Data Storage Structures 1

Instructor: Beom Heyn Kim

beomheynkim@hanyang.ac.kr

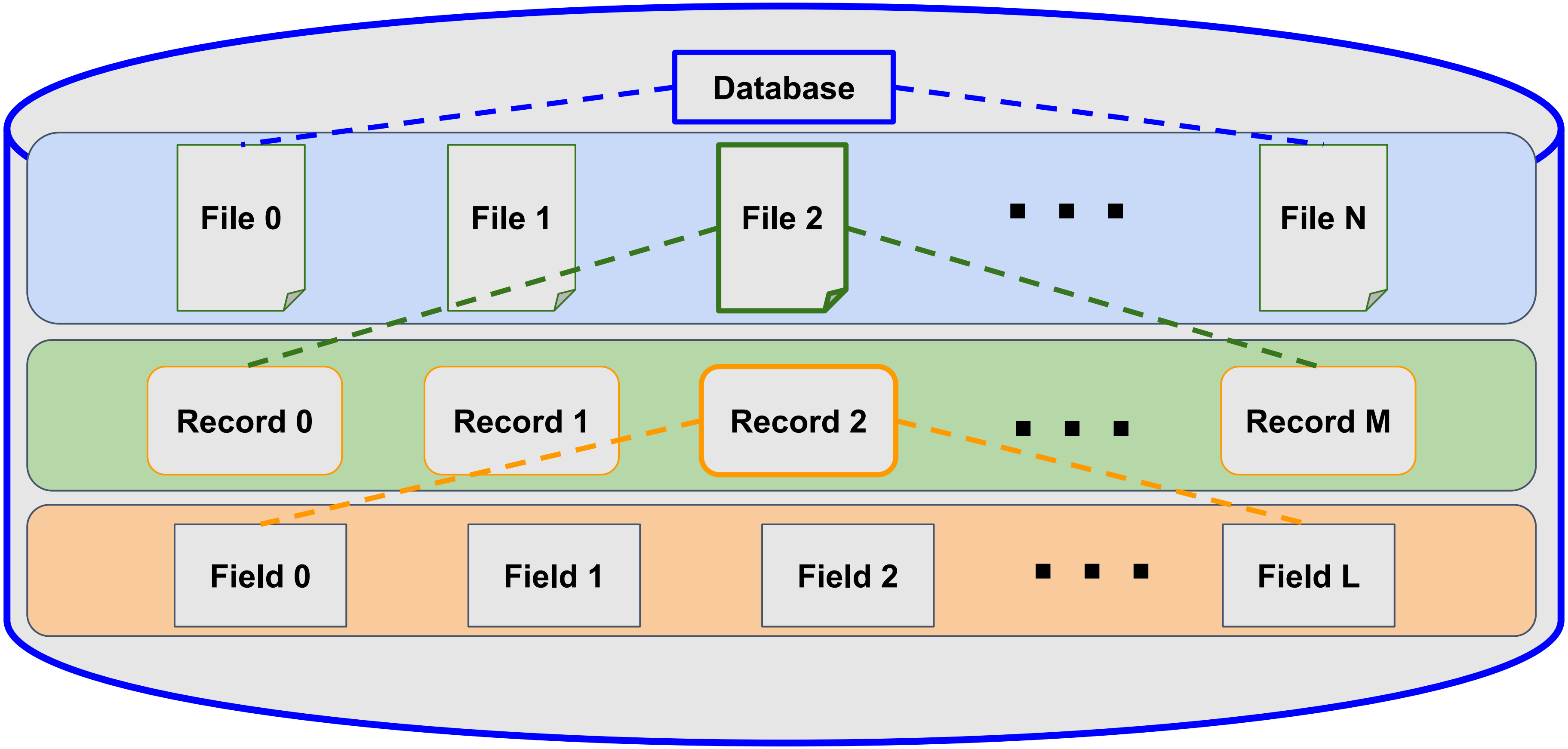
Department of Computer Science

Overview

- File Organization
- Assignments

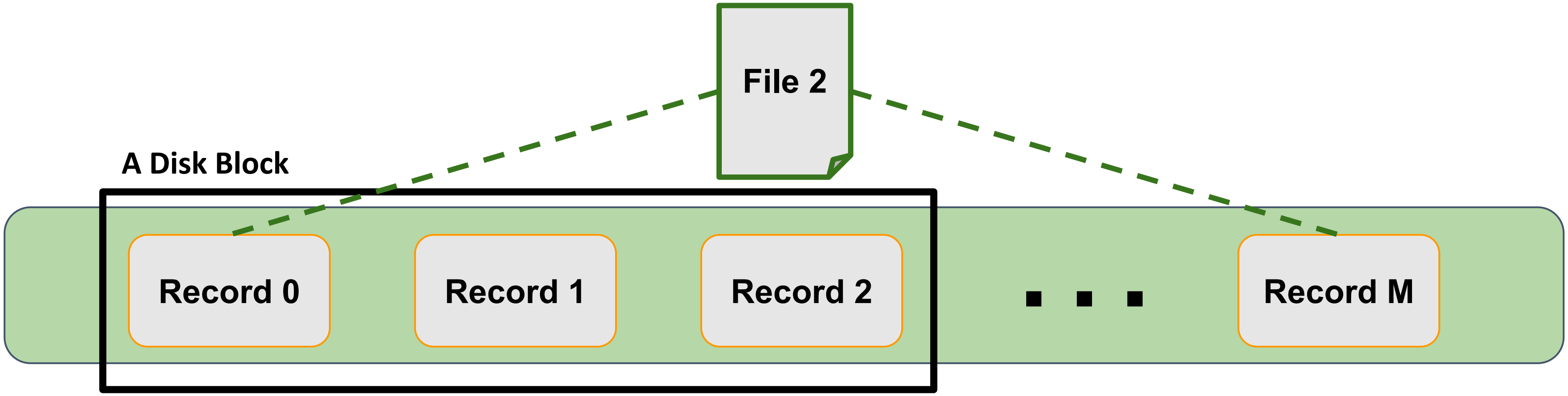
File Organization

- The database is stored as a collection of *files*. Each file is a sequence of *records.* A record is a sequence of fields.



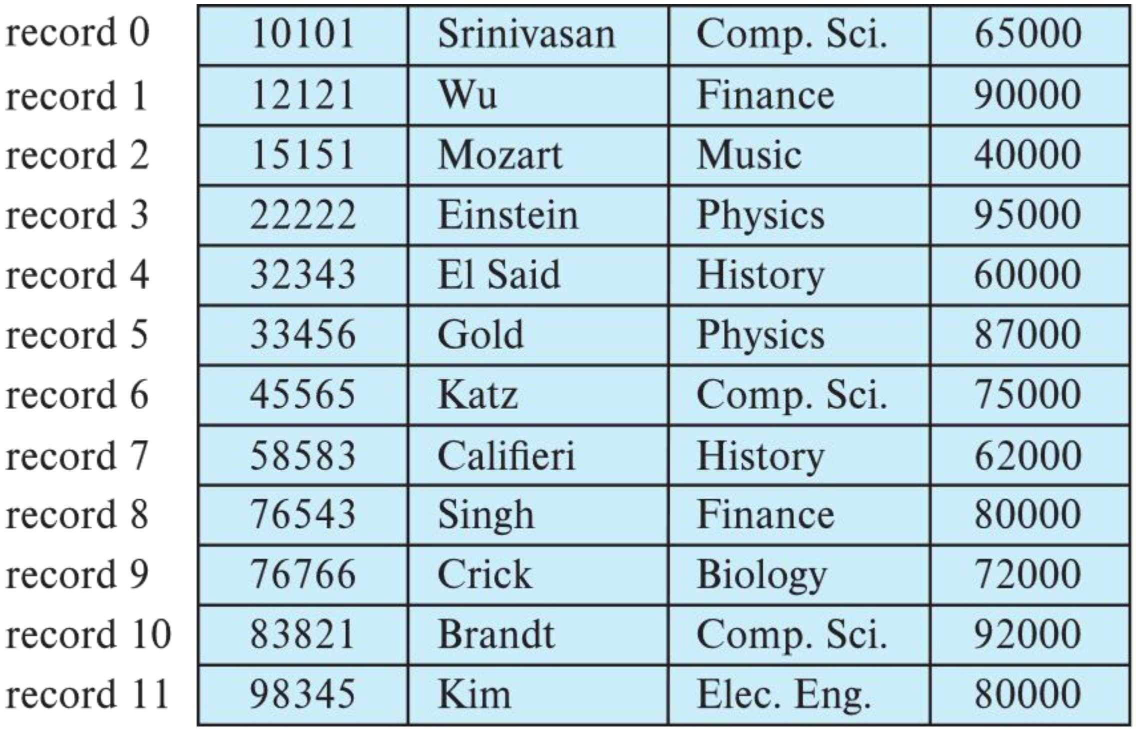
File Organization

- One approach to file organization is as follows:
  - Assume record size is fixed
  - Each file has records of one particular type only
  - Different files are used for different relations
- This case is easiest to implement
  - We will consider variable length records later
- We assume that records are smaller than a disk block



Fixed-Length Records

- How can we store records in a file?
  - Simple approach: Store record *i* starting from byte *n* * *i*, where *n* is the size of each record.



record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Byte Offset 0

Byte Offset n

Byte Offset n*2

Byte Offset n*3

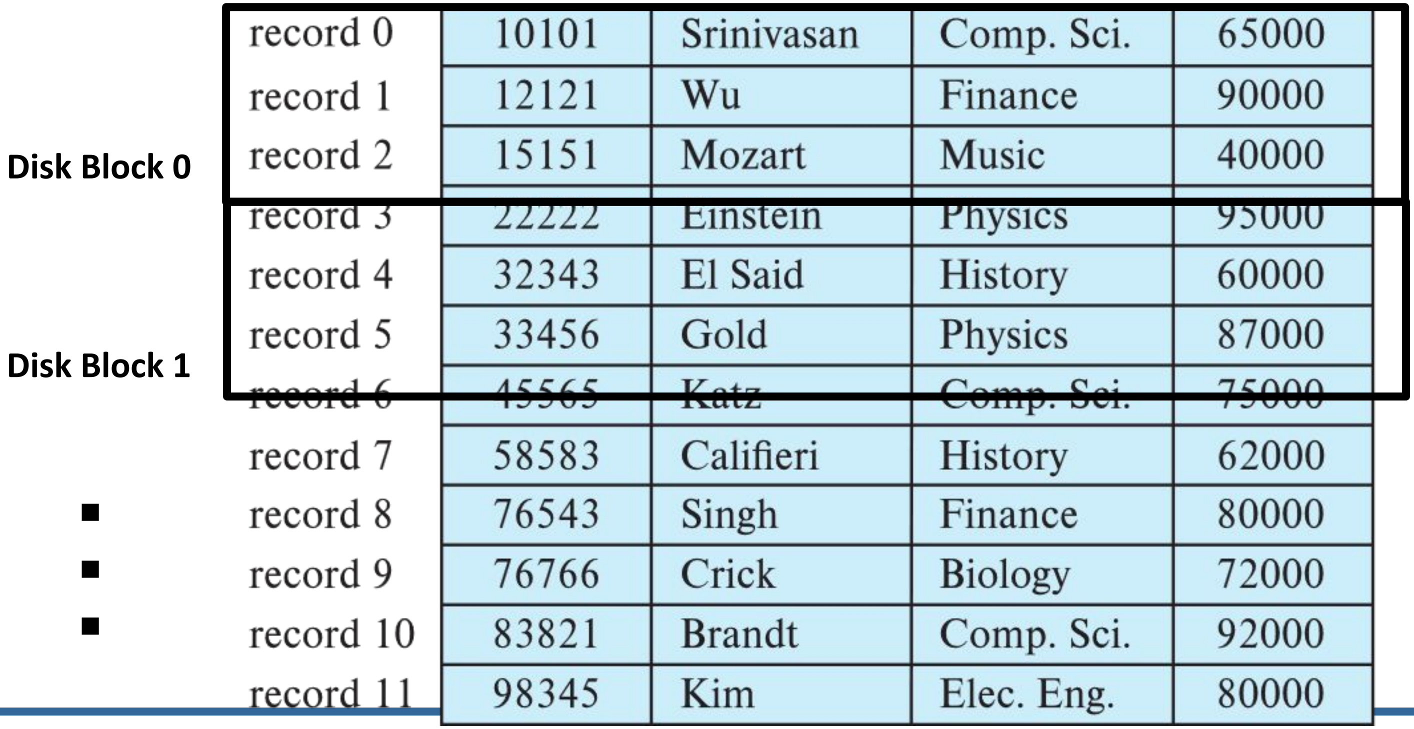
■

■

■

Fixed-Length Records

- One problem is crossing block boundaries
  - Record access is simple but records may cross blocks
    - E.g., suppose each disk block size is 512 bytes and each record's size is 170 bytes
  - Reading a record may require to read 2 disk blocks!
    - E.g., consider record 3 or 6



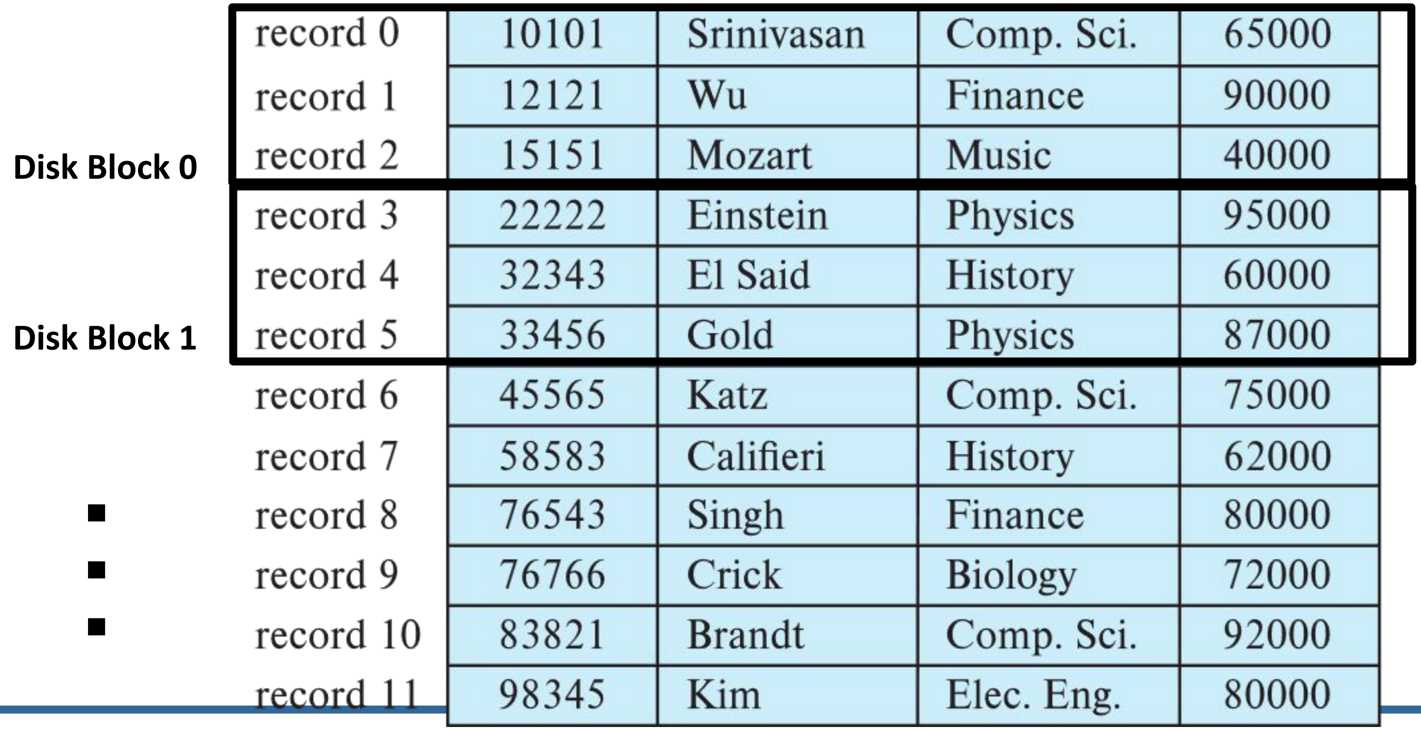
Disk Block 0

Disk Block 1

record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Fixed-Length Records

- Put as many records as each disk block can store without boundary crossing
  - Remaining bytes of each block left unused.
  - Dividing the block size by the record size will and discarding the fractional part will give us how many records can be stored in each block.

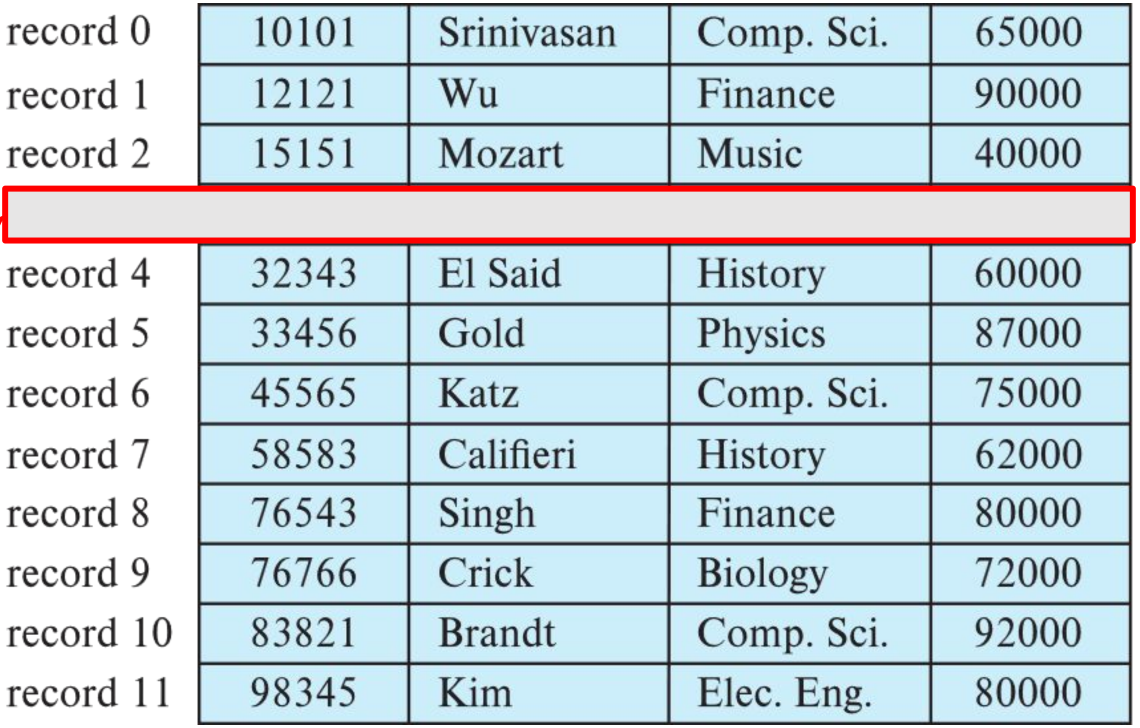


record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

■
■
■

Fixed-Length Records

- Another problem of the simple approach on page 5 is that deleting and reusing spaces are difficult.
  - After the deletion of record 3, a space is freed up



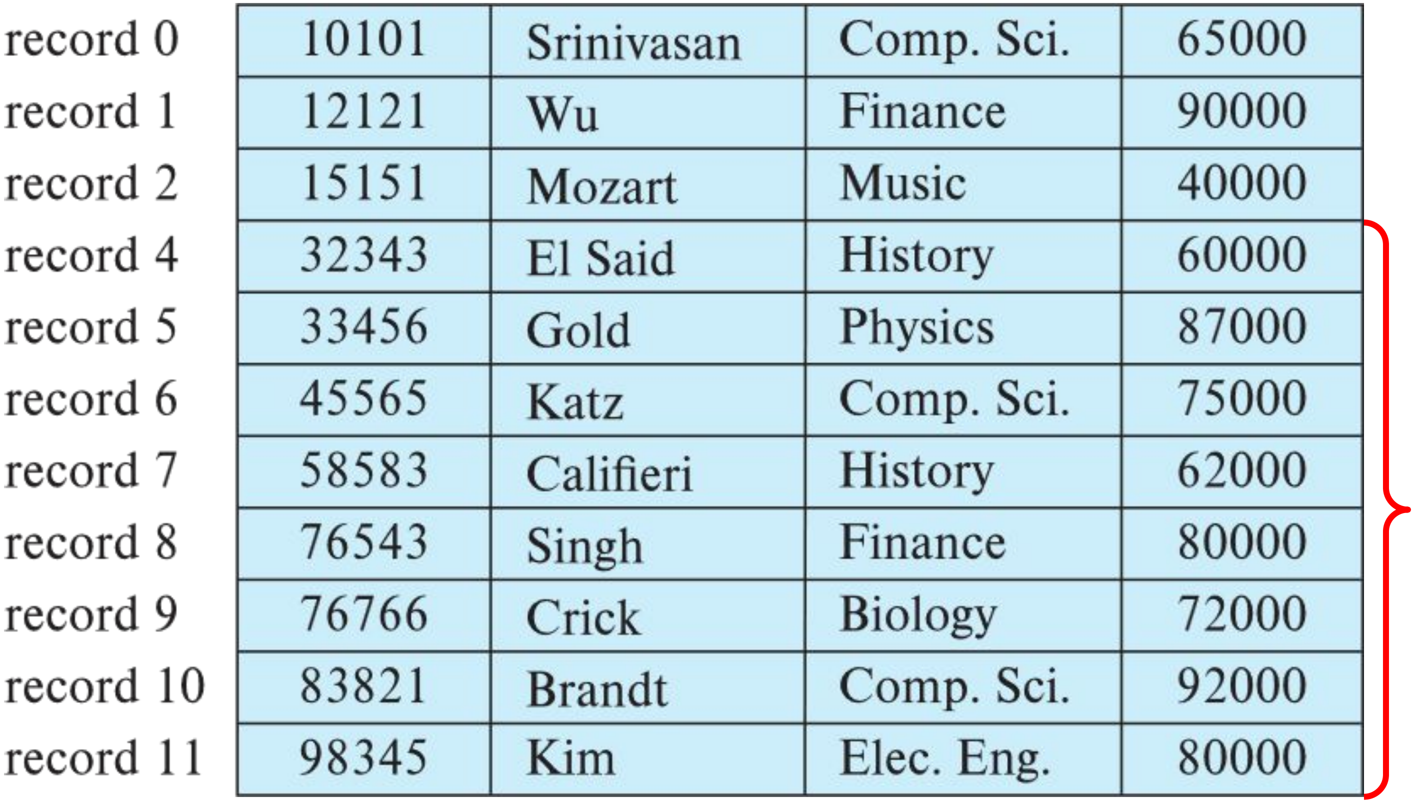
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

What to do to reuse this free space?

Fixed-Length Records

- After the deletion of record i, moving records i+1, …, n to i, …, n-1

Record 3 deleted



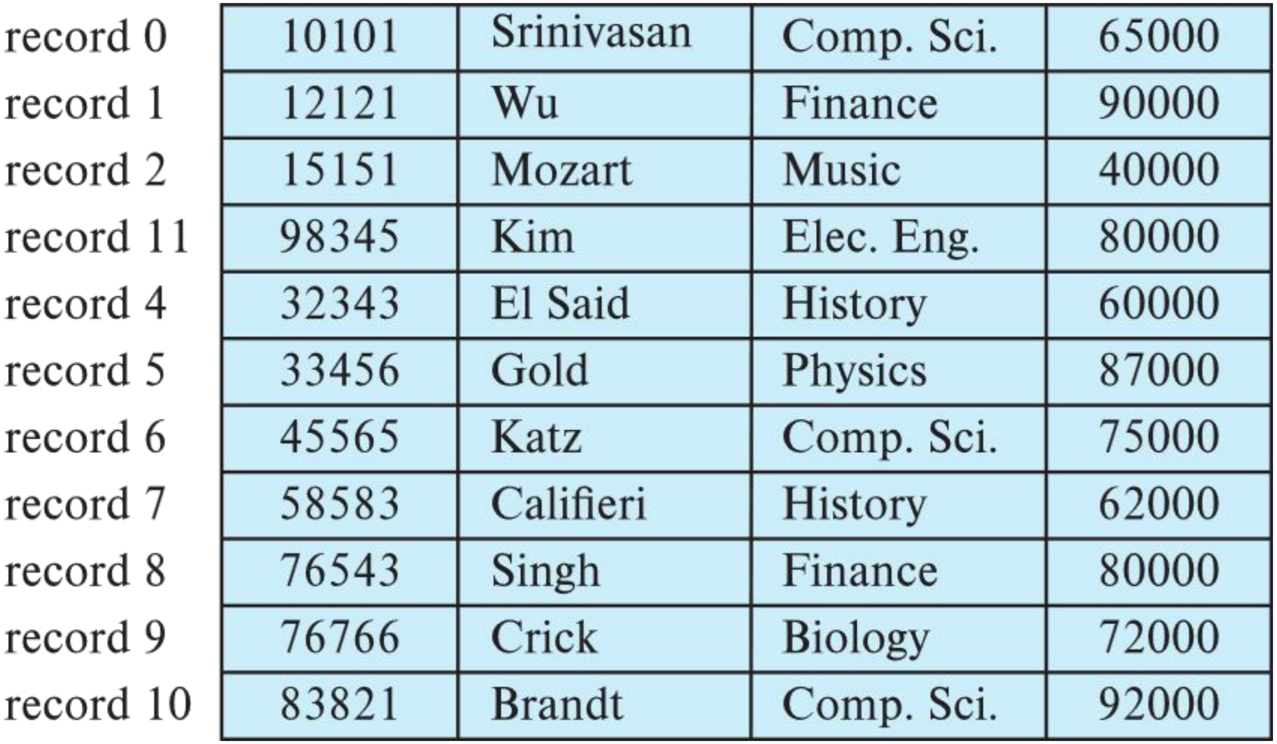
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Moving these records require expensive extra block accesses

Fixed-Length Records

- Moving record n to i

Record 3 deleted and replaced by record 11

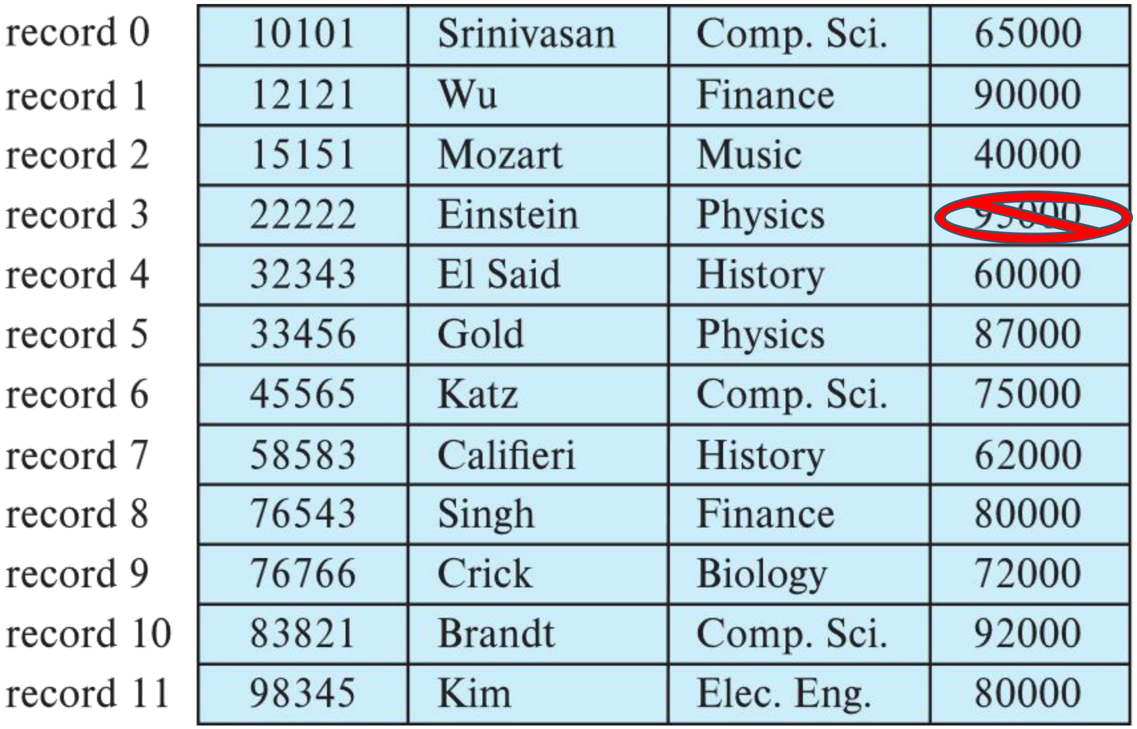


record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 11	98345	Kim	Elec. Eng.	80000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000

Moving few records still require few extra block accesses

Fixed-Length Records

- Considering insertions tend to be more frequent than deletions, leave the space occupied by deleted records but replace them with newly inserted records later
  - Marking deleted records



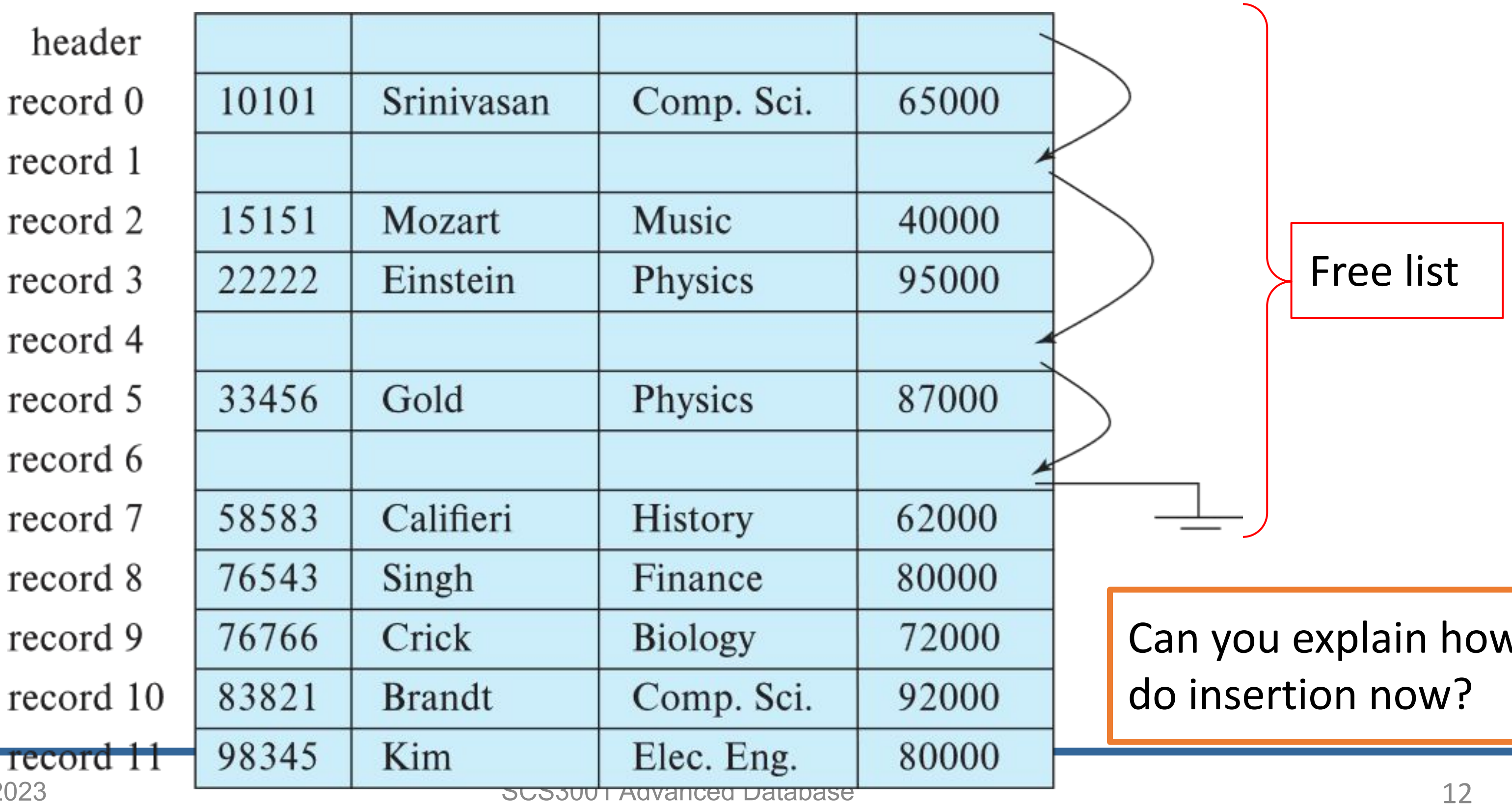
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	~~95000~~
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Markers can tell us which records are deleted, but when we try to find them to replace with new records, it is difficult to find quickly

Fixed-Length Records

- Linking all free records on a free list
    - At the beginning of the file, allocate a file header containing the address of the first of deleted records
    - Each deleted records store addresses to the next deleted records

Record 1, 4, and 6 are deleted and linked on a free list



header				
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1				
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4				
record 5	33456	Gold	Physics	87000
record 6				
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

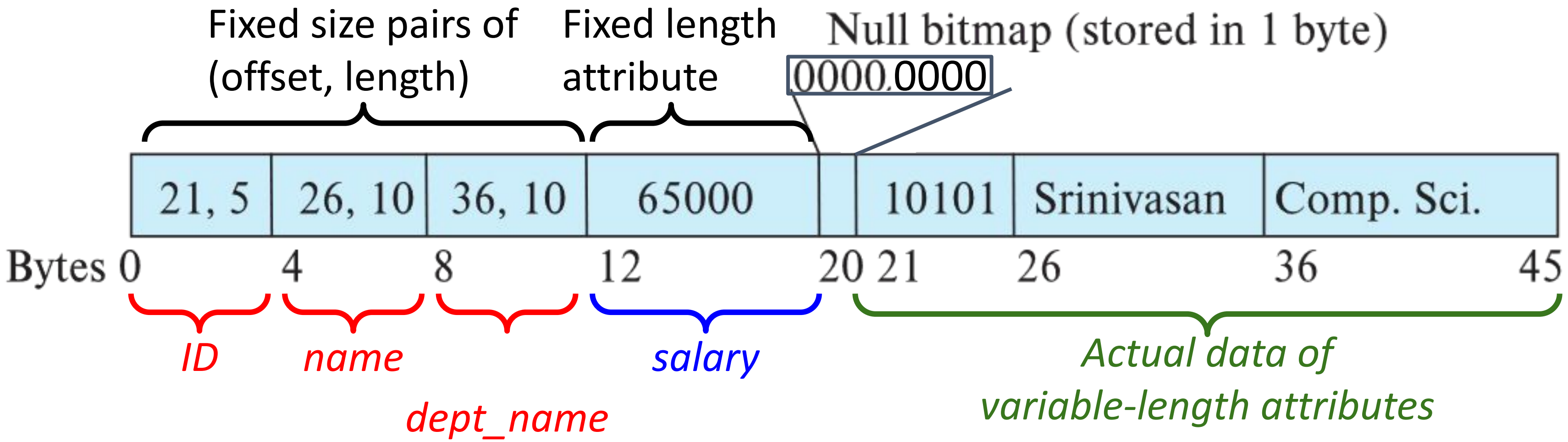
Can you explain how to do insertion now?

Variable-Length Records

- Variable-length records arise in database systems in several ways:
  - Storage of multiple record types in a file.
  - Record types that allow variable lengths for one or more fields such as strings (**varchar**)
  - Record types that allow repeating fields (used in some older data models)
- Two problems:
  - How to represent a single record containing variable length attributes?
  - How to store variable-length records in a block?

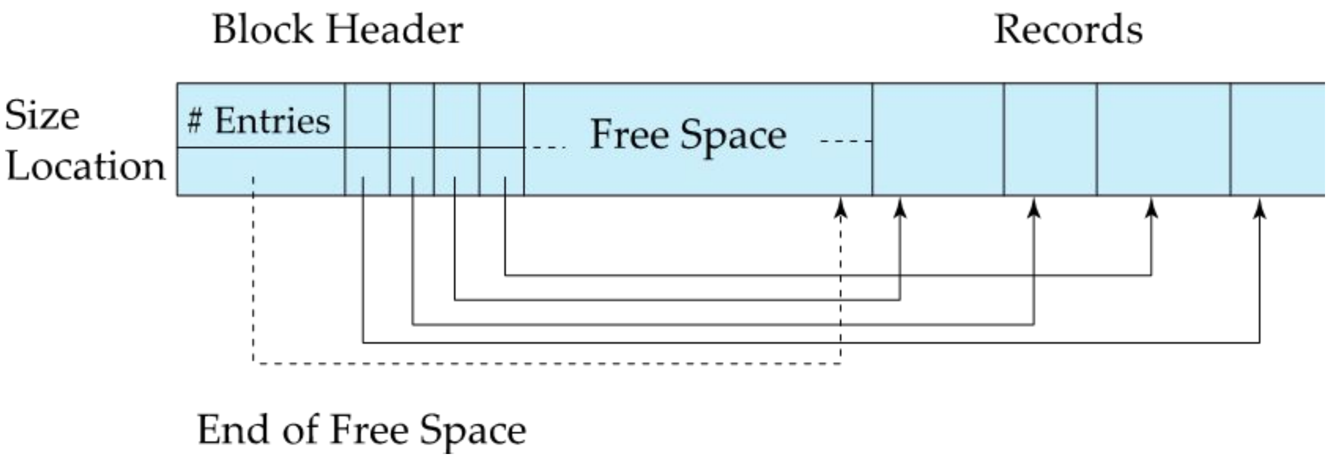
Variable-Length Records

- Representation of a variable-length record (e.g., instructor relation's record)
  - Attributes are stored in order (e.g., *ID, name, dept_name, salary*)
  - Variable length attributes (e.g., *ID, name, dept_name*) represented by fixed size (*offset*, *length*) pairs
  - actual data stored after all fixed length attributes
  - Null values represented by null-value bitmap (size depends on the number of attributes)



Variable-Length Records: Slotted Page Structure

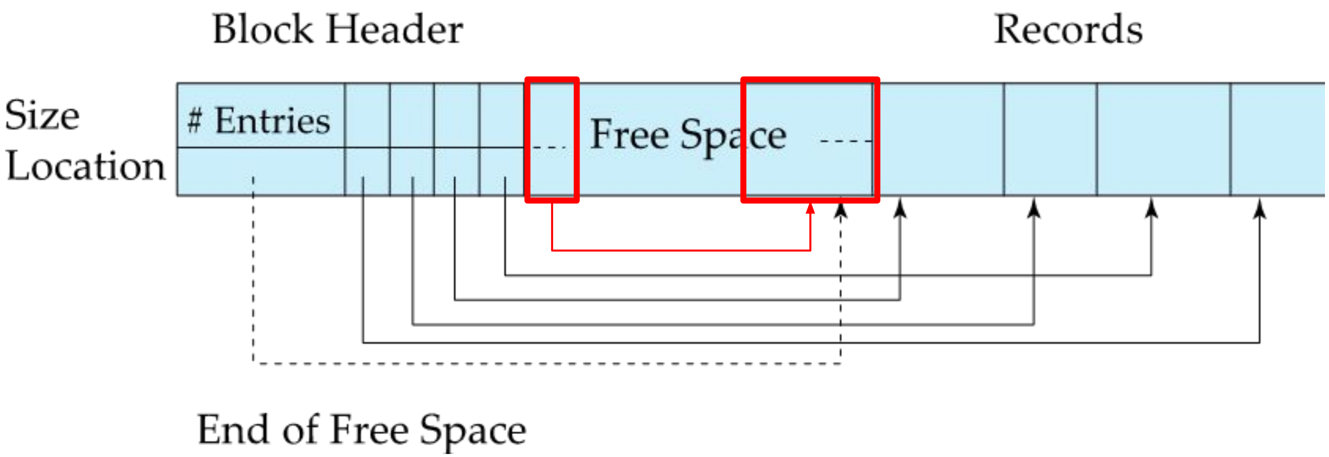
Slotted-Page Structure



- **Slotted-page structure** is commonly used for organizing records within a block.
- There is a header at the beginning of each block that contains:
  - number of record entries
  - end of free space in the block
  - location and size of each record
- The actual records are allocated *contiguously* in the block, starting from the **end** of the block.

Variable-Length Records: Slotted Page Structure

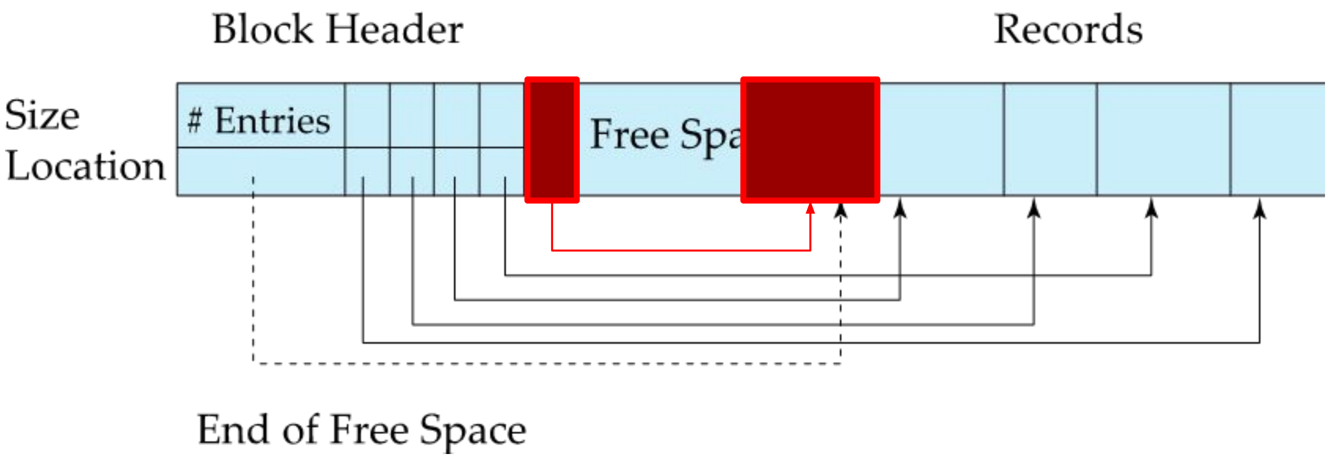
Slotted-Page Structure



- Insertion
  - Space for a new record is allocated at the end of Free Space
  - A new entry containing new record's size and location is added to the header
  - End of Free Space should be updated accordingly

Variable-Length Records: Slotted Page Structure

Slotted-Page Structure



- Deletion
  - The space for a deleted record is freed
  - The corresponding entry is set to deleted (e.g., set the size to be -1)
  - Records can be moved around within a page to keep them contiguous with no empty space between them; entry in the header must be updated.

Assignments

- Reading: Ch13.2
- Practice Excercises: 13.1, 13.2

Solutions to the Practice Excercises:
https://www.db-book.com/Practice-Exercises/index-solu.html

The End