

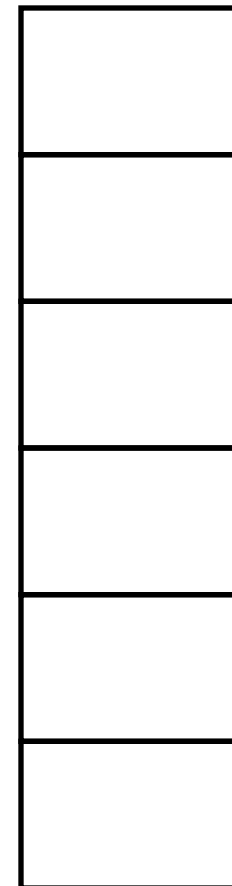
CS143: Files

Professor Junghoo “John” Cho

Files: Main Problem

- How to store tables into disks?
- Q: 512Byte block. 80Byte tuple. How to store?

Jane	CS	3.7
Susan	ME	1.8
June	EE	2.6
Tony	CS	3.1

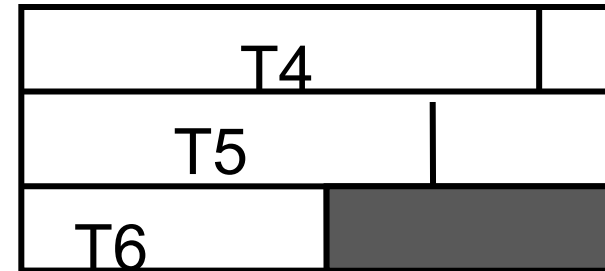
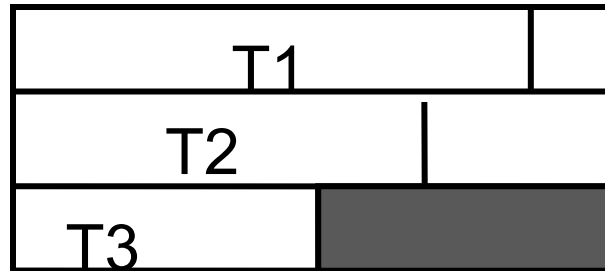


$$\frac{512}{80} = 6.4$$

Spanned vs Unspanned

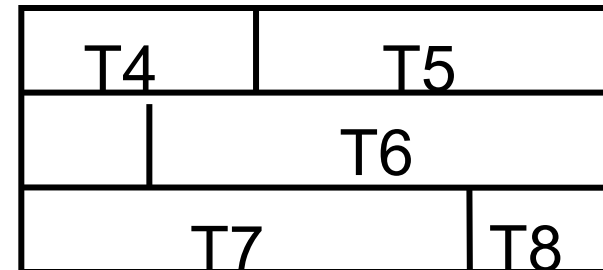
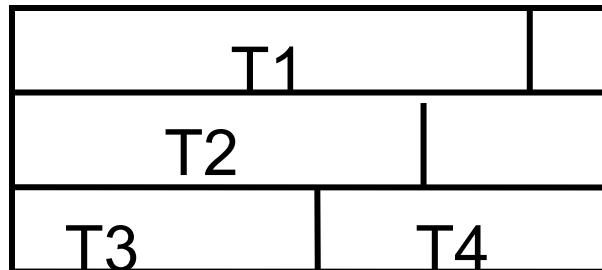
- Unspanned

6 full tuple

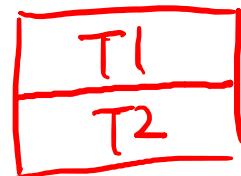


- Spanned

32 bytes



- Q: Maximum space waste for unspanned?

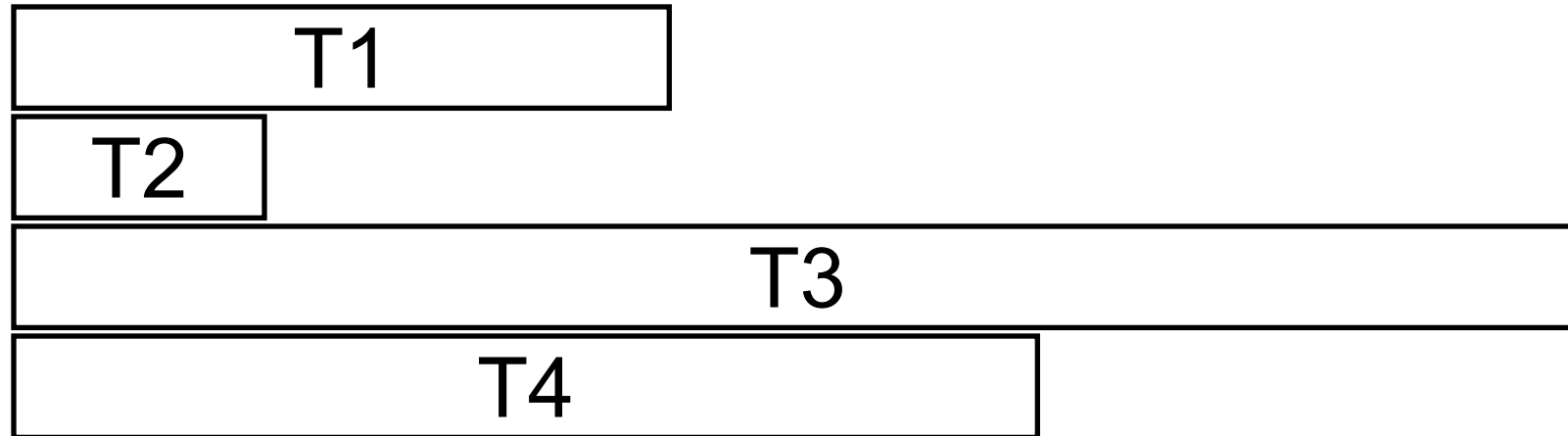


50% = 1 Byte

Variable-Length Tuples

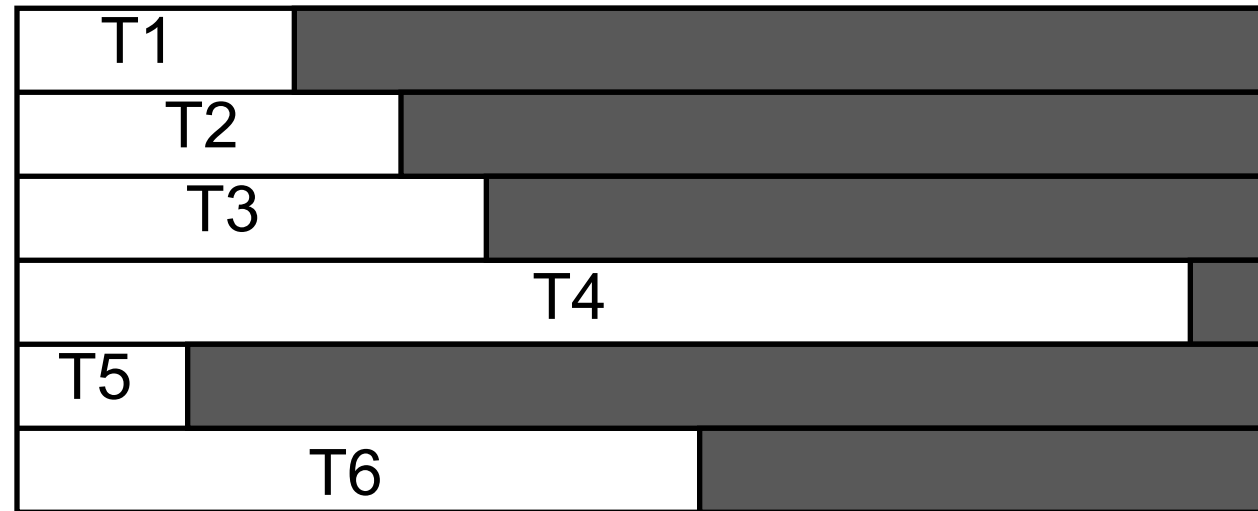
- How do we store them?

int
real
char(16)
varchar(100)



Reserved Space

- Reserve the maximum space for each tuple



A varchar(100)

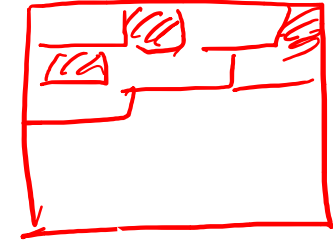
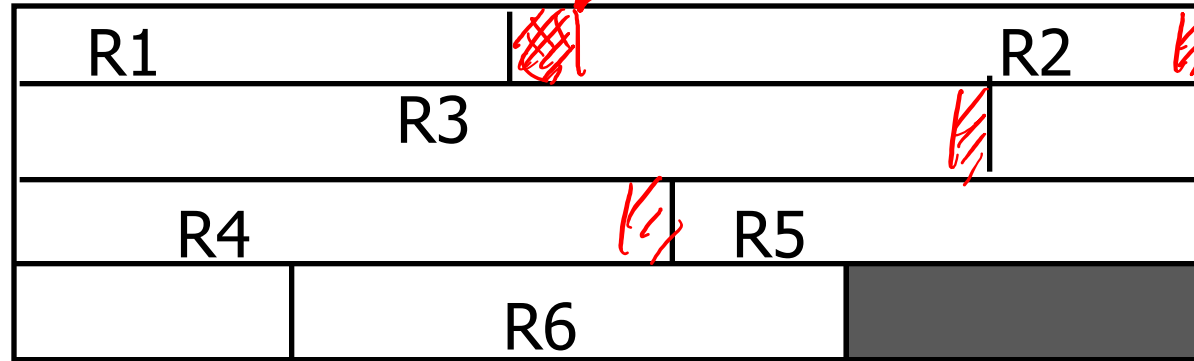
('a')

('b')

('c')

- Q: Any problem?

Variable-Length Space



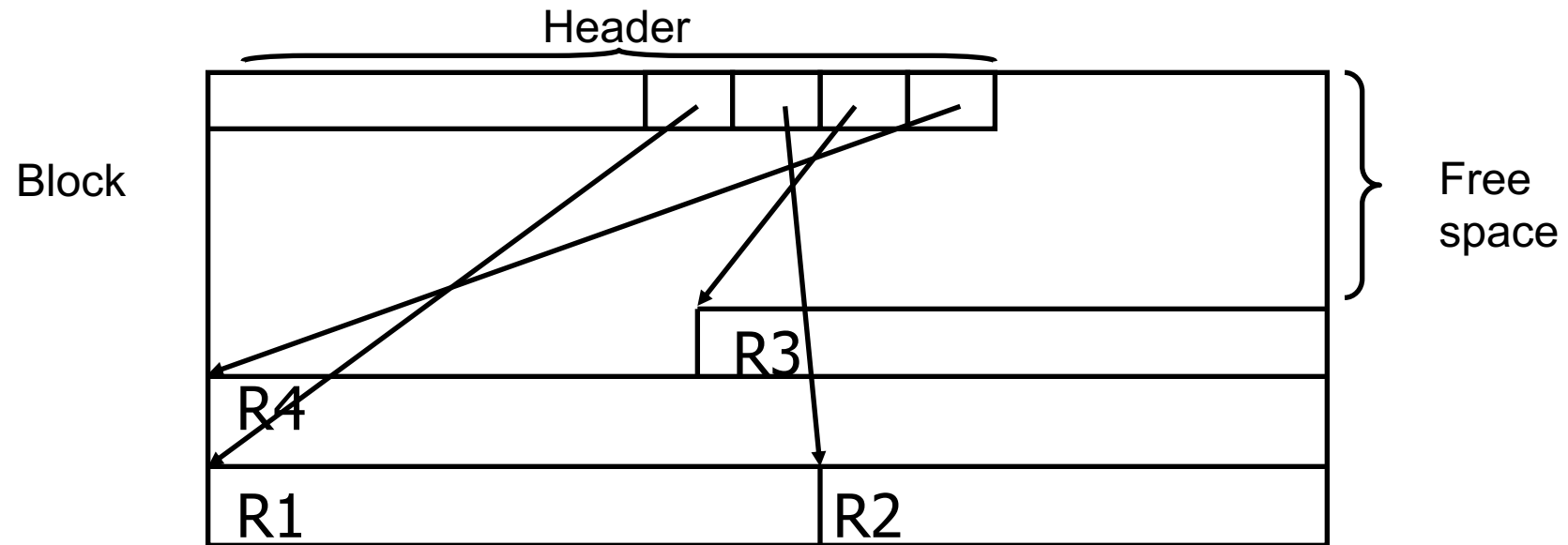
- Pack tuples tightly
- Q: How do we know the end of a tuple?
- Q: What to do for delete/update?
- Q: How can we “point to” to a tuple?

1) end of record marker
2) store the length of the tuple

ca' → 'aaa'

(block#, beginning location of tuple)

Slotted Page



Q: How can we point to a tuple?

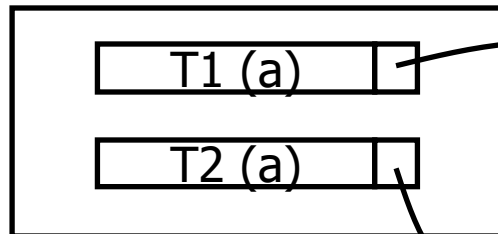
Long Tuples

- ProductReview(
 pid INT,
 reviewer VARCHAR(50),
 date DATE,
 rating INT,
 comments VARCHAR(4000))
- Block size 512B
- How should we store it?

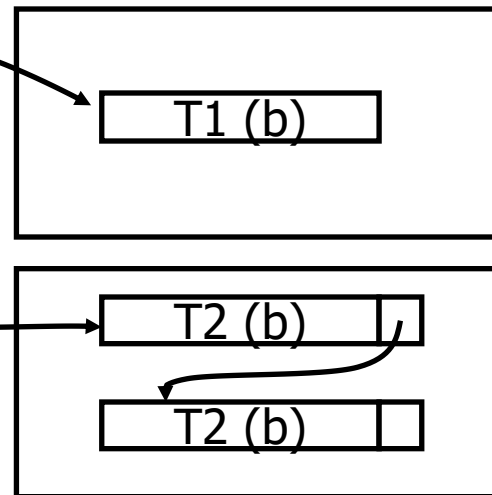
Long Tuples

- Splitting tuples
 - Long attributes are stored separately (often as a separate file)

Block with short attributes.



Block with long attrs.



This block may also have fixed-length slots.

Column-Oriented Storage

- `SELECT name FROM Students WHERE GPA > 3.7`
- For analytical queries, reading the entire row of a tuple may not be needed
 - Row-oriented storage forces us to read the entire row even if most columns are not needed for query processing

Elaine	1 Le Conte	3.7
James	3 Mississippi	2.8
John	12 Wilshire	1.8
Peter	4 Olympic	3.9
Susan	7 Pico	1.0
Tony	12 Sunset	2.4

Column-Oriented Storage

- Store by column, not by row
- Unneeded Columns can be skipped for query processing
 - Better compression and caching behavior
- But
 - Column values of matching rows must be “joined”
 - Insertion/update of a row is more expensive (multiple IOs per row)

Elaine
James
John
Peter
Susan
Tony

1 Le Conte
3 Mississippi
12 Wilshire
4 Olympic
7 Pico
12 Sunset

3.7
2.8
1.8
3.9
1.0
2.4

Sequential File

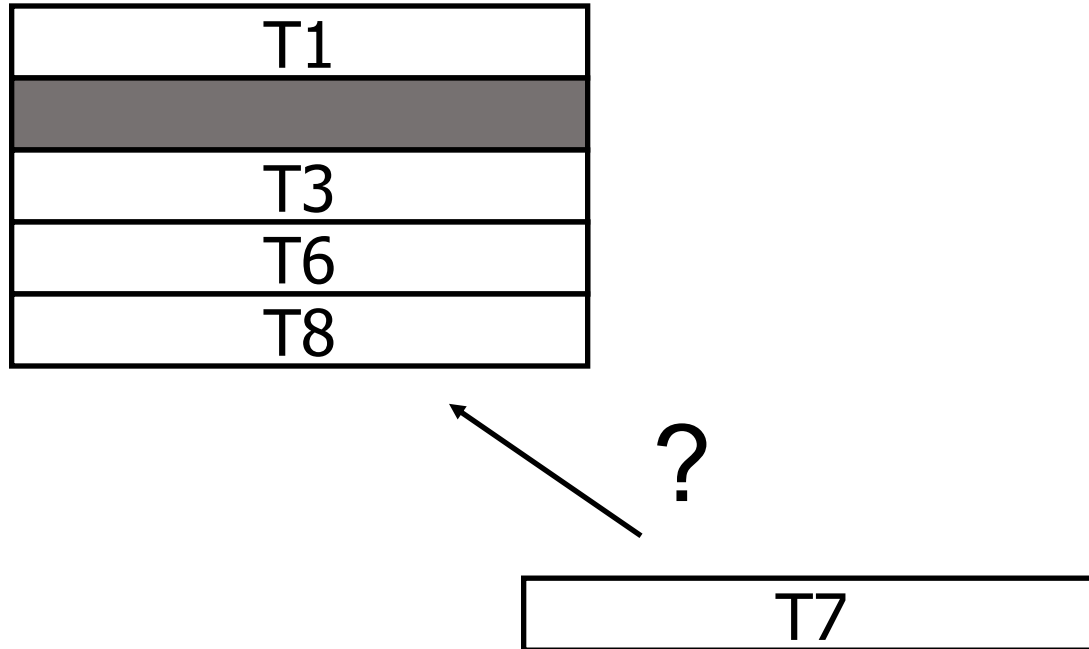
- Tuples are ordered by certain attribute(s) (search key)

Elaine	1 Le Conte	3.7
James	3 Mississippi	2.8
John	12 Wilshire	1.8
Peter	4 Olympic	3.9
Susan	7 Pico	1.0
Tony	12 Sunset	2.4

Search key: Name

Sequencing Tuples

- Inserting a new tuple
 - Easy case

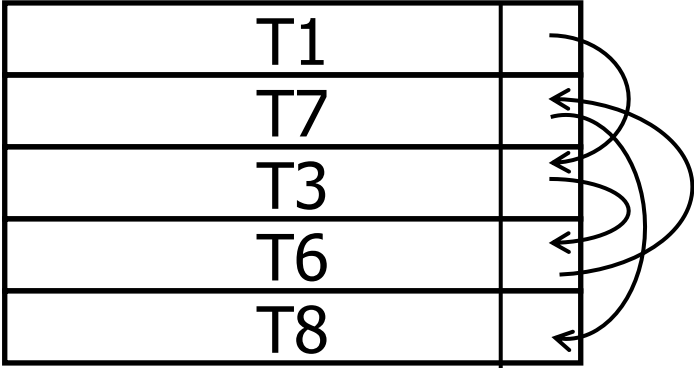


Two Options

1) Rearrange

T1
T3
T6
T7
T8

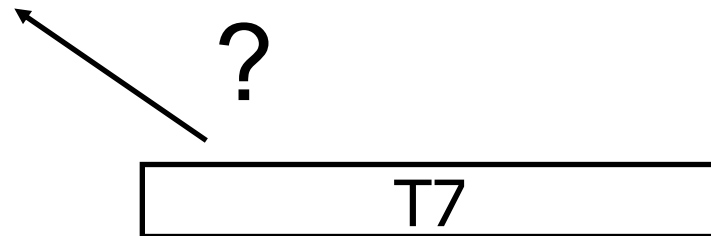
2) Linked list



Sequencing Tuples

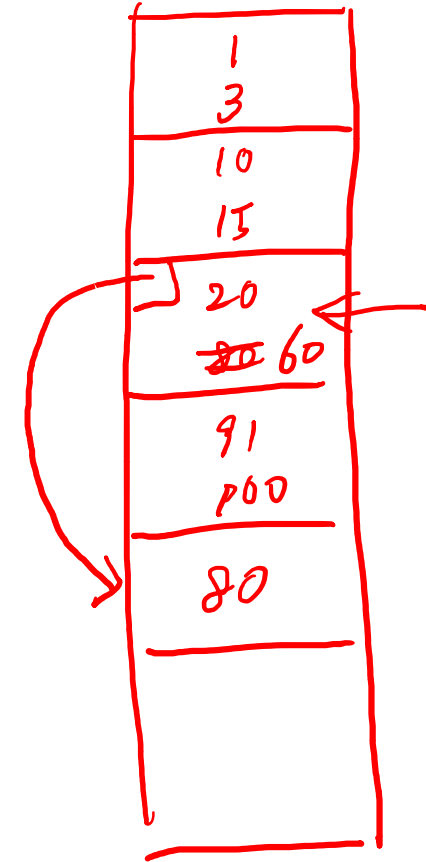
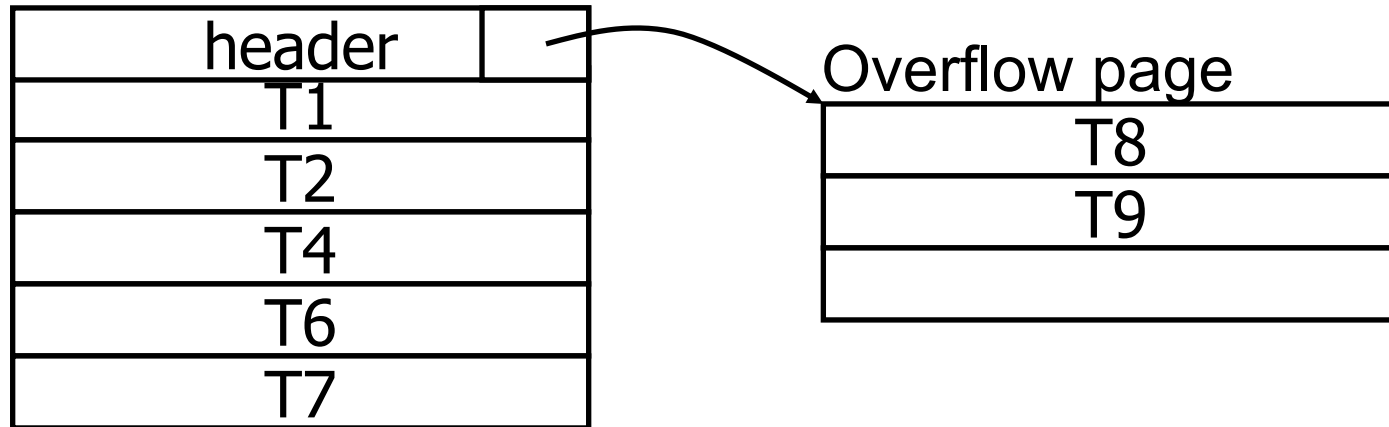
- Inserting a new tuple
 - Difficult case

T1
T4
T5
T8
T9



Sequencing Tuples

- Overflow page



- Reserving free space to avoid overflow
 - PCTFREE in DBMS
- ```
CREATE TABLE R(a int) PCTFREE 40
```



# Things to Remember

- Spanned/unspanned tuples
- Variable-length tuples (slotted page)
- Long tuples
- Row-oriented vs column-oriented storage
- Sequential file and search key
  - Problems with insertion (overflow page)
  - PCTFREE