COSC 460 Lecture 8: Indexing 3: B+Trees and Hash-based indexes

Professor Michael Hay
Fall 2018

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*

B+Tree example shown on board

pollev.com/cosc460

Correct answer: C.

B+trees: deletion

When a deletion causes a node to be under capacity, we looked at two possible actions: merge and redistribute. Which of the following statements is true? Choose the best answer.

A. If you cannot redistribute, you can merge.

B. If you cannot merge, you can redistribute.

C. If you cannot do one, you can do the other.

D. You can always do both.

E. You can do one if and only if you cannot do the other.

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.

Data entry alternatives

Data entry alternatives:

1. (actual record with search key k)

Data stored in index! This can only be done once!

2. (search key k, record id)

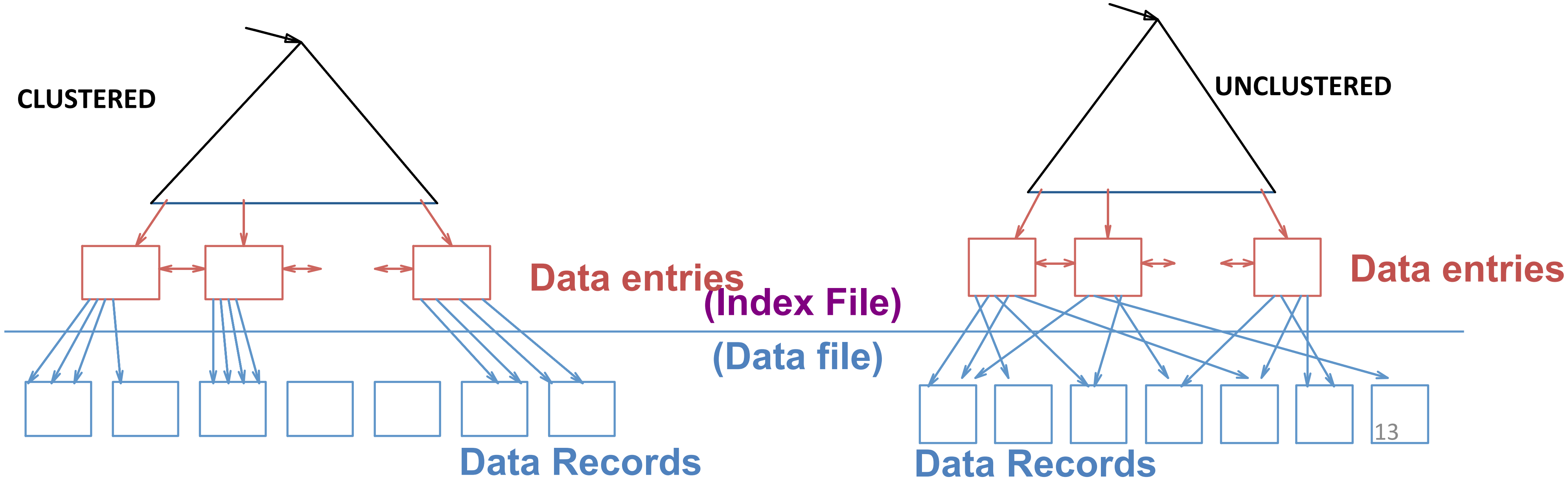
Default approach. Can use record id to find page where record is stored.

3. (search key k, list of record ids)

Data entries are now *variable* length records.

Clustered vs. unclustered

In a clustered index, data ***records*** arranged in ***roughly*** the same order as the data ***entries*** of the index.



Why **"roughly"**? Recent inserts may be out-of-order. Could have batch job that periodically reorders of data records.

Relationship between data entry type and whether index is clustered or unclustered?

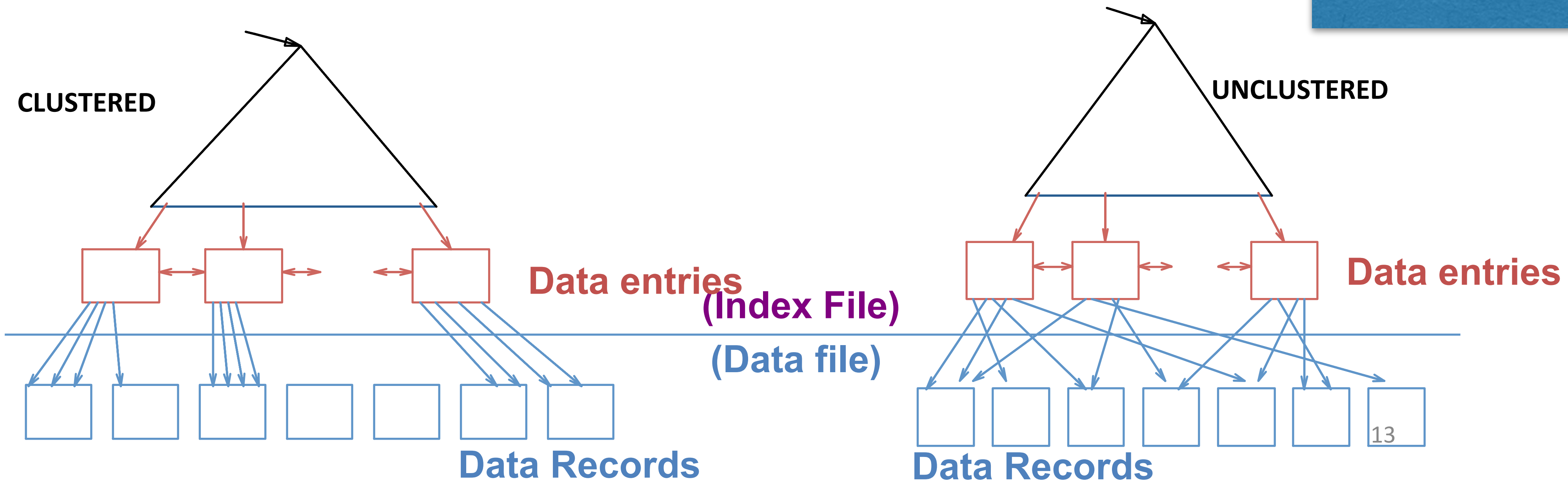
Composite search keys

- Search key may contain multiple attributes. **Example: (major, gpa)**
- Typically used with a tree-like index that *sorts* records
  - B+Tree would order by first attribute, followed by second in case of ties
- Which of these searches supported:
  - Major = ‘CS’?
  - Major = ‘English’ and GPA < 2.3?
  - GPA = 3.99?

Not supported: data sorted first by major. Records with GPA > 3.99 may be spread throughout index.

pollev.com/cosc460

Correct answer: E.



Compare *clustered* vs. *unclustered* indexes. For which operation would a clustered index have the largest performance advantage over an unclustered index?

A. Scan

B. Insert

C. Delete

D. Equality search on a key (sid = 123)

E. Range search (age > 18)

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.