

Techniques used in Evaluating Queries

- ① Indexing: If a join or select, use index to examine just the tuples that satisfy condition
- ② Iteration: Examining all tuples in turn (either of table or index)
- ③ Partition tuples on a sort key via sorting or hashing

Access Paths

An access path is a way of retrieving tuples from a table. Consists of either ① a file scan or ② an index + matching condition.

Considers a simple select

$$\sigma_{\bigwedge_j W_j(x_j) \text{ or } \bigvee_j P_j(y_j)}$$

CNF $<, \leq, =, \neq, \geq$

An index matches a selection condition if the index can be used to retrieve just the tuples that satisfy the condition

EX] A hash index matches a CNF selection if there is an atom of the form attribute=value in the selection for each attribute in the index's search key.

A tree index matches a CNF selection if there is a term of the form
attribute op value for each attribute
in a prefix of the index's search key.
(Suppose search key on $\langle a, b, c \rangle$)
 ~~$\langle a \rangle$~~ $\langle a, b \rangle$, $\langle a, b, c \rangle$ are prefixes of
 ~~$\langle a, b, c \rangle$~~ but $\langle b, c \rangle$ is not.
~~irrelevant~~

The conjuncts an index matches are called
the primary conjunct in the selection

EX Hash index H on $\langle rname, bid, sid \rangle$
 (T)
 $\sigma_{rname='Joe' \wedge bid=5 \wedge sid=3 \wedge bdate='Aug 1970'}$
primary conjunct
 H matches selection

Selectivity of access paths

The selectivity of an access path is the number
of pages retrieved (index pages plus data pages)
if we use this access path to retrieve all
desired tuples

more selective \Rightarrow fewer pages

The fraction of tuples in a table that
satisfy a given conjunct is called the
reduction factor

Ex Let H be on $\langle \text{name}, \text{bid}, \text{sid} \rangle$
 (Sailors), $r_{\text{name}} = \text{Joe} \wedge \text{bid} = 5 \wedge \text{sid} = 3$

Catalog does $N_{\text{Key}}(H)$ and $N_{\text{Pages}}(\text{Sailors})$

$$\text{Reduction Factor} = \frac{N_{\text{Pages}}(\text{Sailors})}{N_{\text{Keys}}(H)}$$

Algorithms for Relⁿ Ops

Selection

given $\sigma_{\text{attr op value}}(R)$ if index use
 else scan R

if have 2
 conjuncts
 disjoint
 easy index
 1st to
 narrow
 select then
 scan result

If ~~collection~~ more than 5% of rows
 returned cheaper to do scan

Projection

- use iteration overtable
 if distinct keyword not
 used. If distinct used
 then need to do partitioning
 (usually via sorting).

Join - $R \bowtie S$

$R.\text{sid} = S.\text{sid}$

could do

Scan R and do a subscan of S
 for each tuple of R :
 nested loop join.

If S has an index then instead of
 subscan use index
 index nested loop join.