Introduction to Databases I

Overview

- Why use a Database?
- The file-based approach
- Disadvantages of the file-based approach
- The database approach
- Comparison

Why use a database ? I

- What is a database
  - computerized record-keeping system

- What can you do with a database
  - store data
  - retrieve data
  - add data
  - delete data
  - update data
  - etc. etc.

Why use a database ? II

- What are the key features of a database
  - Storing ?
  - Retrieval ?
  - Access to data ?

Why use a database ? III

- Why not writing file-based solution?
  - Databases are expensive
  - Easy integration into applications
  - Serialization of objects is easy
    - e.g. a hash table
  - Faster
  - No need for learning about databases

File-Based Approach I

- File-based system
  - A collection of application programs that perform services for the end-users such as the production of reports. Each program *defines* and *manages* its own data.

File-Based Approach II

- Decentralized Approach
- Can grow incrementally
- Easy
- Fast
- Enables simple operations

- Problems?

Disadvantages of the File-based approach

- Separation and isolation of data
  - Hard to link data in several files
- Duplication of data
  - Waste & inconsistency
- Data dependence
  - Closely linked to files & serialization
- Incompatible file formats
  - Different language different files
- No standards for queries
  - Define your own queries
- Data processing is more that get and put
  - How to link multiple files?

Summarize disadvantages

- The definition of data is embedded in the application
- No control over access and manipulation of data

Need a Database Management System !

The database approach I

- Database
  - A shared collection of *logically related* data (and a *description* of the data), designed to meet the information needs of an organization.

- Data + Metadata (Catalog)

The database approach II

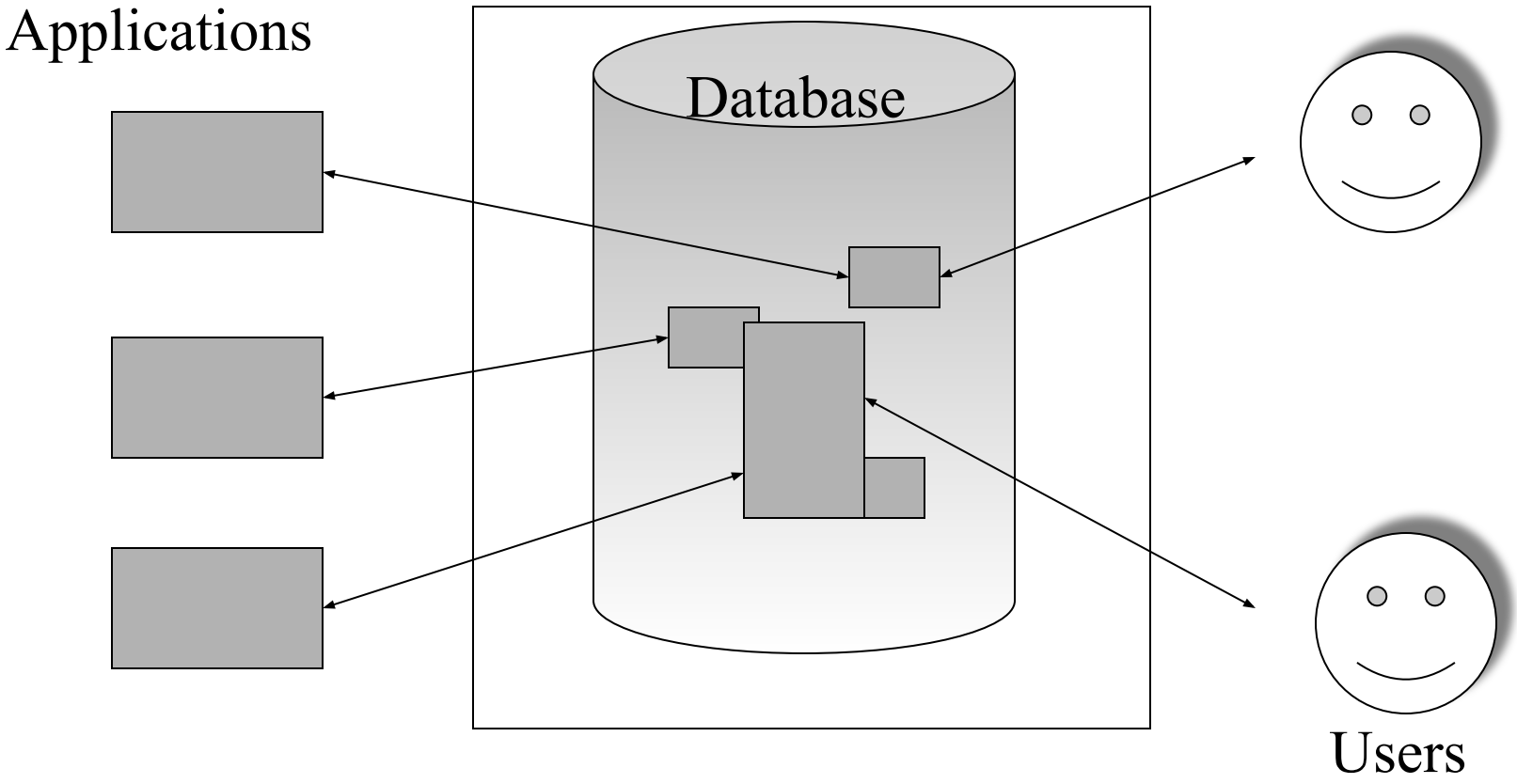
- Data
- Data about data
- Data independence
  - Program
  - Storage
  - Etc.

Database Management System

- DBMS
  - A software that enables the users to define, create and maintain the database and provides controlled access to this database.

- Data definition language (DDL)
- Data manipulation language (DML)

Database management system (DBMS)



Components of a DBMS

- Data
- Hardware & software
- People

Data

- Meta Data
  - data about data
  - types
  - organization
- Data
  - the content
- The DBMS keeps total control over storage and access
- Only queries

Hardware & Software

- Hardware
  - secondary storage volumes
  - processor
  - memory

- Software
  - database manager (DB) / DBMS
    - between files and user/application
    - protecting from storage details
  - utilities
  - application development tools
  - design aids, report writers etc...

People I

- Application programmers
  - write programs that use the DB
  - communication with the DB

People II

- End-User
  - communication via query language e.g. SQL
  - special build-in tools
    - customizable via scripting language

- Database Administrator
  - design of database
  - maintenance of database

Advantages of the database approach

- Control of redundancy
- Data consistency
- More information
- Sharing data
- Data integrity
- Security
- Standards
- Scalability
- Maintenance
- Backup and recovery

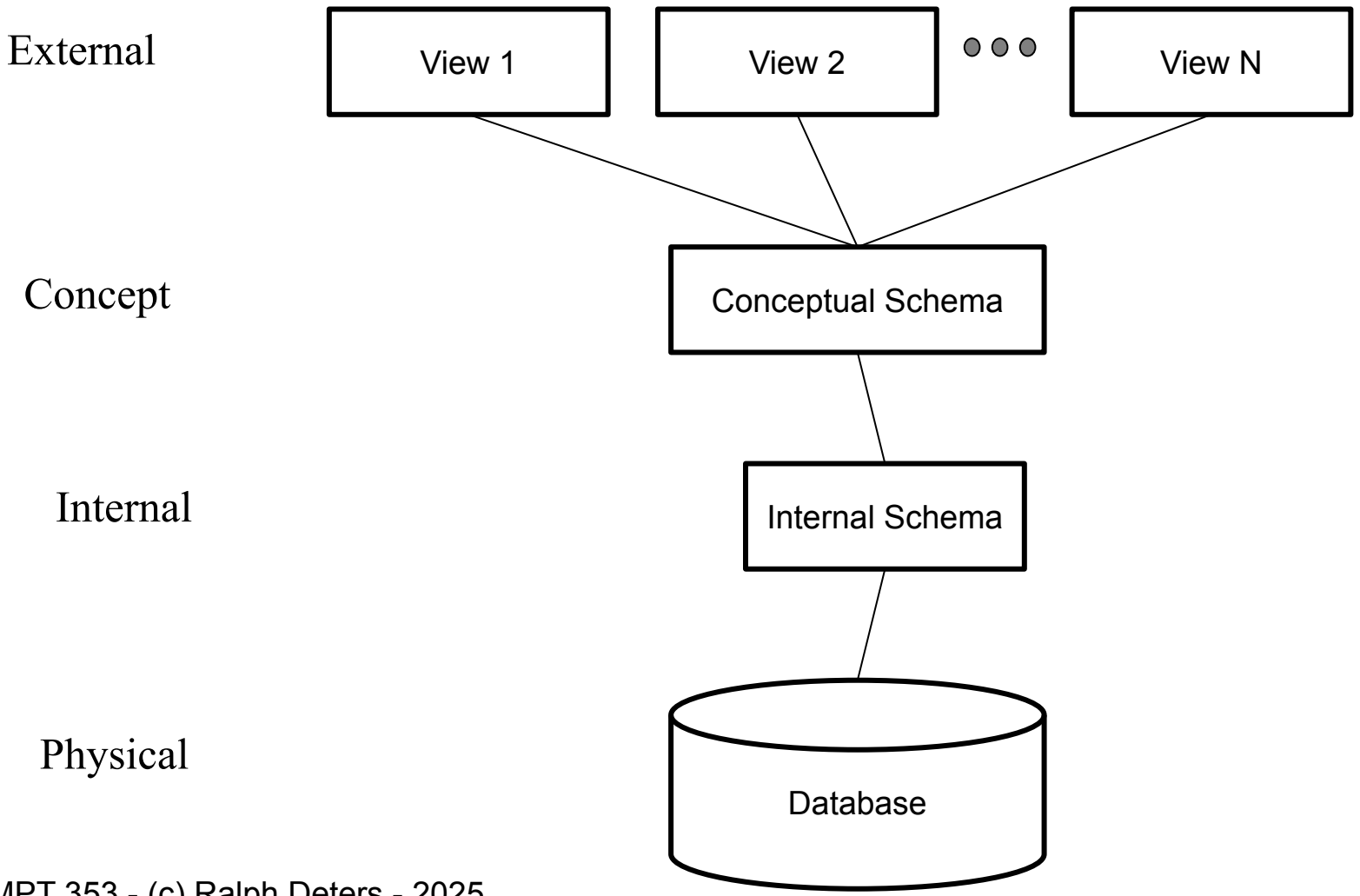
Disadvantages of the database approach

- Complexity
- Size
- Costs
- Performance
- Higher impact of failure

ANSI-SPARC Architecture

- Standards for database systems
  - 1971 DBTG
    - 2 levels
    - System view (schema)
    - User view (user schema)
  - ANSI (1975)
    - 3 levels
    - External, Conceptual, Internal

Three-Level Architecture



3 Levels

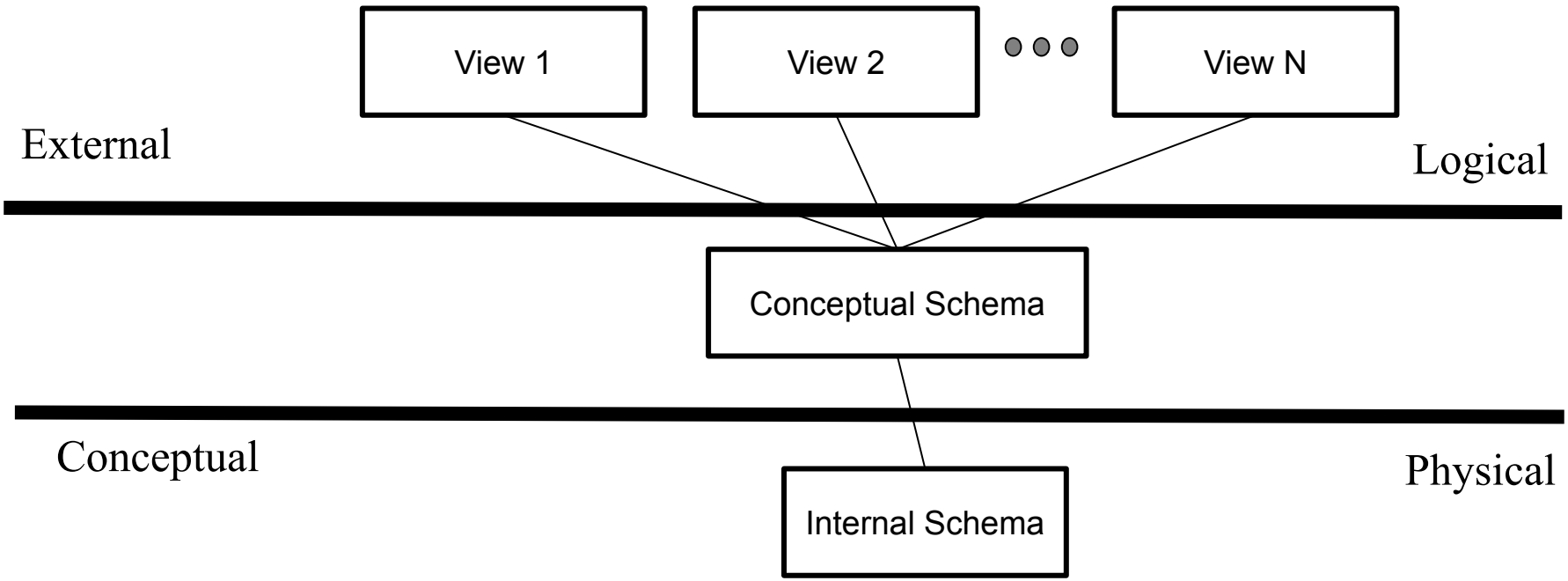
- External level
  - User-view – what should a user see/access
- Conceptual Level
  - Community-view – what data is stored & relationships between data
- Internal Level
  - Physical view – how is the data stored

Mapping & Independence

- Data independence
  - Logical
  - Physical

- Links between the levels
  - 2 mappings
    - External/conceptual mapping
    - Conceptual/internal mapping

Mapping


View 1
View 2
View N
External
Logical
Conceptual Schema
Conceptual
Physical
Internal Schema

Languages

- DDL & DML
  - Data sub languages
    - Embedding into host language
  - Variety of approaches

Data Model

- SQL vs. NoSQL

- Object-based Data Models
- Graph
- Document
- Record-Based Data Models
  - Relational data model
  - Network data model
  - Hierarchical data model

Functions of a DBMS

- Data storage, retrieval, and update
- A user-accessible catalog
- Transaction support
- Concurrency control services
- Recovery services
- Authorizations services
- Support for data communication
- Integrity services
- Services to promote data independence
- Utility services