CMSC 424 – Database design
Lecture 16
Query processing

Mihai Pop

Admin issues

- Questions about midterm?
- Questions about project?

Sample midterm questions

- Do I need to know about: 4NF, multivalued dependencies? - NO
- 1. Given the schema R(A,B,C,D,E), and functional dependencies A->D, B->C, CD->E, A->BC, E->B.
- a) Is the schema in BCNF? If not, list an FD that violates BCNF.
- b) Is the schema in 3NF? If not, list an FD that violates 3NF.
-
- Decompose the schema from problem 1 into BCNF and 3NF.

Oracle: explain plan

```
delete plan_table;
explain plan for
select name
from country
where population > 10000000 ;
```

Explained

```
select
  substr(lpad(' ', level – 1) || operation || ' (' || options || ')', 1, 30) “Operation”,
  object_name “Object”
from
  plan_table
start with id = 0
connect by prior id = parent_id;
```

```
Operation                     Object
----------------------------- -----------------------------
SELECT STATEMENT ()
 TABLE ACCESS (FULL)          COUNTRY
```

How to think about query processing

- n(r), b(r), f(r), V(A, r), SC(A, r) – values that can be computed without knowing what query you might run
- Think about how many results your query might retrieve
- Think about how they are organized on disk:
  - sorted (A is a clustering index)
  - unsorted (A is a secondary index)
- Think about how the index is organized – how many index blocks you need to hit to find the correct answer?
- Usually think of either average or worst-case scenarios.
- When retrieving range – think about what fraction that range represents from the total range of values in database.

Selection / Projection File Scan

- A1: search for equality: R.A=c cost (seq. search rel. sorted)

  $= b(r)/2 + \lceil SC(A,r)/f(r) \rceil - 1$ average successful

  $= b(r)/2$ average unsuccessful
- A2: (binary search)

  $= \lceil \log b(r) \rceil + \lceil SC(A,r)/f(r) \rceil - 1$ average successful
- Size of the result: $n(\sigma(R.A=c)) = SC(A,r) = n(r) / V(A,r)$
- search for inequality: R.A>c
  - cost (file unsorted) = b(r)

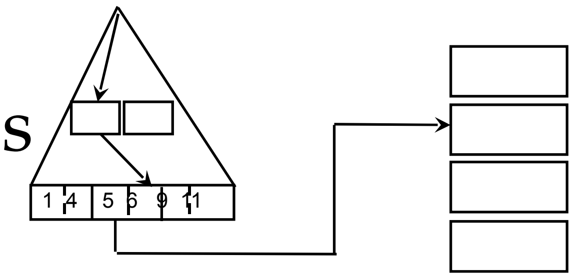
    (sorted on A) = b(r)/2+ b(r)/2 (if we assume that half of the tuples qualify)

  - size of the result: $n(\sigma(R.A>c)) = [max(A,r)-c] * n(r) / [max(A,r) - min(A,r)]$

- projection on A
  - cost as above
  - size of the result: $n(\pi(R,A)) = V(A,r)$

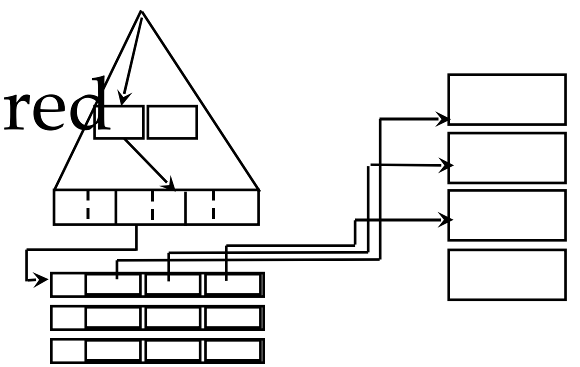
Selection with Indexed Scan R.A=c

- A3: Primary index on key:
  - cost = (height + 1) + 1
    height+1 is needed to get to the leaves
    (unsuccessful stops at the leaves)



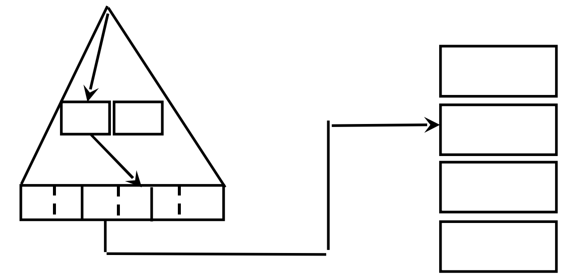
- A4: Primary (clustering) index on non-key:
  - cost = (height + 1) + 1 + ⌈ SC(A,r)/f(r) ⌉
    all tuples with the same value are clustered

- A5: Secondary (non-clustering) index
  - cost = (height + 1) + 1 + SC(A,r)
    tuples with the same value are scattered
  - It can be very expensive



- **size of the result: n(σ(R.A=c))=SC(A,r)=n(r) / V(A,r)**

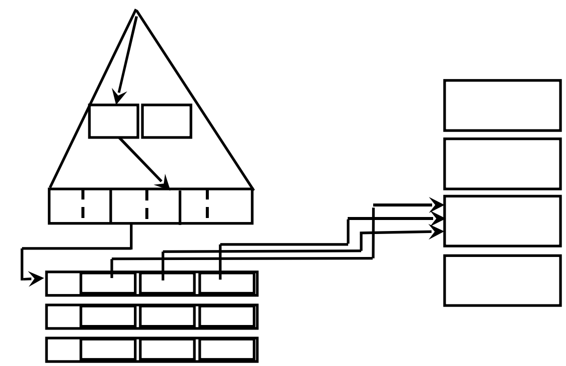
Selection with Indexed Scan R.A>=c



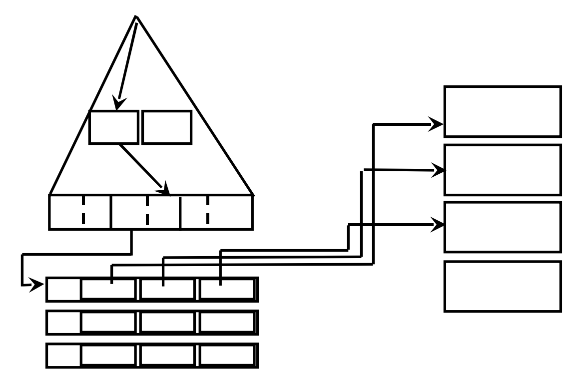
A6: Primary index on key:

- search for A=c; then pick tuples with A >= c

- cost = (height + 1) + b(r)/2 w/o a bound constant c
- = -"- + n(r) (max(A,r)-c)/(max(A,r)-min(A,r))/f(r)



- Primary (clustering) index on non-key:
  - cost = as above (all tuples with the same value are clustered)



A7: Secondary (non-clustering) index

- cost = (height + 1) + B-treeLeaves/2 + n(r)/2 or
- = -"- + -"- +
- + {1 + SC(A,r)}((max(A,r)-c)

tuples with the same value are scattered

can be more expensive than file scan

- size of the result:

n(σ(R.A>c))= [max(A,r)-c] * n(r) / [max(A,r) - min(A,r)]