COSC 460 Lecture 5: Indexes, Part I

Professor Michael Hay
Fall 2018

Review of pset3

(on board)

Cow book exercises

Suppliers(sid,sname,addr)
Parts(pid,pname,color)
Catalog(sid,pid,cost)

1. Find the names of suppliers who supply some red part.

2. Find the sids of suppliers who supply some red or green part.

3. Find the sids of suppliers who supply some red part or are at 221 Packer Street.

4. Find the sids of suppliers who supply some red part and some green part.

5. Find the sids of suppliers who supply every part.

6. Find the sids of suppliers who supply every red part.

Cow book exercises

Suppliers(sid,sname,addr)
Parts(pid,pname,color)
Catalog(sid,pid,cost)

7. Find the sids of suppliers who supply every red or green part.

8. Find the sids of suppliers who supply every red part or supply every green part.

9. Find pairs of sids such that the supplier with the first sid charges more for some part than the supplier with the second sid.

10. Find the pids of parts supplied by at least two different suppliers.

11. Find the pids of the most expensive parts supplied by suppliers named Yosemite Sham.

12. Find the pids of parts supplied by every supplier at less than $200. (If any supplier either does not supply the part or charges more than $200 for it, the part is not selected.)

Indexes: Introduction

- Sometimes, we want to retrieve records by specifying *values in one or more fields*, e.g.,
  - Find all students in the "CS" department
  - Find all students with a gpa > 3.0
  - Find all students in CS with a gpa > 3.0

- **Index**: a *disk-based, auxiliary* data structure that speeds up selections on some *search key fields*.
  - Any subset of the fields of a relation can be the search key for an index on the relation.
  - *Search key* is not the same as (primary) *key (e.g., Search keys don't have to be unique).*
  - *A relation can have multiple indexes*

Indexes: searches supported

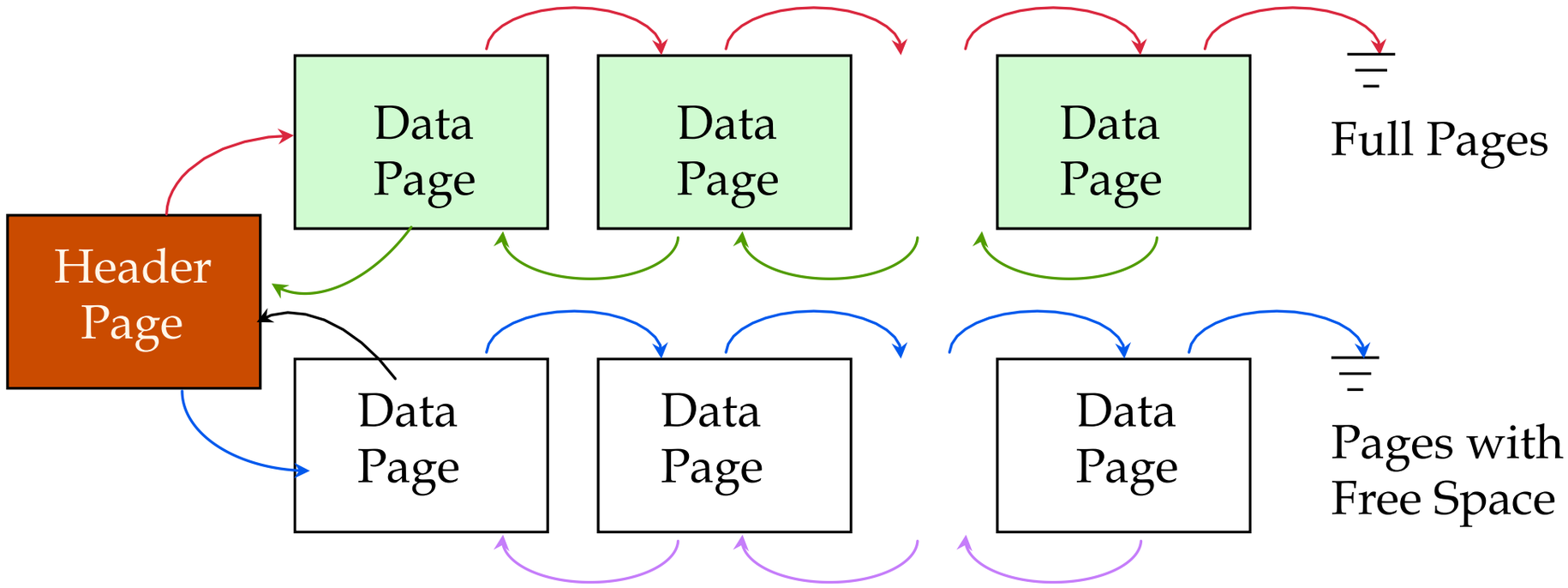
field <*op*> constant

- Equality search (*op* is =)
  - Either "tree" or "hash" indexes help here.
- Range search (*op* is one of <, >, <=, >=, BETWEEN)
  - "Hash" indexes don't work for these.

More exotic searches (not covered in cosc460)

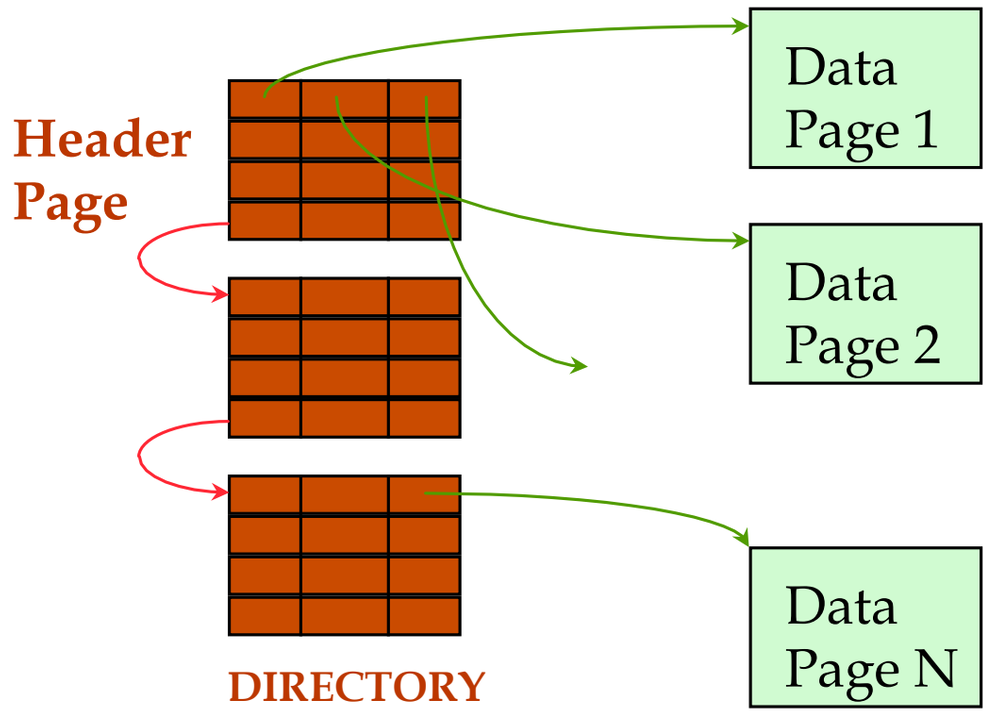
- multi-dimensional ranges ("east of Ithaca and west of Albany and North of NYC and South of Utica")
- multi-dimensional distances ("restaurants near me")
- Ranking queries ("closest gas station")
- Regular expression matches, genome string matches, etc.
- Keyword/Web search - includes "importance" of words in documents, link structure, …

Heap File Implemented as List



- Page id of header page stored in System Catalog
- Page format: requires space for 2 “pointers” (page ids)

Heap File Using Page Directory



We could also use this page organization to maintain a sorted file

- Page id of *first* directory page stored in System Catalog
- Directory page format: directory entries <page id, # free bytes>, plus “pointer” (page id) for next directory page

pollev.com/cosc460

Correct answer: B.

Poll: sorted file range search

Suppose you implement a Sorted File. The file has B pages. When answering a range search (e.g., 18 <= age <= 21), what is the minimum possible cost and maximum possible cost? (Let's ignore cost of reading directory pages.)

A. Min: 1 page; Max: $\log_2$ B pages

B. Min: 1 page; Max: B pages

C. Min: $\log_2$ B pages; Max: $\log_2$ B pages

D. Min: $\log_2$ B pages; Max: B pages

Instructions: *I will give you 1-2 minutes to think on your own.*
Vote 1.
Then you will discuss w/ neighbor (1 min).
Vote 2.
Then we'll discuss as class.

pollev.com/cosc460

Correct answer: D.

Poll: sorted file with list implementation

Would binary search to work effectively if pages were organized using the linked list approach (provided that we don't separate free pages from full ones)?

A. Yes, basically same

B. No, linked list has less overhead

C. No, much faster with linked list approach

D. No, much slower with linked list approach

Instructions: *I will give you 1-2 minutes to think on your own.*
Vote 1.
Then you will discuss w/ neighbor (1 min).
Vote 2.
Then we'll discuss as class.