Database



Dr. Abdelhafid Dakhia

University of Mohamed Khider, Biskra

Faculty of Exact Sciences

Department of Computer science

Email: abdelhafid.dakhia@univ-biskra.dz

2.0

27 April 2025

Table of contents

I - Chapter II : The relational model **3**

1. Specific Objectives for Chapter II 3
2. Database modelling 3
3. The Entity/Association model: Basic Concepts 4
4. The Relational Model 5
5. Basic model concepts 5
6. Cartesian Product of a Set of Domains 6
7. Relation 7

7.1. Generalities 7
7.2. Tuple of a relationship 7
7.3. Cardinality of a Relationship 7
7.4. Degree of a Relationship 7
7.5. The INSCRIPTION Relationship 8
7.6. Relationship Diagram 8

8. Schema of a Database 10

8.1. Generalities 10
8.2. Composite key 10
8.3. Foreign key 10

References **12**

Bibliography **13**

I Chapter II : The relational model

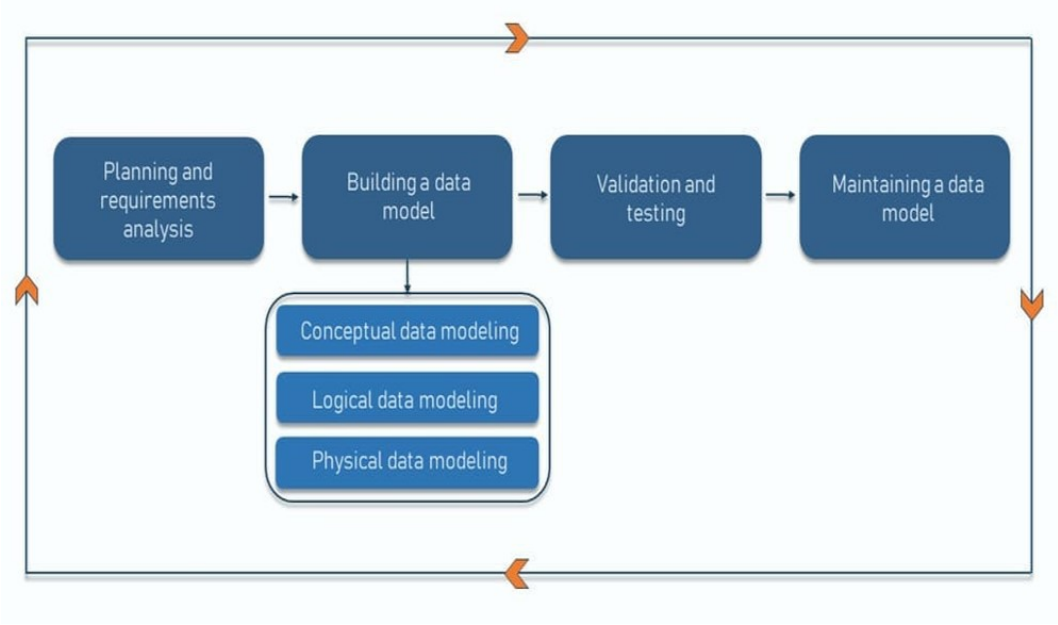
1. Specific Objectives for Chapter II

- Identify the stages of database modeling, including analysis, conceptual design, logical modeling, and implementation.
- Explain the basic elements of the Entity-Relationship (E/R) model such as entities, attributes, keys, and relationships.
- Use relational model concepts like relations, tuples, attributes, degree, and cardinality to represent structured data.
- Analyse different types of keys (primary, candidate, composite, and foreign) and their roles in ensuring data integrity.
- Evaluate database schemas and relationship diagrams to assess their accuracy and effectiveness in modeling real-world scenarios.

2. Database modelling

There are four stages in database design:

- Analysis of existing situation and needs (clarification).
- Creation of a conceptual model to represent all important aspects of the problem.
- Translation of the conceptual model into a logical model (and standardization of this logical model).
- Implementation of a database in a DBMS, based on the logical model (and optimization).



Data modelling Process

Analysis: This consists of studying the problem and recording the needs, choices and constraints in a document known as a clarification note.

Conceptual modeling: This is a non-formal (usually graphic) description of the problem at hand, based on simplifying assumptions. It is not a description of reality, but a simplified representation of reality.

Logical modeling: This describes a solution, taking a formal, general IT approach (typically DBMS type), but independently of specific implementation choices.

Implementation: This corresponds to technical choices, in terms of the DBMS chosen, and their implementation (programming, optimization, etc.).

Note

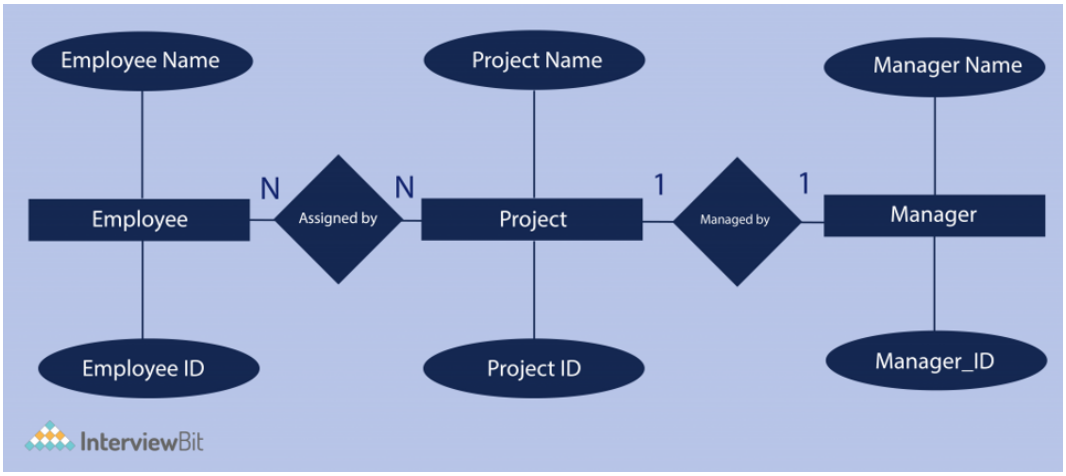
There are several types of conceptual models used in database design. ***The Entity-Association (EA)***, also known as the ***Entity-Relationship (ER) model***, uses entities, attributes, identifiers, and relationships with cardinality to represent data. ***UML (Unified Modeling Language)*** is another modeling approach that uses class diagrams to represent entities and their associations. These models are used to visualize the structure of data and guide the design of databases.

3. The Entity/Association model: Basic Concepts

Definition

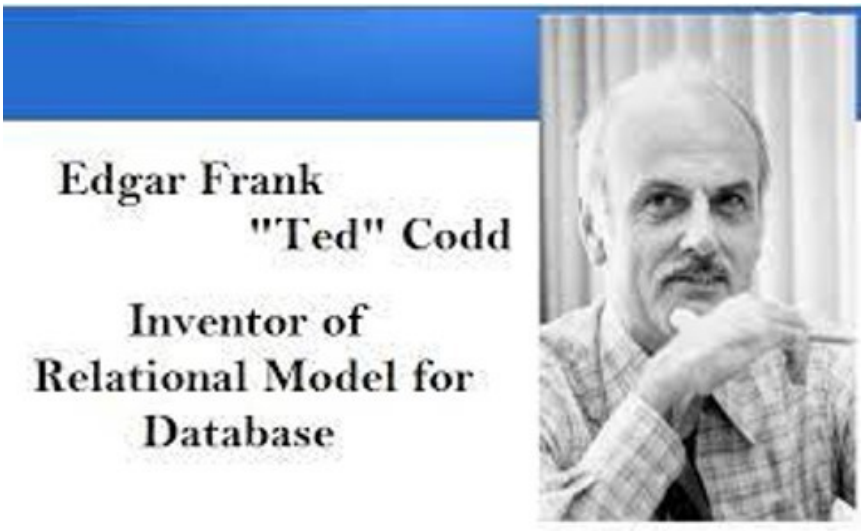
The Entity-Relationship (E/R) model is a ***fundamental framework*** in database design, used to represent real-world data and its structure. At its core, the model is built around the concept of an entity, which refers to any ***distinguishable object*** or thing in the domain being modeled. Each entity is described by a set of properties (also called ***attributes***) that define its characteristics. An occurrence of an entity corresponds to a specific instance of that entity, containing actual data for its properties. To uniquely identify each occurrence, an identifier (or key) is assigned.

Relationships (or associations) are used to illustrate how entities interact with each other, and each relationship is characterized by its cardinality, which defines the number of entity occurrences that can participate in the relationship. This model provides a clear and structured way to visualize and design database systems.



4. The Relational Model

The relational model was proposed by ***Codd*** in the early 70's. It sees the DB as a set of two-dimensional tables called relations.



Among its objectives:*

- Increased independence of programs from data representation.
- Provide a solid basis for dealing with data consistency and redundancy issues.
- Enable the development of non-procedural data manipulation languages based on sound theories.

5. Basic model concepts

Attribut

An attribute is a property or characteristic of a relationship. It is characterized by a unique name in this relationship.

Example

Reference, Description, Couleur→ attributes of the PRODUCT relationship.

Domain

The domain of an attribute corresponds to the set of values that the attribute can take on. This set can be finite, countable or infinite

Example

Domain (colors) = {Blue, Red, White, ...}

Domain (names) = {Ali, Salah, Yasmine...}

6. Cartesian Product of a Set of Domains

The Cartesian product of a set of domains D1, D2, D3,....., Dn denoted by D1 x D2 x D3 x.....x Dn is the set of tuples (v1 , v2, v3 ,....., vn) such that vi ∈ Di for all i = 1, 2, 3,....., n.

Example

Consider the following three domains:

D1=(Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday).

D2= (1, 2, 3,.....31).

D3= (January, February, March, April, May,....December).

Then the Cartesian product

D1 x D2 x D3= {(Monday, 1, January), (Monday, 2, January), (Monday, 3, January), ... (Monday, 31, January), (Monday, 1, February), ... (Monday, 31, February), ...

(Monday, 31, December), (Tuesday, 1, January), ... (Sunday, 1, December), ... (Sunday, 31, December)

7. Relation

7.1. Generalities

A relationship is defined by a list of attributes A1 , A2 , A3 ,...., An whose respective domains are D1 , D2 , D3 ,..... Dn .

It is noted R(A1 , A2 , A3 ,..... , An) where R is the name of the relationship.

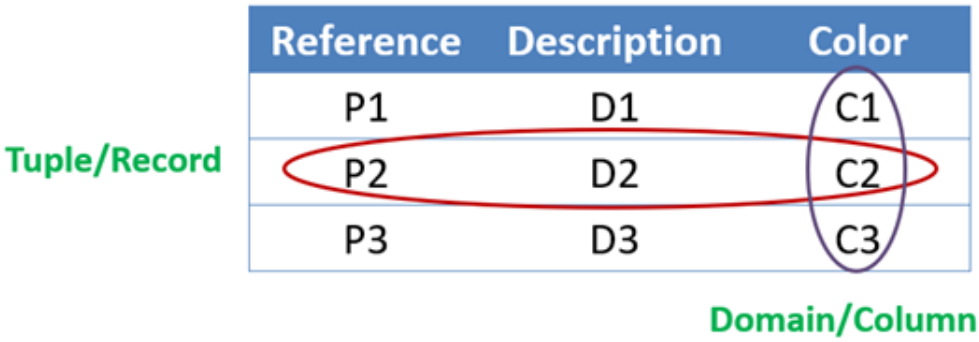
It is composed of a set of tuples (a1 , a2 , a3 ,....., an) or ai ∈ Di ∀ i = 1, 2, 3,....., n, this set constituting a subset of the Cartesian product :

D1 x D2 x D3 x..... x Dn.

7.2. Tuple of a relationship

Definition

A tuple in a relationship is simply a row in the table representing that relationship. An attribute, on the other hand, designates a column in the same table.



Relation PRODUCT

7.3. Cardinality of a Relationship

Definition

The cardinality of a relation* is the number of tuples in that relation. The cardinality of a relation R will therefore be an integer and is given by:|R|.

7.4. Degree of a Relationship

This is the number of columns (domains) in a relationship.

R ⊂ D1 × D2 × ... × Dn

n is the degree of the relation R.

Example

The domains :

NAME = { Zaoui, Badaoui }

FIRST NAME = { Ali, Nora, Larbi }

DATE_NAISS = {Date between 1/1/1990 and 31/12/2020}

SPORT_NAME = { judo, tennis, soccer }

La relation ELEVE

ELEVE ⊂ NOM × PRENOM × DATE_NAISS

ELEVE ={{(Zaoui, Ali,1/1/1992), (Badaoui, Nora , 2/2/1994) }.

7.5. The INSCRIPTION Relationship

REGISTRATION ⊂ NAME × SPORT_NAME

REGISTRATION= {(Zaoui, judo), (Zaoui, soccer), (Badaoui, tennis)}

To easily visualize the contents of a relationship, we use tabular representation.

Each line corresponds to a vector

Each column corresponds to a domain

Example

INSCRIPTION ⊂ NOM × NOM_SPORT.

Name	Sport
Zaoui	Judo
Zaoui	Foot
Badoui	Tennis

7.6. Relationship Diagram

The schema of a relationship is defined by :

- The name of the relationship
- List of attributes

Note: R (A1, A2, ... , An).

Example

Student (last name, first name, birth)

Product (Reference, designation, color)

Let the relationship

CATALOG_PRIX (Product_Code , Description , Supplier , Price).

This relationship is defined by its attributes, which are : Product_Code , Description, Supplier and Price.

Its name is: CATALOG_PRIX.

A table representation of this relationship, where the columns correspond to the attributes and the rows to the tuples of the relationship, would be :

Product-code	Description	SUPPLIER	PRICE
P001	CHELF	ALI	1500
P002	WARDROBE	ALI	5400
P003	DRAWER	ALI	750
P030	ARMCHAIR	BRAHIM	3000
P003	DRAWER	KAMEL	600
P150	CHAIR	KAMEL	1000
P150	CHAIR	OMAR	900

8. Schema of a Database

8.1. Generalities

Definition

The schema of a database is defined by :

- The set of relationship schemas that make up the database,
- The database schema shows how data is organized in the database.

Key to a relationship

One of the ***integrity constraints*** of a schema is the unique identification of the tuples of a relation. This unique identification is ensured by the notion of ***relationship key.***

- A key can be made up of a single attribute or a list of attributes that uniquely characterizes a tuple of the relationship.
- A relationship can have several keys. A key with a minimum number of attributes is chosen as the primary key; other possible keys are called ***candidate keys.***
- By convention, the primary key of a relationship is underlined in a relationship diagram.

Primary key: One or more attributes of a relationship, enabling its tuples (records) to be fully differentiated.

In other words, a primary key is one or more attributes that uniquely identify records.

Candidate key: One or more attributes that can be qualified as a unique key in a relationship.

8.2. Composite key

Another type of key is called a composite key, which is a combination of two or more attributes in a table that can be used to uniquely identify each row in the table.

Example

PRODUCT (NProd, nom_prod, prixUHT)

NProd → nom_prod

NProd → priceUHT

Then ***NProd is a key***

Example

Employee {ID, First_name, Last_name, Salary, Age, Address}

We have two candidate keys (ID) and (First_name, Last_name)

Because they uniquely identify each employee in the EMPLOYEE relationship. Here, we'll choose ID as the primary key, because sometimes it can happen that two employees have the same first and last name, so it will be easy to find an employee with his ID.

8.3. Foreign key

Definition

A foreign key is a set of one or more columns in a relationship that refers to a primary key in another relationship.

All foreign key values appear in another relationship as key values.

- This is a ***referential integrity constraint.***
- By convention, the foreign key of a relationship is preceded (or followed) by the # symbol in a relationship schema

Example

Let's consider the following relationship diagrams:

Customer (NumCl, NomCl, AdrCl) Refers to all customers.

Order (NCmde, DateCmde, #NumCl) Refers to all orders.

The NumCl attribute in the Order relationship (table) is a foreign key.

It takes its values from the value domain of the NumCl attribute in the Customer relationship diagram.

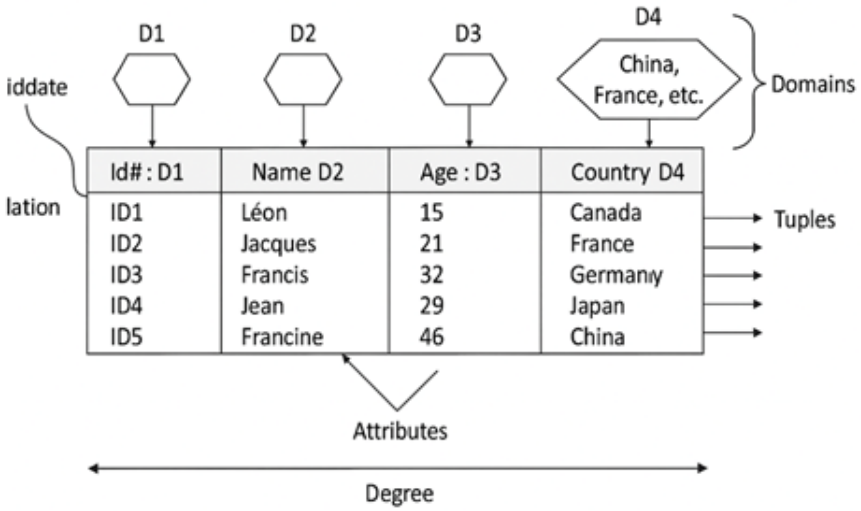
An order is always placed by an existing Customer in the database.

An attribute can be both a primary and a foreign key:

Film (numFilm, title, year).

Actor (numActor, last name, first name).

Casting (#numFilm, #numActor, character).



Data modelling process

References

(1)
Ramakrishnan, Raghu, Johannes Gehrke, and Johannes Gehrke. Database management systems. Vol. 3. New York: McGraw-Hill, 2003.

Bibliography

Silberschatz, Abraham, Henry F. Korth, and Shashank Sudarshan. "Database system concepts." (2011).