SLOTS 01

CHAPTER 1

THE WORLDS OF DATABASE SYSTEMS

OBJECTIVES

Understand concepts of:

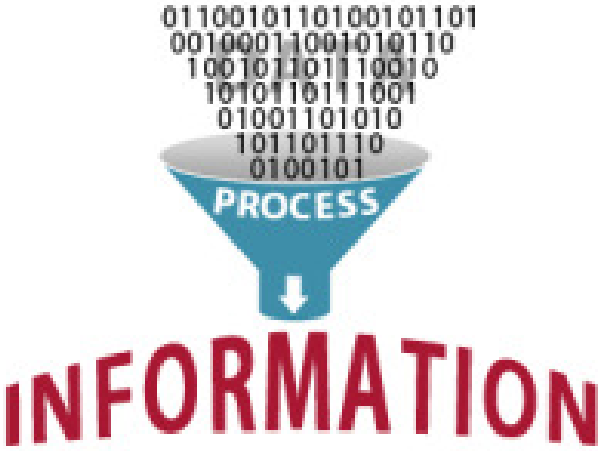
- **Information, Data, Database**
- **Database Management System (DBMS)**
- **Database System**

CONTENT

- The evolution of Database Systems
- Overview of database management systems

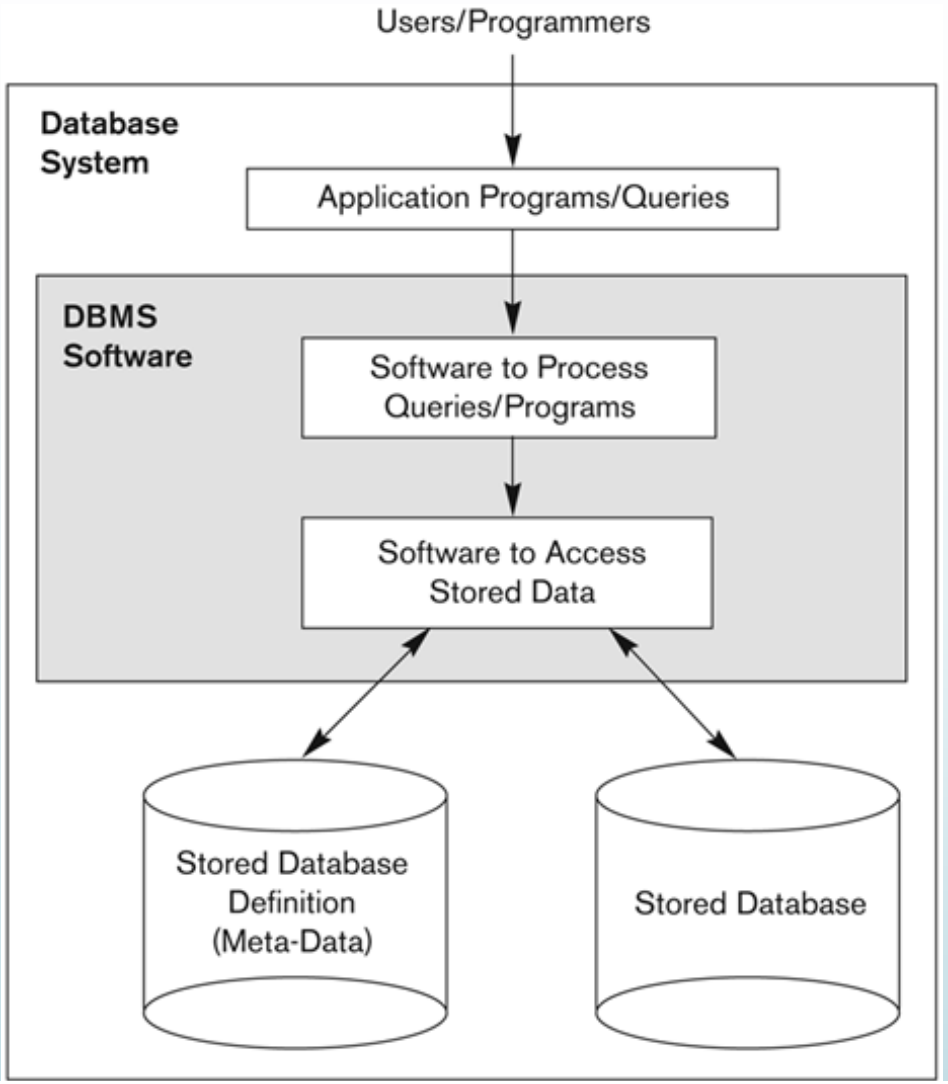
THE EVOLUTION OF DATABASE SYSTEMS

- Data
- Information
- What are the differences between data and information?



1.1 THE EVOLUTION OF DATABASE SYSTEMS

- **Database**
  - A collection of information that exists over a long period of time.
  - A collection of related data.
  - managed by a DBMS
- **Database Management System (DBMS)**
  - A software package/system to facilitate the creation and maintenance of a computerized database
- **Database System**
  - The DBMS software together with the data itself. Sometimes, the applications are also included.



1.1 THE EVOLUTION OF DATABASE SYSTEMS

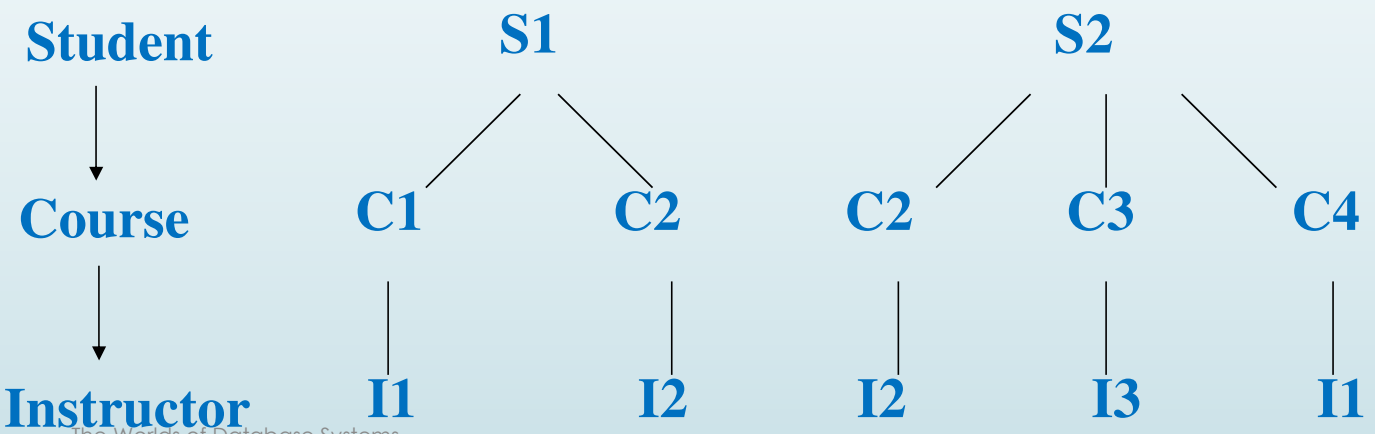
- **The DBMS is expected to**
  1) Allow users to create new databases and specify their schemas
  2) Give users the ability to query the data
  3) Support the storage of very large amounts of data
  4) Enable durability
  5) Control access to data from many users at once

- Early DBMS
  - 1960s, the first DBMS based on file system

Responsibility	Yes/No
(1)	Limited
(2)	Not directly supported
(3)	Yes
(4)	Not always supported
(5)	No

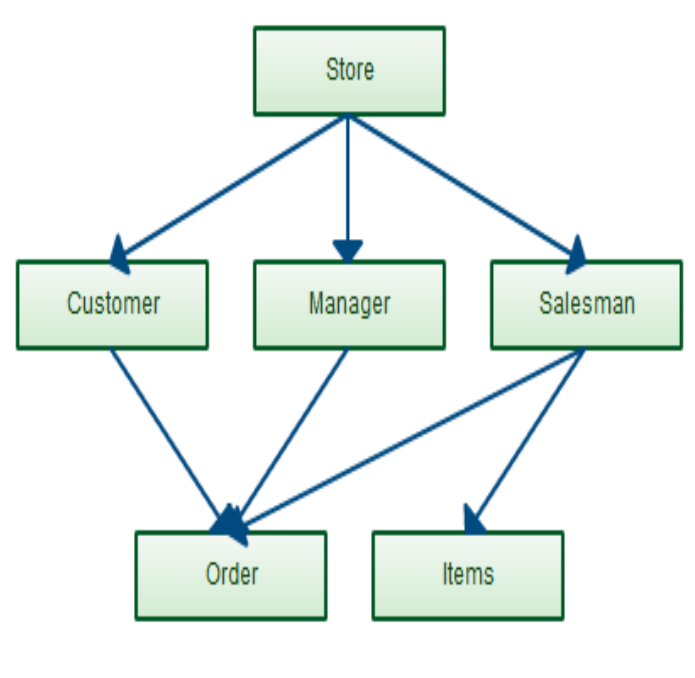
1.1 THE EVOLUTION OF DATABASE SYSTEMS

- **Hierarchical data model (tree-based model)**
  - Was used in early mainframe DBMS
  - The IBM Information Management System (IMS) is example of a hierarchical database system



1.1 THE EVOLUTION OF DATABASE SYSTEMS

- **Network data model (graph-based model)**
  - Charles Bachman invented in the late 1960s
  - standard specification published in 1969 by the Conference on Data Systems Languages (CODASYL) Consortium
  - The network model allows each record to have multiple parent and child records
- → Not support high-level query language



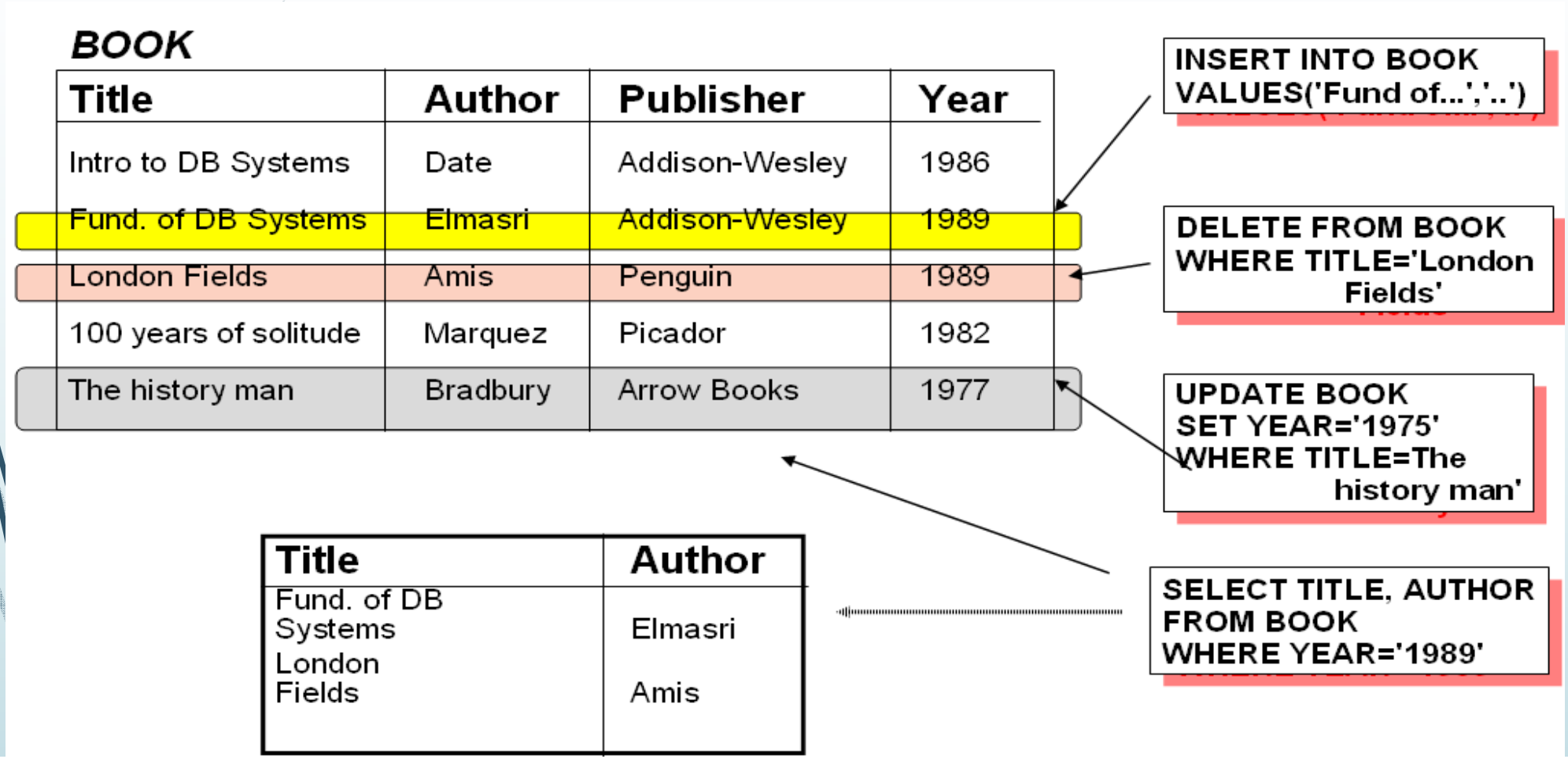
1.1 The Evolution of Database Systems

Relational Database Systems

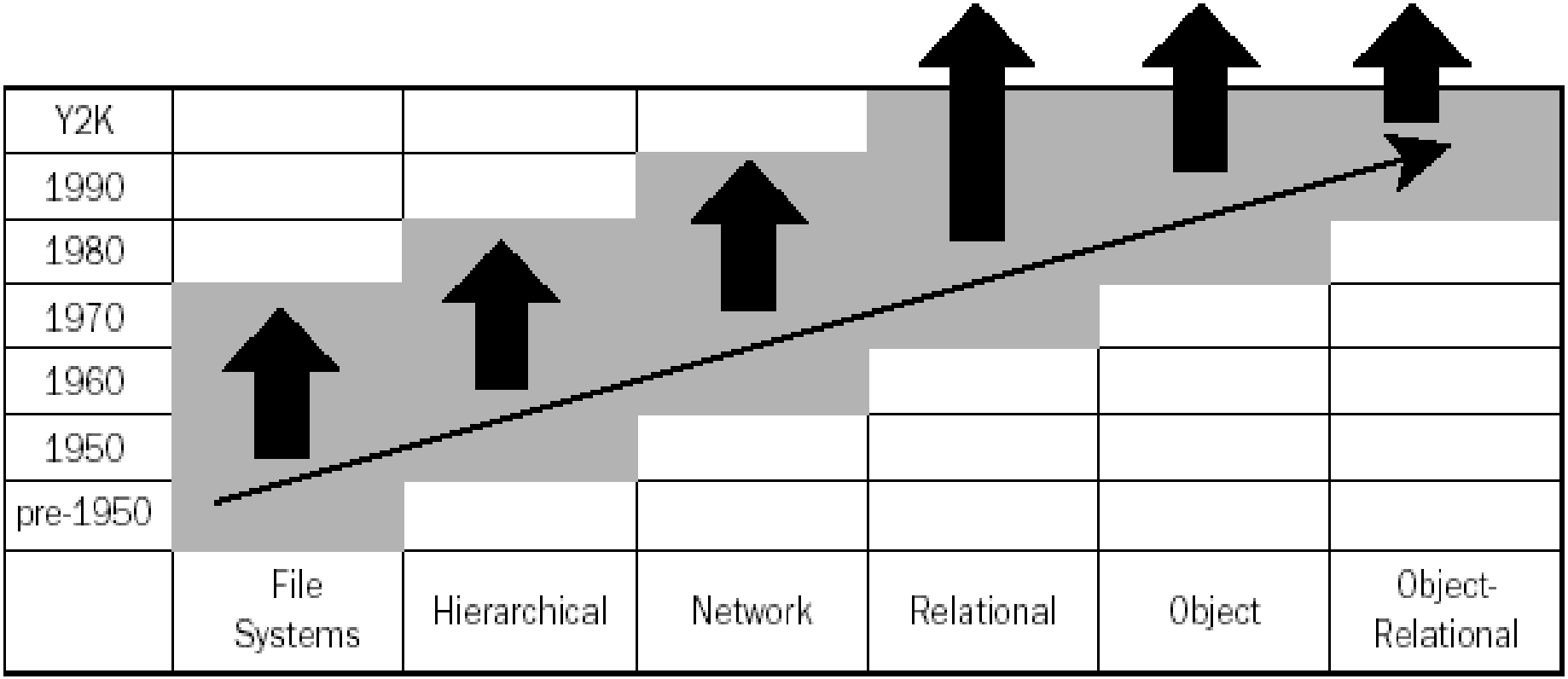
- 1970s, Edgar Frank "Ted" Codd defined relational model based on relations (*)
  - Revolutionary idea of DBMS activity
  - at IBM (System R, DB2)
  - at Universities like Berkeley (Ingres)
- SQL, the most important query language, was developed by IBM in 1974
- 1979, Oracle v.2, the first commercial RDBMS product using SQL

1.1 THE EVOLUTION OF DATABASE SYSTEMS

- Book relation example



1.1 THE EVOLUTION OF DATABASE SYSTEMS



The evolution of database modeling techniques.

→ 2000s-now: NoSQL , newSQL

1.1 The Evolution of Database Systems

- **Smaller and Smaller Systems**
  - Originally, DBMS's were large, expensive software running on large computers
  - Today, DBMS can run on PC, Mobile, …

  ⇒DB systems based on the **relational model** are available for even very small machines
- **Bigger and Bigger Systems**
  - Size of data has been increasingly continuously
  - Many databases store petabytes and serve it all to users

1.1 The Evolution of Database Systems

Information Integration

- Join the information contained in many related databases into a whole
  - Example: a large company has many divisions, each division have built its own database of products and employees on different DBMS's and different structures
  - How we join these databases without any matters
- Need to build structures on top of existing databases, with the goal of integrating the information distributed among them

1.1 The Evolution of Database Systems

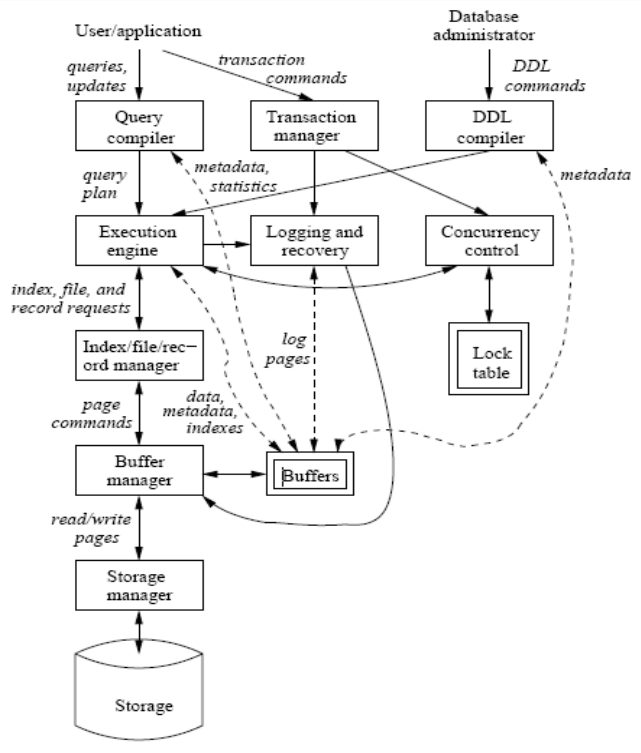
- **Information Integration (con't.)**
  - Two popular approaches
    - Creation of **data warehouses**, where information from many databases is copied periodically, with the appropriate translation, to a central database
    - Implementation of a middleware (mediator) that support an integrated model of the data of the various databases, while translating between this model and the actual models used by each database

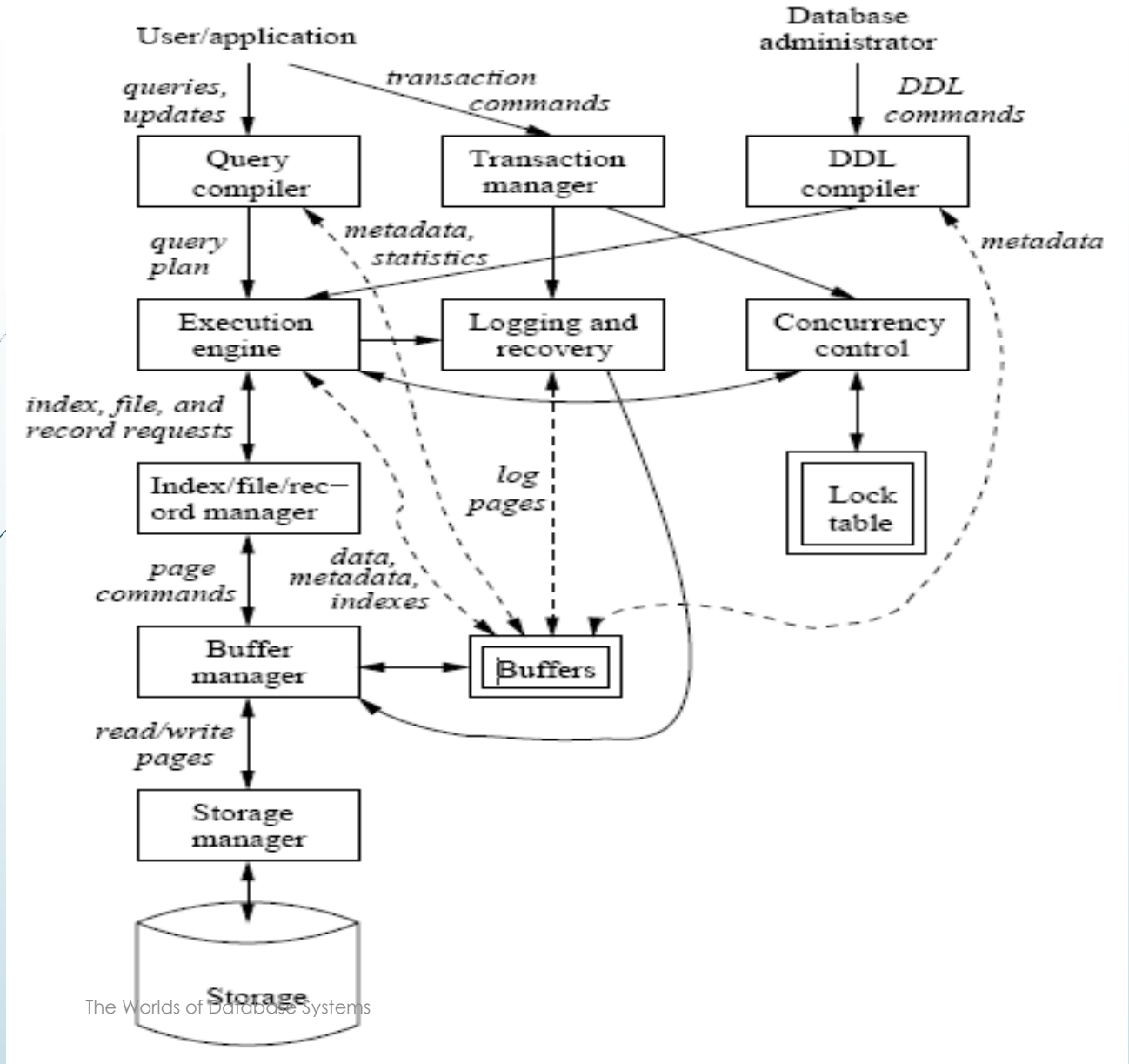
1.2 OVERVIEW OF DBMS

- **Database Management System**
  - DBMS components
  - Database Users
  - Database language
  - Relational databases

1.2 OVERVIEW OF DBMS

- **DBMS components**
  - **Single box**: system component
  - **Double box**: memory data structure
  - **Solid line**: control & data flow
  - **Dashed line**: data flow only




User/application
Database administrator
queries, updates
transaction commands
DDL commands
Query compiler
Transaction manager
DDL compiler
query plan
metadata, statistics
metadata
Execution engine
Logging and recovery
Concurrency control
index, file, and record requests
Index/file/record manager
log pages
Lock table
page commands
data, metadata, indexes
Buffer manager
Buffers
read/write pages
Storage manager
Storage

1.2 OVERVIEW OF DBMS

- **Database Users**
  - **Database Administrators**, authorize access to database, coordinate, monitor its use, acquiring software, and hardware resources, …
  - **Database Designers**, define the content, the structure, the constraints, and functions or transactions against the database
  - **Database End users**, use data for queries, reports and some of them actually update the database content.

1.2 OVERVIEW OF DBMS

- **DDL - Data Definition Language Commands**
  - DBA needs special authority to execute schema-altering commands
  - Schema-altering commands are known as DDL commands, and used for defining data structure
  - These commands are parsed by a DDL compiler and passed to the execution engine, then goes through the index/file/record manager to alter the metadata (schema information for the database)
  - Examples: **CREATE, ALTER, DROP**

1.2 OVERVIEW OF DBMS

- **DML - Data Manipulation Language Commands**
  - Are used by computer programs or DB users to **retrieve, insert, delete,** and **update data**
  - Not affect the schema of the database, but **affect the content** of the database or **extract data** from database
  - DML has two separate subsystems
    - Answering the query
    - Transaction processing

1.2 OVERVIEW OF DBMS

1. **Answering the query**
   - Query is parsed and optimized by the *query compiler* which the result is *query plan*
   - *Query plan* is passed to execution engine to execute
2. **Transaction processing (will be discussed in the next chapters)**
   - Transaction is a group of some database operations.
   - Transaction is processed by *transaction manager*.

THE TRENDS OF DB DESIGN AND DBMS

- Non relational databases (NoSQL)
  - MongoDB
  - Redis
- Multi-model databases
  - Oracle database
  - Arango DB