Database Management Systems

BITS Pilani
Hyderabad Campus

Dr.R.Gururaj
CS&IS Dept.

Lecture Session-11
Data Storage

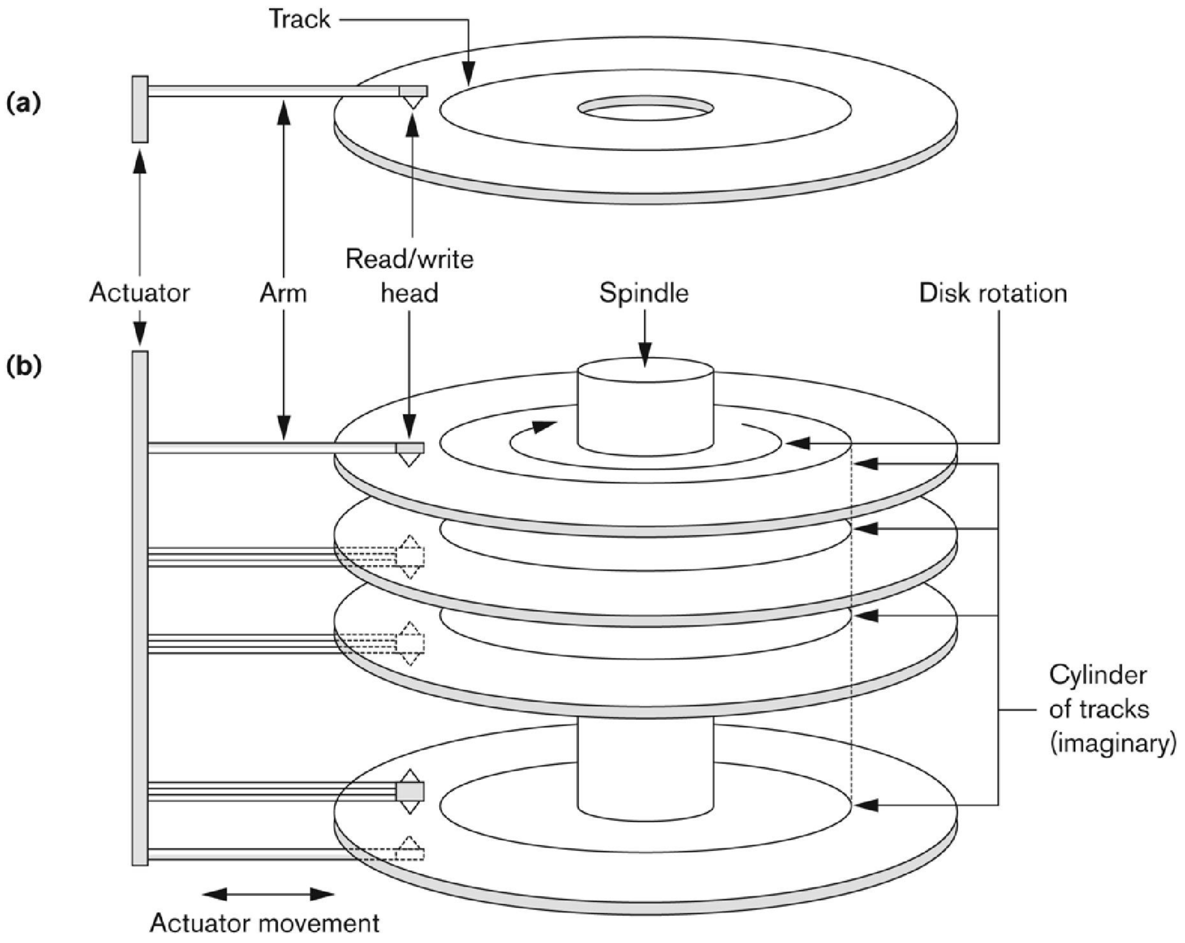
Content

- ❑ *Disk pack features*
- ❑ *Records and Files*
- ❑ *File operations*
- ❑ *Ordered and unordered features*

Disk Storage

- Disk is the preferred secondary storage device for high storage capacity and low cost.
- Data stored as magnetized areas on magnetic disk surfaces.
- A **disk pack** contains several magnetic disks connected to a rotating spindle.
- Disks are divided into concentric circular **tracks** on each disk **surface**.
    - Track capacities vary typically from 4 to 50 Kbytes or more

Figure 13.1
(a) A single-sided disk with read/write hardware. (b) A disk pack with read/write hardware.



- ❑ A track is divided into smaller **blocks** or **sectors.**
- ❑ The division of a track into **sectors** is hard-coded on the disk surface and cannot be changed.
- ❑ A track is divided into **blocks**.
  1. The block size B is fixed for each system.

     Typical block sizes range from B=512 bytes to B=4096 bytes.
  2. Whole blocks are transferred between disk and main memory for processing.

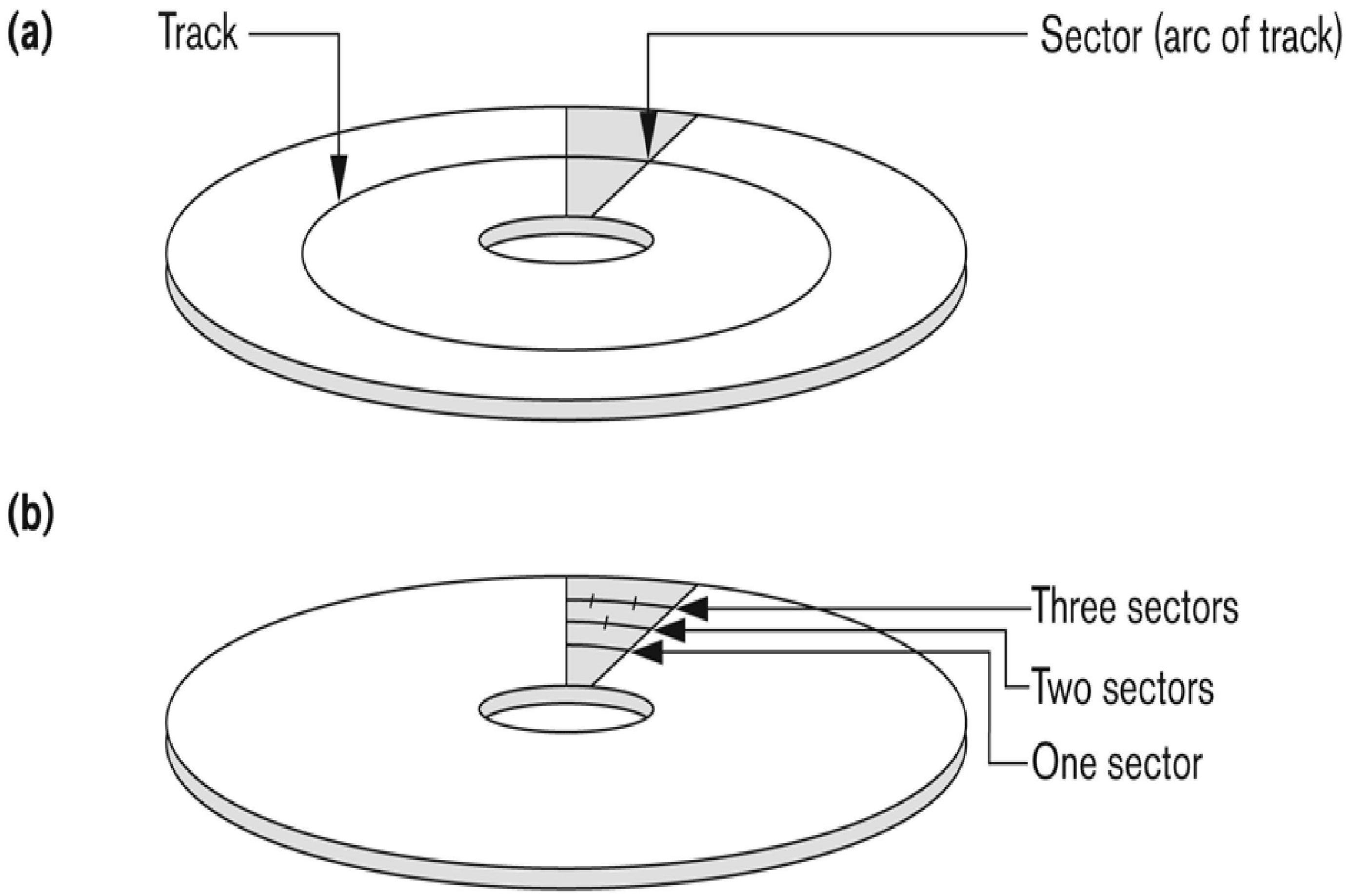


Figure 13.2 Different sector organizations on disk. (a) Sectors subtending a fixed angle. (b) Sectors maintaining a uniform recording density.

- ❑ A **read-write head** moves to the track that contains the block to be transferred.

  Disk rotation moves the block under the read-write head for reading or writing.

- ❑ A physical disk block (hardware) address consists of:
  - ➢ a cylinder number (imaginary collection of tracks of same radius from all recorded surfaces)
  - ➢ the track number or surface number (within the cylinder)
  - ➢ and block number (within track).
- ❑ Reading or writing a disk block is time consuming because of the seek time s (time to position the head on required track)

  3-7msec and rotational delay (latency) – time to position at the beginning of the required block **rd**.

  3-4 msec with 15000rpm

Block transfer time. Smaller than above two.

Files and Records

- A **file** is a *sequence* of records, where each record is a collection of data values (or data items).
- A **file descriptor** (or **file header**) includes information that describes the file, such as the *field names* and their *data types*, and the addresses of the file blocks on disk.
- Records are stored on disk blocks.
- The **blocking factor (bfr)** for a file is the (average) number of file records stored in a disk block.
- A file can have **fixed-length** records or **variable-length** records.

- File records can be **unspanned** or **spanned**
  - **Unspanned**: no record can span two blocks
  - **Spanned**: a record can be stored in more than one block
- The physical disk blocks that are allocated to hold the records of a file can be *contiguous, linked*.
- In a file of fixed-length records, all records have the same format. Usually, unspanned blocking is used with such files.
- Files of variable-length records require additional information to be stored in each record, such as **separator characters** and **field types**.
  - Usually spanned blocking is used with such files.

File operations

Typical file operations include:

- **OPEN**: Readies the file for access, and associates a pointer that will refer to a *current* file record at each point in time.
- **FIND**: Searches for the first file record that satisfies a certain condition, and makes it the current file record.
- **FINDNEXT**: Searches for the next file record (from the current record) that satisfies a certain condition, and makes it the current file record.
- **READ**: Reads the current file record into a program variable.
- **INSERT**: Inserts a new record into the file & makes it the current file record.
- **DELETE**: Removes the current file record from the file, usually by marking the record to indicate that it is no longer valid.
- **MODIFY**: Changes the values of some fields of the current file record.
- **CLOSE**: Terminates access to the file.
- **REORGANIZE**: Reorganizes the file records.

  For example, the records marked deleted are physically removed from the file or a new organization of the file records is created.
- **READ_ORDERED**: Read the file blocks in order of a specific field of the file.

Unordered Files

Also called a *heap* or a *pile* file.

New records are inserted at the end of the file.

A *linear search* through the file records is necessary to search for a record.

- This requires reading and searching half the file blocks on the average, and is hence quite expensive.

Record insertion is quite efficient.

Reading the records in order of a particular field requires sorting the file records.

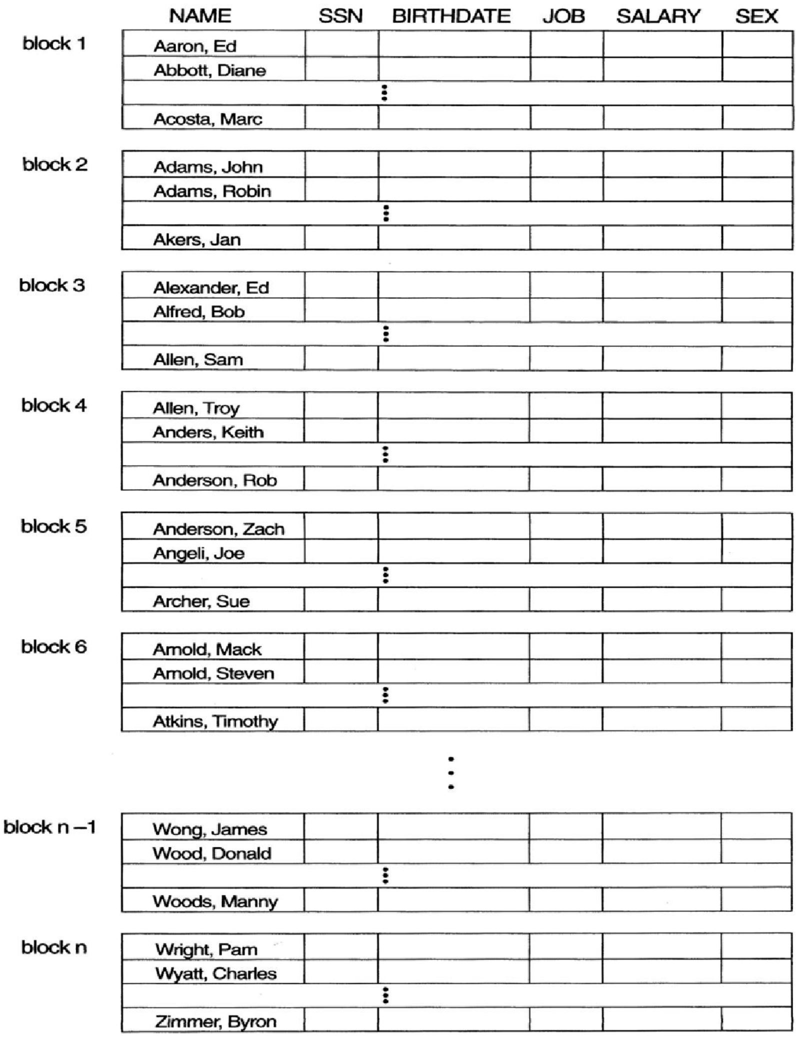
Ordered Files

- Also called a *sequential* file.
- File records are kept sorted by the values of an *ordering field*.
- Insertion is expensive: records must be inserted in the correct order.

A *binary search* can be used to search for a record on its *ordering field* value.

  - This requires reading and searching $\log_2$ of the file blocks on the average, an improvement over linear search.

- Reading the records in order of the ordering field is quite efficient.



	NAME	SSN	BIRTHDATE	JOB	SALARY	SEX
block 1	Aaron, Ed					
	Abbott, Diane					
			⋮			
	Acosta, Marc					
block 2	Adams, John					
	Adams, Robin					
			⋮			
	Akers, Jan					
block 3	Alexander, Ed					
	Alfred, Bob					
			⋮			
	Allen, Sam					
block 4	Allen, Troy					
	Anders, Keith					
			⋮			
	Anderson, Rob					
block 5	Anderson, Zach					
	Angeli, Joe					
			⋮			
	Archer, Sue					
block 6	Arnold, Mack					
	Arnold, Steven					
			⋮			
	Atkins, Timothy					
			⋮			
block n –1	Wong, James					
	Wood, Donald					
			⋮			
	Woods, Manny					
block n	Wright, Pam					
	Wyatt, Charles					
			⋮			
	Zimmer, Byron					

Summary

- ✓ *What is Disk storage*
- ✓ *Disk characteristics*
- ✓ *Disk pack structure*
- ✓ *Files and Records*
- ✓ *Ordered and unordered files*