Database Management Systems

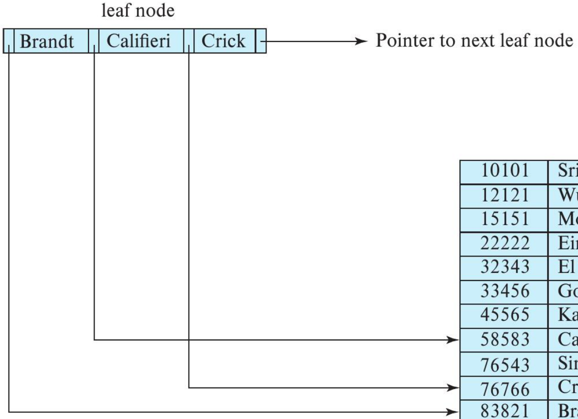
Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Sai University

Lecture 16, 20 October 2023

- Leaf nodes form a dense index — linked list of leaves, each one block

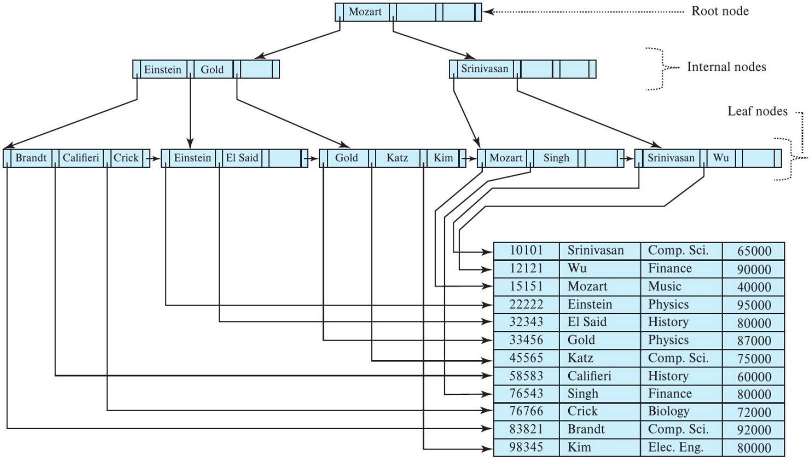


10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	80000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	60000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

instructor file

B+ trees

- Leaf nodes form a dense index — linked list of leaves
- Non-Leaf nodes form a sparse index



10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	80000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	60000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

B+ trees

- Leaf nodes form a dense index — linked list of leaves
- Non-leaf nodes form a sparse index
- Constraints — assume n keys and pointers can fit in a block
  - Each leaf has at least $\lceil (n-1)/2 \rceil$ key values
  - Each non-leaf has at least $\lceil n/2 \rceil$ pointers
  - Height of the tree is proportional to $\log_{n/2}(N)$

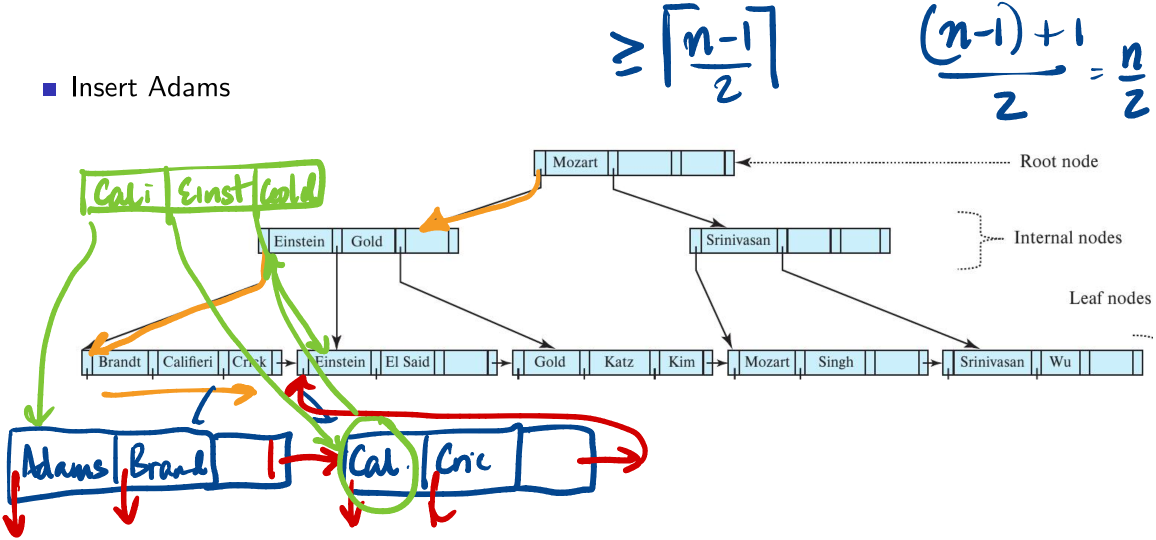
n ptrs

$n-1$ keys

$P_1\ K_1\ \ldots\ P_{n-1}\ K_{n-1}\ P_n$

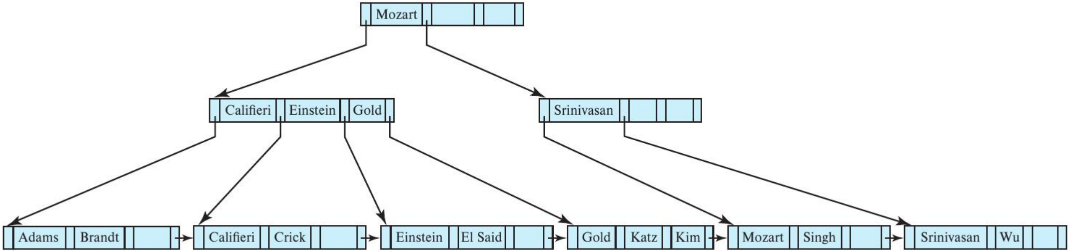
B+ trees — insertion

- Insert Adams



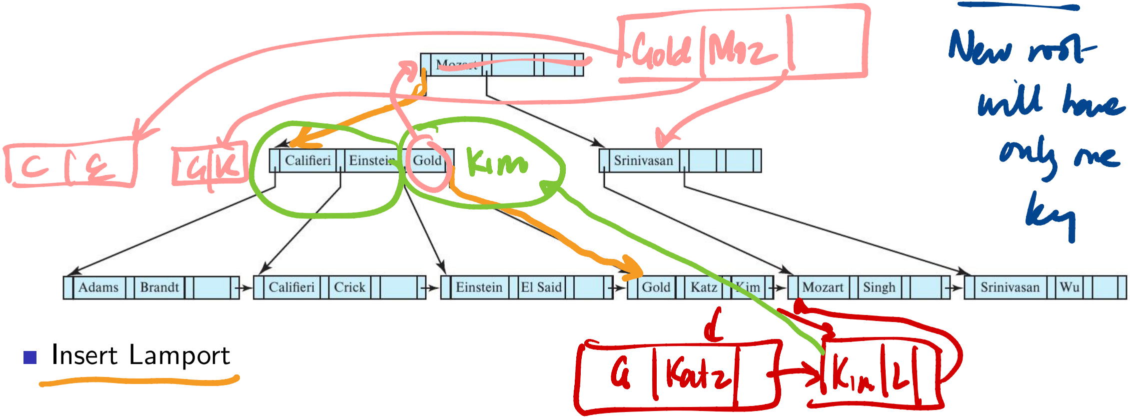
B+ trees — insertion

- Insert Adams



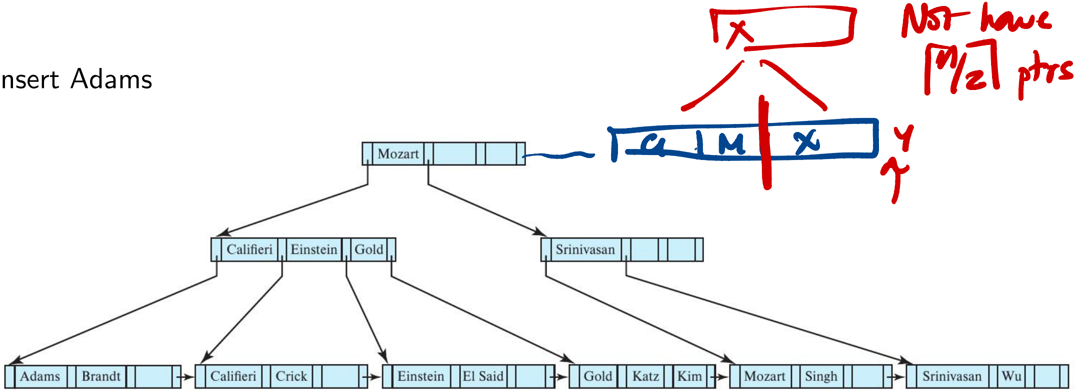
B+ trees — insertion

- Insert Adams



- Insert Lamport

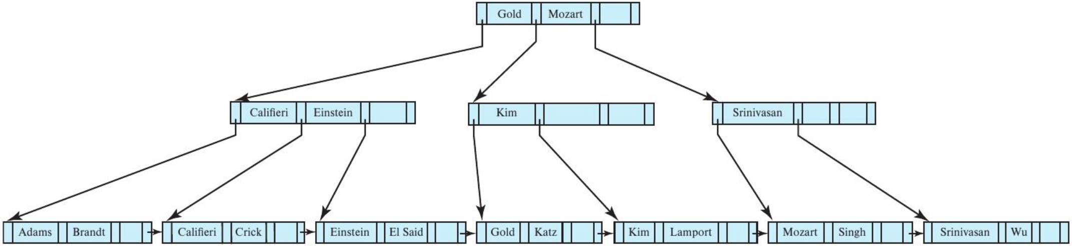
- Insert Adams



- Insert Lamport

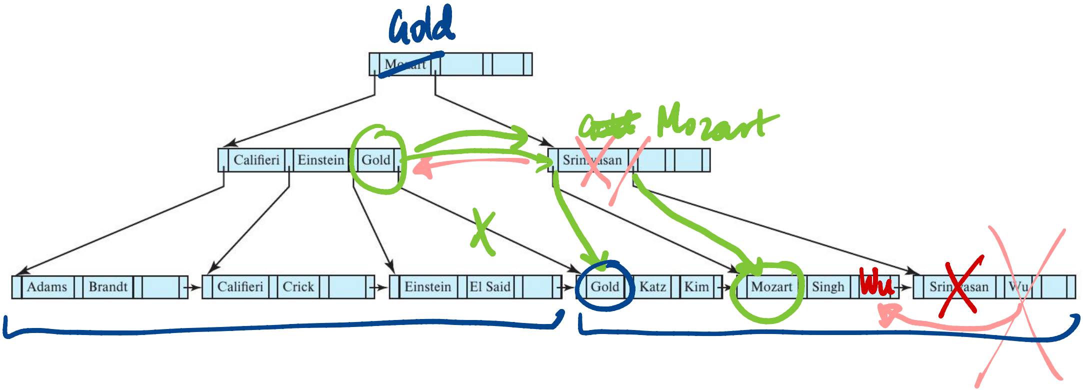
B+ trees — insertion

- Insert Adams
- Insert Lamport



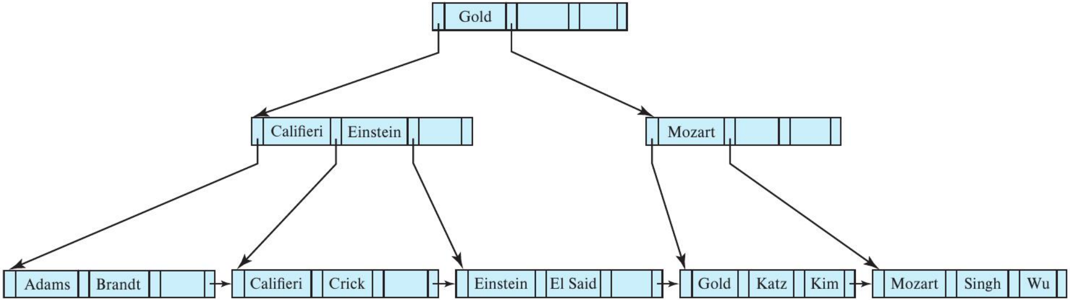
B+ trees — deletion

- Delete Srinivasan



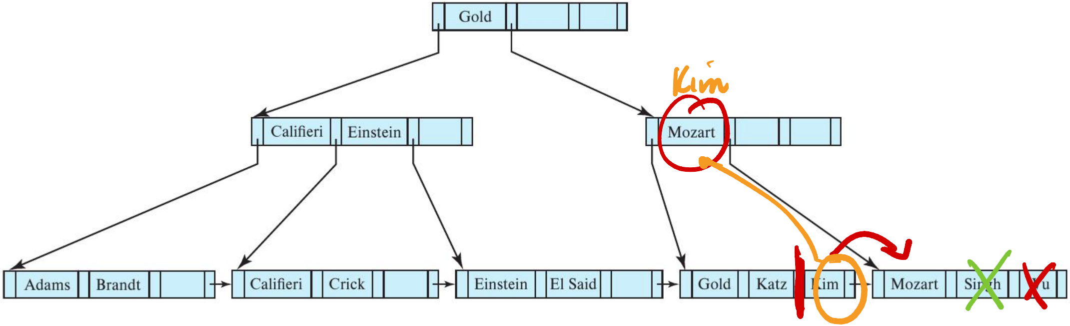
B+ trees — deletion

- Delete Srinivasan



B+ trees — deletion

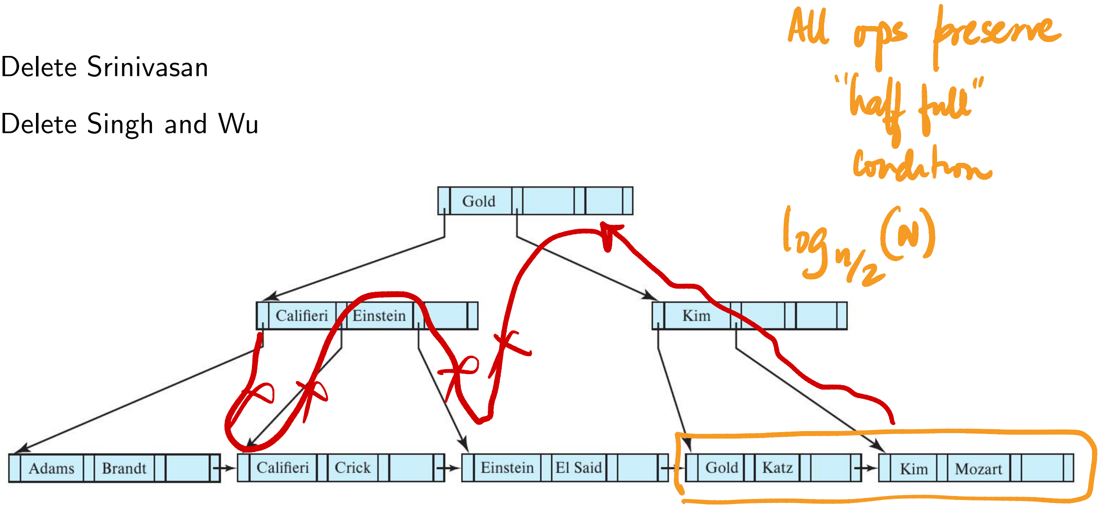
- Delete Srinivasan



- Delete Singh and Wu

B+ trees — deletion

- Delete Srinivasan
- Delete Singh and Wu



Query processing

- Translate the query from SQL into relational algebra
- Evaluate the relational algebra expression

Like a compiler

Query → Machine code

Query → Algorithm

Query processing

- Translate the query from SQL into relational algebra
- Evaluate the relational algebra expression
- Challenges

Query processing

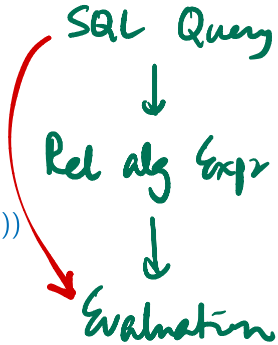
- Translate the query from SQL into relational algebra
- Evaluate the relational algebra expression
- Challenges
  - Many equivalent relational algebra expressions

    $\sigma_{salary<75000}(\pi_{salary}(instructor))$ vs $\pi_{salary}(\sigma_{salary<75000}(instructor))$

Query processing

- Translate the query from SQL into relational algebra
- Evaluate the relational algebra expression
- Challenges
  - Many equivalent relational algebra expressions

    $\sigma_{salary<75000}(\pi_{salary}(instructor))$ vs $\pi_{salary}(\sigma_{salary<75000}(instructor))$
  - Many ways to evaluate a given expression

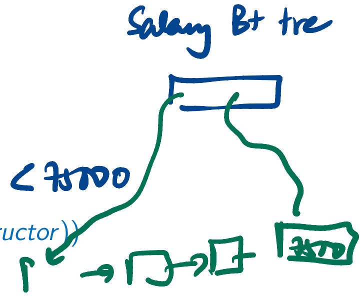


Query processing

- Translate the query from SQL into relational algebra
- Evaluate the relational algebra expression
- Challenges
  - Many equivalent relational algebra expressions

    $\sigma_{salary<75000}(\pi_{salary}(instructor))$ vs $\pi_{salary}(\sigma_{salary<75000}(instructor))$
  - Many ways to evaluate a given expression
- Query plan
  - Annotate the expression with a detailed evaluation strategy ~~key values~~

Query processing



- Translate the query from SQL into relational algebra
- Evaluate the relational algebra expression
- Challenges
  - Many equivalent relational algebra expressions

    $\sigma_{salary<75000}(\pi_{salary}(instructor))$ vs $\pi_{salary}(\sigma_{salary<75000}(instructor))$
  - Many ways to evaluate a given expression
- Query plan
  - Annotate the expression with a detailed evaluation strategy ~~key values~~
    - Use index on *salary* to find instructors with $salary < 75000$

Query processing

- Translate the query from SQL into relational algebra
- Evaluate the relational algebra expression
- Challenges
  - Many equivalent relational algebra expressions

    $\sigma_{salary<75000}(\pi_{salary}(instructor))$ vs $\pi_{salary}(\sigma_{salary<75000}(instructor))$
  - Many ways to evaluate a given expression
- Query plan
  - Annotate the expression with a detailed evaluation strategy key values
    - Use index on *salary* to find instructors with $salary < 75000$
    - Or, scan entire relation, discard rows with $salary \geq 75000$

Query optimization

- Choose plan with lowest cost

We know what the tables look like

Query optimization

- Choose plan with lowest cost
- Maintain database catalogue — number of tuples in each relationn, size of tuples, ...

Query optimization

- Choose plan with lowest cost
- Maintain database catalogue — number of tuples in each relationn, size of tuples, ...
- Assess cost in terms of disk access and transfer, CPU time, ...

10^{-3} 10^{-9} Ignore

SSDs 10^{-6}

Query optimization

- Choose plan with lowest cost
- Maintain database catalogue — number of tuples in each relationn, size of tuples, ...
- Assess cost in terms of disk access and transfer, CPU time, ...
- For simplicity, ignore in-memory costs (CPU time), restrict to disk access

Query optimization

- Choose plan with lowest cost
- Maintain database catalogue — number of tuples in each relationn, size of tuples, ...
- Assess cost in terms of disk access and transfer, CPU time, ...
- For simplicity, ignore in-memory costs (CPU time), restrict to disk access
- Disk accesses
  - Relation r occupies b_r blocks
  - Disk seeks — time t_S per seek
  - Block transfers — time t_T per transfer

Read vs write

Query optimization

- Choose plan with lowest cost
- Maintain database catalogue — number of tuples in each relationn, size of tuples, ...
- Assess cost in terms of disk access and transfer, CPU time, ...
- For simplicity, ignore in-memory costs (CPU time), restrict to disk access
- Disk accesses
  - Relation r occupies b_r blocks
  - Disk seeks — time t_S per seek
  - Block transfers — time t_T per transfer
- Other factors — buffer management etc

Which blocks to keep in memory?

Selection

(A1) Linear search

Read b_r blocks

"Seek" only first block

$b_r \cdot t_T + 1 \cdot t_S$

$\sigma_\theta(r)$

Selection

(A1) Linear search

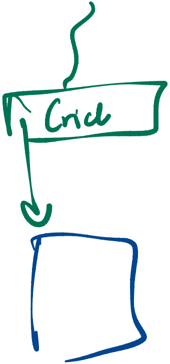
(A2) Clustering index, equality on key — index height h_i

single value

← # of index queries to reach specific record

h_i index lookups $t_S + t_T$

Actual record $t_S + t_T$



$$(h_i + 1) \cdot (t_S + t_T)$$

$$\sigma_\theta(r)$$

Selection

(A1) Linear search

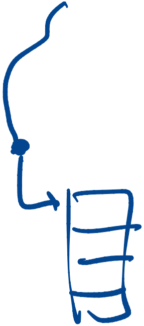
(A2) Clustering index, equality on key — index height h_i

(A3) Clustering index, equality on nonkey

table is sequential wrt key

h_i index lookups $t_S + t_T$

b blocks with matching records 1 $t_S + b t_T$



$$h_i(t_S + t_T) + t_S + b \cdot t_T$$

Selection

(A1) Linear search

(A2) Clustering index, equality on key — index height h_i

(A3) Clustering index, equality on nonkey

(A4) Secondary index (key, non-key)

Search key order ≠ table order

$h_i(t_S + t_T)$ + 1 block lookup

$(h_i + 1)(t_S + t_T)$, as before

Selection

(A1) Linear search

(A2) Clustering index, equality on key — index height h_i

(A3) Clustering index, equality on nonkey

(A4) Secondary index (key, non-key)

↳ n records match this search key

each is in a different block

$$h_i(t_S + t_T) + n(t_S + t_T)$$

Expensive

Selection

(A1) Linear search

(A2) Clustering index, equality on key — index height h_i

(A3) Clustering index, equality on nonkey

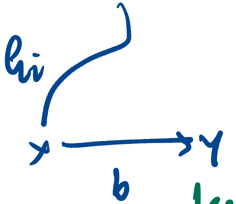
(A4) Secondary index (key, non-key)

(A5) Clustering index, comparison — sorted on A

$\sigma_{salary < 75000}$

$x < salary < y$

$h_i (t_S + t_T) + (t_S + b t_T)$



1 seek, b transfers

Selection

(A1) Linear search

(A2) Clustering index, equality on key — index height h_i

(A3) Clustering index, equality on nonkey

(A4) Secondary index (key, non-key)

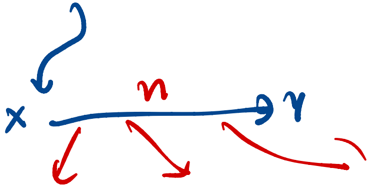
(A5) Clustering index, comparison — sorted on A

(A6) Clustering index, comparison — not sorted on A

Linear Scan

$h_i(t_S + t_T)$ index

$n \cdot (t_S + t_T)$



Complex selections

Conjunctions, disjunctions and negations

(A7) Conjunctive selection using one index

$\sigma_{\theta_1 \wedge \theta_2 \wedge \dots \wedge \theta_k}(r)$

↑ Index (on θ_2)

$\sigma_{\theta_2}(r)$ → check $\theta_1 \wedge \theta_3 \wedge \dots \wedge \theta_k$

$\sigma_{\theta_1 \wedge \theta_2}(r)$ — and

$\vee$ — or

$\neg\theta$ — not

Complex selections

Conjunctions, disjunctions and negations

(A7) Conjunctive selection using one index

(A8) Conjunctive selection using composite index

Complex selections

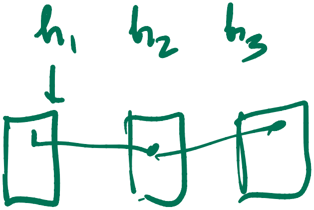
Conjunctions, disjunctions and negations

(A7) Conjunctive selection using one index

(A8) Conjunctive selection using composite index

(A9) Conjunctive selection using intersection of pointers

$$\sigma_{\theta_1 \wedge \theta_2 \wedge \theta_3}(r)$$



Complex selections

Conjunctions, disjunctions and negations

(A7) Conjunctive selection using one index

(A8) Conjunctive selection using composite index

(A9) Conjunctive selection using intersection of pointers

(A10) Disjunctive selection by union of pointers

Linear scan

$\sigma_{\theta_1 \vee \theta_2 \vee \theta_3 \ldots}(r)$

$h_1 \quad h_2 \quad \ldots \quad h_k$

Complex selections

Conjunctions, disjunctions and negations

(A7) Conjunctive selection using one index

(A8) Conjunctive selection using composite index

(A9) Conjunctive selection using intersection of pointers

(A10) Disjunctive selection by union of pointers

(Neg) Negation

Linear scan

σ π ✓

Join

$\sigma_\theta(r_1 \times r_2)$

Sorting

In memory sort

vs External (disk based) sort