

# CSE 444: Database Internals

## Section 2: Indexing

# Plan for the Sections

- We will go through examples together
- Should be a good practice for the homework problems
- Ideas, suggestions, comments, feedback are always welcome
  - write your thoughts on discussion board

Indexes: Useful for search query/range  
query/joins

Revisit Tweet Example:

Tweets(tid, user, time, content)

# Tweet Relation in a Sequential File

tid	user	time	content
10	1	05:03:00	"....."
20	2	12:05:07	"....."
30	2	18:12:00	"....."
40	3	00:16:13	"....."
50	4	10:10:13	"....."
60	1	04:09:07	"....."
70	2	12:08:34	"....."
80	4	11:08:09	"....."

— 1 record

— 1 page

- File is sorted on “tid”

# (Lec 5) Index Classification

- Primary/secondary
- Dense/sparse
- Clustered/unclustered
- Question: Draw a secondary dense index on “user”

# Ex1. Secondary Dense Index

tid	user	time	content
-----	------	------	---------

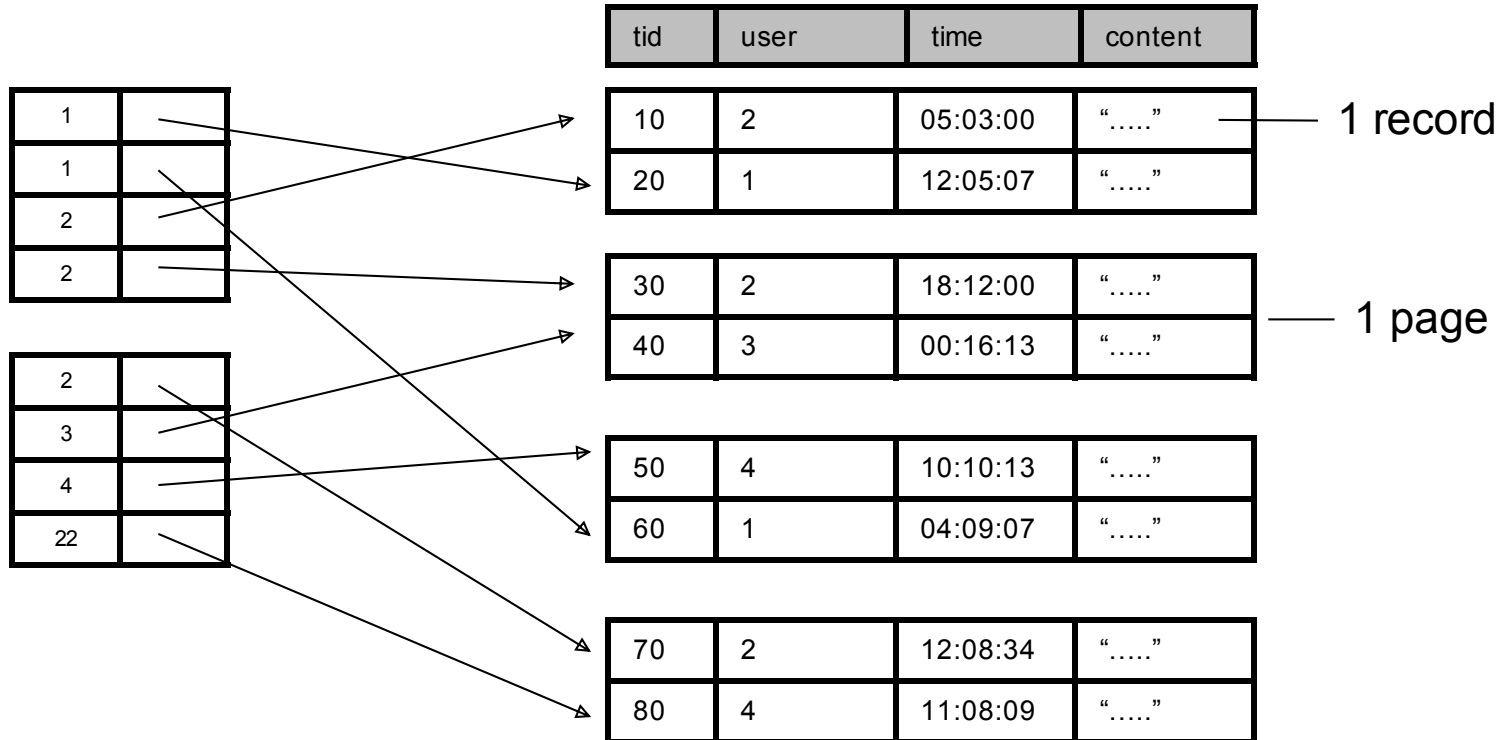
10	2	05:03:00	"...."	— 1 record
20	1	12:05:07	"...."	

30	2	18:12:00	"...."	— 1 page
40	3	00:16:13	"...."	

50	4	10:10:13	"...."
60	1	04:09:07	"...."

70	2	12:08:34	"...."
80	4	11:08:09	"...."

# Ex1. Secondary Dense Index (user)



- **Dense:** an “index key” (not database key) for every database record
- **Secondary:** cannot reorder data, does not determine data location
- Also, **Unclustered:** records close in index may be far in data

- Question: Draw a primary dense index on “user”



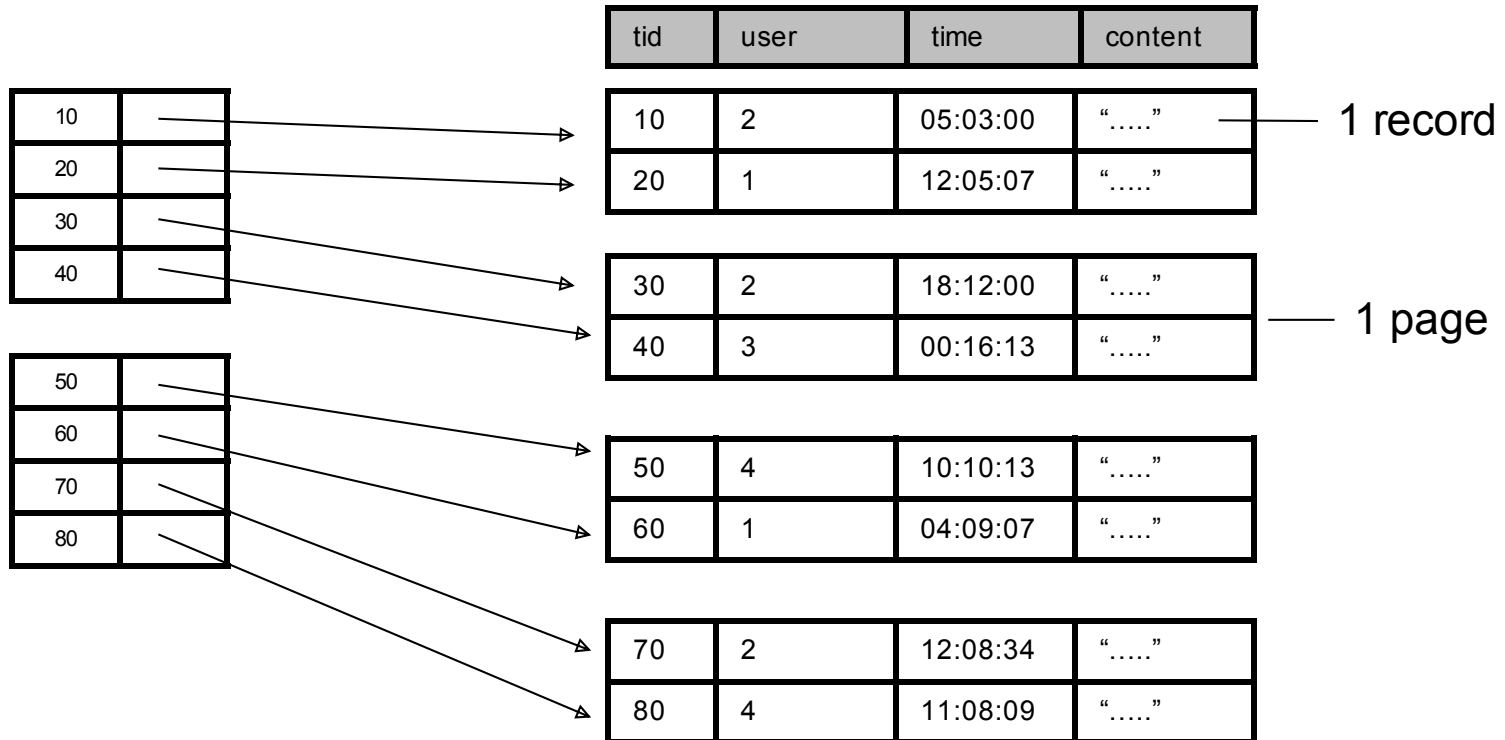
# Ex2. Primary Dense Index (tid)

tid	user	time	content
10	1	05:03:00	"....."
20	2	12:05:07	"....."
30	2	18:12:00	"....."
40	3	00:16:13	"....."
50	4	10:10:13	"....."
60	1	04:09:07	"....."
70	2	12:08:34	"....."
80	4	11:08:09	"....."

— 1 record

— 1 page

# Ex2. Primary Dense Index (tid)



- **Dense:** an “index key” for every database record
  - (In this case) every “database key” appears as an “index key”
- **Primary:** determines the location of indexed records
- Also, **Clustered:** records close in index are close in data

Primary Clustered Index

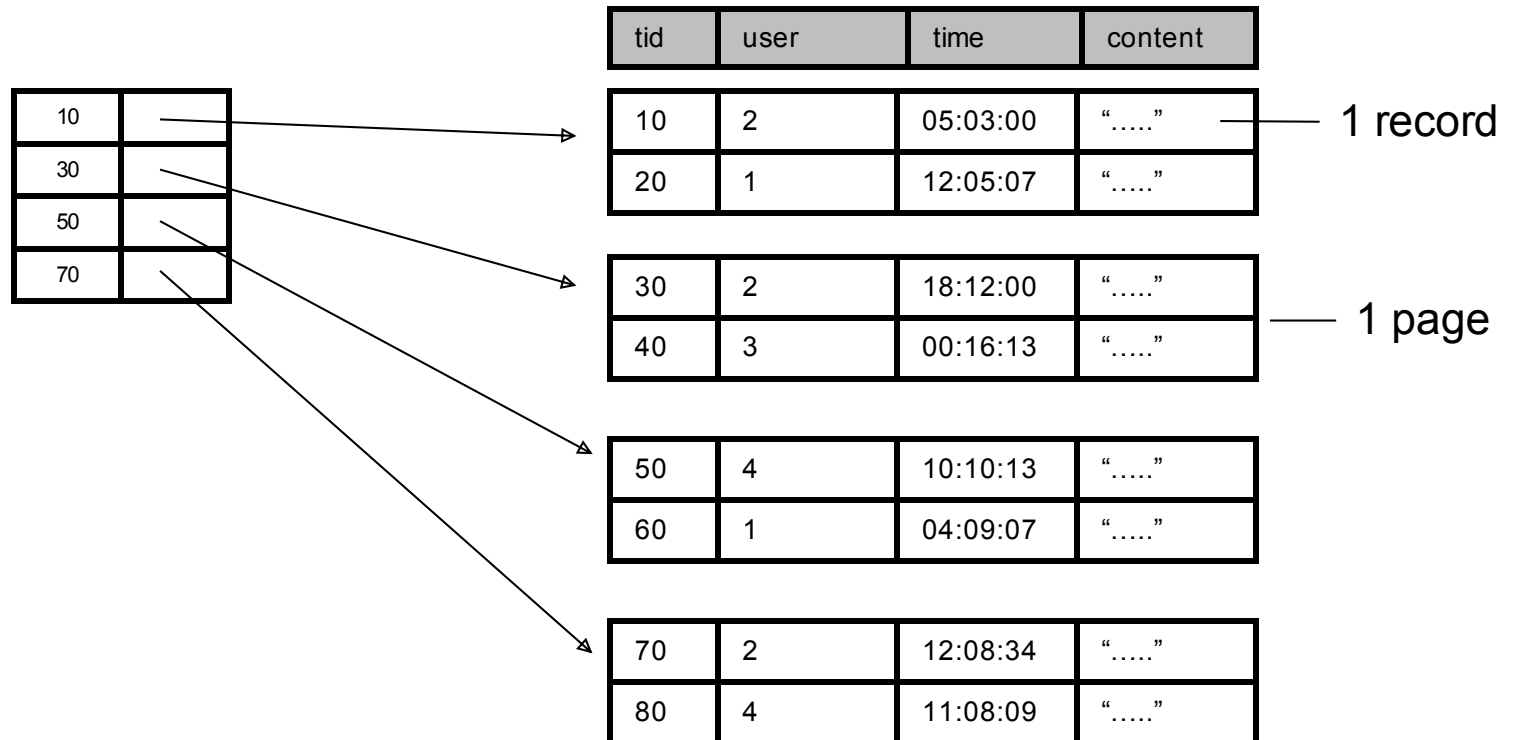
Vs.

Secondary Unclustered Index?

Clustered Index can be made Sparse  
(normally one key per page)

- Question: Draw a primary sparse index on “user”

# Ex3. Primary Sparse Index (tid)

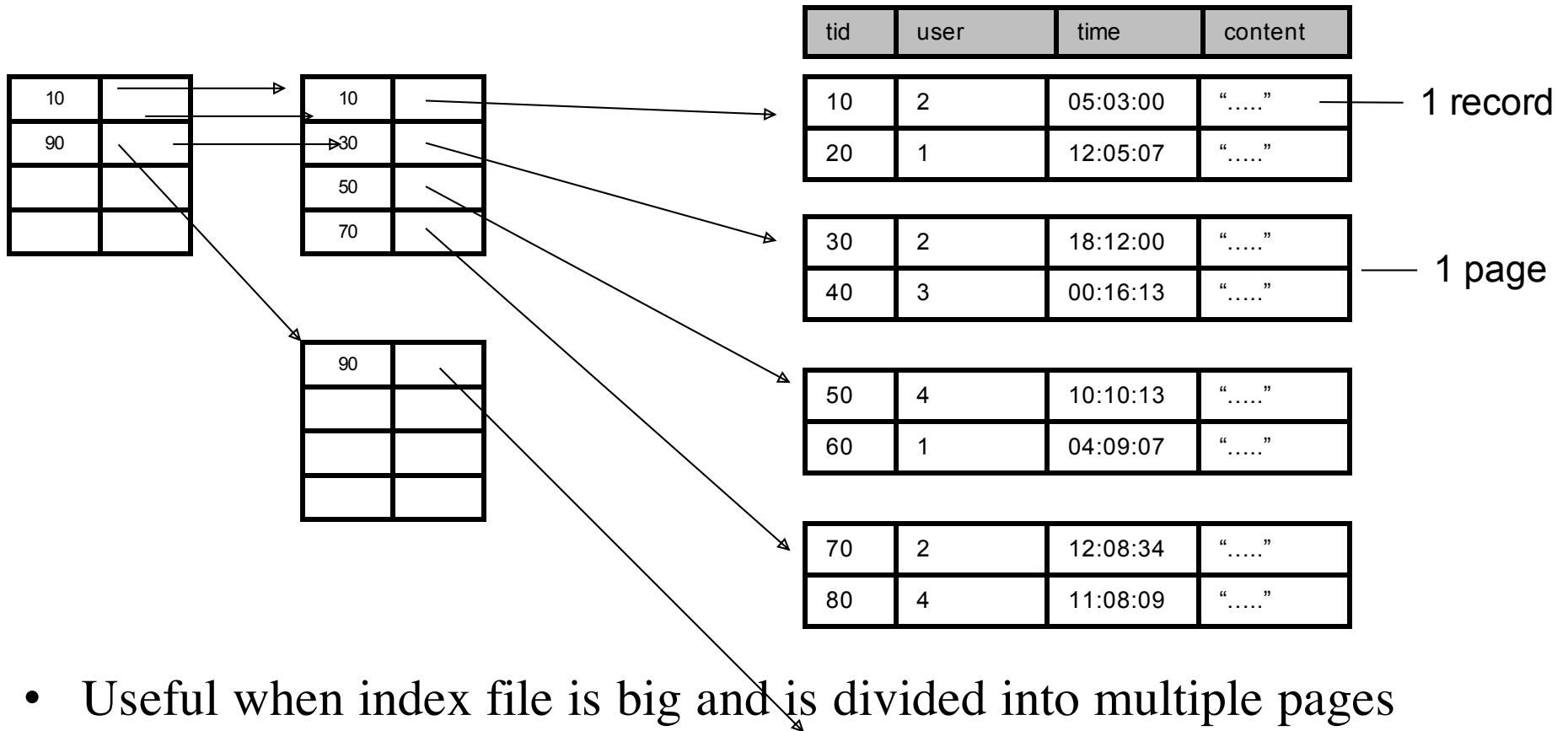


- Only one index file page instead of two

# Discussion

- **Primary/secondary**
  - Primary: common in queries, efficiency (one tuple/key)
  - Secondary: more useful when “almost a key”
- **Clustered/unclustered**
  - Clustered:
    - fewer data page read, can have sparse index
    - expensive to maintain, at most one per file
- **Dense/sparse**
  - Sparse: smaller, only for clustered index, at most one per file
  - Dense: multiple dense indexes, useful in some optimization (inverted data file)
- **How to decide which indexes to create**
  - Overhead (read/write index page, updates)
  - Depends on workload (Example in sec 8.4)

# Index to index



- Useful when index file is big and is divided into multiple pages
- Efficient and standard implementation: B+ trees
  - balanced, good for both range and search query

- Tomorrow – Lec 6:
  - B+ Trees
  - Hash-based Index
    - Not good for range query