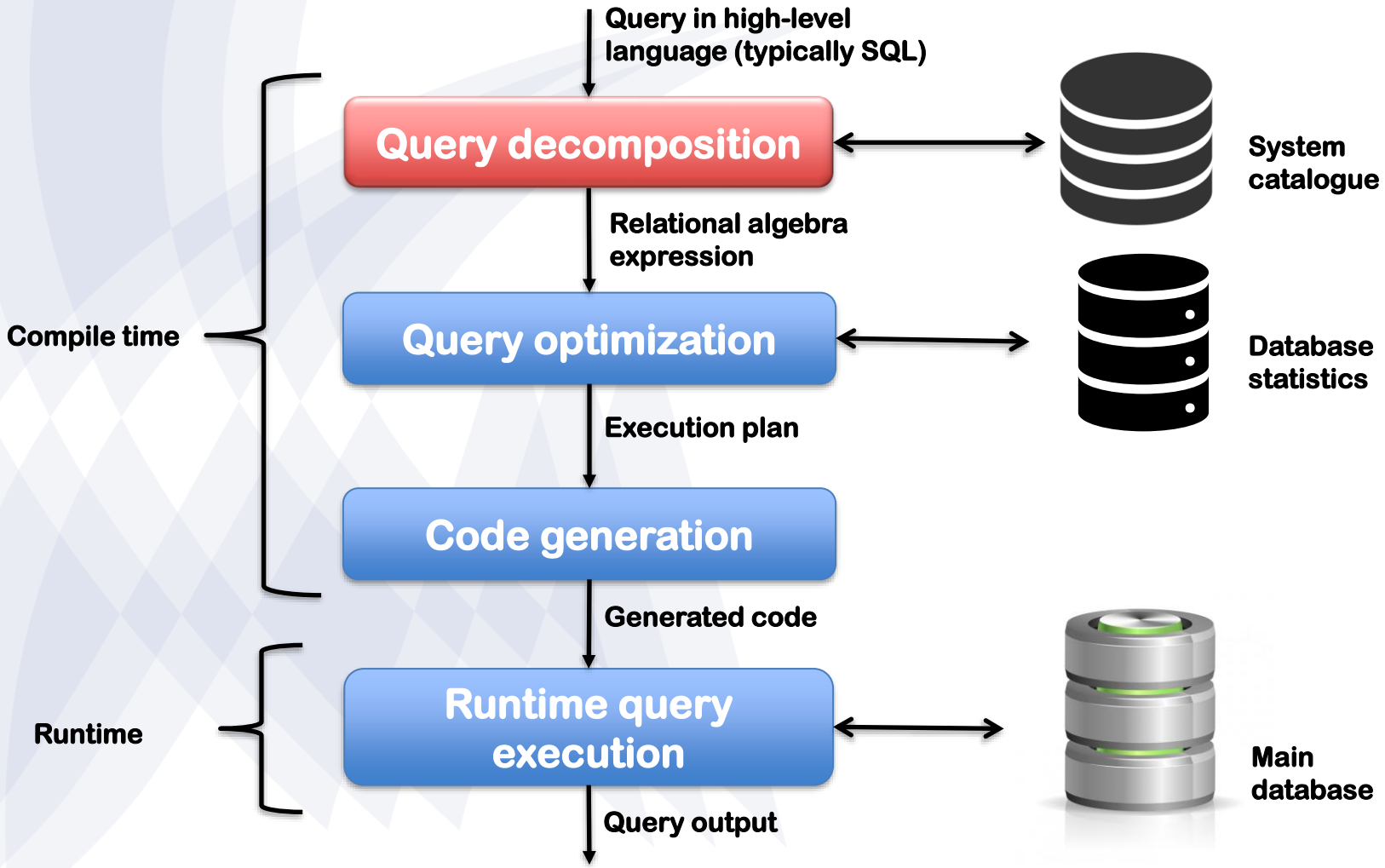
Stages of Query Processing

- The other stages of query processing in addition to query optimization are (in order):
  1. Query decomposition
  2. Query optimization
  3. Code generation
  4. Runtime query execution

Stages of Query Processing



Stages of Query Processing



Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Decomposition

- Query decomposition aims to transform a high-level query into a low-level relational algebra query and to check whether the query is syntactically and semantically correct
- Query decomposition is in turn made up of four stages:
  - Analysis
  - Normalization
  - Semantic analysis
  - Query restructuring

Query Decomposition: Analysis

Strathmore UNIVERSITY

- The analysis stage verifies that any operations applied to database objects are appropriate for the object type
- It also verifies that the relations and attributes specified in the query are defined in the system catalogue
- The result of this stage is a transformation of the high-level, non-procedural SQL into some internal representation that is more suitable for processing
- This internal representation is a form of a query tree called a relational algebra tree

Query Decomposition: Analysis

Strathmore UNIVERSITY

- For example, given the query:

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

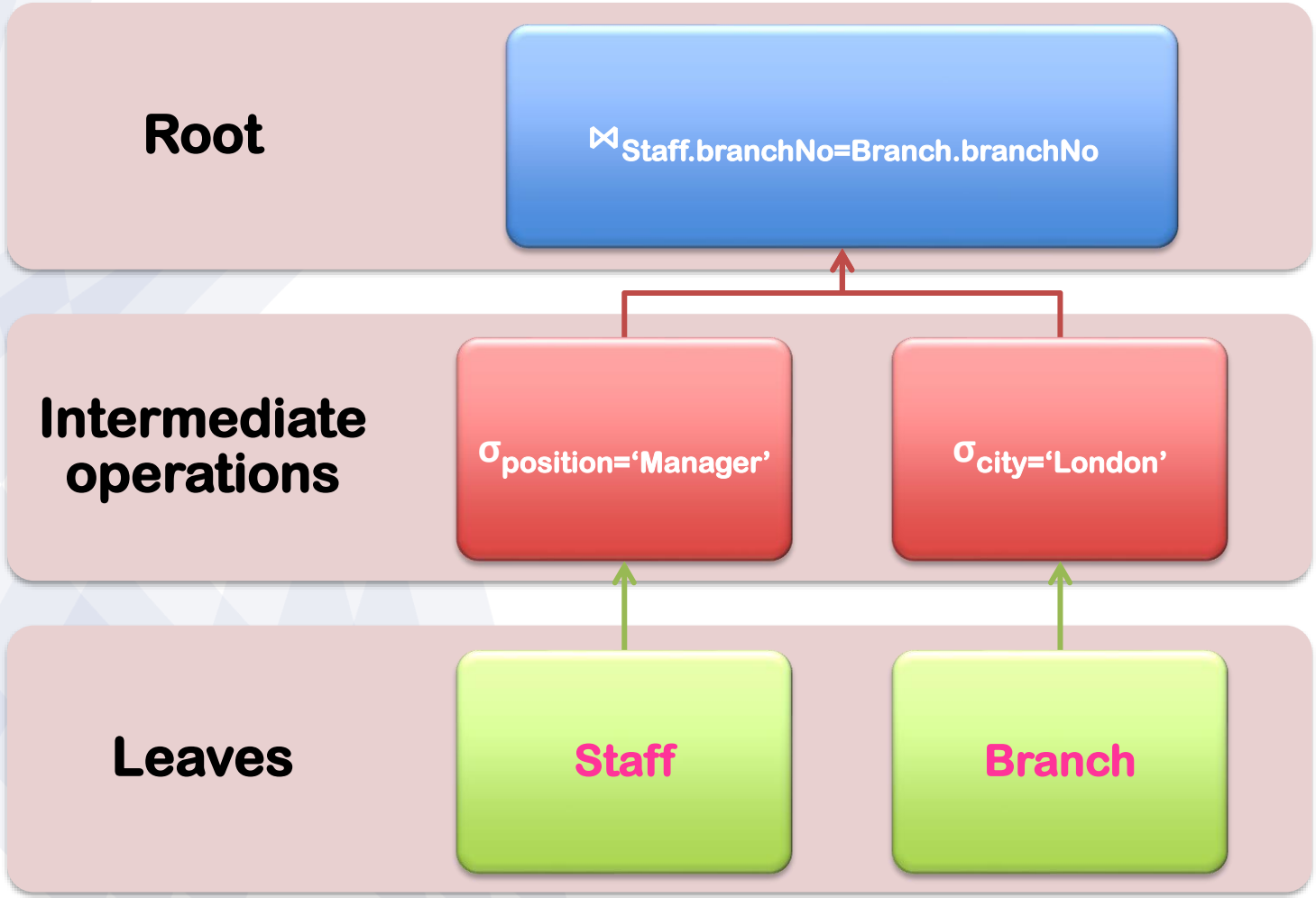
Query Decomposition: Analysis

- The resulting relational algebra is:

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

- The resulting relational algebra tree will be as shown on the next slide
- Note: The sequence of operations is directed from the leaves to the root (bottom-up)

Query Decomposition: Analysis



Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Decomposition: Normalization

Strathmore UNIVERSITY

- The normalization stage of query decomposition converts the query into a normalized form that can be more easily manipulated
- It focuses on the WHERE clause, also known as the predicate
- The WHERE clause can be converted into the following forms through the application of transformation rules:
  - Conjunctive normal form
  - Disjunctive normal form

Query Decomposition: Normalization

Strathmore UNIVERSITY

- **Conjunctive normal form**: Conjuncts that are connected with the **AND** operator, i.e. ∧ e.g.

  (position='Manager' ∨ salary > 20000) ∧ branchNo = 'B003'

- **Disjunctive normal form**: Disjuncts that are connected with the **OR** operator, i.e. ∨ e.g.

  (position='Manager' ∧ branchNo = 'B003') ∨ (salary > 20000 ∧ branchNo = 'B003')

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Decomposition: Semantic Analysis

Strathmore UNIVERSITY

- This stage of query decomposition rejects normalized queries that are incorrectly formulated or contradictory
- Incorrectly formulated queries contain WHERE clauses that have components which do not contribute to the generation of the result
- Contradictory queries contain WHERE clauses that cannot be satisfied by any tuples

Query Decomposition: Semantic Analysis

Strathmore UNIVERSITY

- An example of an incorrectly formulated query:

```
SELECT
    p.propertyNo, p.street
FROM
    Client c,
    Viewing v,
    PropertyForRent p
WHERE
    c.clientNo = v.clientNo
        AND c.maxRent >= 500
        AND c.prefType = 'Flat'
        AND p.ownerNo = 'CO93'
```

Query Decomposition: Semantic Analysis

Strathmore UNIVERSITY

- An example of a contradictory query:

```
SELECT
    p.propertyNo, p.street
FROM
    Client c,
    Viewing v,
    PropertyForRent p
WHERE
    c.maxRent > 500
        AND c.clientNo = v.clientNo
        AND v.propertyNo = p.propertyNo
        AND c.prefType = 'Flat'
        AND c.maxRent < 200
```

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Decomposition: Simplification

Strathmore UNIVERSITY

- This stage of query decomposition detects redundant qualifications, eliminates common subexpressions, and transforms the query into a semantically equivalent but more easily and efficiently computed form
- It focuses on access restrictions, view definitions, and integrity constraints
- Queries where the user does not have access to all components of the query are rejected

Presentation Outline

Strathmore UNIVERSITY

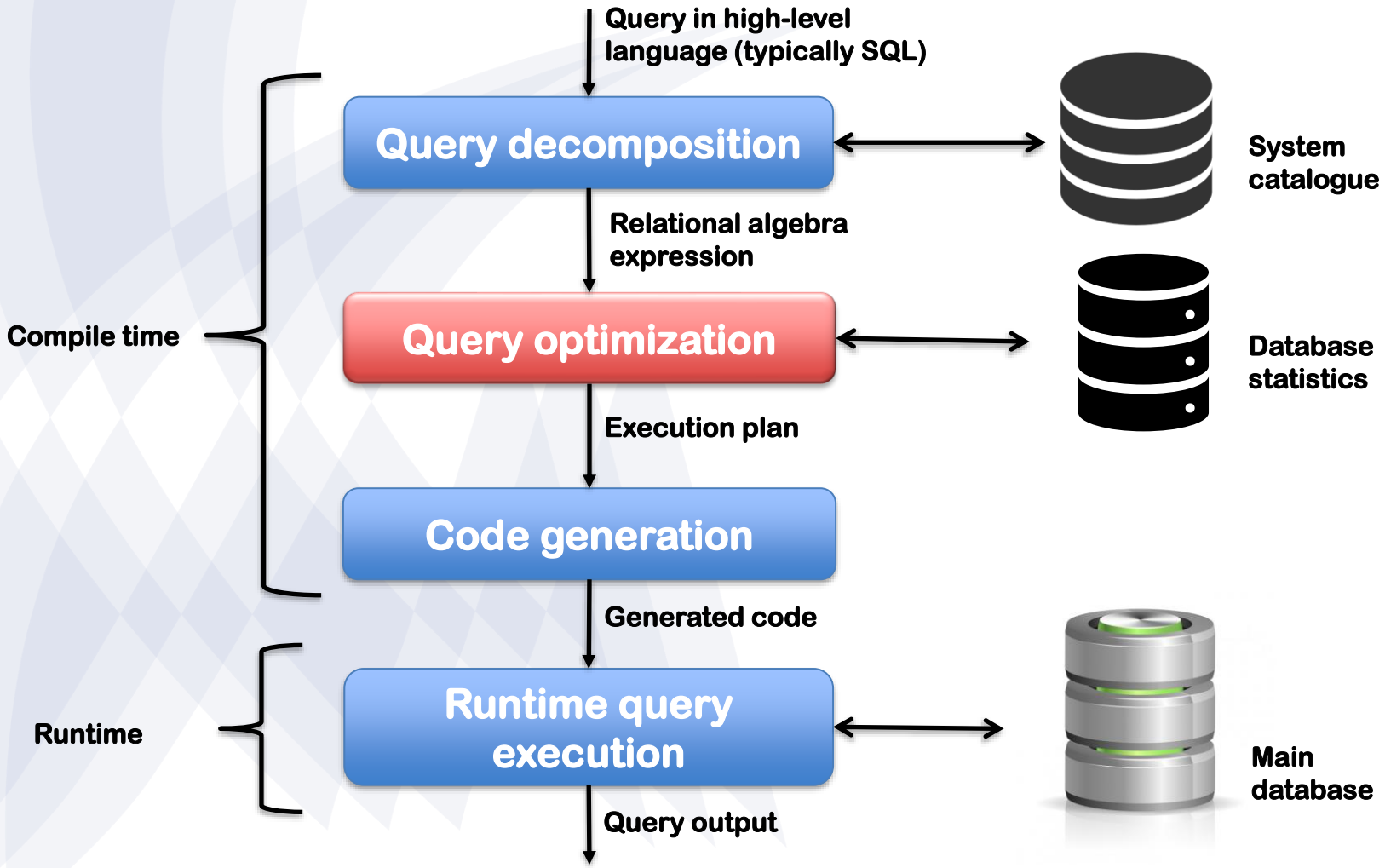
- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Decomposition: Query Restructuring

Strathmore UNIVERSITY

- This stage of query decomposition restructures the query to provide a more efficient implementation

Stages of Query Processing



Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Optimization

- Query optimization applies transformation rules to convert one relational algebra expression into an equivalent expression that is known to be more efficient
- The heuristic approach to query optimization orders the operations in a query using transformation rules that are known to generate good execution strategies
- The cost estimation approach to query optimization compares different strategies based on their relative costs and selects the one that minimizes resource usage

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Optimization: Heuristic Approach

Strathmore UNIVERSITY

- **Transformation rules that can be applied are:**

1. **Cascade of selection**

   $\sigma_{p \wedge q \wedge r}(R) = \sigma_p(\sigma_q(\sigma_r(R)))$

   E.g.

   $\sigma_{branchNo='B003' \wedge salary>15000}(Staff) = \sigma_{branchNo='B003'}(\sigma_{salary>15000}(Staff))$

2. **Commutativity of selection operations**

   $\sigma_p(\sigma_q(R)) = \sigma_q(\sigma_p(R))$

   E.g.

   $\sigma_{branchNo='B003'}(\sigma_{salary>15000}(Staff)) = \sigma_{salary>15000}(\sigma_{branchNo='B003'}(Staff))$

Query Optimization: Heuristic Approach

Strathmore UNIVERSITY

3. In a sequence of Projection operations, only the last in the sequence is required

$$\Pi_L (\Pi_M \ldots \Pi_N (R)) = \Pi_L (R)$$

E.g.

$$\Pi_{lName} (\Pi_{branchNo,\ lName}(Staff)) = \Pi_{lName}(Staff)$$

Query Optimization: Heuristic Approach

Strathmore UNIVERSITY

4. Commutativity of Selection and Projection

If the predicate p involves only the attributes in the projection list, then the Selection and Projection operations can commute as follows:

$\Pi_{A_1,\ldots,}\Pi_{A_m}(\sigma_p(R)) = \sigma_p(\Pi_{A_1,\ldots,}\Pi_{A_m}(R))$ where $p \in \{A_1, A_2, \ldots, A_n\}$

E.g.

$\Pi_{fName,\ lName}(\sigma_{lName='Lee'}(Staff)) = \sigma_{lName='Lee'}(\Pi_{fName,\ lName}(Staff))$

Query Optimization: Heuristic Approach

Strathmore UNIVERSITY

5. Commutativity of Theta join (and Cartesian product)

Note: This rule also applies to the equijoin and the natural join because they are special cases of the Theta join

$R \bowtie_p S = S \bowtie_p R$

$R \times S = S \times R$

E.g.

$Staff \bowtie_{Staff.branchNo=Branch.branchNo} Branch = Branch \bowtie_{Staff.branchNo=Branch.branchNo} Staff$

Query Optimization: Heuristic Approach

Strathmore UNIVERSITY

6. Commutativity of Selection and Theta join (or Cartesian product)

$\sigma_p (R \bowtie_r S) = (\sigma_p (R)) \bowtie_r S$

$\sigma_p (R \times S) = (\sigma_p (R)) \times S$

Where $p \in \{A_1, A_2, \ldots, A_n\}$ are attributes of one of the relations being joined

If the selection predicate is a conjunctive predicated of the form $(p \wedge q)$, where p involves only attributes of R, and q involves only attributes of S, then the Selection and Theta join commute as:

$\sigma_{p \wedge q} (R \bowtie_r S) = (\sigma_p (R)) \bowtie_r (\sigma_q (S))$

$\sigma_{p \wedge q} (R \times S) = (\sigma_p (R)) \times (\sigma_q (S))$

Query Optimization: Heuristic Approach

Strathmore UNIVERSITY

6. *…cont.* **Commutativity of Selection and Theta join (or Cartesian product)**

E.g.

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

Query Optimization: Heuristic Approach

Strathmore UNIVERSITY

7. Commutativity of Projection and Theta join (or Cartesian product)

If the projection list is of the form $L = L_1 \cup L_2$, where L_1 involves only attributes of R, and L_2 involves only attributes of S, then provided the join condition contains only attributes of L, the Projection and Theta join operations commute as follows:

$$\Pi_{L_1 \cup L_2}(R \bowtie_r S) = (\Pi_{L_1}(R)) \bowtie_r (\Pi_{L_2}(S))$$

E.g.

$$\Pi_{position,\ city,\ branchNo}(Staff \bowtie_{Staff.branchNo=Branch.branchNo} Branch) =$$
$$(\Pi_{position,\ branchNo}(Staff)) \bowtie_{Staff.branchNo=Branch.branchNo} (\Pi_{city,\ branchNo}(Branch))$$

Query Optimization: Heuristic Approach

Strathmore UNIVERSITY

- **Transformation rules that can be applied are:**

8. **Commutativity of Union and Intersection (but not Set Difference)**

   $R \cup S = S \cup R$

   $R \cap S = S \cap R$

9. **Commutativity of Selection and set operations (Union, Intersection, and Set Difference)**

   $\sigma_p (R \cup S) = \sigma_p (S) \cup \sigma_p (R)$

   $\sigma_p (R \cap S) = \sigma_p (S) \cap \sigma_p (R)$

   $\sigma_p (R - S) = \sigma_p (R) - \sigma_p (S)$

Query Optimization: Heuristic Approach

Strathmore UNIVERSITY

- **Transformation rules that can be applied are:**

10. **Commutativity of Projection and Union**

$$\Pi_L(R \cup S) = \Pi_L(S) \cup \Pi_L(R)$$

11. **Associativity of Theta join (and Cartesian product)**

$$(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$$

$$(R \times S) \times T = R \times (S \times T)$$

If the join condition q involves only attributes from the relations S and T, then the Theta join is associative in the following manner:

$$(R \bowtie_p S) \bowtie_{q \wedge r} T = R \bowtie_{p \wedge r} (S \bowtie_q T)$$

Query Optimization: Heuristic Approach

Strathmore UNIVERSITY

- **Transformation rules that can be applied are:**

…cont. **E.g.**

$$(\text{Staff} \bowtie_{\text{Staff.staffNo=PropertyForRent.staffNo}} \text{PropertyForRent}) \bowtie_{\text{ownerNo=Owner.ownerNo} \wedge \text{Staff.lName=Owner.lName}} \text{Owner} = \text{Staff} \bowtie_{\text{Staff.staffNo=PropertyForRent.staffNo} \wedge \text{Staff.lName=lName}} (\text{PropertyForRent} \bowtie_{\text{ownerNo=ownerNo}} \text{Owner})$$

Query Optimization: Heuristic Approach

Strathmore UNIVERSITY

- **Transformation rules that can be applied are:**
  - **12. Associativity of Union and Intersection (but not Set difference)**

$$(R \cup S) \cup T = S \cup (R \cup T)$$

$$(R \cap S) \cap T = S \cap (R \cap T)$$

(optional for an undergraduate degree) For a more detailed, original proof of the heuristic transformation rules, please refer to Alfred Vaino Aho's articles published in 1979

Query Optimization: Heuristic Approach

Strathmore UNIVERSITY

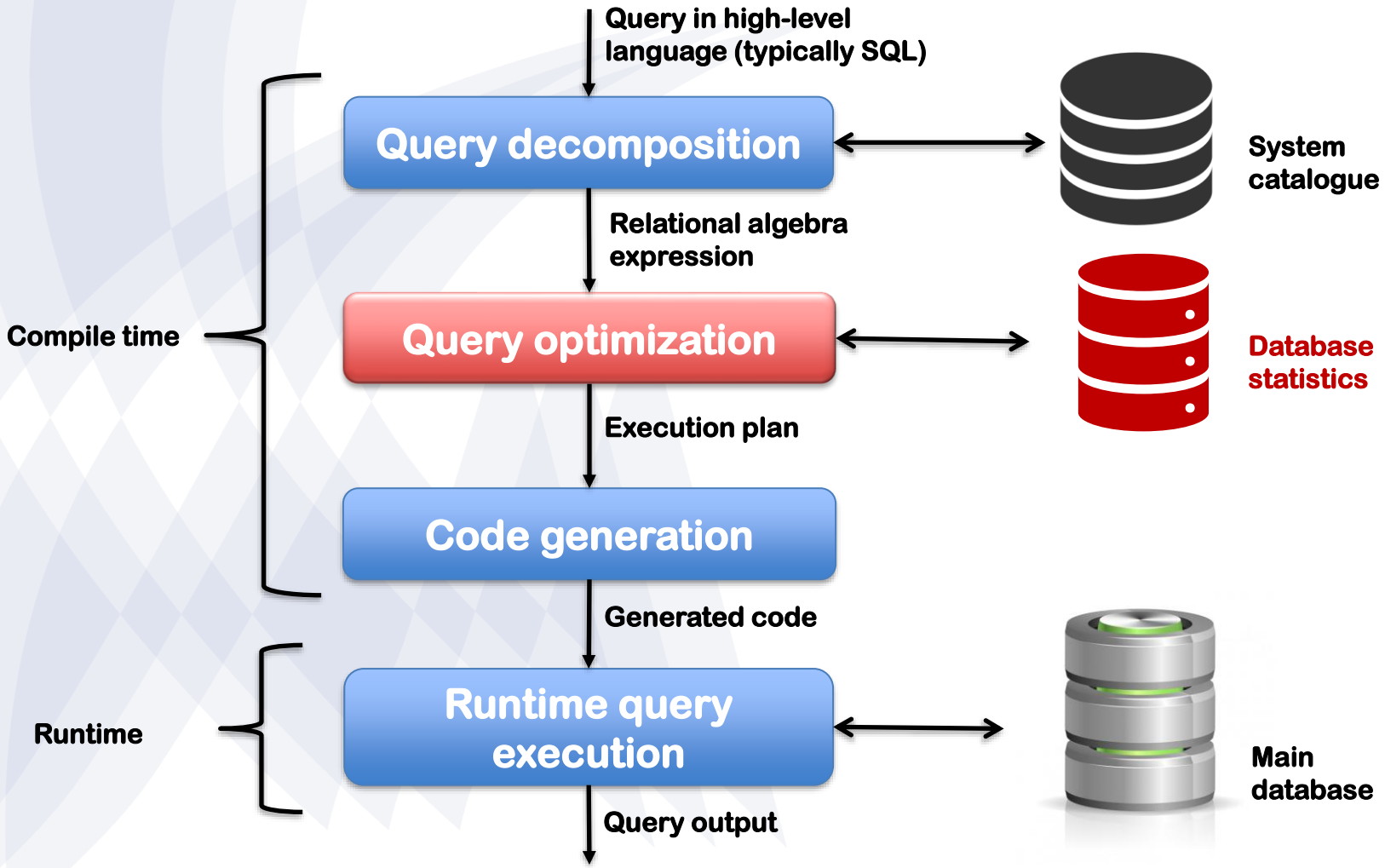
- A summary of the Heuristic processing strategies that can be applied by the DBMS are:
  - Perform Selection operations as early as possible
  - Combine the Cartesian product with a subsequent Selection operation whose predicate represents a join condition into a join operation
  - Use associativity of binary operations to rearrange leaf nodes so that the leaf nodes with the most restrictive Selection operations are executed first
  - Perform Projection operations as early as possible
  - Commute common expressions once

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Stages of Query Processing



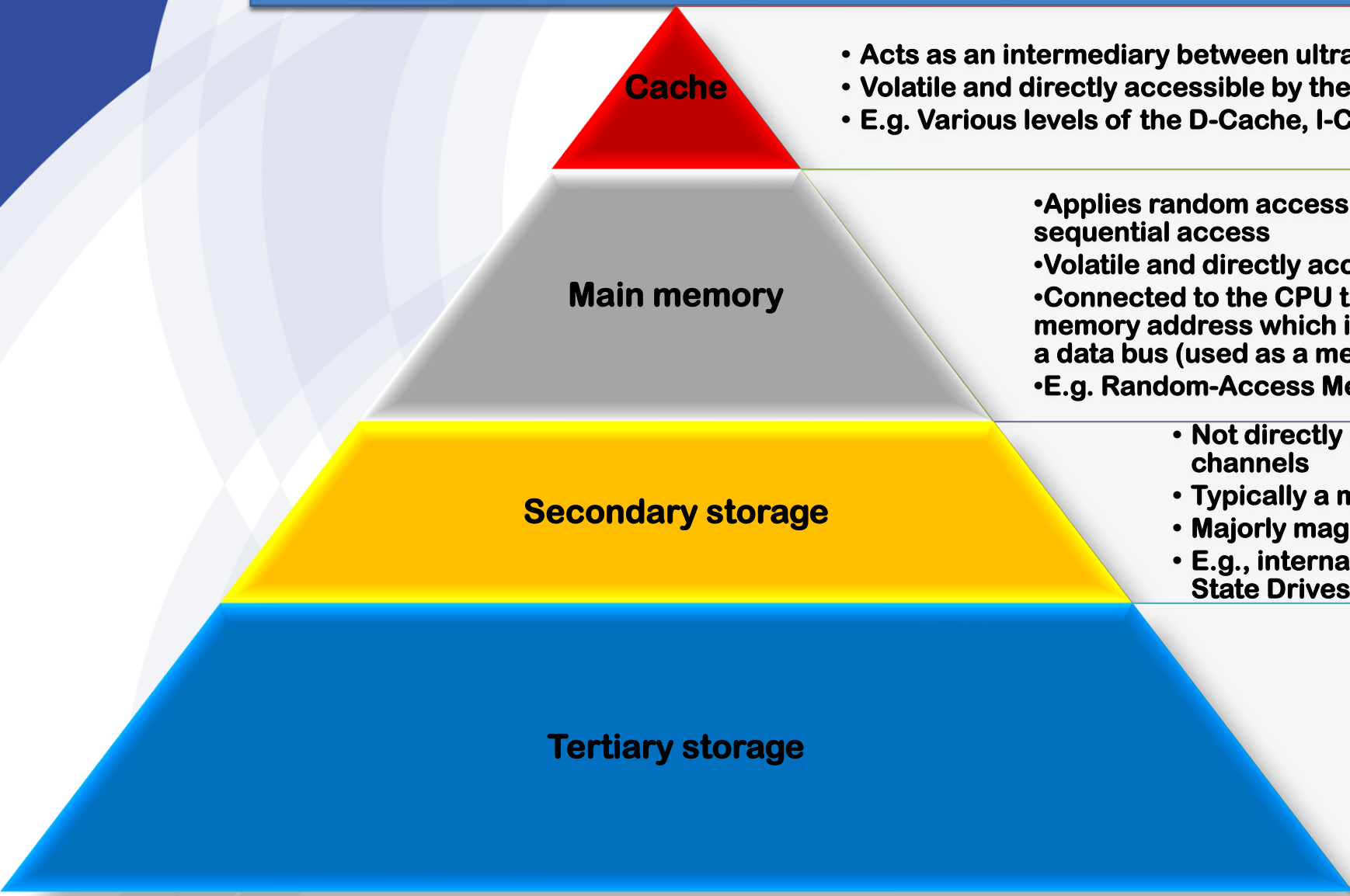
Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

- The cost-based approach uses formulae that estimate the costs for a number of options and select the one with the lowest cost
- This is done as part of the query optimization stage of query processing so as to choose the most efficient way to execute a query
- The most dominant cost in query processing is IO cost (a.k.a. disk access), which can either be from tertiary storage, secondary storage, main memory, or cache (the most common source is from secondary storage, i.e. HDD or SSD)

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY



- Acts as an intermediary between ultra fast processor registers and main memory
- Volatile and directly accessible by the CPU
- E.g. Various levels of the D-Cache, I-Cache, and Translation Lookaside Buffer (TLB)

- Applies random access (also known as direct access) as opposed to sequential access
- Volatile and directly accessible by the CPU
- Connected to the CPU through an address bus (used to send a memory address which indicates the location of the desired data) and a data bus (used as a medium to read/write data from main memory)
- E.g. Random-Access Memory (RAM)

- Not directly accessible by the CPU; accessed through I/O channels
- Typically a million times slower than main memory
- Majorly magnetic disks
- E.g., internal Hard Disk Drives (HDD) and internal Solid-State Drives (SSD)

- Large capacities and smaller cost per Byte
- Can be offline or nearline
- Primarily used for archiving, transferring data and backups
- E.g., magnetic tape, optical disks, Network Attached Storage (NAS), cloud storage and flash memory devices

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

- Most of the cost estimates are based on the cardinality of the relation
- Various database statistics are stored by the DBMS in the system catalogue
- These statistics are used in cost estimation

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

For each base relation R:

- **nTuples(R)**: the number of tuples in relation R; the cardinality of R
- **bFactor(R)**: the number of tuples that fit into one block; the blocking factor of R
- **nBlocks(R)**: the number of blocks required to store R. If the tuples of R are stored together, then:

  nBlocks(R) = [nTuples(R)/bFactor(R)]

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

For each attribute A of base relation R:

- $nDistinct_A(R)$: the number of distinct values that appear for attribute A in relation R
- $min_A(R)$, $max_A(R)$: the minimum and maximum possible values for the attribute A in relation R
- $SC_A(R)$: The average number of tuples that satisfy an equality condition on attribute A; i.e. the selection cardinality of attribute A

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

For each multilevel index "I" on attribute set "A":

- $nLevels_A(I)$: the number of levels in I
- $nLfBlocks_A(I)$: the number of leaf blocks in I

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

- It would be useful to have the statistics up to date
- However, updating the statistics every time an insertion, update, or deletion operation is performed can have a negative impact on performance
- As a better alternative, DBMSs update the statistics on a periodic basis, e.g. when the system is idle

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

(1) Linear search on a heap file that has no index

- Assuming that tuples are uniformly distributed into blocks, then on average half the blocks will be searched before the specific tuple is found

  [nBlocks(R)/2]

- For any other condition, the entire file may have to be searched with a cost of

  nBlocks(R)

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

(2) Binary search on a sorted file that has no index

- The cost of finding the first tuple that satisfies the predicate will be:

  $[\log_2(nBlocks(R))]$

- $SC_A(R)$ will give us the average number of tuples expected to satisfy the predicate. These tuples will occupy $[SC_A(R)/bFactor(R)]$ blocks, such that the cost of searching the first block has already been catered for in $[\log_2(nBlocks(R))]$
- The cost estimate where more than 1 tuple satisfies the predicate will therefore be:

  $[\log_2(nBlocks(R))] + [SC_A(R)/bFactor(R)] - 1$

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

(3) Equality search on an attribute that has a hash index

- The hash function is applied on the attribute to determine the exact location of the target address
- If there is no overflow, then the estimated cost is 1
- If there is an overflow, then additional cost will be incurred in order to traverse through the overflow

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

(4) Equality search on an indexed primary key

- A clustered index can be used to retrieve the tuple that satisfies the predicate if the predicate involves an equality condition on the primary key
- The cost is therefore based on retrieving each level of the index and retrieving the tuple itself:

$$nLevels_A(I) + 1$$

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

(5) Inequality search on an indexed primary key

- Inequality conditions include: <, <=, >, >=
- If the index is sorted, the target tuples can be found by accessing all tuples before the target
- Assuming a uniform distribution, then on average half the blocks containing tuples will be searched before the specific tuple is found
- The estimated cost would therefore be:

  $nLevels_A(I) + [nBlocks(R)/2]$

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

(6) Equality search on an indexed (clustered) non-primary key

- A clustered index that has been defined on a non-primary key attribute can be used to locate the attribute's values
- $SC_A(R)$ will give us the average number of tuples expected to satisfy the predicate. These tuples will occupy $[SC_A(R)/bFactor(R)]$ blocks
- Therefore, the total estimated cost will be: $nLevels_A(I) + [SC_A(R)/bFactor(R)]$

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

(7) Equality search on an indexed (non-clustered) non-primary key

- A non-clustered index that has been defined on a non-primary key attribute can be used to locate the attribute's values
- The cost of accessing the index is $nLevels_A(I)$
- $SC_A(R)$ will give us the average number of tuples expected to satisfy the predicate. Given that it is a non-clustered index and assuming the tuples are on different blocks, the cost for this will be $SC_A(R)$
- The total cost is therefore $nLevels_A(I) + [SC_A(R)]$

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

(8) Inequality search on an indexed (B^+Tree) non-primary key

- Inequality conditions, on attribute A given condition x include: A<x, A<=x, A>x, A>=x
- This would involve a scan from the smallest value to x (for < or <=) and from x up to the maximum value (for > or >=)
- Subsequently resulting in half the leaf blocks being accessed, and half the actual tuples being accessed. The cost of accessing the B+Tree index is also added. The cost estimate is therefore:

$$nLevels_A(I) + [nLfBlocks_A(I)/2 + nTuples(R)/2]$$

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

Projections

- Based on the definition of a projection, its implementation involves the following steps:
  - Step 1: Removal of attributes that are not required
  - Step 2: Elimination of any duplicate tuples that are produced from the previous step
- Each step has a cost associated with it
- This is beyond the scope of this unit at undergraduate level however, if you wish, you can conduct research on the estimated cost of each step in a projection

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Query Optimization: Cost-Based Approach

Strathmore UNIVERSITY

- Join operations are the second-most time consuming operations to process (second to the Cartesian product)
- Although also beyond the scope of this unit at undergraduate level, if you wish, you can conduct research on the estimated cost of the following joins:
  - Block nested join
  - Indexed nested loop join
  - Sort-merge join
  - Hash join

Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Activity 2: Writing-Based Learning

- You **must** be able **to think**, **to read**, and **to write** by the end of your 4-year undergraduate degree
- In as much as you have a right to quality education, you also have a **responsibility** to use this ability to make the world a better place
- The reason why you have rights is so that you can execute your responsibilities
- Other specialized knowledge is learnt through on-the-job-training (OJT) over years of experience
- One of the ways you can develop your thinking capacity is through writing

Activity 2: Writing-Based Learning

Strathmore UNIVERSITY

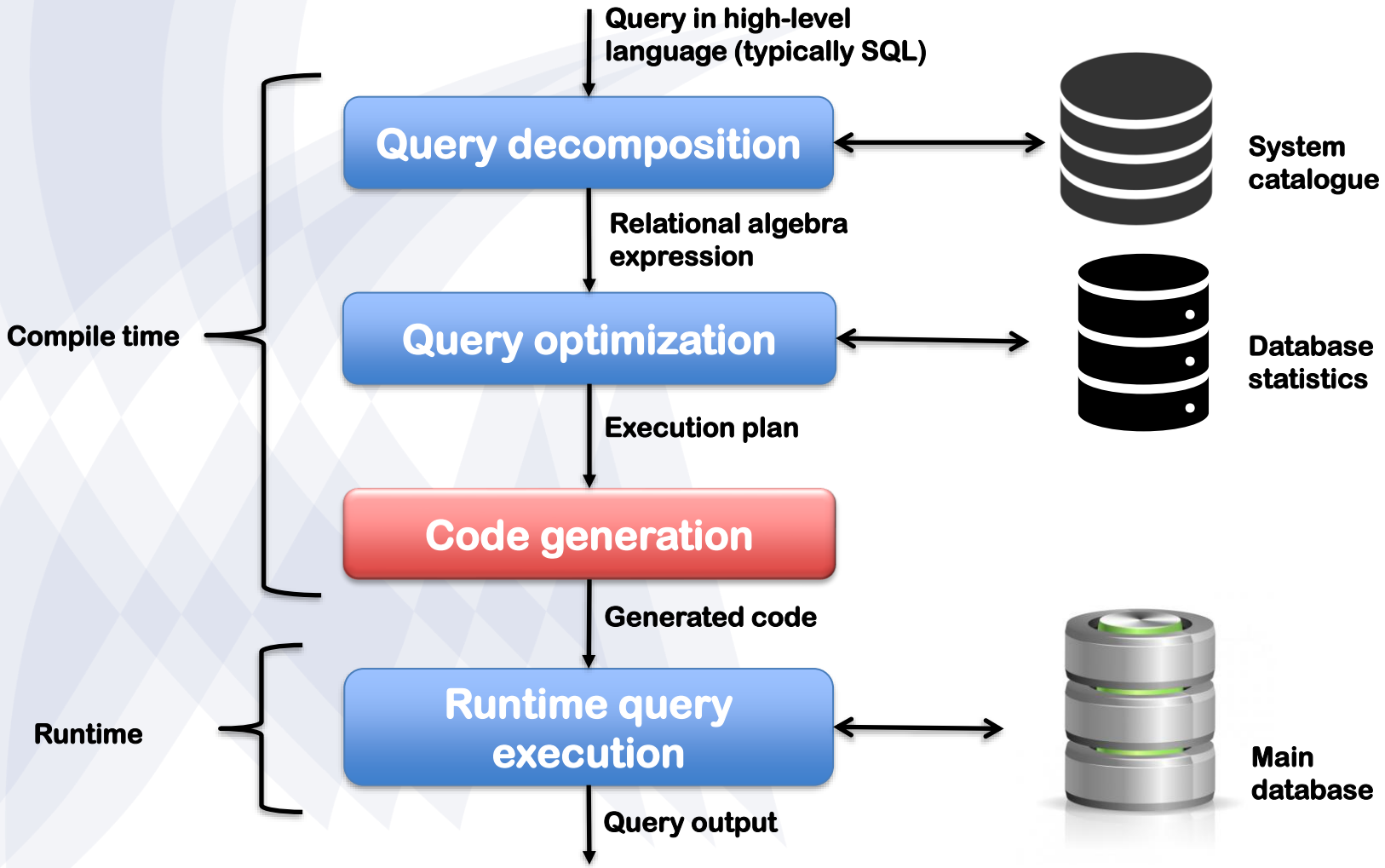
- **Writing is formalized thinking**. It involves:
  - Picking the right words to include in a sentence
  - Grouping sentences correctly to form a paragraph
  - Developing a coherent argument by sequencing your paragraphs correctly
  - Ironing out the contradictions between the concepts you have learnt
  - Critiquing your own ideas as intensely as you critique the ideas of others
  - Breaking down old ideas to allow new ideas to form
  - Discovering and revealing the truth
- Write a 700 – 2,000-word **essay** on how each of the 8 cost estimation techniques (refer to slides 75 to 89) can be combined with various heuristic rules during query optimization by the DBMS and the effect this has on reducing the workload of a DBA

Activity 2: Writing-Based Learning

- **Upload** your work via the link provided on eLearning
- **Approximate time required: 1 week**

Strathmore UNIVERSITY

Stages of Query Processing

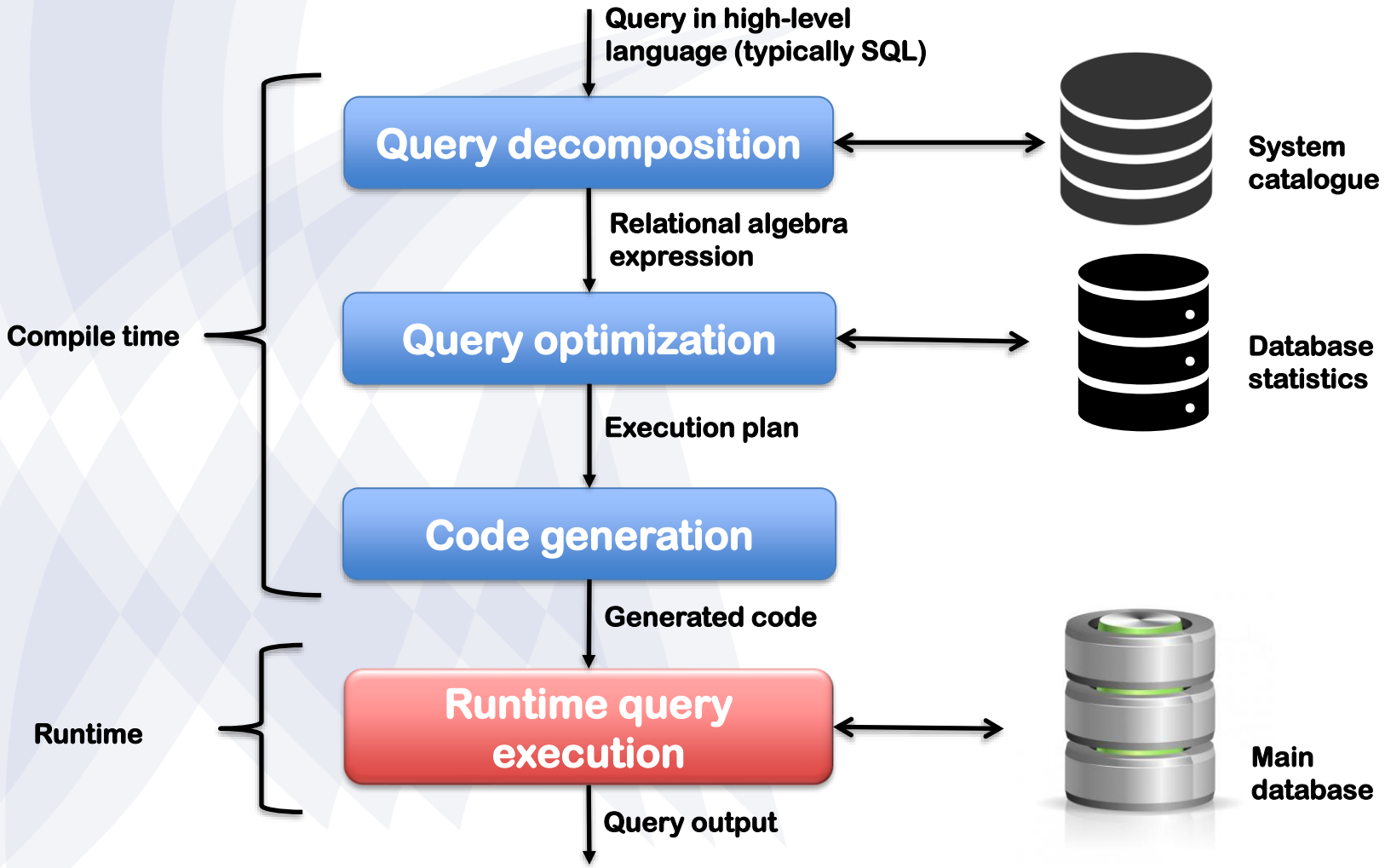


Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Stages of Query Processing



Presentation Outline

Strathmore UNIVERSITY

- ✓ Introduction to Query Processing
  - Activity 1: Class Discussion on High-Level to Low-Level Query Transformation
- ✓ Query Decomposition
  - Analysis
  - Normalization
  - Semantic Analysis
  - Simplification
  - Query Restructuring
- ✓ Query Optimization
  - Heuristic Approach
  - Cost-Based Approach
    1) Linear search on a heap file that has no index
    2) Binary search on a sorted file that has no index
    3) Equality search on an attribute that has a hash index
    4) Equality search on an indexed primary key
    5) Inequality search on an indexed primary key
    6) Equality search on an indexed (clustered) non-primary key
    7) Equality search on an indexed (non-clustered) non-primary key
    8) Inequality search on an indexed (B^+ Tree) non-primary key
  - Cost Estimation of Projection Operations
  - Cost Estimation of Join Operations
  - Activity 2: Writing-Based Learning: Essay on query optimization techniques
- ✓ Code Generation
- ✓ Runtime Query Execution

Further Reading and References

Strathmore UNIVERSITY

Connoly, T., & Begg, C. (2015). Query Processing. In *Database systems: A practical approach to design, implementation, and management* (6th ed.). Addison-Wesley.

Curtis, D. (2016). *Among the clouds*. https://www.darrencurtismusic.com/relaxing

Pixabay (2015). *Gray Scale Photo of Gears. Free stock photo* [Photograph]. Retrieved from https://www.pexels.com/photo/gray-scale-photo-of-gears-159298/