**Chapter 7: QUERY PROCESSING & OPTIMIZATION**

**Query Processing**

* Conversion of **high-level** queries into **low-level** machine instructions.

**Steps in Query Processing (Summary)**

* Parsing & translation
* Optimization
* Evaluation

**Steps in Query Processing (Brief)**

Parsing and translation:-

* **SQL** is **human readable**, while **relational algebra** is **machine readable**.
* So, we first convert **SQL** **statements** to **relational algebra**.
* It involves use of ***parser*** and ***translator***.

Optimization:-

* Uses a program called ***optimizer***.
* **Optimizer** selects the best way to **evaluate** a given query.
* **Query evaluation plan:** Query tree **+** Algorithm

Evaluation engine:-

* **Evaluates** all query plans & selects the **best** one from it.
* Then **performs** its operation on the database.
* And **fetches/stores** requested data from database.

**Measures of Query Cost**

* **Query cost:** Total time taken to **execute** & **fetch/store** data from database.

Factors affecting query cost are:-

* **Communication cost** (bandwidth/electricity)
* **CPU cycles** (clock cycles)
* **Disk access** (covers most of total time taken)
* **Disk access cost** (number of bytes read/written)

***\*Cost of writing is more than reading\****

**Selection Operation**

* **File scan:** The search algorithm used to **find** & **fetch** a data.
* Symbol for ***selection operator*** is **σ**.

Syntax example:-

**σcondition (relation)**

* Searching algorithms used in selection operations:
  + Linear search
  + Binary search

Linear search:-

* **Key attributes:** Primary attributes with only **single instance** in record.
* For **key attributes**, the search can stop immediately if the record is found.
* This costs **BR/2** in **best case** (**BR** is the number of blocks).
* Otherwise, for **non-key attributes** the linear search keeps running with **average case**.

Binary search:-

* Relations are **ordered** as per their attributes.
* This costs **log2(BR)** on average case.

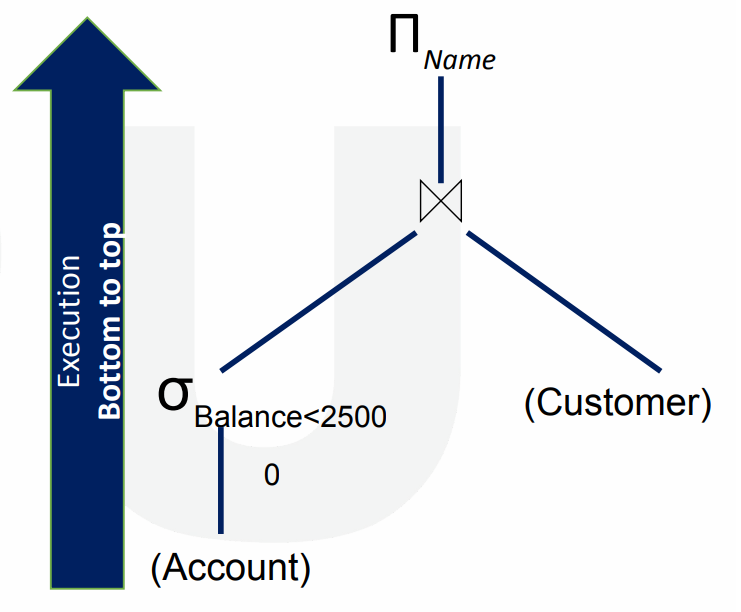
**Evaluation of Expressions**

* Query & expression are the **same** thing.

Methods to evaluate multiple operations in expression:-

* ***Materialization***
* ***Pipelining***

**Materialization**



* Starts from the **bottom** of the expression & does operations **one-by-one**.
* **Small parts** of the equation are **solved** & **stored** before moving to **bigger part**.

**Materialization cost = Cost of individual eqn + Cost of writing each small part of that eqn**

Disadvantages of materialization:-

* **High cost** in general
* Creates **too many relations** for intermediate results (**small parts** of the eqn)

**Pipelining**

* An **output** of an operation is used as an **input** in another operation.
* Thus, it **reduces** the number of **temporary relations**.
* And that results in **reduction** in **total cost**.
* Two ways to execute pipelining are:
  + ***Demand driven***
  + ***Producer driven***

Demand driven:-

* Also known as **lazy driven**.
* Requests for tuple/queue at **top** with **final inputs** of pipeline.

Producer driven:-

* Also known as **eager driven**.
* Operations produce **tuple/queue** without waiting for requests.

**Query Optimization**

* A **better** optimization plan has **lower cost**.
* Query optimization approaches:
  + ***Cost-based approaches***
  + ***Heuristic-based approaches***

Cost-based optimization:-

* This approach lists **all possible plans** & chooses the one with least cost.
* Also known as ***exhaustive search optimization***.

Heuristic-based optimization:-

* Comparatively **less expensive** method.
* Three heuristic rules:-
  + **Rule 1:** Make **early selection** to reduce the **number of tuples/queues**.
  + **Rule 2:** Make **early projection** to reduce the **number of attributes**.
  + **Rule 3:** Make **smallest operations** first.

**Transformation of Relational Expression**

* Two **relational algebra expressions** are **equal** if they produce **same** set of tuples.

**Cost Based Optimization Approach**

* We call a plan **optimal** if it **answers fast** or occupies **low space**.
* The **cost** of an algorithm also depends on **cardinality** of the input.
* **Query tree:** A **tree** data structure containing all operation which when **linked**, perform the **query** the ordered way.
* **Cardinality** is important because it provides the output as input in **pipelining**.

Dependencies of cost-based optimization:-

* **Cardinality:** Number of rows **returned** as per the chosen **optimal plan**.
* **Selectivity:** Number of rows **selected** as per the given **condition**.
* **Cost**

Cost components of query execution:-

* **Secondary memory access cost**
* **Memory usage cost:** Number of memory **buffers** used.
* **Storage cost:** Files/data produced when producing final results (**intermediate results**).
* **Computational cost**
* **Communication cost:** Communicating results across system & transfer on internet etc.