Chapter 02: Data Model

1. Introduction:

The data model is defined as a set of concepts and rules for composing these concepts to describe data. There are actually four main data models: the hierarchical model, the network model, Entity-relationship model and the relational model.

2. Hierarchical model:

The ***hierarchical database model*** *organizes data in a* ***tree-like structure*** *where:*

Data is stored as ***records (nodes)***

Each parent can have ***many children***

Each child has ***only one parent***

Relationships are ***1-to-many***

It is similar to a ***family tree*** *or* ***organizational chart****.*

This model was widely used in early database systems.

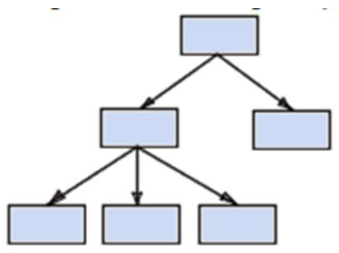

Figure 2.1: Hierarchical model

Example: Company Organization Database

Entities

Company (Root)

Departments (Financial department, Human resource department)

Employees

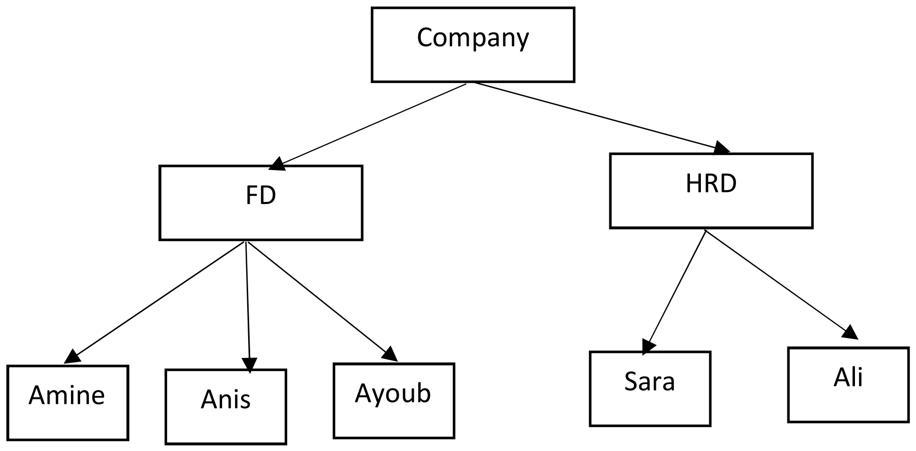

Figure 2.2: Hierarchical model of company organization database.

☺ ***Advantages***

✓*Simple to understand*
✓*Fast for hierarchical queries*
✓*Good performance for predictable relationships*
✓*Enforces parent-child integrity*

⚠ ***Limitations***

✗ *A child cannot have multiple parents*
✗ *Difficult to modify structure*
✗ *Poor support for many-to-many relationships*
✗ *Requires traversal from root each time*

These limitations led to the evolution of ***network*** *and later* ***relational models****.*

3. Network model:

The basic elements of this model are the concepts of sets of entities (logical records) and associations between sets of entities (links).

Logical Record and Link

The concept of a logical record is not specific to DBMSs; it is found in most file management systems and programming languages. It allows grouping elementary data into logical records.
Example:

student = record

Student_ID: integer, key

Name: character(10)

Address: character(40)

End

The second fundamental element in the network model is the link. A link can be defined as the representation of an association; the association is an abstract perception of reality, while the link is its materialization.

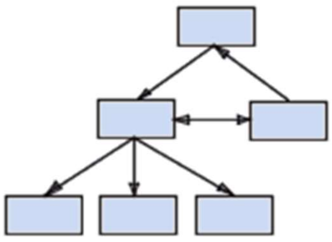

Figure 2.3: network model

Example:

As an example, we will present the diagram of a database of drug users composed of the following records:

-TOBACCO, composed of the following elementary data: Tobacco Number and Tobacco Name

-SMOKER, composed of the data: Smoker Number, Last Name, and First Name

-ABUSE, describing, for each tobacco product, the quantity smoked by a smoker

-PRODUCER, defining, for each tobacco product, the name of the producer

-ORDERS, specifying the tobacco orders placed by smokers with producers

The links between these different records are:

-PRODUCTION, which associates a producer with the tobacco products produced

-SALE, which associates a tobacco product with the corresponding orders

-PURCHASE, which associates a smoker with their orders

-SMOKING, which associates a smoker with the quantity of tobacco smoked

-CONSUMPTION, which associates a tobacco product with all the quantities smoked by the different smokers.

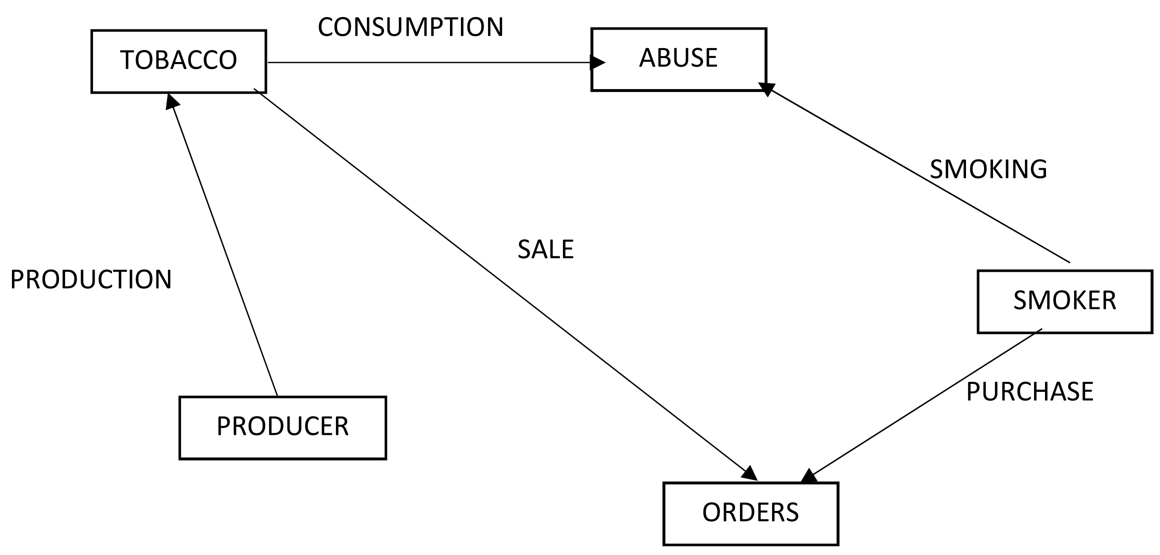

Figure 2.4: Diagram of a database of drug users.

☺ ***Advantages***

✓*Supports* ***many-to-many relationships***
✓*More* ***flexible*** *than the hierarchical model*
✓***Fast data access*** *using pointers*
✓*Reduces* ***data redundancy***
✓*Maintains* ***strong data integrity***

⚠ *Limitations*
✗ ***Complex design and navigation***
✗ *Hard for users to understand*
✗ ***Structural dependence*** *— changes are costly*
✗ *No high-level query language (navigational access)*
✗ ***Difficult and expensive to maintain***

4. Entity Relationship model

1. Definition 2.1:

- ***EA in French**, **ER in English** (Entity-Relationship): A formalism adopted by ISO to describe the conceptual aspect of data using entities and relationships.*
- *The **Entity-Relationship (E/R) model** is a **graphical formalism** for data modeling.*

Success of the Entity-Relationship Model is due to:

- ***Graphical language** for better visualization and understanding.*
- ***Simple concepts**: entities and relationships.*
- ***Grouping of similar items** into classes: **entity classes** and **relationship classes**.*

2. Basic Concepts:

2.1 Concept of Entity:

Entities are objects that can be distinctly and unambiguously identified.
***Examples**: A student, a teacher, a course.*

*Each entity has a set of **attributes**. An **attribute** is a property associated with an entity. In other words, an attribute is a fundamental piece of data that we observe about an entity.*

***Example**: The entity student has the following attributes: student ID, name, first name, address, degree.*

- *A specific student: **02030005, Benali, Mohamed, Cité Daksi Constantine, Computer Science**, is an **instance** or **occurrence** of the entity student.*

2.2 Primary Key:

- *A **key** is an attribute or a group of attributes whose value must uniquely identify an entity.*
  ***Example**: For the entity student, the primary key is the student ID (number-student).*
- *With a **key**, each occurrence must be uniquely and unambiguously identifiable to distinguish it from all others.*
- *The **primary key**, also called an **identifier**, is a property or a group of properties whose value uniquely identifies an entity without ambiguity.*

***Note**: Declaring an attribute as a key is a decision made by the database designer.*
Conclusion:

*For an entity type **E** with a set of attributes **A**, a **primary key** of **E** is a minimal subset of **A** that uniquely identifies any instance of the entity among all other occurrences of the entity **E**.*

Remark:

*It is possible to have multiple keys for the same set of entities. In such cases, one is choosen as the **primary key**, and the others are designated as **secondary keys**. The choice of the primary key is crucial for the quality of the database schema.*

Characteristics of a good primary key:

✍ *Its value is **known for every entity**.*

✍ *It should **never require modification**.*

✍ *For performance reasons, its **storage size** should be as small as possible.*

Graphical Representation:

- *An **entity** is always represented by a **rectangle**.*
- *The **primary key** is always **underlined** in the graphical model.*

For example:
If the entity is Student, and the primary key is Student ID, it would be represented as:

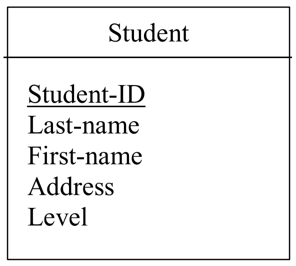

Figure 2.5: graphical representation of entity

(Student-ID is underlined to indicate it as the primary key).

2.3 Concept of Association:

*An **association** is a link between two or more entities.*

- *It defines a **relationship** that connects entities within a model.*
- *An association can also have its own **specific attributes** that describe the nature of the relationship.*

Example:

In a university database:

- *An association might link the entity Student with the entity Course through an association called Enrollment.*
- *The Enrollment association could have attributes like Enrollment Date.*

Graphical representation: *associations are often represented by* ***ellipses*** *connecting the* **related entities.**

2.4 Cardinalities:

The ***cardinality*** *of an association for a constituent entity is defined by a* ***minimum bound*** *and a* ***maximum bound***:

- ***Minimum cardinality***:
  The minimum number of times an occurrence of the entity participates in occurrences of the association, usually ***0*** *or* ***1***.
- ***Maximum cardinality***:
  The maximum number of times an occurrence of the entity participates in occurrences of the association, usually ***1*** *or* ***n***.

Examples of Cardinality:

1. ***1,1***: *Mandatory and unique participation (e.g., a country has exactly one capital).*
2. ***0,n***: *Optional and multiple participation (e.g., a customer can place zero or multiple orders).*
3. ***1,n***: *Mandatory and multiple participation (e.g., an employee works on at least one project but can work on many).*

In graphical representation:

- *Cardinalities are often placed near the lines connecting entities and associations. For example:*

- *A **1,n** relationship between Student and Course (through Enrollment) means a student can enroll in one or many courses, but each course must have at least one student.*

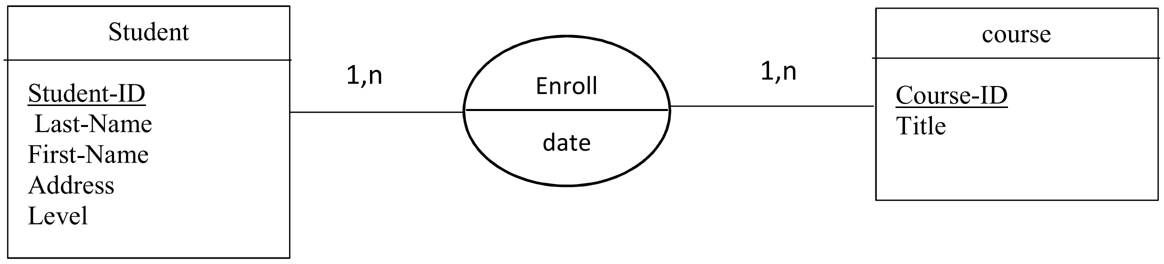

Figure 2.6: example of cardinality

1. Weak Entities:

*Until now, we have considered entities that are independent of each other. Each entity has its own identifier and can be treated separately. However, there are cases where an entity can only exist in close association with another, and its identity is dependent on that other entity. Such an entity is referred to as a **weak entity**.*

3.1 Example of a Weak Entity:

*Consider a **cinema** and its **rooms**. We can think of each room as an entity with attributes like **capacity**, **room number**, etc. However, it is difficult to imagine representing a room without associating it with the cinema it belongs to. It is at the **cinema level** that we find general information such as the **cinema name**, **address**, or **phone number**.*

*In this case, the **room** (weak entity) depends on the **cinema** for its identification, and its existence is tied to the cinema. It is more natural to number rooms with an internal number specific to each cinema. In this case, the **primary key** of a **weak entity** like a room is composed of **two parts**:*

1. ***Cinema Key (name and address)**: This indicates which cinema the room belongs to.*
2. ***Room Number**: This is the number of the room within that cinema.*

*Together, these two parts form the **primary key** for the room entity.*

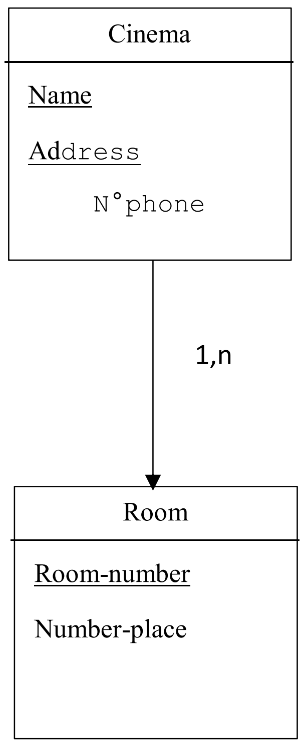

Figure 2.7: Example of weak entity

2. Example of Entity-Relationship Model:

Considering the modeling of a library, in this library, a member can borrow a maximum of 3 books, and automatically, a copy can be taken by only one member on a given date.

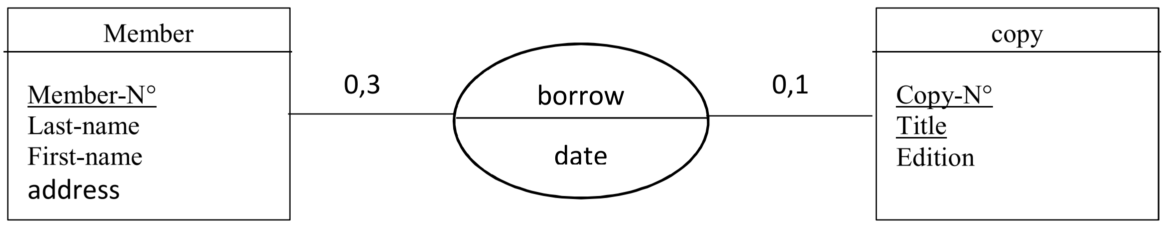

Figure 2.8: Example of Entity-Relationship Model:

In next example the model represents the fact that each actor can play a role in multiple films, and a film is produced by only one producer, but a producer can produce multiple films. A user can rate none or several films, and a film can be rated by none or several users.

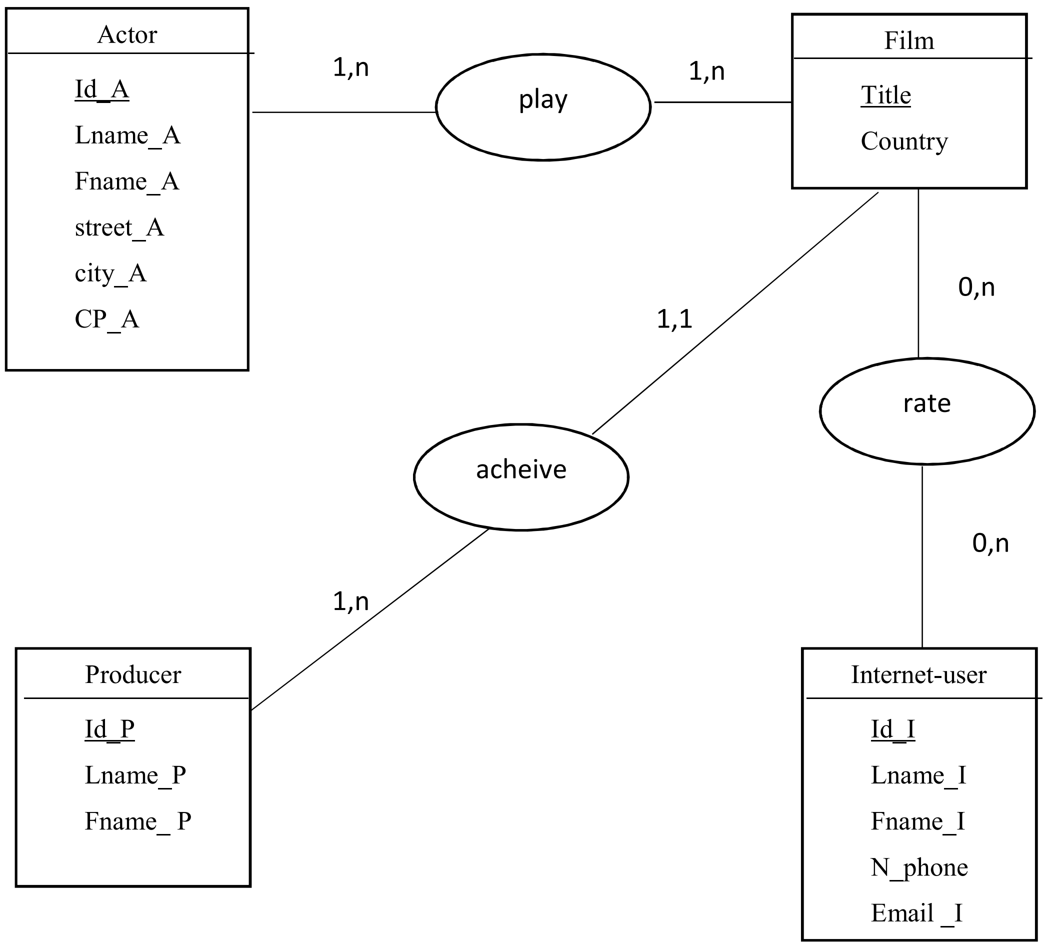

Figure 2.9: example of production films

3. Advantages and Disadvantages of the Entity-Relationship Model:

The Entity-Relationship (E/R) model is simple and practical.

- *There are only 3 concepts:* ***entities***, ***associations***, *and* ***attributes***.
- *It is well-suited for an* ***intuitive graphical representation***, *even though there are many conventions.*
- *It allows for the* ***quick modeling of not-too-complex structures***.
- *In the context of databases, the E/R model is used during the* ***design phase***. *It helps specify the structure of the information to be stored in the database and provides an* ***abstract representation*** *independent of the logical model that will be chosen later.*

Unfortunately, there are several disadvantages. First, it is ***non-deterministic****: there is no absolute rule for determining a unique entity-relationship model, and generally, there are multiple possible views.*