CENG 351
Introduction to Data Management and File Structures

CENG 351- Fall 2024

- **Instructors:**
  - Section 1: İsmail Sengör Altıngövde Office:A203 (altingovde@ceng.metu.edu.tr)
  - Section 2: Pınar Karagöz Office:A404 (karagoz@ceng.metu.edu.tr)

- **Lecture Hours:**

Section 1:	Tuesday 15.40, 16.40;	Thursday 15.40	(BMB 1)
Section 2:	Tuesday 13.40, 14.40;	Friday 10.40	(BMB 3)

- **Course Web page:**

  http://odtuclass.metu.edu.tr

- **Teaching Assistants:**

Ardan Yılmaz	ardan@ceng.metu.edu.tr
Emre Külah	kulah@ceng.metu.edu.tr
Haktan Sarıtepe	haktans@ceng.metu.edu.tr
İbrahim Tarakçı	tarakci@ceng.metu.edu.tr

References

- Raghu Ramakrishnan, Database Management Systems (3rd. ed.), McGraw Hill, 2003 (text book).
- R. Elmasri, S.B. Navathe, Fundamentals of Database Systems, 4th edition, Addison-Wesley, 2004.
- Michael J. Folk, B. Zoellick, File Structures, 2nd ed., Addison-Wesley Longman Ltd., 1991.

Course Outline

1. Introduction to Relational Ratabase Systems
2. Relational Model and E/R Modeling, Normalization
3. Relational Algebra, Relational Calculus
4. Structural Query Language (SQL)
5. Secondary Storage Media
6. Sequential File Processing
7. External Sorting of Large Files
8. Indexing: Multilevel Indexing and B+ trees
9. Hashing (static, linear, extendible hashing)

Grading

- In-class written assignments 25% (4 x 6.25% each)
  - ICA: Week of Oct 21, 2024
  - ICA2: Week of Nov 4, 2024
  - ICA3: Week of Nov 11, 2024 (Do not miss the SQL Lab Demo)
  - ICA4: Week of Dec 25, 2022

- Programming assignments 20% (2 x 10% each)
- Midterm Exam 25%
- Final Exam 30%

Tentative date for the midterm: Week of Nov. 18, 2024

Course Conduct

- In-class Assignments:
  - All in-class assignments will be conducted in the class.

- Programming Assignment:
  - Programming assignments will be offline such that the students will submit their solutions on course home page (odtuclass) by the deadline.

Course Conduct

- Midterm and Final Exams:
  - Midterm and Final Exams will be conducted face to face in the class.

Grading Policies

- Policy on missed midterm:
  - You may miss the midterm exam or written assignments only if you inform the instructors BEFORE the exam/class and you have a legal excuse (e.g. medical report). There will be a make-up exam right after the week following the end of the time period covering the legal excuse.
- Lateness policy:
  - You have a 5 day late submission opportunity for the programming assignments. (on the basis of day granularity)
- All assignments and programs are to be your own work. No group projects or assignments are allowed.

Grading Policies

- Final Exam Eligibility:
  - **A student can take the final exam if and only if the average of his/her in-class written assignments is at least 30 points. Otherwise; the student is not allowed to take the final exam and hence will get "NA".**

- Missing the final exam without a legal excuse means FAILING the course directly (i.e., you will get "NA")!

Introduction to the Basic Concepts

Motivation

- How to handle *large amount* of data:
  - Storage of data (disk resident)
  - Organization of data
  - Access to data
  - Processing of data

- With a system point of view

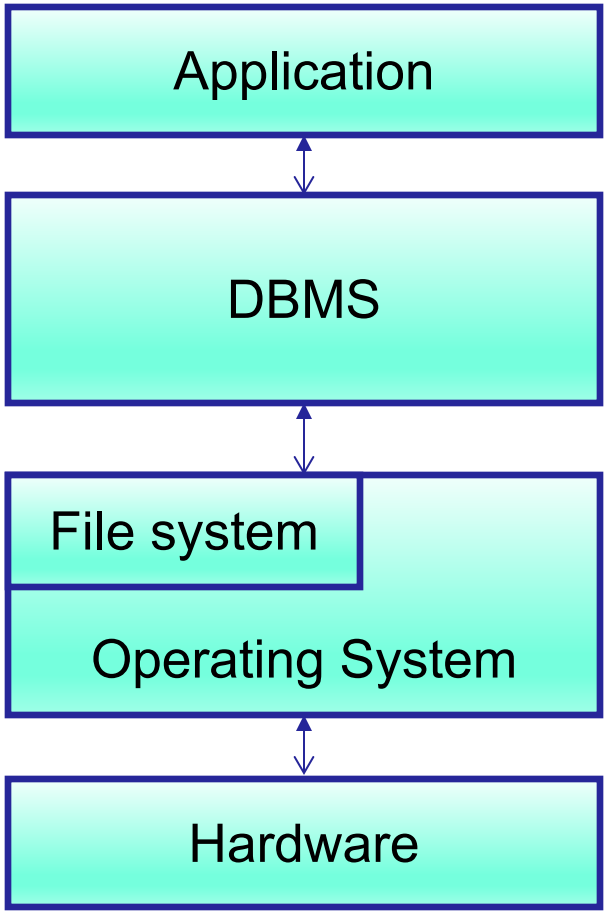
Data Structures vs File Structures

- Both involve:
  - Representation of Data

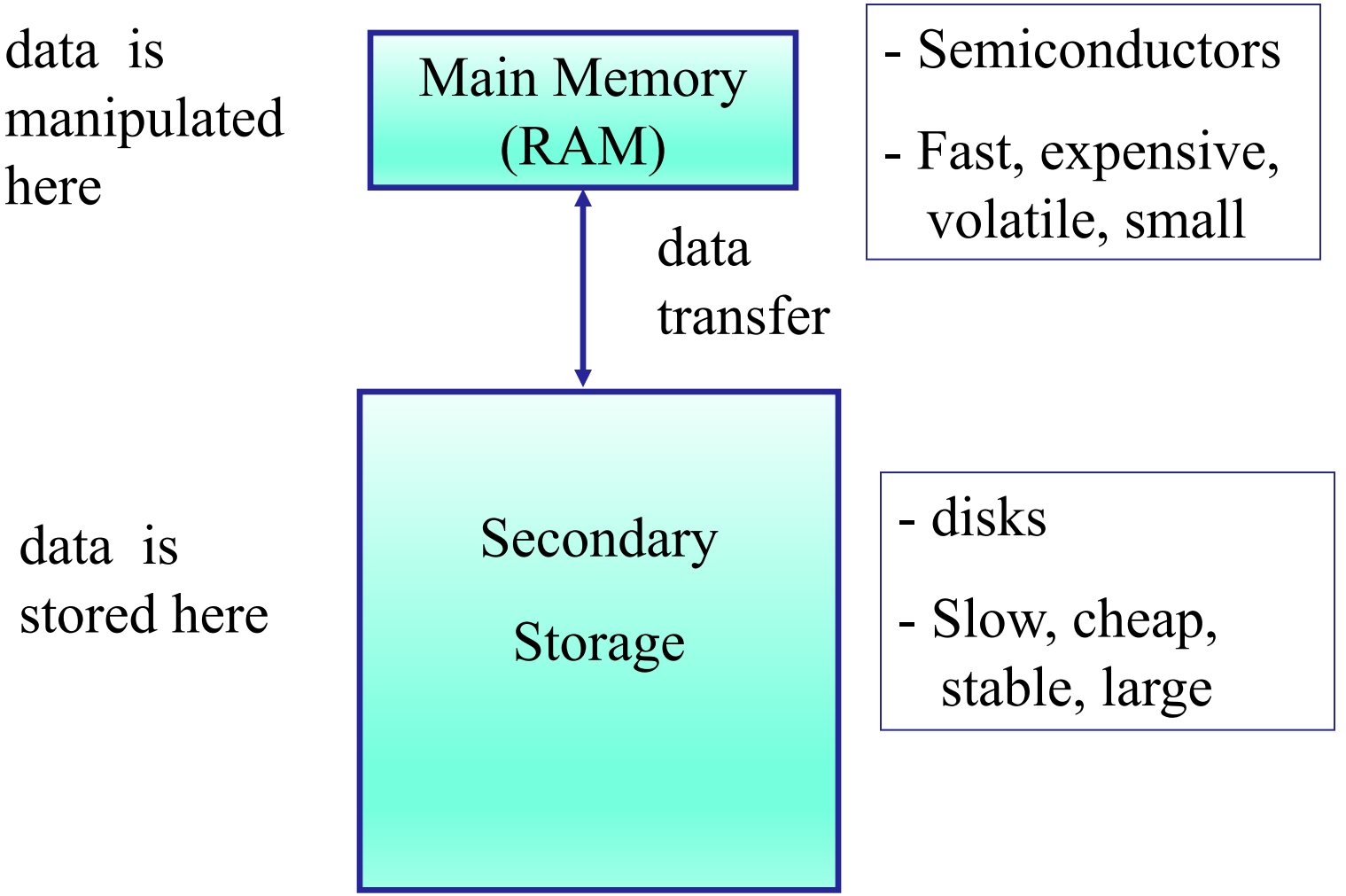
    +
  - Operations for accessing data

- Difference:
  - Data structures: deal with data in main memory
  - File structures: deal with data in secondary storage

Where do File Structures fit in Computing?



Computer Architecture



Advantages

- Main memory is fast
- Secondary storage is big (because it is cheap)
- Secondary storage is stable (non-volatile) i.e. data is not lost during power failures

Disadvantages

- Main memory is small. Many databases are too large to fit in main memory (MM).
- Main memory is volatile, i.e. data is lost during power failures.
- Secondary storage is slow (10,000 times slower than MM)

How fast is main memory?

- Typical time for getting info from:

  Main memory: ~10 nanosec = 10 x 10^{-9} sec

  Magnetic disks:

  ~5-10 milisec (HD) = 10 x 10^{-3} sec

  ~25-100 microsec (SSD) = 100 x 10^{-6} sec
- Keeping same time proportion as above:

  MM: 1 sec

  SSD: 2.7 hours

  HD: 11.57 days

Normal Arrangement

- Secondary storage (SS) provides reliable, long-term storage for large volumes of data
- At any given time, we are usually interested in only a small portion of the data
- This data is loaded temporarily into main memory, where it can be rapidly manipulated and processed.
- As our interests shift, data is transferred automatically between MM and SS, so the data we are focused on is always in MM.

Goal of the file structures

- Minimize the number of trips to the disk in order to get desired information
- Grouping related information so that we are likely to get everything we need with only one trip to the disk.

Database

What is a database ?

Database

What is a database ?

- A collection of files storing related data.
- Models real-world enterprise (such as a university, hospital, library, etc.)

Examples of databases.

- METU's students database, Amazon's products database, THY airline reservation database, Isbank accounts db, Instragram postings db, Walmart payroll database

Database Management System

What is a DBMS ?

- A software package that allows us to store and manage efficiently a large database and allows it to persist over long periods of time.

Examples of DBMSs.

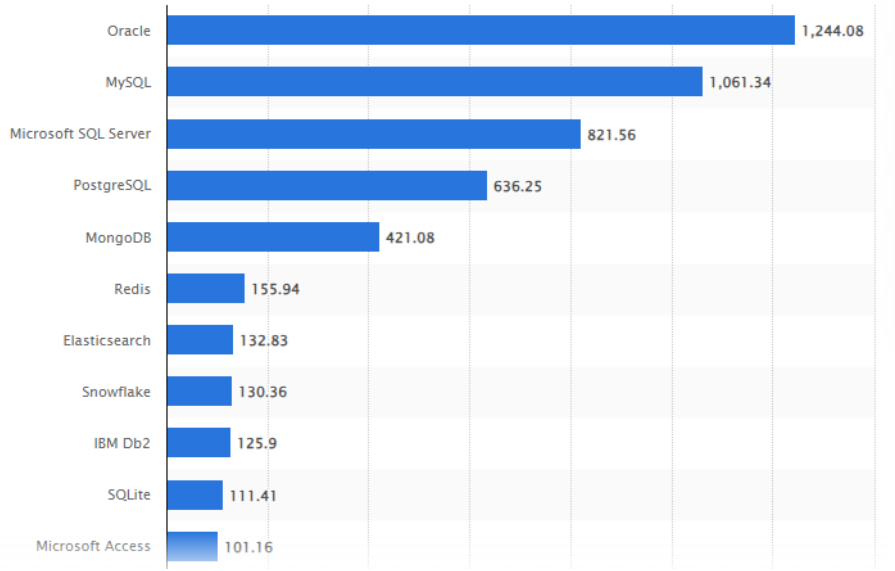
- Oracle, IBM DB2, Microsoft SQL Server, Vertica
- Open source: MySQL (Sun/Oracle), PostgreSQL

Database Management System

The global **enterprise data management market size** was valued at USD 89.34 billion in 2022

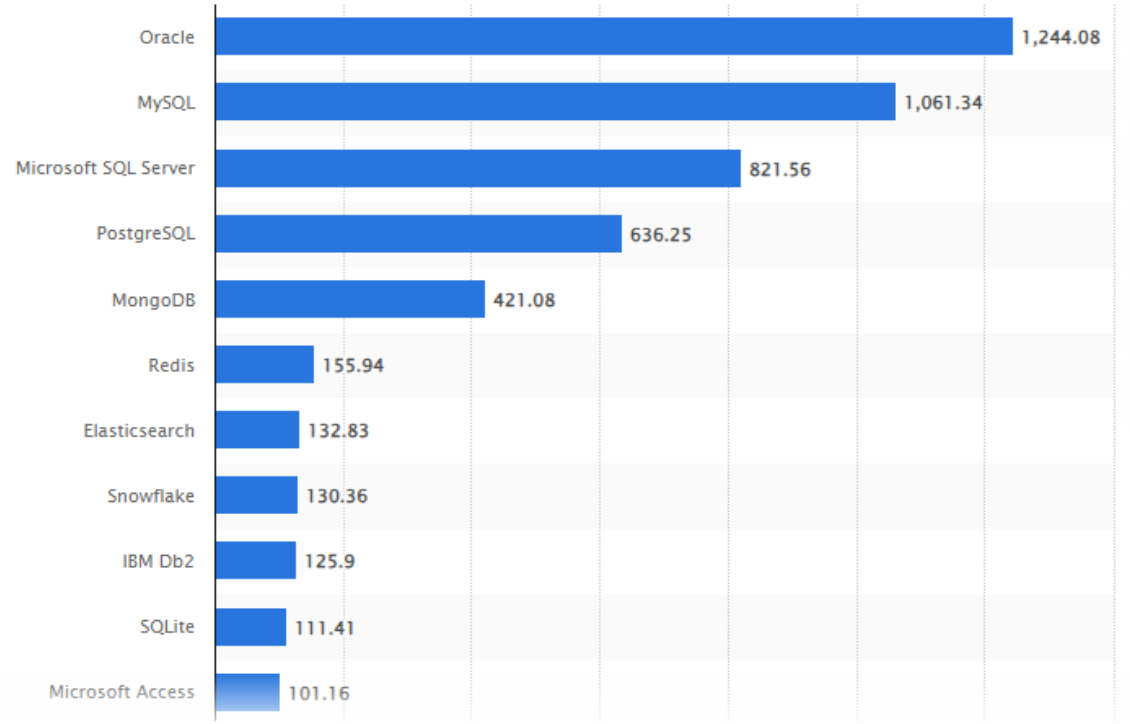
expected to grow at a compound annual growth rate (CAGR) of 12.1% from 2023 to 2030.

Source: grandviewresearch.com



Source: statista.com 2024

Database Management System



We will focus on ***Relational*** **DBMSs**

An Example: Online Bookseller

- What data do we need?
  - Data about books, customers, pending orders, order histories, trends, preferences, etc.
  - Data about sessions (clicks, pages, searches)
  - Note: data must be persistent!
  - Also note that data is large… won't fit all in memory
- What capabilities on the data do we need?
  - Insert/remove books, find books by author/title/etc.,
  - Analyze past order history, recommend books, …
  - Data must be accessed efficiently, by many users
  - Data must be safe from failures and malicious users

Required Data Management Functionality

1. Describe real-world entities in terms of stored data
2. Persistently store large datasets
3. Efficiently query & update
   - Must handle complex questions about data
   - Must handle sophisticated updates
   - Performance matters
4. Change structure (e.g., add attributes)
5. Concurrency control: enable simultaneous updates
6. Crash recovery
7. Security and integrity

DBMS Benefits

- Expensive to implement all these features inside the application.
- DBMS provides these features (and more)
- DBMS simplifies application development.

Key Data Management Concepts

- **Data models**: how to describe real-world data
  - Relational, XML, graph data (RDF), …
- **Schema v.s. data**
- **Declarative query language**
  - Say what you want not how to get it
- **Data independence**
  - Physical independence: Can change how data is stored on disk without maintenance to applications
  - Logical independence: can change schema w/o affecting apps
- **Query optimizer** and compiler
- **Transactions**: isolation and atomicity

Structure of a DBMS

- A typical DBMS has a layered architecture.
- This is one of several possible architectures,
- each system has its own variations.