

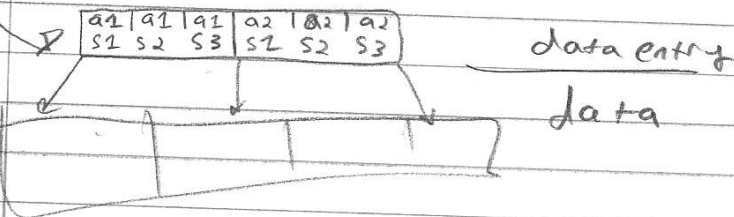
3/4/2008

Project 3 - replace clock replacement with CRU  
Ref: pg. 320

index =  $\langle \text{age}, \text{sal} \rangle$

Query 1:  $\text{age} = 25 \text{ AND } \text{sal} = 75k$  (uses the whole index)

Query 2:  $\text{sal} = 75k$  (index won't help - a better index  $\langle \text{sal}, \text{age} \rangle$  would help)



Rule 1: For point queries on attributes  $P = \{P_1, P_2, \dots\}$   
an index  $\langle a, b, c, \dots \rangle$  is useful if  $a \in P$

Range Queries:

Query:  $29 < \text{age} \leq 30 \text{ AND } 30k < \text{sal} < 50k$

Index:  $\langle \text{age}, \text{sal} \rangle \quad \langle \text{sal}, \text{age} \rangle$

both will use the index;  
performance depends on # of tuples returned  
after applying index condition (selectivity)

Rule 2: higher selectivity in higher-ranked attributes in the index

Rule 3: index helps iff  $p_i \in P$  can form a prefix of  $I$   
where  $I$  is the index and  $I = \{i_1, i_2, i_3, \dots\}$

Index-only plan: query can be processed w/o reading data pages.

(more about aggregates)

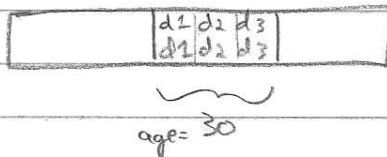
Example: Select E.dno, Count(\*)  
From Empl E  
Group By E.dno;

Empl Index <dno>

---

Select E.dno, count(\*)  
From E  
Where E.age = 30  
Group By E.dno

A good index would be <age, dno>



---

Select E.dno, Count(\*)  
From E  
Where E.age > 30  
Group By E.dno

If E.age > 30 is small, Index <age, dno>  
would be best. If not, Index <dno, age>  
might be better.

---

Sometimes basic stats of the index are cached in the nodes.

- I/O is not much of a factor
- Fan out is increased