B tree

* In B tree we can store both the key and data are stored in the internal leaf node
* It wastes space
* Searching becomes difficult
* B-tree does not stores redundant search key

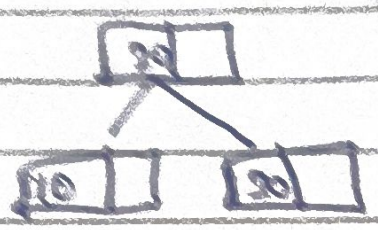
Example: Construct a B tree of order 3 for the following

10, 18, 30, 15, 11, 40, 20

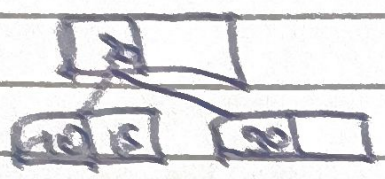
Step 1: Insert 10, 18 for ascending order

Step 2: Insert 30

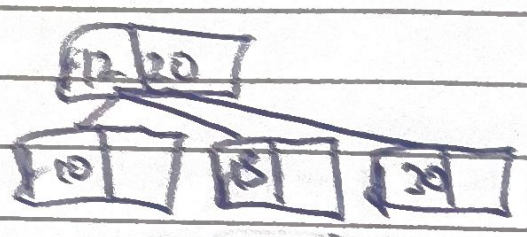
10 18 30



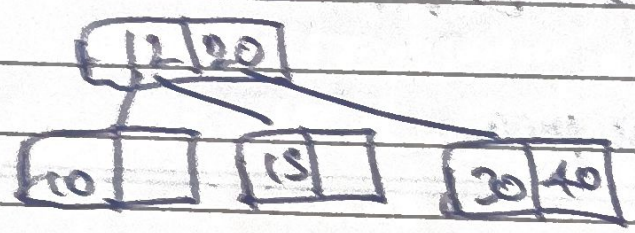
Step 3: Insert 15



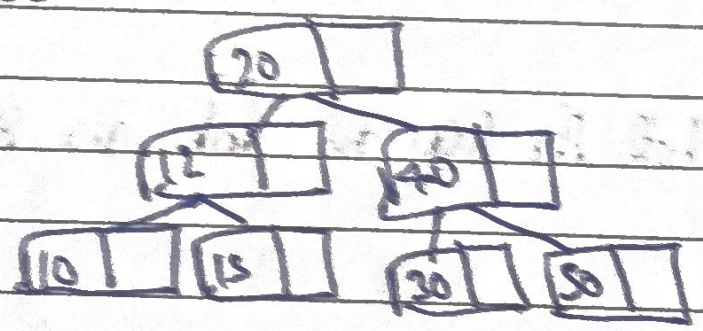
Step 4: Insert 11



Step 4 Insert 40



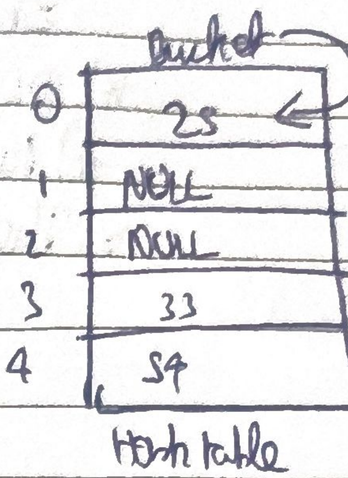
Step 6: Insert 50



Page No.:
Date:
YOUVA

Concept of Hashing

*) These are used to allocate indexes to files based on hash function



Basic terms:

- o) Hash table
- o) Hash function
- o) Bucket the storage condition
- o) Collision
- o) Node the process of calculating address
- o) Synonyms: Keys that hash to the same bucket

Static and Dynamic Hashing

*) It has Data, Data Address and Bucket
*) The Address are fixed and always same
*) The final digit of the data will be address of the data in index

EX 78 Mod(10) = 8

*) To find the Address Just, check the end value
*) Insertion also same

Closed Hashing:

*) If same Address is required for both data it Just stores to adjacent

Open Hashing:

*) It store to the next Address if collision occurs

Dynamic Hashing

* It is used to satisfy the overflow caused by static hashing
* one bit addressing checks the last one digit for addresses
* two bit addressing checks for the last two digits
* The Bucket grows or shrink dynamically

Query processing overview

* It converts high level queries with low-level instructions

Steps in query processing

1) Parsing and Translation

* The query is checked for correct syntax using parser
* The parsed query is translated into an internal representation, tree structure
* Example: SQL query

2) Optimization:

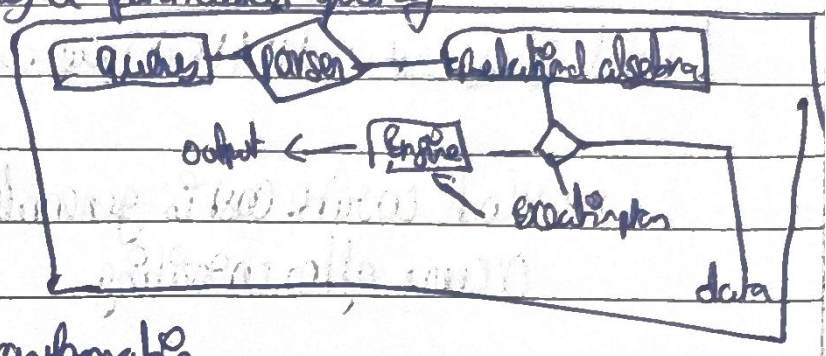
* Multiple query evaluation plans are created from relational algebra expression
* Each plan assigned a cost (heuristics)
* The optimizer chooses the plan with the lowest estimated cost

3. Evaluation:

*) The chosen query execution plan is executed by the query Engine
*) The engine retrieves results by performing operations.

Query cost:

*) The amount of space and time taken by a particular query



Key steps and operations:

o) Selections, projections, Joins, sorting
o) Query trees and relational algebra transformation
o) Cost-based heuristic-based optimization

Measure of query cost

*) The query cost helps system choose the best execution plan

Factors Contributing:

Disk Access: most influencial factor due to slower speed compared to CPU
CPU time: Computation time to executed queries
Communication cost: Cost of data transfer
Memory usage: Cost to store and process data in RAM

Disk Access dominates Query cost:

*) Its the slowest and easily measurable
*) Qr considers
  o) Number of disk seeks
  o) Number of blocks read
  o) Number of blocks written

Typical query cost formula:

- b : number of blocks transferred
- S : number of seeks
- t_T : time to transfer one block
- t_S : time to perform one seek

Total query cost: $$\text{Total query cost} = (b \times t_T) + (S \times t_S)$$

- Block write cost is generally higher than read because verification occurs after writing.

Further cost components:

- Access cost to secondary storage
- Computation cost (CPU)
- Communication cost
- Memory usage cost

Heuristic Cost-Estimation

*) Heuristic is a rule that leads to least cost in most of cases
*) Systems may use heuristics to reduce the number of choices that must be made in cost based fashion
*) Heuristic optimization transforms the query tree by using a set of rules that typically improve execution performance

The rule one

(1) performs selection early
(2) performs projection early
(3) performs most restrictive

Steps in Heuristics

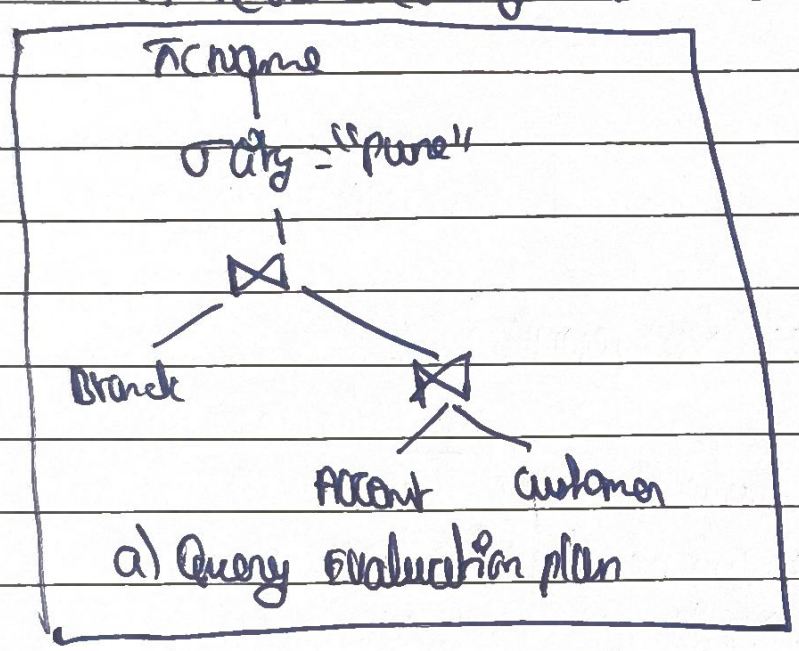
1) Scanner and parser generate initial query representation
2) Representation is optimized according to heuristic rules
3) Query execution plan is developed

Ex:

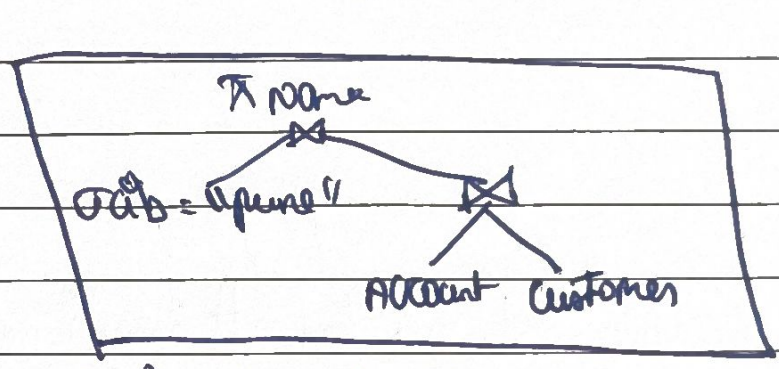
Suppose there are two relational Algebra

(1) $\sigma_{City = "Pune"}(\Pi_{Cname, Branch}(Branch \bowtie Account \bowtie Customer))$

(2) $\Pi_{Cname}(\sigma_{City = "Pune"}(Branch \bowtie Account \bowtie Customer))$



a) Query evaluation plan



b) It reach faster than (a)