Physical Storage

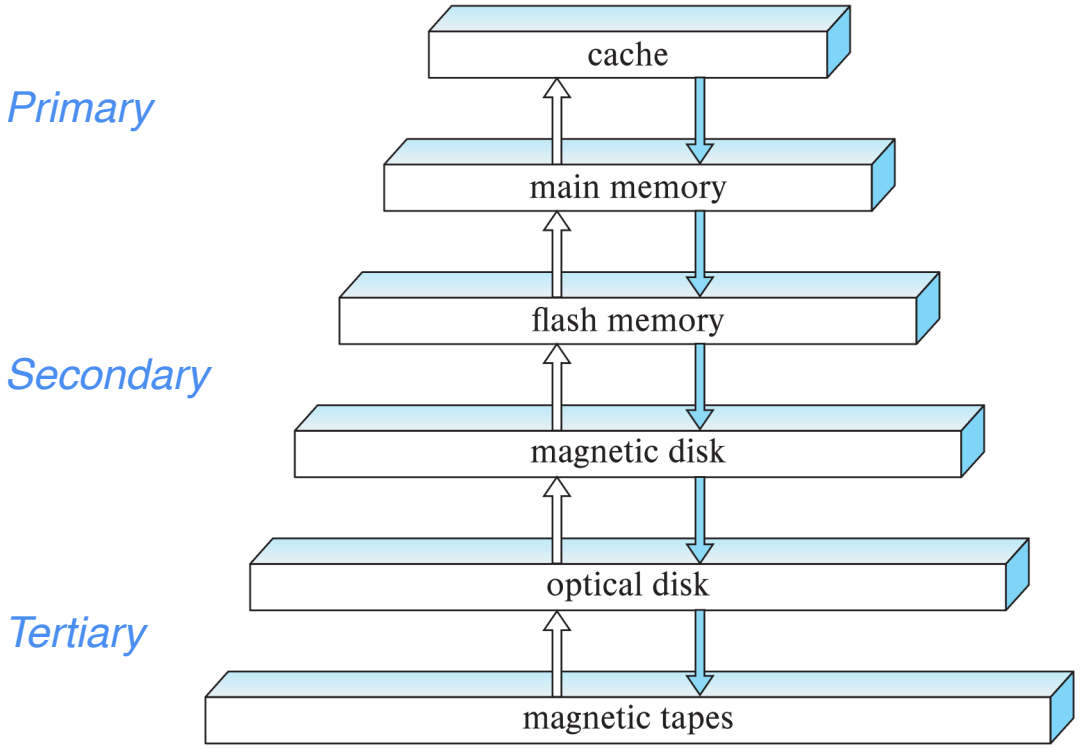
DB physical level

- Data is physically saved in **storage** devices
  - 2 types, regarding **persistence**:
    - **volatile:** loses content when power is *turned off*
    - **non-volatile**: content persists
  - Other possible classifications: Regarding speed of data access, cost (per unit of data), reliability, etc.

- Our view of the DB is at the **logical** level
  In the relational model, as a collection of tables

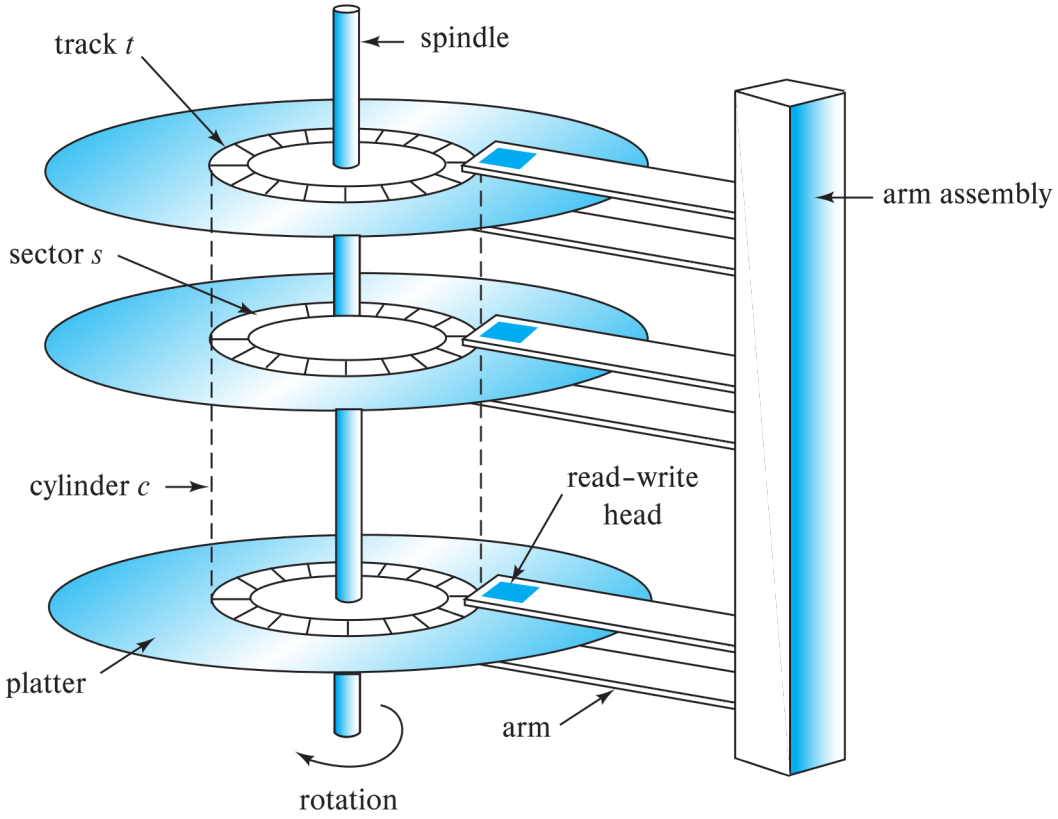
- Goals of a database system:
  - Simplify and facilitate access to data
  - Avoid burdening users with physical details

Storage Hierarchy



- **Primary storage**: Fastest media, but volatile
- **Secondary storage (online)**: Non-volatile, moderately fast access time
- **Tertiary storage (offline)**: Non-volatile, slow access time, used for *archive*

Magnetic Disks



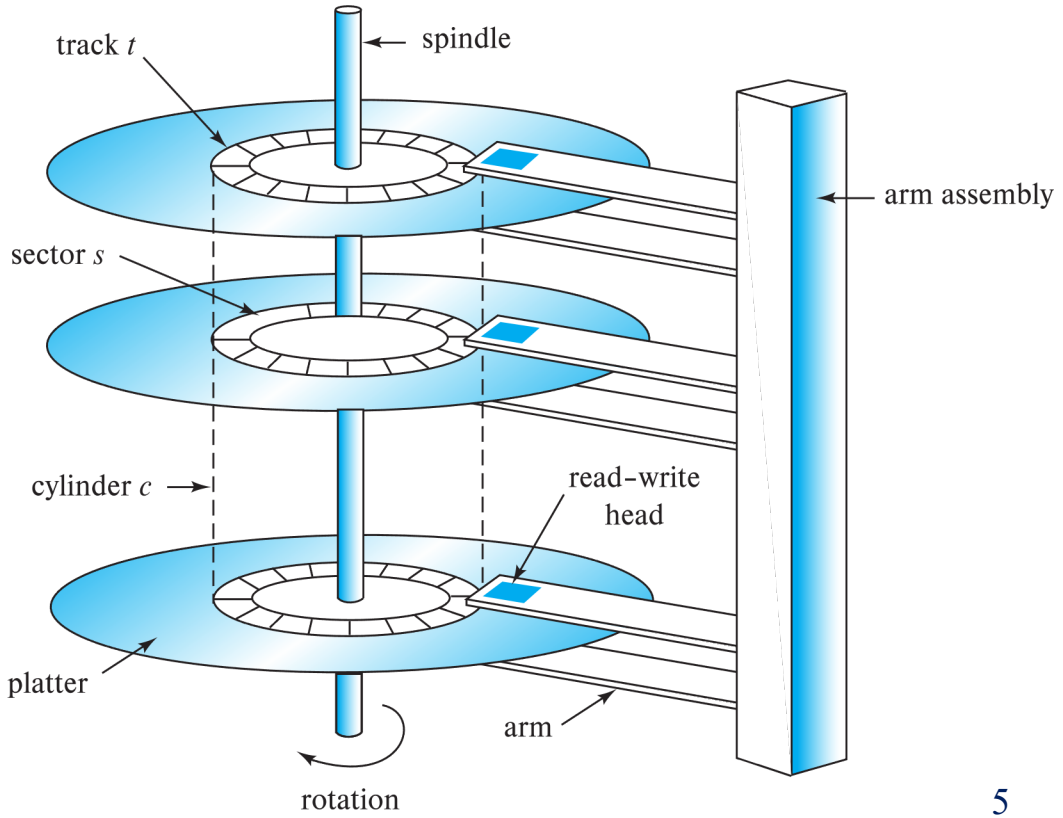
Magnetic Disks

- **Platter**: saves info in both surfaces (magnetic material)
- **Read-write head**: stores info as reversals of directions of magnetization
- Head-disk assemblies
  - multiple platters: usually 1-5
  - one head per platter-surface, mounted on a common arm (move together)

- Surface of platter divided into circular **tracks**
  Over 50K-100K tracks per platter on typical hard disks
- Tracks are divided into **sectors**: smallest unit of data that can be read/written
  ~512 bytes; 500-1000 sectors on inner tracks, 1000-2000 on outer tracks
- i^{th} **cylinder** consists of i^{th} track of all the platters

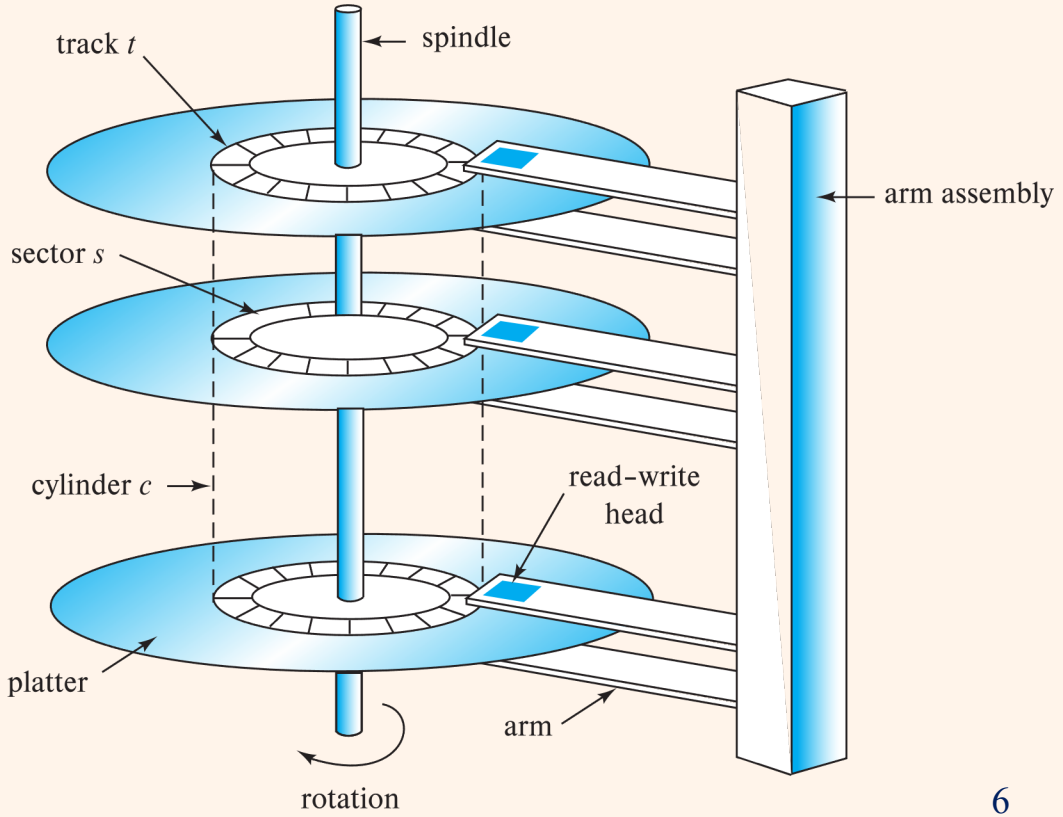
Magnetic Disks

- How to read/write a sector?
  1. disk arm places the head on right track
  2. while platter spins continually, data is read/written as sector passes under head



Exercise: No. sectors per track

- The no. sectors per track is smaller in inner-tracks and larger in outer-tracks.
  - Which implications does this have in terms of performance?



HDDs: an impressive piece of technology

- With a width of less than a hundred nanometers and a thickness of about ten, the **head** flies above the platter at a *speed* of up to 15,000 RPM, at a *height* that is the equivalent of 40 atoms.
- Consider this small comparison.
  If the read/write head were a Boeing 747, and the hard-disk platter were the surface of the Earth:
  - The head would fly at Mach 800
  - At less than 1 cm from the ground
  - And count every blade of grass
  - Making fewer than 10 unrecoverable counting errors in an area equivalent to the whole Ireland

*Source: Matthieu Lamelot, Tom's Hardware.

Important concepts

- **Disk block**: logical unit for storage retrieval/allocation (a.k.a. *page*)
  Size of 4-6 KB
  - *Larger blocks* reduce no. transfers from disk, but space is wasted (partially filled blocks)

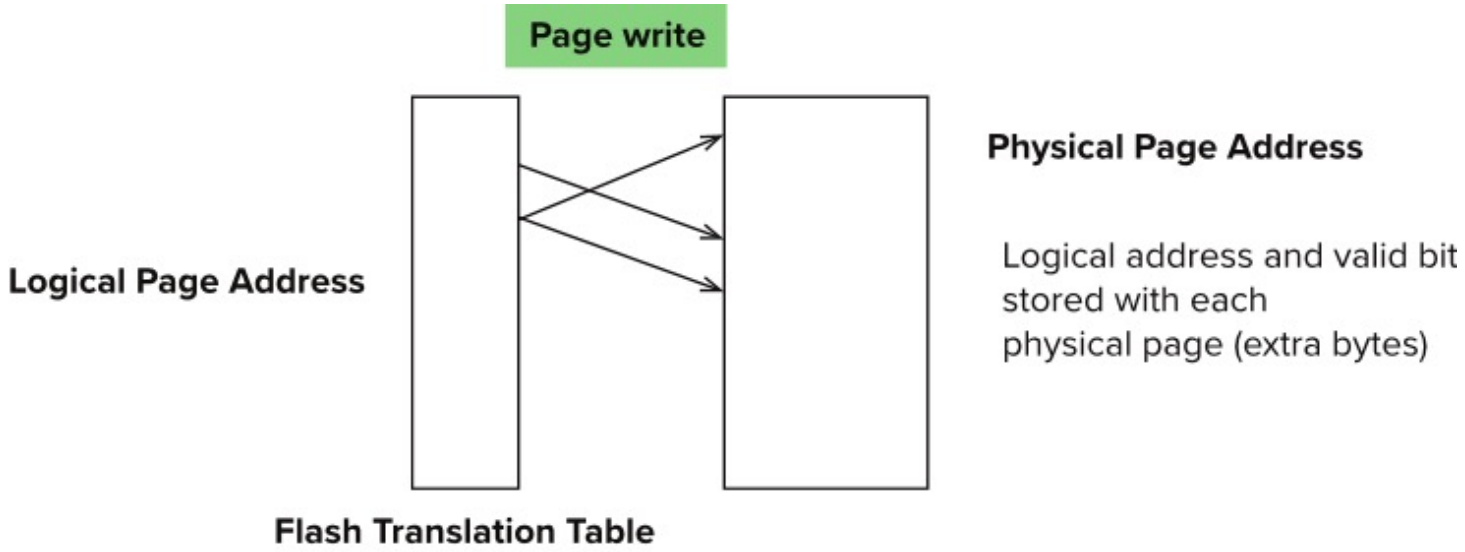
- **Access pattern**
  - **Sequential**: successive requests are for successive disk blocks
    - Disk seek required only for first block (best transfer rates)
  - **Random**: successive requests are for blocks from all over the disk
    - Each access requires a seek (too many seeks harm transfer rate)

Flash Storage

- **Solid state disks (SSD)**
  - Use standard block-oriented disk interfaces
  - Much faster random access than HDDs
    Latency of 20-100 **microseconds** for a page retrieval
  - High data transfer rate
    up to 500MB/s with SATA, up to 3 GB/s with NVMe PCIe
- Flash memory
  - **Block**: smallest data unit that can be read
    512 bytes - 4 KB
  - Blocks need to be erased before written!
    - **Erase block**: smallest unit of erasable data
      256 KB to 1 MB (128 to 256 pages)
      2-5 millisecs
  - After 0.1-1 million erases, *erased blocks* become unreliable

Flash Storage

- **Translation table** tracks mapping by means of **flash translation layer**



- **Remapping** logical blocks to different physical blocks avoids waiting for erase before writing
  - **Wear leveling**: logical blocks frequently modified are assigned to physical blocks modified few times
    hot data / cold data

Exercise: Translation table size

- We have a flash storage system with:
  - Total size = 64GB
  - Block size = 4KB
  - Memory address = 4 bytes
- Which is the size of the translation table?

Performance Measures for Disks

- **Mean time to failure (MTTF)**: average time it is expected to run without failure (3-5 years)
- **Access time**: from requesting read/write to beginning of data transfer.

HDD

  - **Seek time**: time to reposition the arm over the correct track.
    Average seek time is 1/2 the seek time of the worst case (4-10 milliseconds)
  - **Rotational latency**: time for the correct sector to appear under head
    Average latency is 1/2 a full-rotation time.

- **Data-transfer rate**: rate at which data can be retrieved/stored to disk
  - HDDs: 25 to 200 MB/s max. (**lower** for inner tracks)
  - SSDs: 400 MB/s (SATA), 2-3 GB/s (NVMe PCIe)
- **I/O ops. per second (IOPS)**: no. *random* block reads/writes per second
  - HDDs: 50-200 IOPS
  - SSDs: Read: 10,000 IOPS; Write: 40,000 IOPS

Improvement of Disk-Block Access

Main goal: to minimize the no. (random) accesses

Mainly in HDDs, but also in SSDs

- **Buffering:** in-memory buffer to store temporary read disk blocks
  Done by database systems, but also by operating systems
- **Read-ahead** extra blocks from a track to a buffer anticipating they will be requested soon
  Not so useful for random block access
- **Disk-arm-scheduling** re-orders block requests to minimize disk arm movement
  Results may be returned in a different order from the request order

Elevator algorithm:

Improvement of Disk-Block Access

- **File organization**: Allocate blocks of a file as contiguously as possible.
  - Files may get **fragmented**
    E.g., if free blocks on disk are scattered, so will be blocks of new files
  - Some systems allow to **defragment** the file system
    Files are backup and restored in a more contiguous way

- **Non-volatile write buffers**:
  1. Disk controller first writes blocks to a non-volatile buffer
  2. It subsequently writes the data to disk
     - Can increase efficiency by minimizing disk arm movement
  - On recovery from a system crash, interrupted operations won't be lost (will be pending in the buffer)

Physical Storage