1. Introduction to Databases and DBMS
DATA AND DATABASES

- A **database** (DB) is a set of related data and the way it is organized.

- A **database management system** (DBMS) is a type of software specialized in managing and administering databases.

1. Introduction to Databases and DBMS

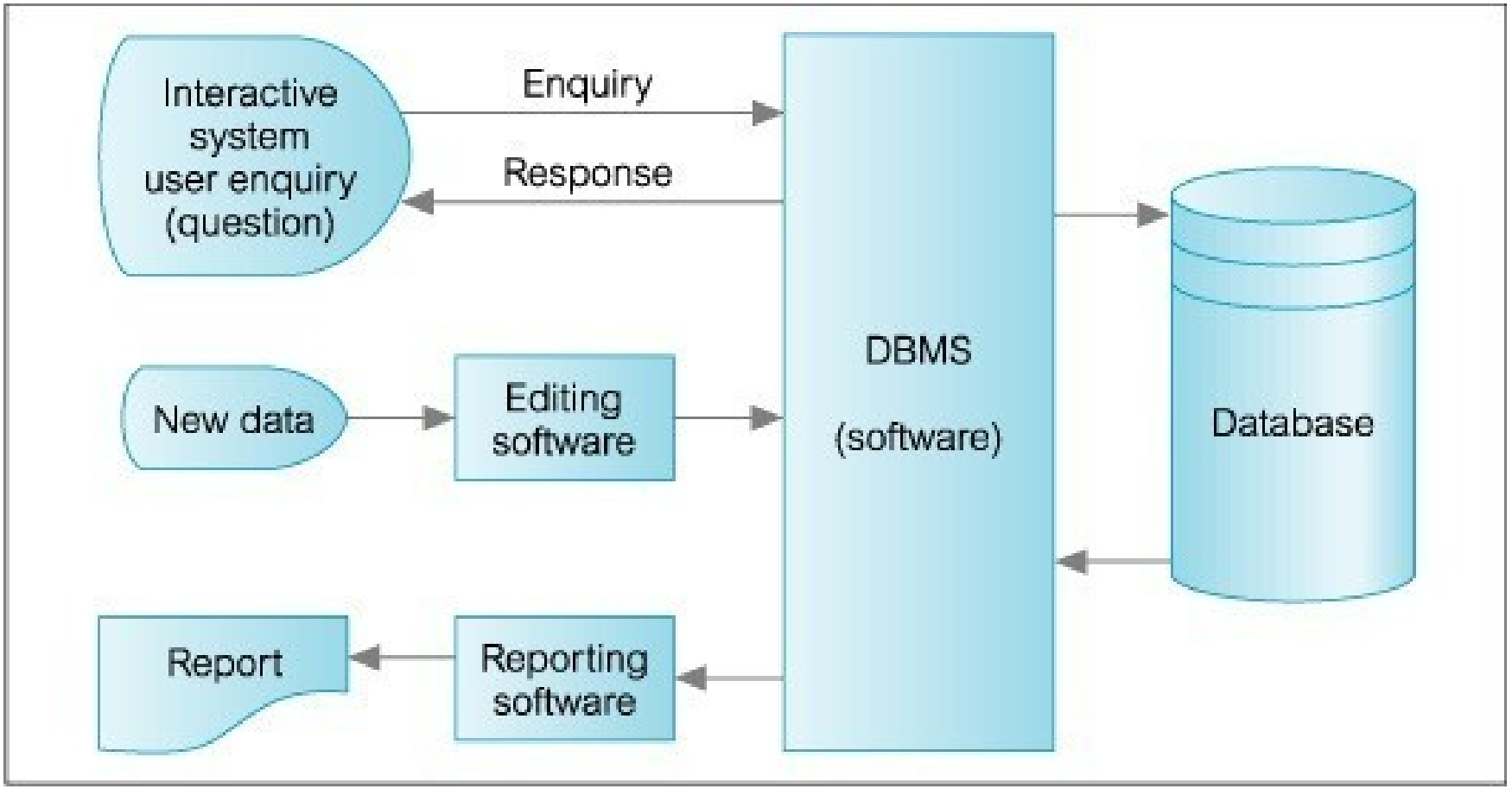
DBMS

definition:

The database management system (DBMS) is a program that allows users to **define**, **create** and **maintain** the database and provides **controlled** access to it. This tool provides an **interface** between the user and the databases.

1. Introduction to Databases and DBMS

DBMS



1. Introduction to Databases and DBMS
DBMS

Hardware evolution → provides → DBMS evolution

A brief timeline:

1950s

Sequential data processing

- Flat file systems
- Batch processing
- Magnetic tapes for storage
- Punched card for input

1960s-1970s

Centralized systems

- Mainframe with network of non-intelligent terminals
- First transactional applications
- Programs closely linked to physical design

1980s

Relational DBMS

- Major commercialization of relational DBMS
- Independent programs from physical design
- SQL becomes industrial standard
- Personal DB for PC

1990s

- Distributed DB
- Client-server arquitecture
- 4th generation language

- Distributing computing
- Shared everything architectures emerged as techniques to increase database performance

present+

- Object Orientation
- Internet
- Multimedia elements

- Post relational DBMS (OLAP, multivalue, ...)
- Internet based applications
- Data warehouse
- Relational DBMS with XML support
- Management systems for XML native database

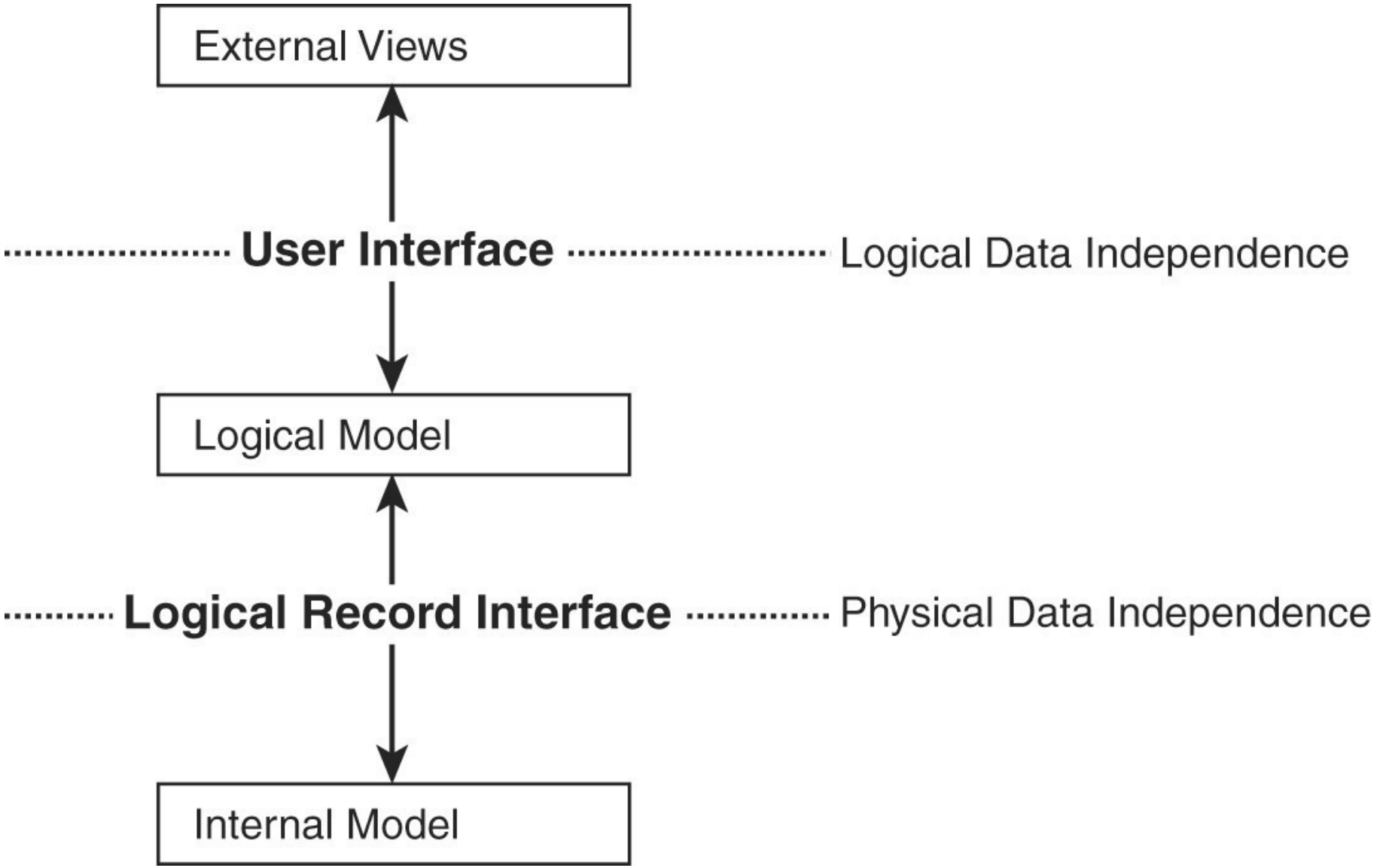
1. Introduction to Databases and DBMS
DBMS

Objectives of a DBMS:

- No predefined queries of any complexity
- Ensure three abstraction levels: physical, logical and external.
- Guarantee physical and logical data independence.
- Avoid or solve problems due to data redundancy.
- Data consistency (restrictions, backups, ...).
- Security of access to data (access rights, authorization levels -global, entity, attribute, operation types-, encriptation, ...)
- Allow user concurrency using 2 techniques:
  - Transactions
  - Locking

1. Introduction to Databases and DBMS

DBMS



1. Introduction to Databases and DBMS
DBMS

DBMS languages:

- **DDL** (Data Definition Language). Define and describe the objects of the DB (structure, relationships and constraints).
- **DML** (Data Manipulation Language). Query, insert, delete or update data on DB. Types:
  - Procedural (user specifies what data is required and how to get those data)
  - Declarative (user specifies what data is required without specifying how to get those data)

Procedural are more complex, but more efficient than declarative

1. Introduction to Databases and DBMS
DBMS

SQL (Structured Query Language) includes instructions for all these types of languages and due to its simplicity and power has become the standard language for relational DBMS.

Access to DB also from:

- 4th generation languages
- Visual interfaces
- Programming languages (through ODBC and JDBC)

1. Introduction to Databases and DBMS
DBMS

Users of a DBMS:

- **External users**. They work on the external level through views or portions of the DB. They are clients of databases. They use them without knowing at all its operation and organization. They are people with little or no computer skills.
- **Developers** of customized applications for external users.
- **Sophisticated users** working on DB through visual interfaces:
  - Writing ad hoc queries to answer their own needs.
  - Running ad hoc reports for executives / managers.
- **Administrators**. They take care of the physical design, implementation and maintenance of the DB.

1. Introduction to Databases and DBMS
DBMS

Administrator functions:

- Starting up and shutting down the databases.
- Schema definition.
- Granting of authorization for using resources.
- Backing up the databases periodically and restoring the databases as needed.
- Manage available space on hard drives.
- Take care of data consistency.
- Tuning the system performance changing physical design as needed.
- Monitoring the use of resources.
- Teach programmers and users about using DB.
- Considering software and hardware upgrades.
- Solve emergencies.

1. Introduction to Databases and DBMS
DBMS

A DBMS is partitioned into modules that deal with each of the responsibilities of the overall system. The **functional components** of a DBMS can be broadly divided into:

- the Storage Manager:

PROBLEM: movement of data to and from disk is slow relative to the speed of the CPU → **SOLUTION**: minimize the need to move data between disk and main memory

- the Query Processor
  - translate updates and queries written in a nonprocedural language, at the logical level, into an efficient sequence of operations at the physical level

1. Introduction to Databases and DBMS
DBMS

Storage Manager is responsible for storing, retrieving and updating data in the database. Componets:

- Authorization and integrity manager: integrity constraints and authority of users to access data.
- Transaction manager: ensures the database remains in a consistent state.
- File Manager: allocation of space on disk storage and the data structures.
- Buffer manager: enables the database to handle data sizes that are much larger than the size of main memory.

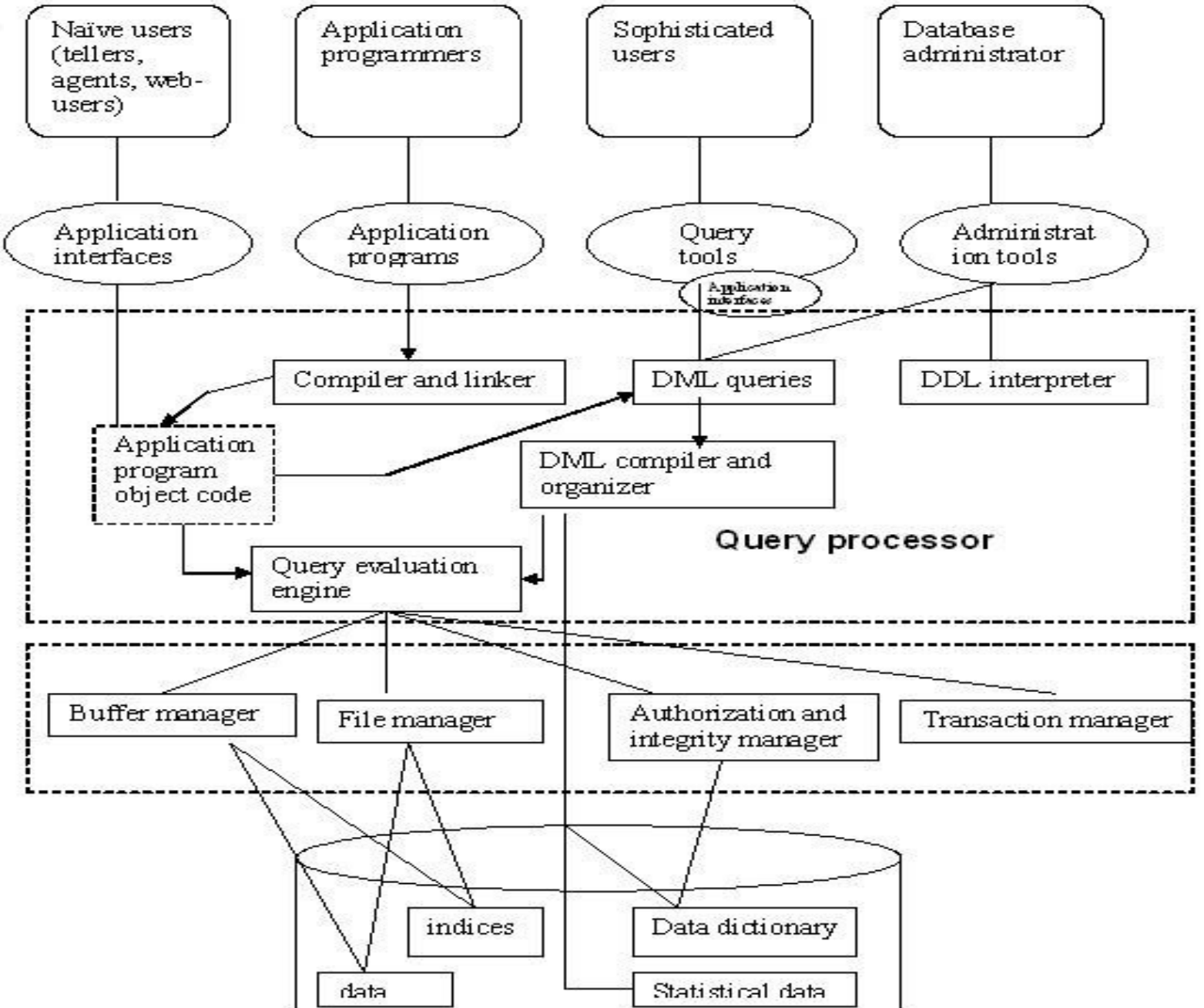
1. Introduction to Databases and DBMS
DBMS

Query Processor components:

- DDL interpreter, which interprets DDL statements and records the definitions in the data dictionary.
- DML compiler, which translates DML statements in a query language into an evaluation plan consisting of low level instructions that the query evaluation engine understands.
- Query evaluation engine, which executes low level instructions generated by the DML compiler.

1. Introduction to Databases and DBMS

DBMS



1. Introduction to Databases and DBMS
DBMS

A **data dictionary**, or metadata repository, is a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format." Contains:

- The definitions of all schema objects in the database (tables, views, indexes, clusters, synonyms, sequences, procedures, functions, packages, triggers, and so on)
- How much space has been allocated for, and is currently used by, the schema objects
- Default values for columns
- Integrity constraint information
- The names of users
- Privileges and roles each user has been granted
- Auditing information (ex. who has accessed or updated various schema objects)
- Other general database information

2. Database models

Data models are a set of tools to describe logical data, their relationships, their meaning and constraints we can apply to ensure consistency.

These tools are:

- Data structure
- Integrity rules.
- Operations performed on the data.

2. Database models
THE MOST COMMON DATABASE MODELS

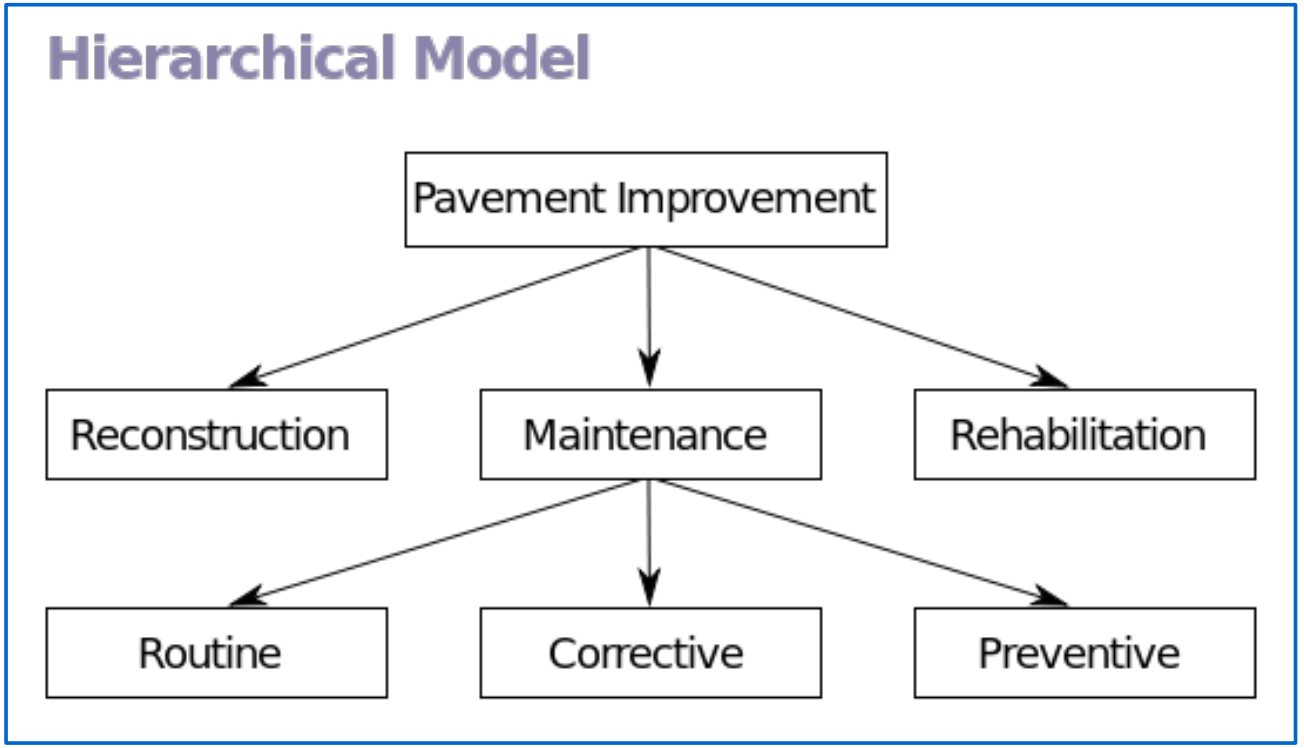
The data models used by the most popular DBMS products are:

- Hierarchical
- Network
- Relational
- Relational with objects
- Object-oriented

2. Database models
THE MOST COMMON DATABASE MODELS

A **hierarchical** database model is a data model in which the data is organized into a tree-like structure. The data is stored as records which are connected to one another through links.

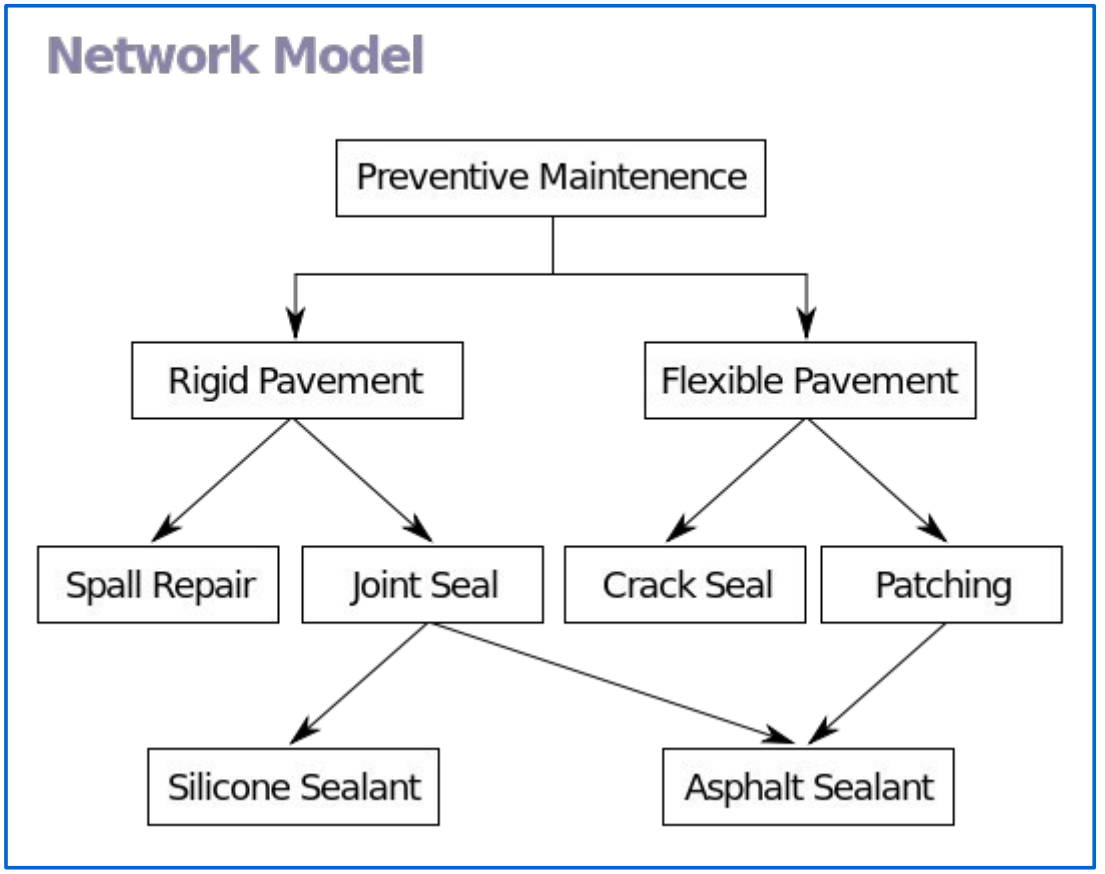
This model mandates that each child record has only one parent, whereas each parent record can have one or more child records.



2. Database models
THE MOST COMMON DATABASE MODELS

The **network** model expands upon the hierarchical structure, allowing many-to-many relationships in a tree-like structure that allows multiple parents.

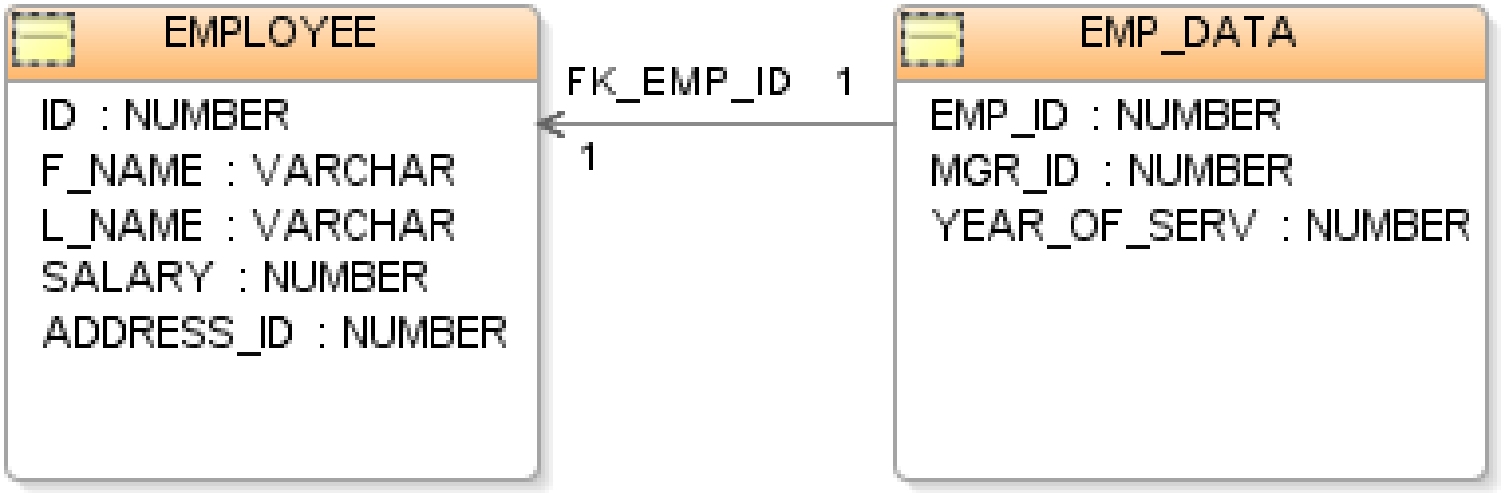
The network model allows each record to have multiple parent and child records, forming a generalized graph structure.



2. Database models
THE MOST COMMON DATABASE MODELS

The **relational model** was introduced as a way to make DBMS more independent of any particular application. It is a mathematical model defined in terms of predicate logic and set theory, and systems implementing it have been used by mainframe, midrange and microcomputer systems.

All data is represented in terms of tuples, grouped into relations



2. Database models
THE MOST COMMON DATABASE MODELS

- Object-oriented models define a database as a collection of objects with features and methods.
- The relatively new object-relational database model is the simple relational database model extended by some basic object-oriented concepts. These allow us to work with the widely know relational database model but also have some advantages of the object-oriented model without its complexity (supports extension of the data model with custom data-types and methods).
- The multidimensional model is a specialized adaptation of the relational model used to represent data in data warehouses in a way that data can be easily summarized using online analytical processing, or OLAP queries.
  - MultiValue database is a type of multidimensional database. They have features that support and encourage the use of attributes which can take a list of values, rather than all attributes being single-valued. They are often categorized within the category of post-relational databases.