Course code : **CSE2007**
Course title : **Database Management System**
Module : **6**
Topic : **1**

Placing File Records on Disk

Objectives

This session will give the knowledge about

- Disk Storage, Basic File Structures
- Placing file records into the disks

Disk Storage

- Databases typically stored on magnetic disks
  - Accessed using physical database file structures
- Storage hierarchy
  - Primary storage
    - CPU main memory, cache memory
  - Secondary storage
    - Magnetic disks, flash memory, solid-state drives
  - Tertiary storage
    - Removable media

Disk Storage

Cache memory

- Static RAM
- DRAM

Mass storage

- Magnetic disks
- CD-ROM, DVD, tape drives

Flash memory

- Nonvolatile

Disk Storage

Types of Storage with Capacity, Access Time, Max Bandwidth (Transfer Speed), and Commodity Cost

Type	Capacity*	Access Time	Max Bandwidth	Commodity Prices (2014)**
Main Memory- RAM	4GB–1TB	30ns	35GB/sec	$100–$20K
Flash Memory- SSD	64 GB–1TB	50µs	750MB/sec	$50–$600
Flash Memory- USB stick	4GB–512GB	100µs	50MB/sec	$2–$200
Magnetic Disk	400 GB–8TB	10ms	200MB/sec	$70–$500
Optical Storage	50GB–100GB	180ms	72MB/sec	$100
Magnetic Tape	2.5TB–8.5TB	10s–80s	40–250MB/sec	$2.5K–$30K
Tape jukebox	25TB–2,100,000TB	10s–80s	250MB/sec–1.2PB/sec	$3K–$1M+

*Capacities are based on commercially available popular units in 2014.

**Costs are based on commodity online marketplaces.

Storage Organization of Databases

Persistent data

- Most databases

Transient data

- Exists only during program execution

File organization

- Determines how records are physically placed on the disk
- Determines how records are accessed

Secondary Storage Devices: Hard disk drive

Bits (ones and zeros)

- Grouped into bytes or characters

Disk capacity measures storage size

- Capacity = no.of cylinders * no.of sectors/track * no.of surface * allocation unit

Disks may be single or double-sided

Concentric circles called tracks

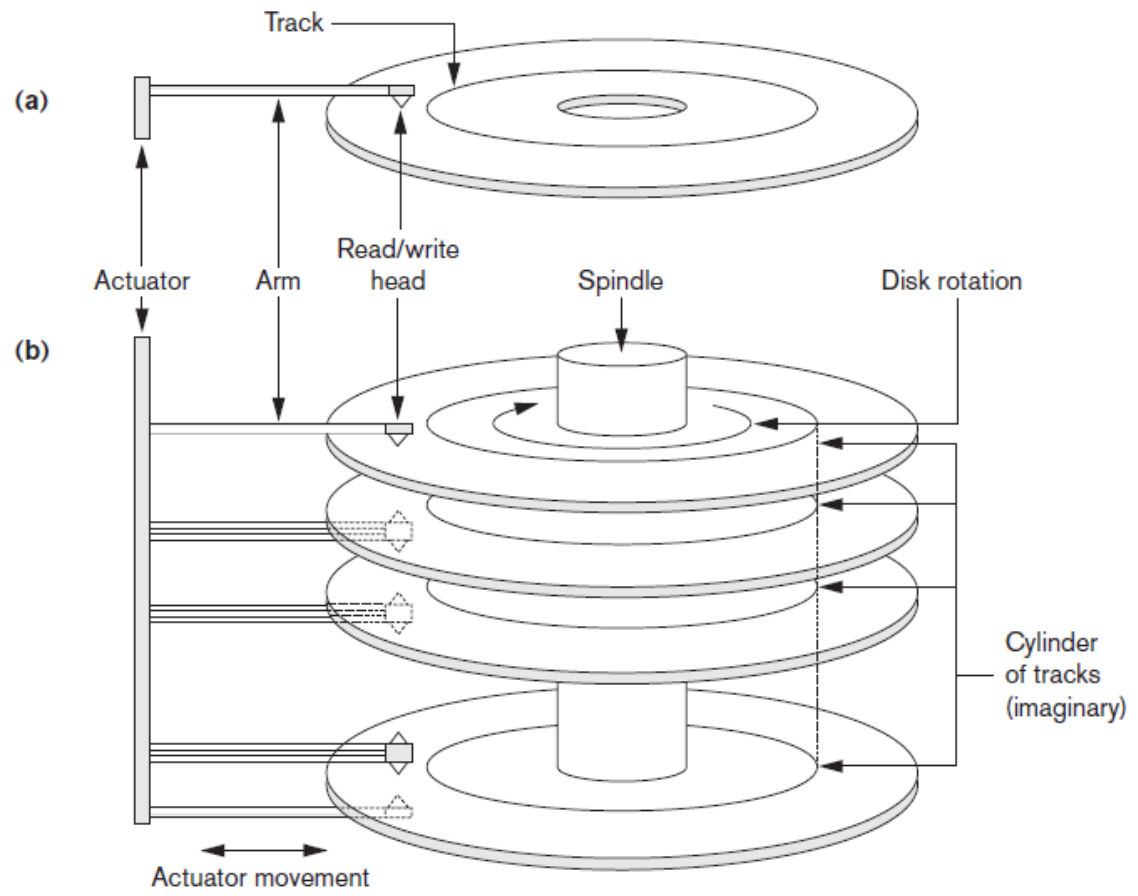
- Tracks divided into blocks or sectors

Disk packs

- Cylinder

Single-Sided Disk and Disk Pack

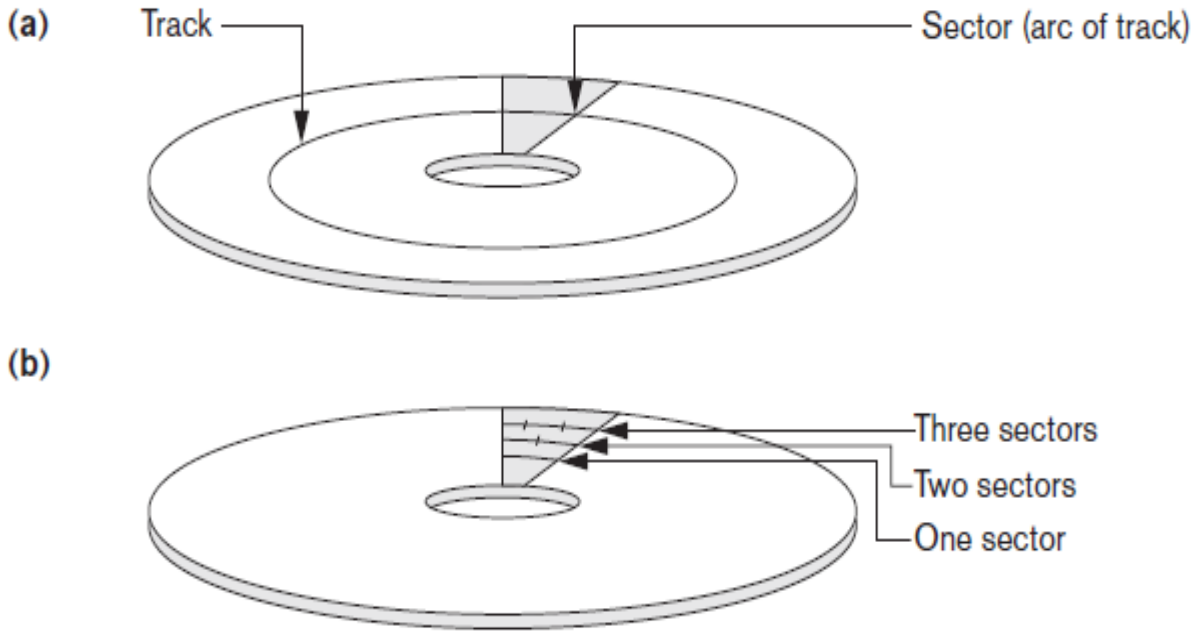
Figure: (a) A single-sided disk with read/write hardware (b) A disk pack with read/write hardware



Sectors on a Disk

Figure: Different sector organizations on disk

(a) Sectors subtending a fixed angle (b) Sectors maintaining a uniform recording density



Seek time: time required to put R/W Head on respective track

Latency: time required to put R/W Head on respective sector

Secondary Storage Devices (cont'd.)

Formatting

- Divides tracks into equal-sized disk blocks
- Blocks separated by interblock gaps

Data transfer in units of disk blocks

- Hardware address supplied to disk I/O hardware

Buffer

- Used in read and write operations

Read/write head

- Hardware mechanism for read and write operations

Secondary Storage Devices (cont'd.)

Disk controller

- Interfaces disk drive to computer system
- Standard interfaces
  - SCSI - Small Computer System Interface
  - SATA - Serial Advanced Technology Attachment
  - SAS - Serial Attached SCS

Bad Sector

Disk Partitioning

Disk Fragmentation

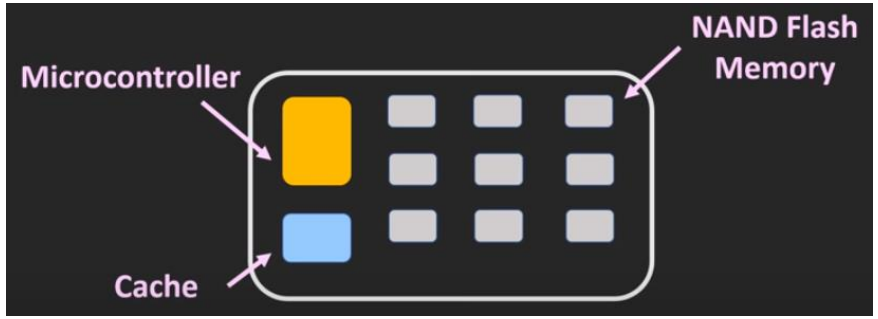
Secondary Storage Devices (cont'd.)

Techniques for efficient data access

- Data buffering
- Proper organization of data on disk
- Reading data ahead of request
- Proper scheduling of I/O requests
- Use of log disks to temporarily hold writes
- Use of SSDs or flash memory for recovery purposes

Solid State Device Storage

- Sometimes called flash storage
- Main component: controller
- Set of interconnected flash memory cards
- No moving parts
- Data less likely to be fragmented
- More costly than HDDs
- DRAM-based SSDs available
    - Faster access times compared with flash



Magnetic Tape Storage Devices

Sequential access

- Must scan preceding blocks

Tape is mounted and scanned until required

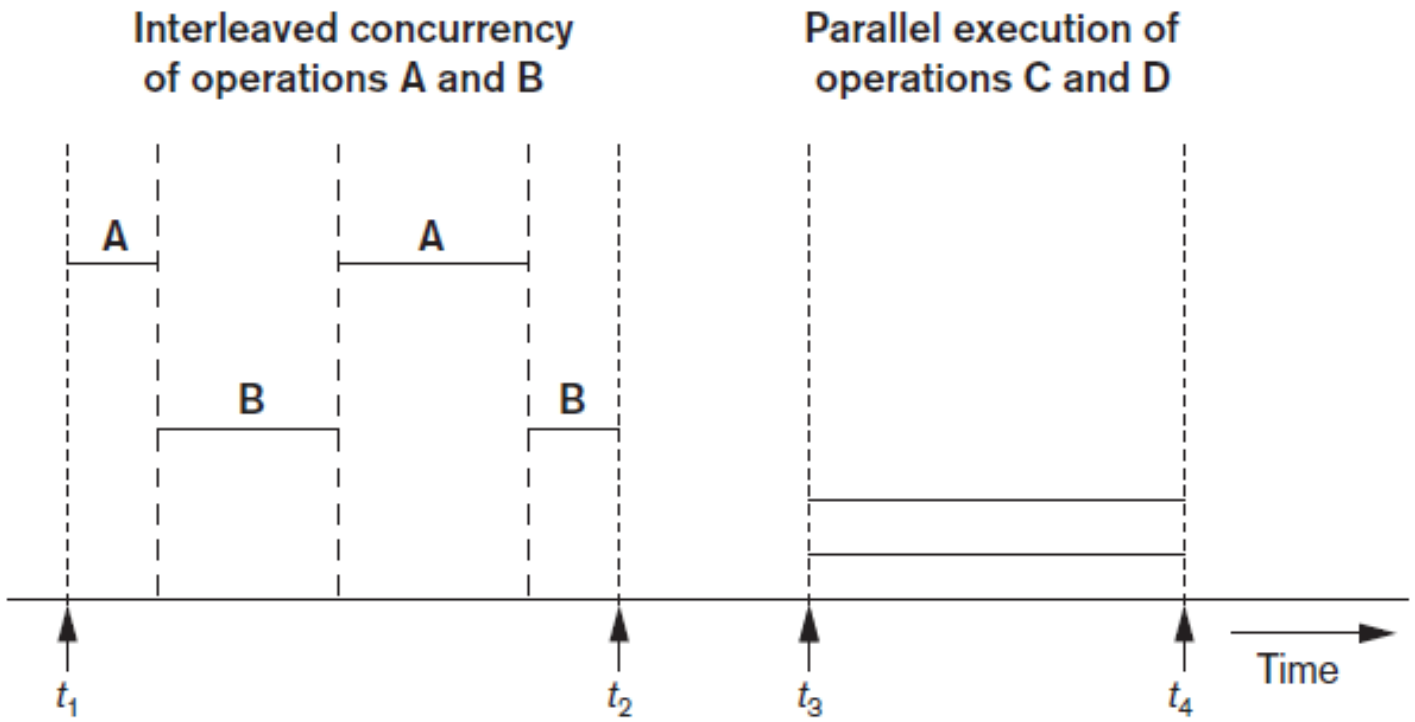
block is under read/write head

Important functions

- Backup
- Archive

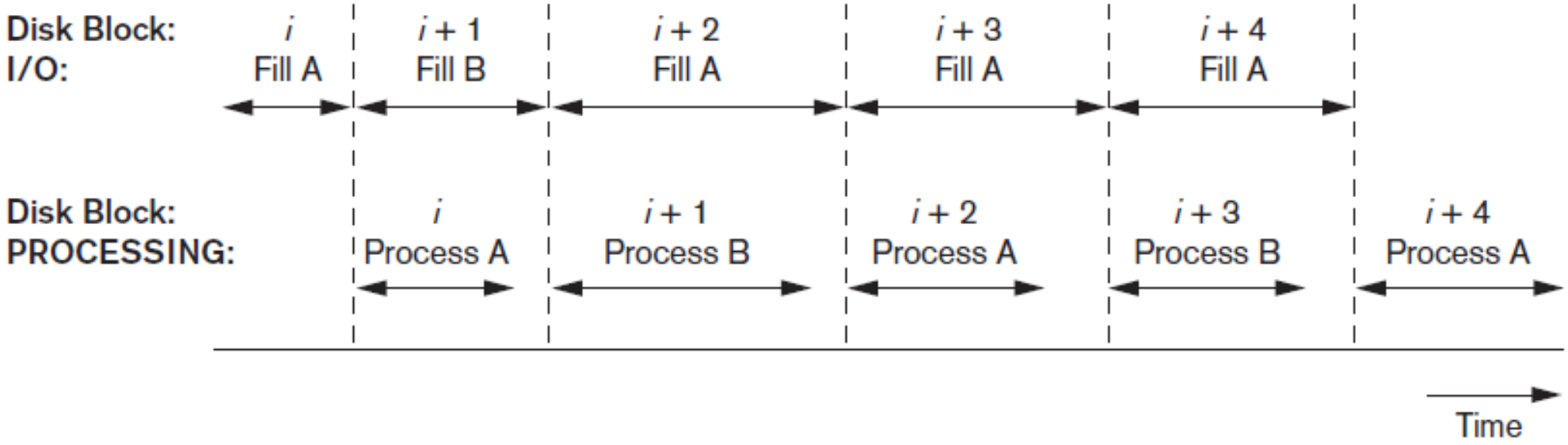
Buffering of Blocks

Buffering most useful when processes can run concurrently in parallel



Buffering of Blocks

Double buffering can be used to read continuous stream of blocks



Use of two buffers, A and B, for reading from disk

Buffer Management and Replacement Strategies

Buffer management information

- Pin count
- Dirty bit

Buffer replacement strategies

- Least recently used (LRU)
- Clock policy
- First-in-first-out (FIFO)

Placing File Records on Disk

Record: collection of related data values or items

- Values correspond to record field

Data types

- Numeric
- String
- Boolean
- Date/time

Binary large objects (BLOBs)

- Unstructured objects

Placing File Records on Disk

For example, an EMPLOYEE record type may be defined—using the C programming language notation—as the following structure:

```
struct employee{
char name[30];
char ssn[9];
int salary;
int job_code;
char department[20];
} ;
```

Records and Record Types

A file is a sequence of records. In many cases, all records in a file are of the same record type.

If every record in the file has exactly the same size (in bytes), the file is said to be made up of fixed-length records.

If different records in the file have different sizes, the file is said to be made up of variable-length records.

Fixed-Length Records

Simple approach:

- Store record i starting from byte n * (i – 1), where n is the size of each record.
- Record access is simple but records may cross blocks
- Modification: do not allow records to cross block boundaries

Deletion of record i: alternatives:

1. move records i + 1, . . ., n to i, . . . , n – 1
2. move record n to i
3. do not move records, but link all free records on a free list

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

Option-1

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

Option-2

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

Option-3

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

Variable-length records

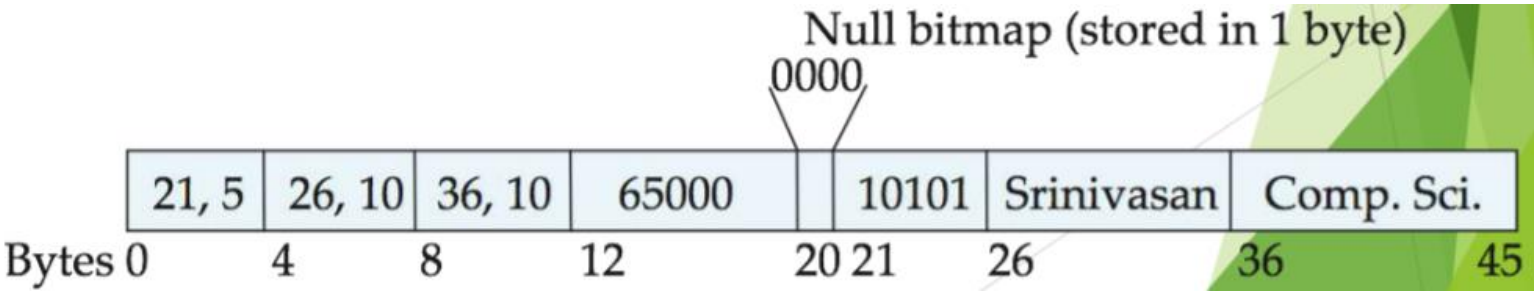
A file may have variable-length records for several reasons:

- One or more of the fields are of varying size (variable-length fields).
- Fields may have multiple values for individual records; such a field is called a repeating field and a group of values for the field is often called a repeating group.
- One or more of the fields are optional
- The file contains records of different record types and hence of varying size (mixed file).

Variable-length records

Variable-length records :

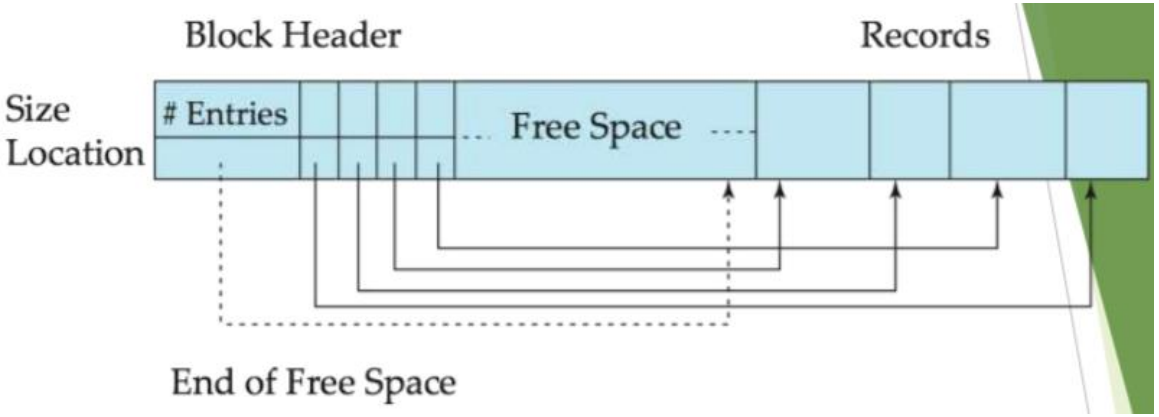
- Attributes are stored in order
- Variable length attributes represented by fixed size (offset, length), with actual data stored after all fixed length attributes
- Null values represented by null-value bitmap



Variable-Length Records: Slotted Page Structure

Slotted page header contains:

- number of record entries
- end of free space in the block
- location and size of each record

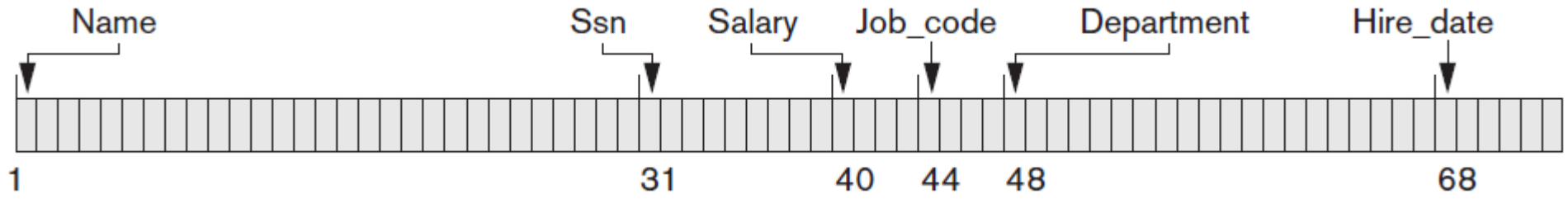


Records can be moved around within a page to keep them contiguous with no empty space between them; entry in the header must be updated.

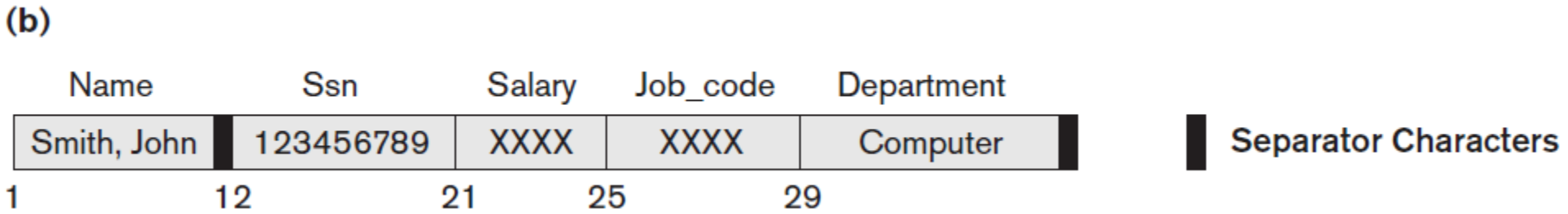
Pointers should not point directly to record — instead they should point to the entry for the record in header.

Records and Record Types

A fixed-length record with six fields and size of 71 bytes.



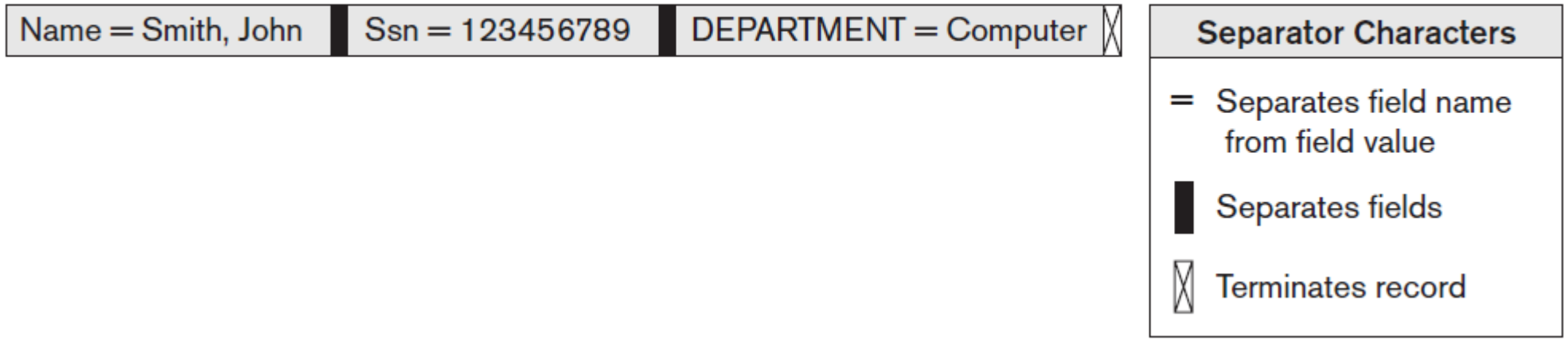
A record with two variable-length fields and three fixed-length fields.



Records and Record Types

A variable-field record with three types of separator characters.

(c)



Organization of Records in Files

Heap – a record can be placed anywhere in the file where there is space

Sequential – store records in sequential order, based on the value of the search key of each record

Hashing – a hash function computed on some attribute of each record; the result specifies in which block of the file the record should be placed

Records of each relation may be stored in a separate file. In a multi-table clustering file organization records of several different relations can be stored in the same file

Motivation: store related records on the same block to minimize I/O

Record Blocking and Spanned versus Unspanned Records

File records allocated to disk blocks

Spanned records

- Larger than a single block
- Pointer at end of first block points to block containing remainder of record

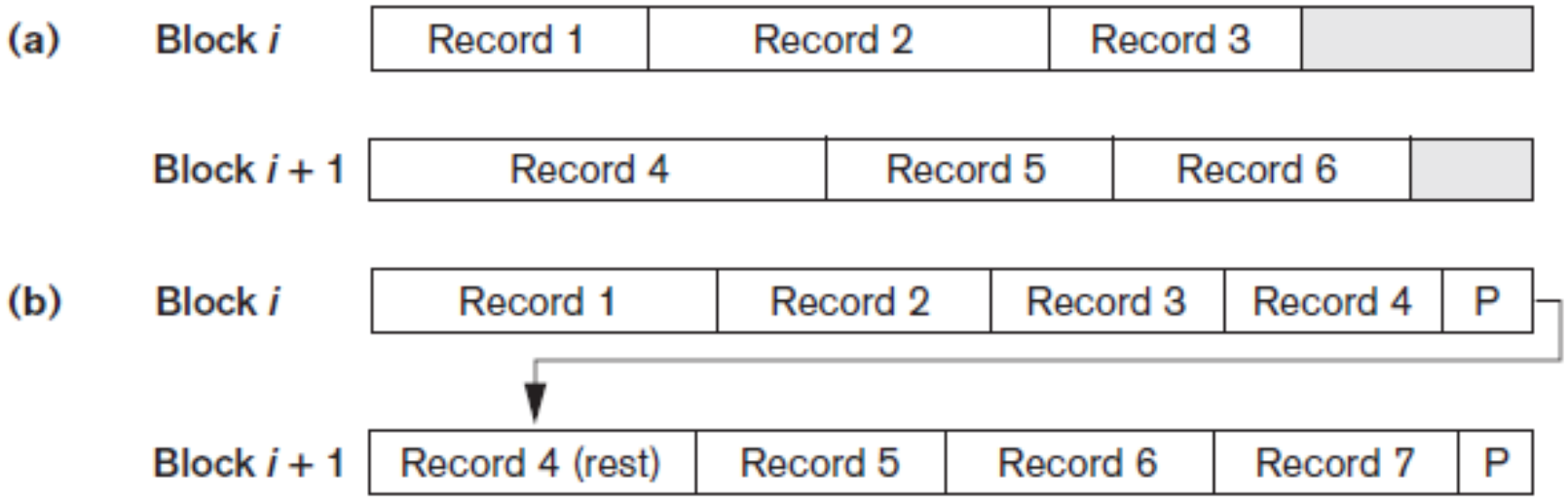
Unspanned

- Records not allowed to cross block boundaries

Record Blocking and Spanned versus Unspanned Records

Blocking factor

- Average number of records per block for the file



Types of record organization (a) Unspanned (b) Spanned

Allocating File Blocks on Disk

Allocating file blocks on disk

- Contiguous allocation - the file blocks are allocated to consecutive disk blocks
- Linked allocation - each file block contains a pointer to the next file block
- A combination of the two allocates clusters of consecutive disk blocks, and the clusters are linked. Clusters are sometimes called file segments or extents.
- Indexed allocation - use indexed allocation, where one or more index blocks contain pointers to the actual file blocks

File Headers

Contains file information needed by system programs

- Disk addresses
- Format descriptions

To search for a record on disk, one or more blocks are copied into main memory buffers.

Programs then search for the desired record or records within the buffers, using the information in the file header.

If the address of the block that contains the desired record is not known, the search programs must do a linear search through the file blocks.

Summary

This session will give the knowledge about

- Disk Storage, Basic File Structures
- Placing file records into the disks