Conceptual Model

CE3101 - Databases

Disclaimer / Descargo de Responsabilidad

Esta presentación corresponde a una guía usada por el profesor durante las clases. La misma ha sido modificada para ser utilizado en el modelo de cursos asistidos por tecnología. No es una versión final, por lo que la misma podría requerir todavía hacer algunos ajustes. Para aspectos de evaluación esta presentación es solo una guía, por lo que el estudiante debe profundizar con el material de lectura asignado y lo discutido en clases para aspectos de evaluación.

This presentation corresponds to a guide material used by the professor during classes. It has been modified to be used in the model of technology-assisted courses. It is not a final version, so it may still require some adjustments. For evaluation aspects, this presentation is only a guide, so the student should delve with the assigned reading material and what has been discussed in class.

Database Design Process

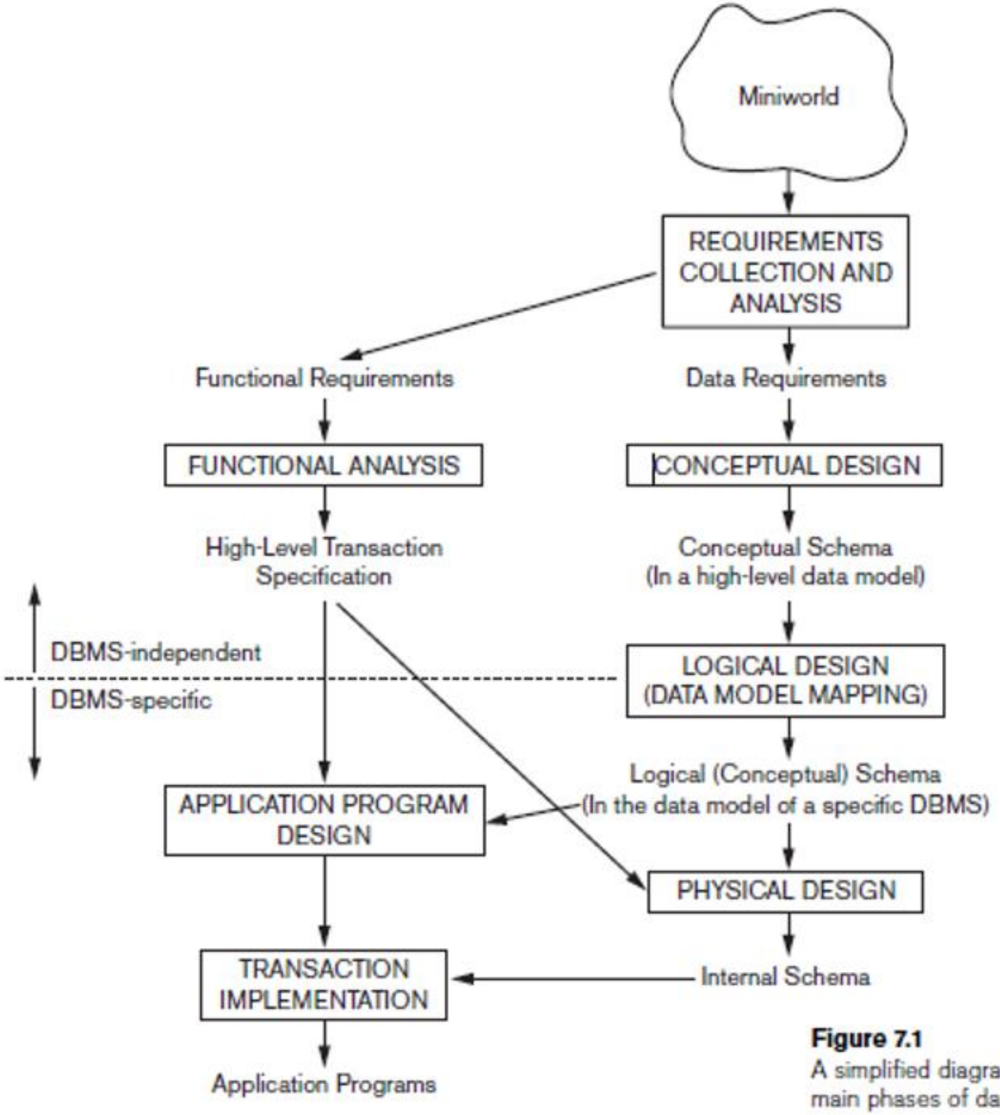

Miniworld
REQUIREMENTS COLLECTION AND ANALYSIS
Functional Requirements
Data Requirements
FUNCTIONAL ANALYSIS
CONCEPTUAL DESIGN
High-Level Transaction Specification
Conceptual Schema (In a high-level data model)
DBMS-independent
DBMS-specific
LOGICAL DESIGN (DATA MODEL MAPPING)
Logical (Conceptual) Schema (In the data model of a specific DBMS)
APPLICATION PROGRAM DESIGN
PHYSICAL DESIGN
TRANSACTION IMPLEMENTATION
Internal Schema
Application Programs

Figure 7.1
A simplified diagram to illustrate the main phases of database design.

Think about a car…

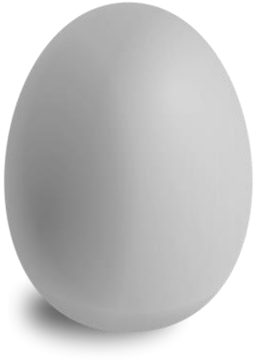
- The car **is:**
  - Black
  - Has doors that open upward
  - Lamborghini
  - Made of metal
  - Heavy
  - Expensive
  - Not a Hyundai
  - The same I have in my garage!

Let’s bake a cake

- ➔ What objects are involved?

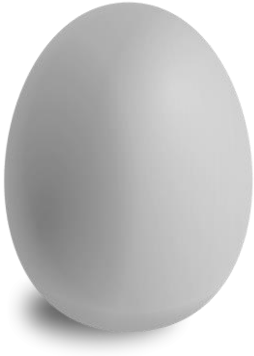
Let’s bake a cake

- ➔ What objects are involved?



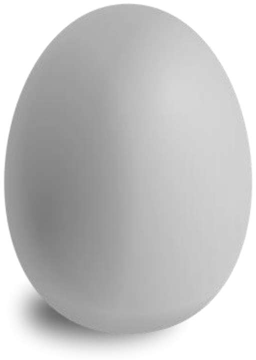
Let’s bake a cake

- ➔ What objects are involved?



Let’s bake a cake

➔ What objects are involved?

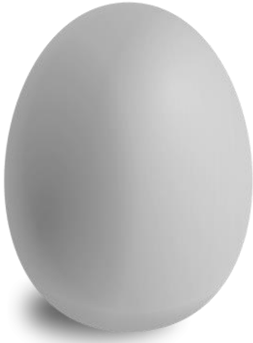


Let’s bake a cake

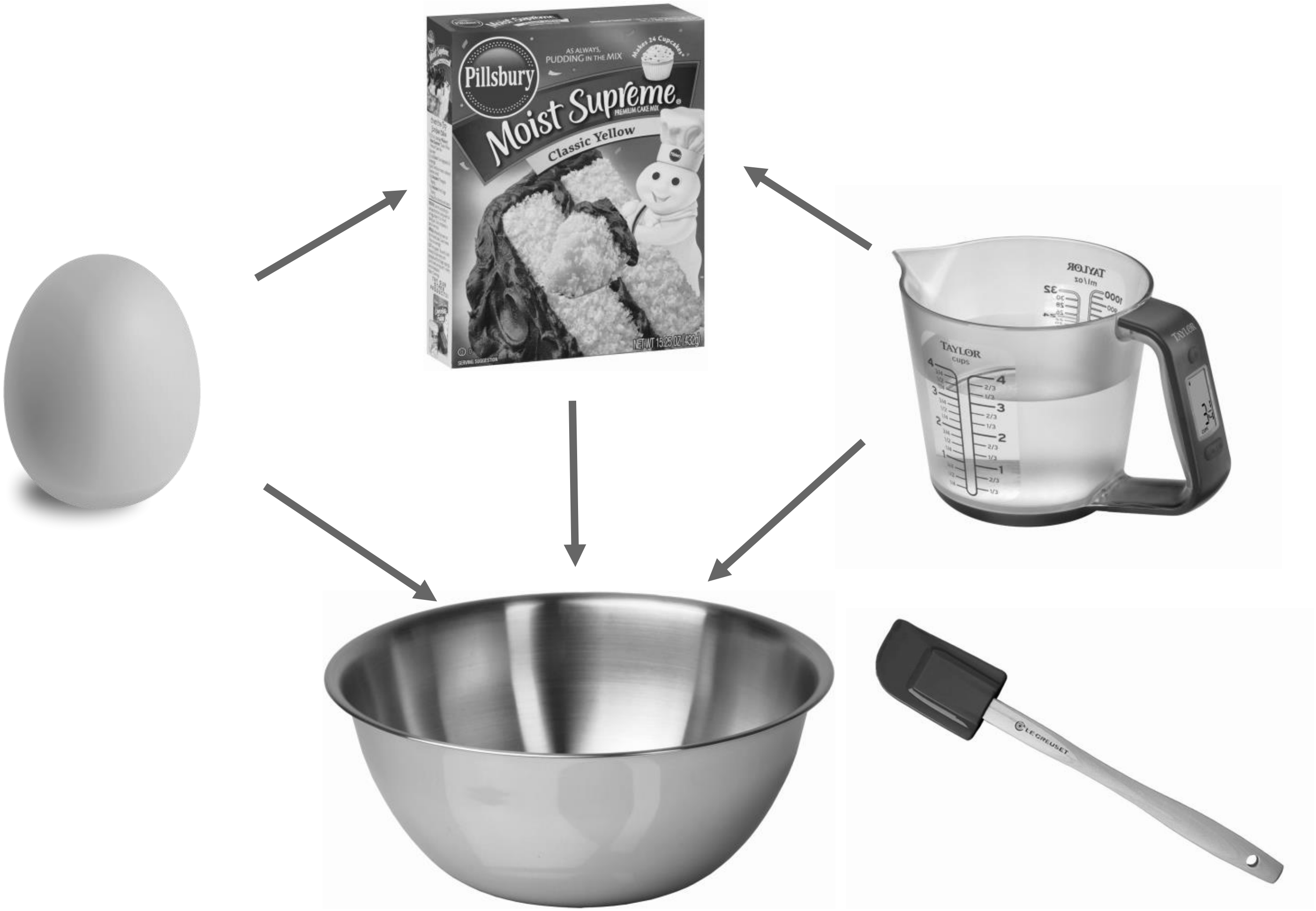
- ➔ What objects are involved?

Let’s bake a cake

➔ What objects are involved?



Now define the relationship between the objects

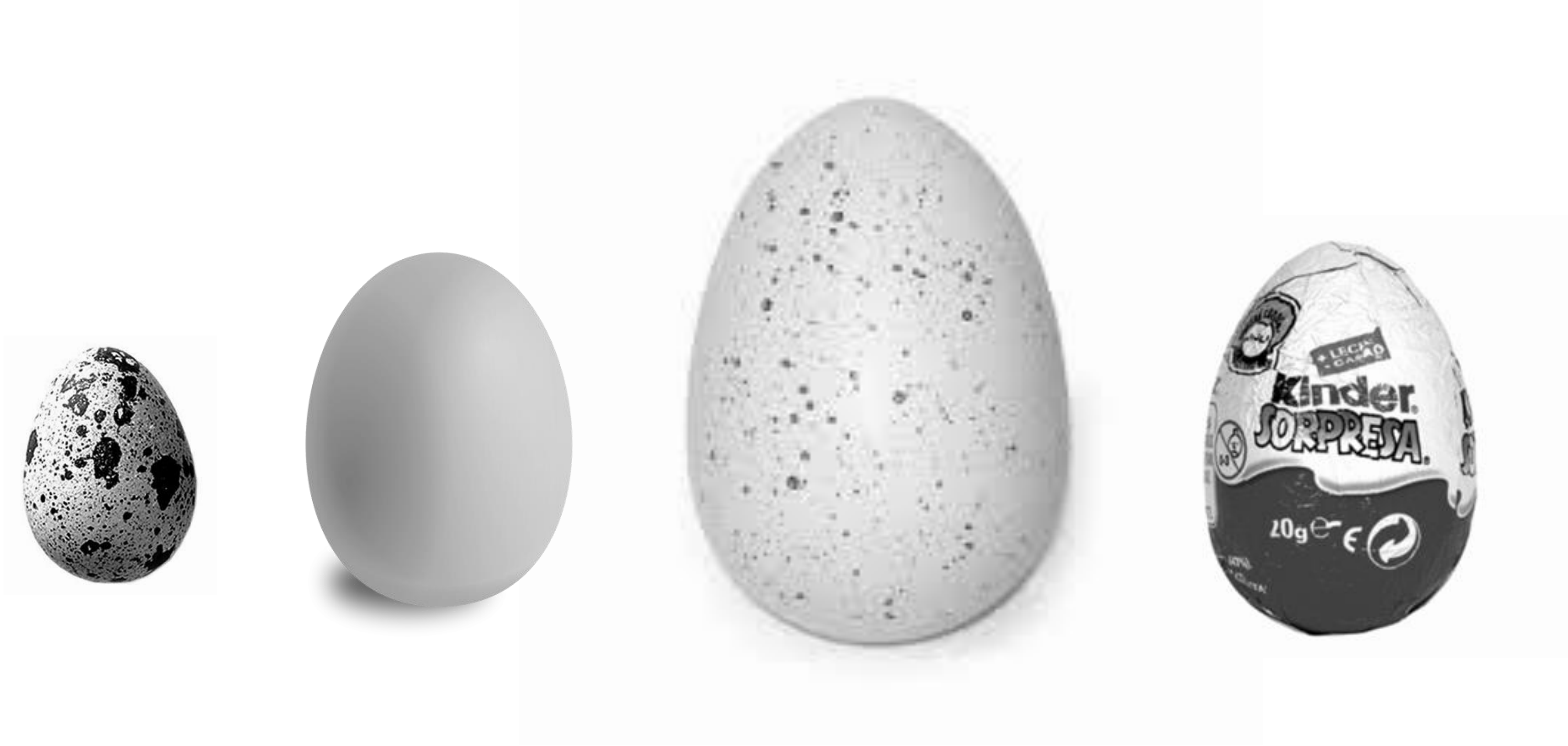


We have to define the attributes of each one

- ➔ Ok… you need an egg, but what kind of egg?

We have to define the attributes of each one

➔ Ok… you need an egg, but what kind of egg?



Ejercicio Modelado del Contexto de un Problema

Entities and Attributes

- The basic object that the ER Model represents is an **entity**, which is a *thing* in the real world with an independent existence.

- Each entity has **attributes**. Particular properties that describe it.

Entities and Attributes

ENTITY

➔ The basic object that the ER Model represents is an **entity**, which is a *thing* in the real world with an independent existence.

EXAMPLE

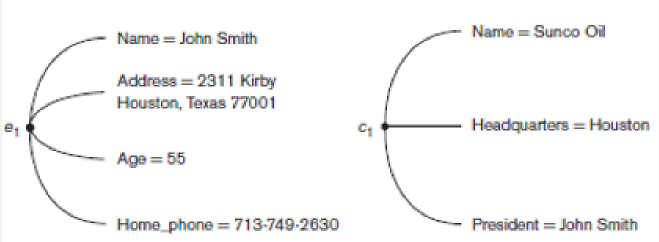
REPRESENTED BY

Entities and Attributes

ENTITY

- ➔ The basic object that the ER Model represents is an **entity**, which is a *thing* in the real world with an independent existence.

EXAMPLE



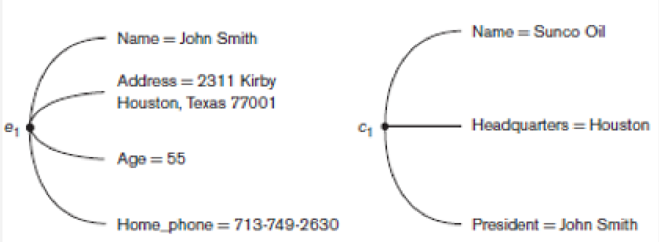
REPRESENTED BY

Entities and Attributes

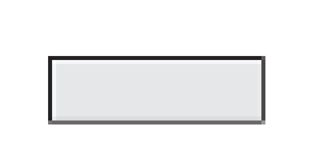
ENTITY

➔ The basic object that the ER Model represents is an **entity**, which is a *thing* in the real world with an independent existence.

EXAMPLE



REPRESENTED BY



Entities and Attributes

ATTRIBUTES

➔ Each entity has **attributes**. Particular properties that describe it.

EXAMPLE

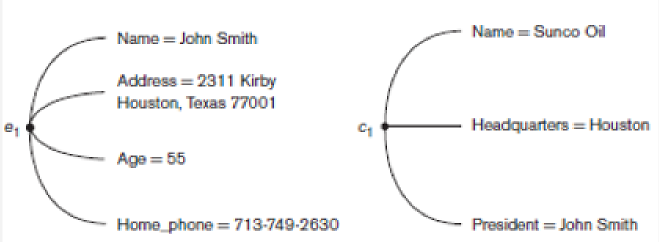
REPRESENTED BY

Entities and Attributes

ATTRIBUTES

- ➔ Each entity has **attributes**. Particular properties that describe it.

EXAMPLE



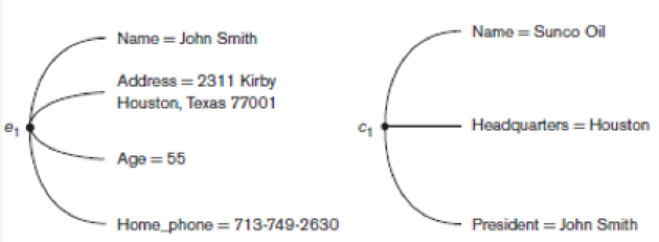
REPRESENTED BY

Entities and Attributes

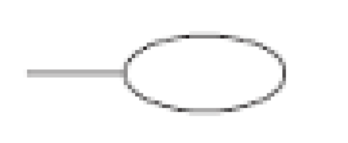
ATTRIBUTES

➔ Each entity has **attributes**. Particular properties that describe it.

EXAMPLE



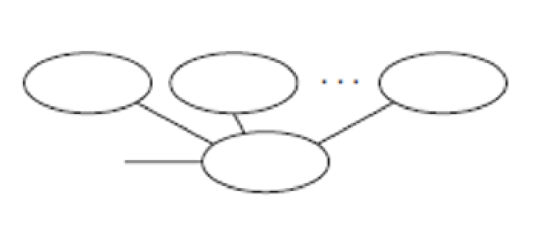
REPRESENTED BY



Attributes

COMPOSITE ATTRIBUTES

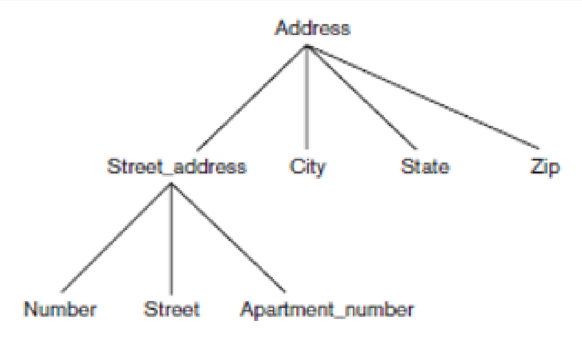
➔ Attribute divided into smaller parts. Represents more basic attributes with independent meanings.



Attributes

COMPOSITE ATTRIBUTES

- ➔ For example, the address attribute of the EMPLOYEE

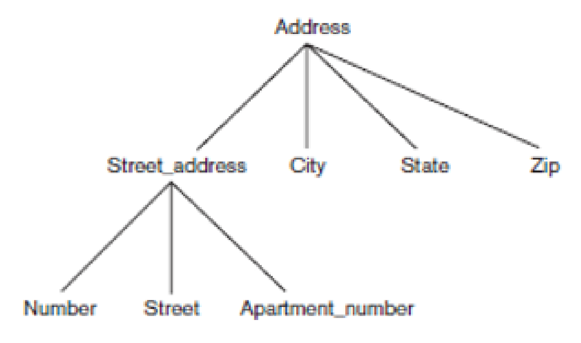


Attributes

Are useful to model situations on which a user sometimes refers to the composite attribute as a unit but at other times refers specifically to its components.

COMPOSITE ATTRIBUTES

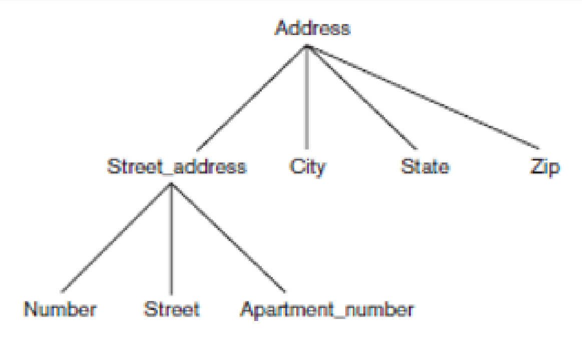
- ➔ For example, the address attribute of the EMPLOYEE



Attributes

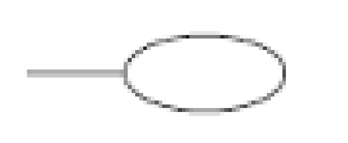
COMPOSITE ATTRIBUTES

- ➔ For example, the address attribute of the EMPLOYEE



SINGLED-VALUED ATTRIBUTES

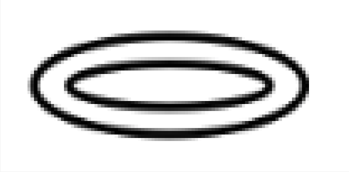
- ➔ Has a single value for particular entity.



Attributes

MULTIVALUED ATTRIBUTES

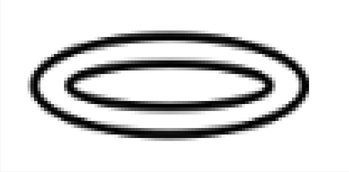
- ➔ Can have a set of values for the same entity.



Attributes

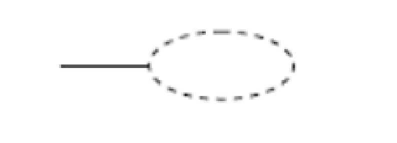
MULTIVALUED ATTRIBUTES

- ➔ Can have a set of values for the same entity.



STORED AND DERIVED

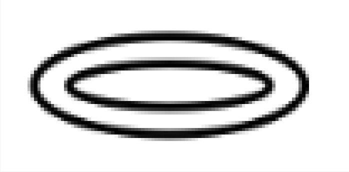
- ➔ In some cases, two (or more) attribute values are related.
- ➔ **Stored.** Attribute that has the value.
- ➔ **Derived**. Attribute that depends of other attribute to assign a value.



Attributes

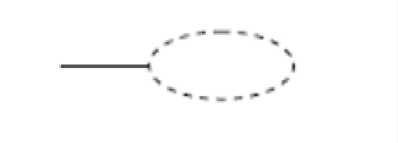
MULTIVALUED ATTRIBUTES

- ➔ Can have a set of values for the same entity.



STORED AND DERIVED

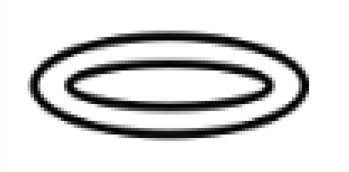
- ➔ In some cases, two (or more) attribute values are related.
- ➔ **Stored.** Attribute that has the value.
- ➔ **Derived**. Attribute that depends of other attribute to assign a value.



Attributes

MULTIVALUED ATTRIBUTES

➔ Can have a set of values for the same entity.



STORED AND DERIVED

➔ For a particular person entity, the value of Age can be determined from the current (today's) date and the value of that person's Birth date. The Age attribute is hence called a derived attribute and is said to be derivable from the Birth date attribute, which is called a stored attribute.

Attributes

NULL VALUES

- ➔ In some cases, a particular entity may not have an applicable value for an attribute.

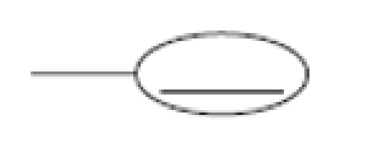
Attributes

NULL VALUES

- ➔ In some cases, a particular entity may not have an applicable value for an attribute.

KEY ATTRIBUTES

- ➔ Uniqueness constraint.
- ➔ One or more attributes whose values are distinct for each individual entity in the entity set.

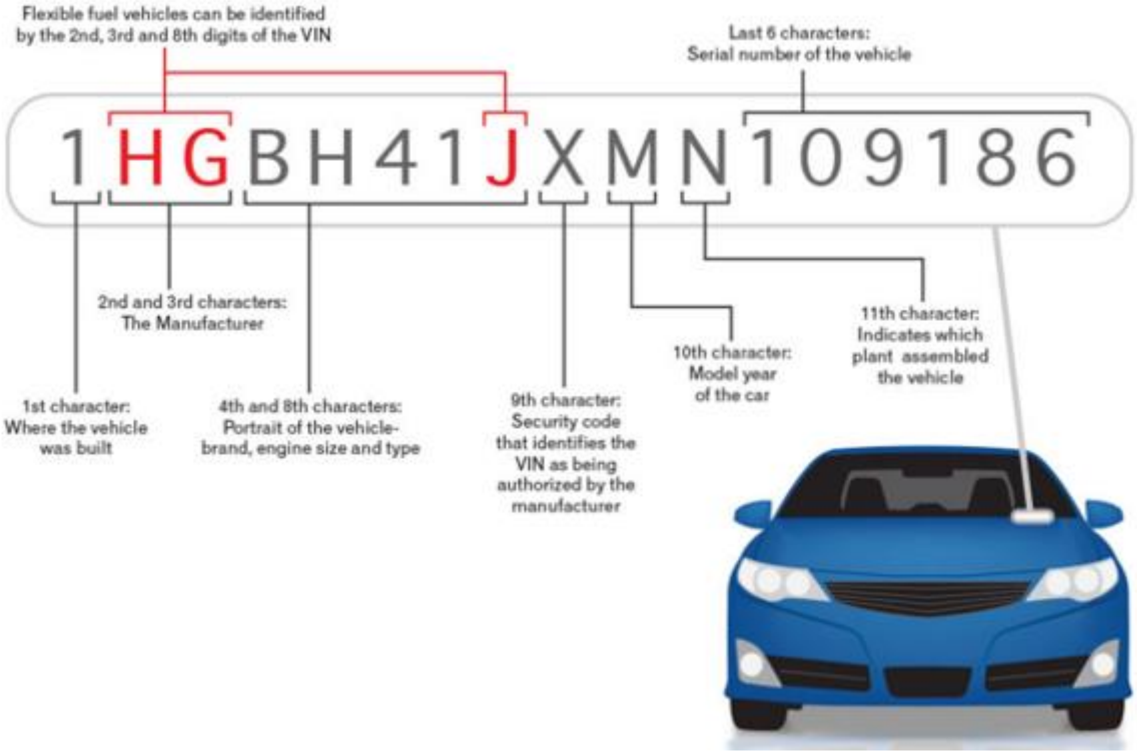


Key Attributes

- ➔ The car’s vehicle identification number (VIN) is the identifying code for a specific automobile. The VIN serves as the car’s fingerprints, as no two vehicles in operation have the same VIN.

Key Attributes

➔ The car’s vehicle identification number (VIN) is the identifying code for a specific automobile. The VIN serves as the car’s fingerprints, as no two vehicles in operation have the same VIN.


Flexible fuel vehicles can be identified by the 2nd, 3rd and 8th digits of the VIN
Last 6 characters: Serial number of the vehicle
1 H G B H 4 1 J X M N 1 0 9 1 8 6
2nd and 3rd characters: The Manufacturer
11th character: Indicates which plant assembled the vehicle
10th character: Model year of the car
1st character: Where the vehicle was built
4th and 8th characters: Portrait of the vehicle-brand, engine size and type
9th character: Security code that identifies the VIN as being authorized by the manufacturer

Key Attributes of an Entity Type

- ➔ An important constraint on the entities of an entity type is the **key** or **uniqueness constraint** on attributes.

- ➔ An entity type usually has one or more attributes whose values are distinct for each individual entity in the entity set. Such an attribute is called a key attribute.

Value Sets (Domains) of Attributes

- Each simple attribute of an entity type is associated with a value set (or domain of values), which specifies the set of values that may be assigned to that attribute for each individual entity.

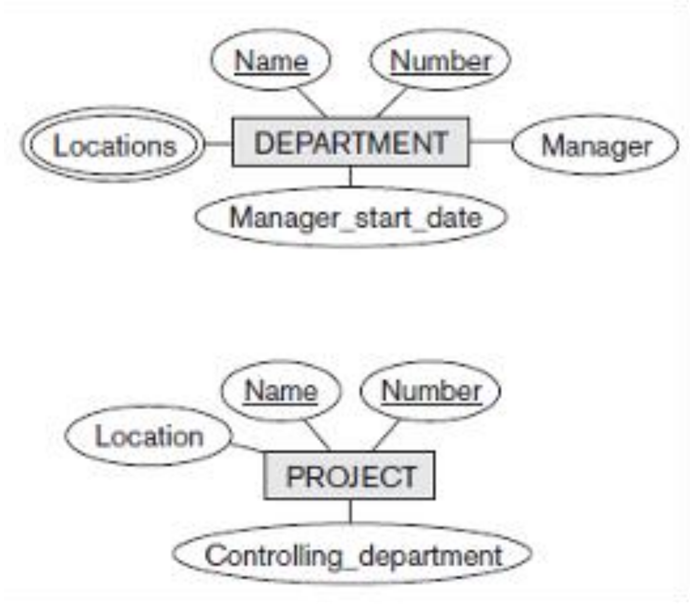
Pregunta inicial

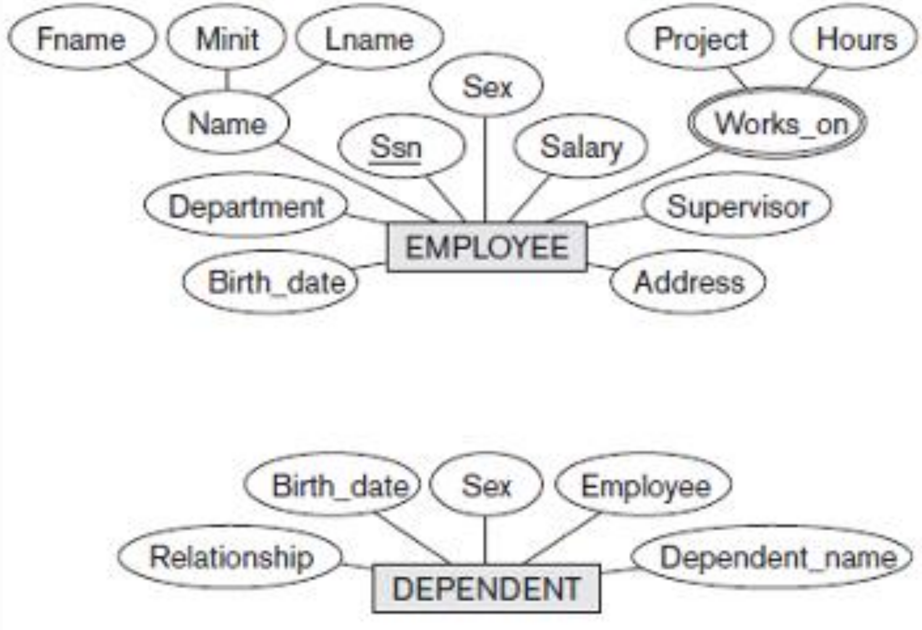
Cuál es el único animal que solo tiene un oido?

Study Case

- The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.
- A department controls a number of projects, each of which has a unique name, a unique number, and a single location.
- We store each employee’s name, Social Security number,2 address, salary, sex(gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. We keep track of the current number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee (who is another employee).
- We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent’s first name, sex, birth date, and relationship to the employee.

Study Case


Name
Number
Locations
DEPARTMENT
Manager
Manager_start_date
Name
Number
Location
PROJECT
Controlling_department


Fname
Minit
Lname
Project
Hours
Sex
Name
Ssn
Salary
Works_on
Department
Supervisor
EMPLOYEE
Birth_date
Address
Birth_date
Sex
Employee
Relationship
Dependent_name
DEPENDENT

Respuesta a pregunta inicial

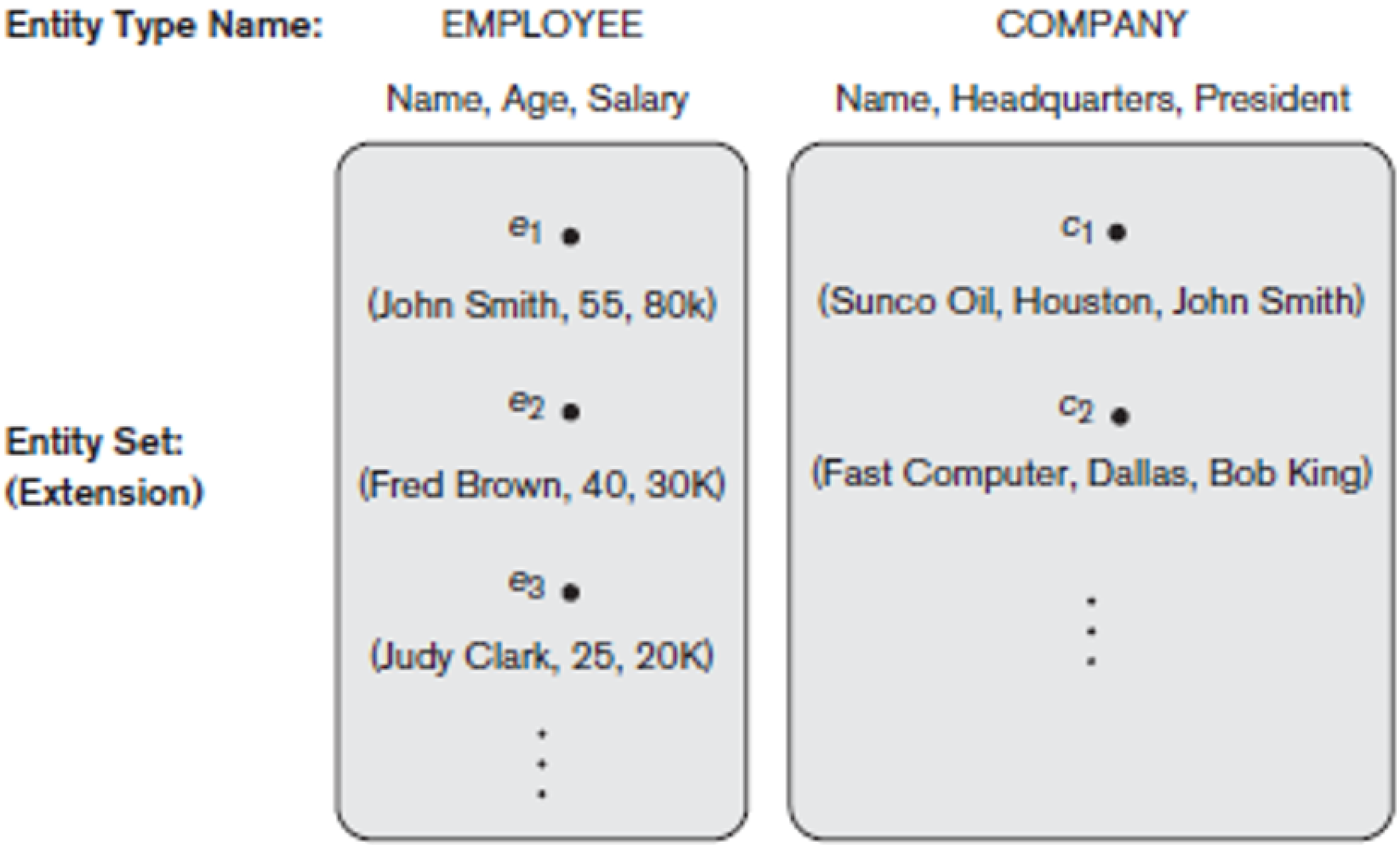
La mantis religiosa. Situado en el torax y no en la cabeza.

Pregunta inicial

Aparte de las huellas dactilares qué otras partes del cuerpo humano sirven como marca de identidad?

Relationship Types, Relationship Sets, Roles and Structural Constraints

Entity Type, Instances and Entity Sets



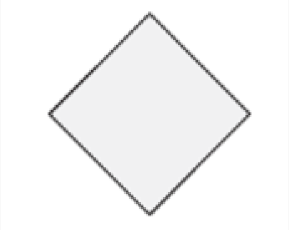
Relationship

DEFINITION

➔ Whenever an attribute of one entity refers to another entity type, some relationship exists.

➔ In the ER model, these references should not be represented as attributes but as **relationships**.

SYMBOL



Relationship Types

- ➔ A relationship type R among n entity types defines a set of associations among entities from these entity types.

E1, E2, E3…, En

- ➔ **Formally.** Relationship set is a mathematical relation on

E1, E2, E3, …, En

can be defined as a subset of the Cartesian Product of the entity sets

E1 x E2 x E3 x … x En

Each of the entity types E1, E2, E3, …, En is said to participate in the relationship type R.

Relationship Types

➔ A relationship type R among n entity types defines a set of associations among entities from these entity types.

E1, E2, E3…, En

➔ **Formally.** Relationship set is a mathematical relation on

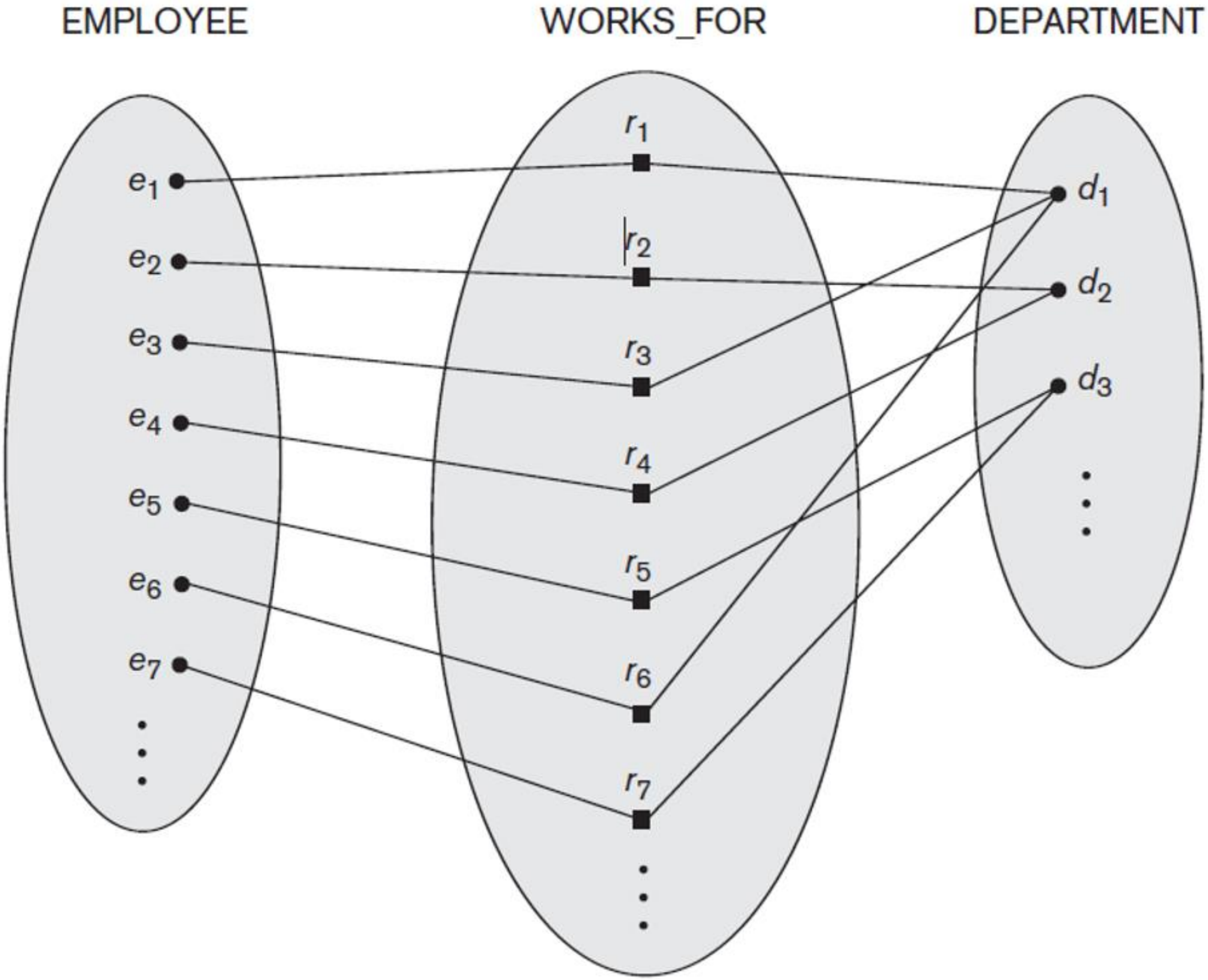
E1, E2, E3, …, En

can be defined as a subset of the Cartesian Product of the entity sets

E1 x E2 x E3 x … x En

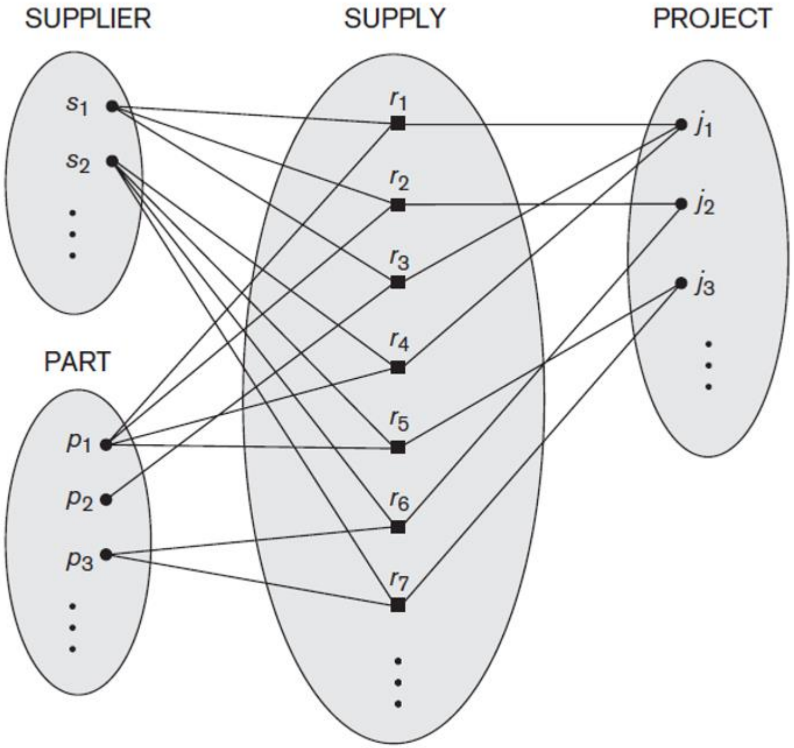
(Informally) Each relationship instance r_i in R is an association of entities, where the association includes exactly one entity from each participating entity type.

Relationship Types



Degree of a Relationship Type

- ➔ The degree of a relationship type is the number of participating entity types.
- ➔ A relationship type of degree two is called **binary**, and one of degree three is called **ternary**.



Role Name and Recursive Relationships

ROLE NAME

➔ Helps to explain what the relationship means. Role that participating entity plays in relationship instance.

RECURSIVE RELATIONSHIP

➔ The same entity participates more than once in a relationship.

Role Name and Recursive Relationships

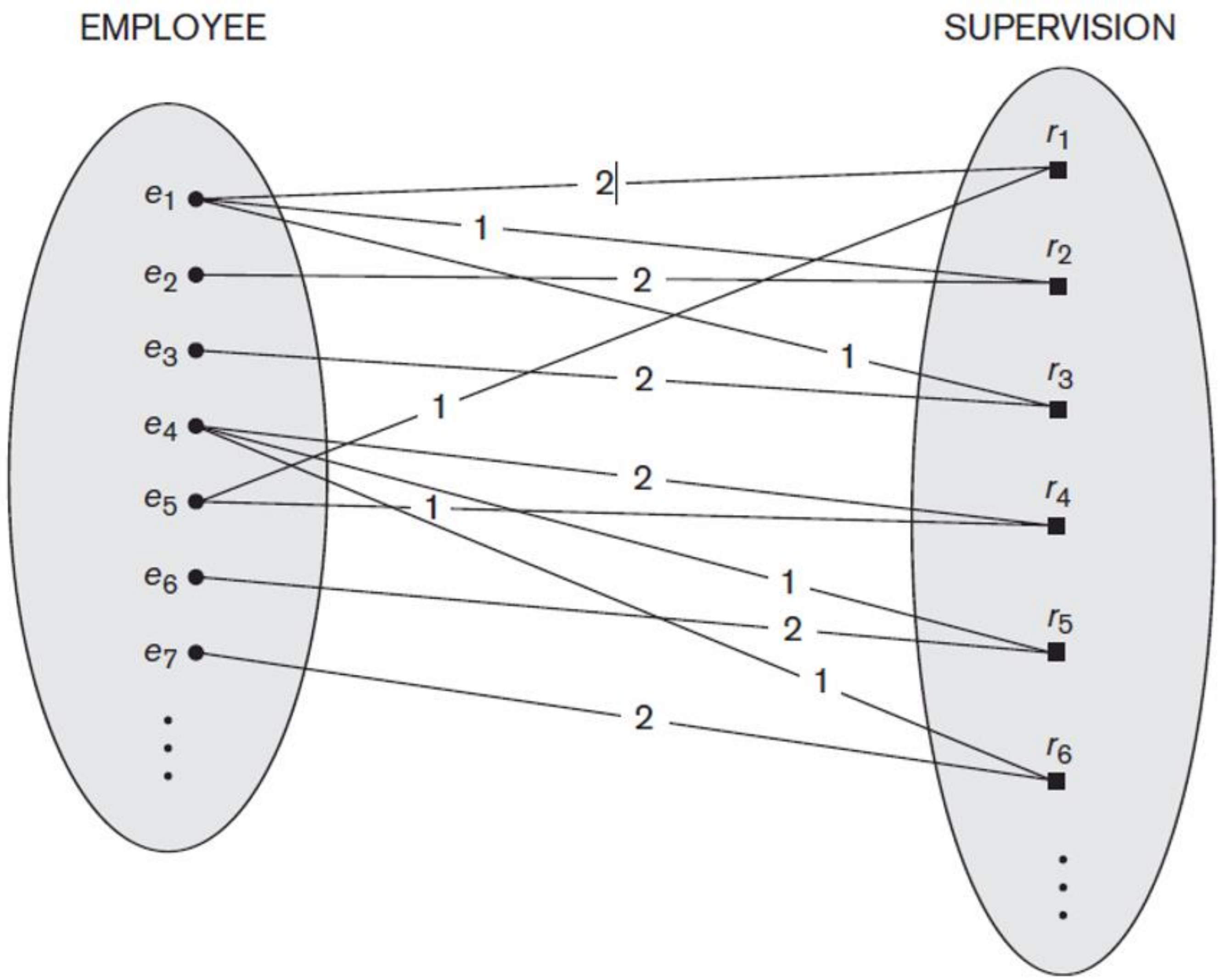


Figure 7.11
A recursive relationship SUPERVISION between EMPLOYEE in the *supervisor* role (1) and EMPLOYEE in the *subordinate* role (2).

Constraints on Binary Relationship Types

- Relationship types usually have certain constraints that limit the possible combinations of entities that may participate in the corresponding relationship set.

- These constraints are determined from the miniworld situation that the relationships represent.

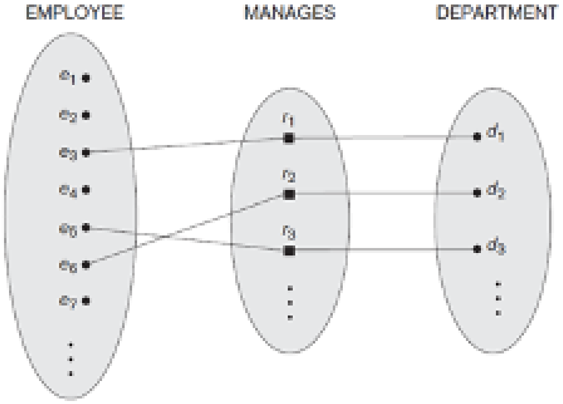
- We can distinguish two main types of binary relationship constraints :
  - Cardinality radio.
  - Participation.

Constraints on Binary Relationship Types

- Relationship types usually have certain constraints that limit the possible combinations of entities that may participate in the corresponding relationship set.

- These constraints are determined from the miniworld situation that the relationships represent.

- We can distinguish two main types of binary relationship constraints :
  - **Cardinality radio**. Specifies the maximum number of relationship instances that an entity can participate in.
  - Participation.

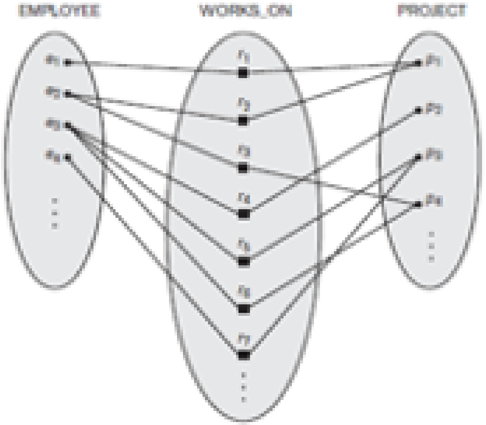


Constraints on Binary Relationship Types

- ➔ Relationship types usually have certain constraints that limit the possible combinations of entities that may participate in the corresponding relationship set.

- ➔ These constraints are determined from the miniworld situation that the relationships represent.

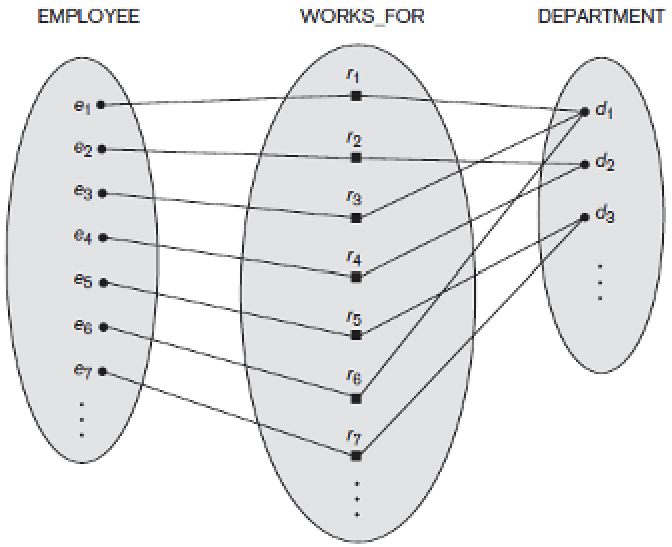
- ➔ We can distinguish two main types of binary relationship constraints :
    - ◆ **Cardinality radio**. Specifies the maximum number of relationship instances that an entity can participate in.
    - ◆ Participation.



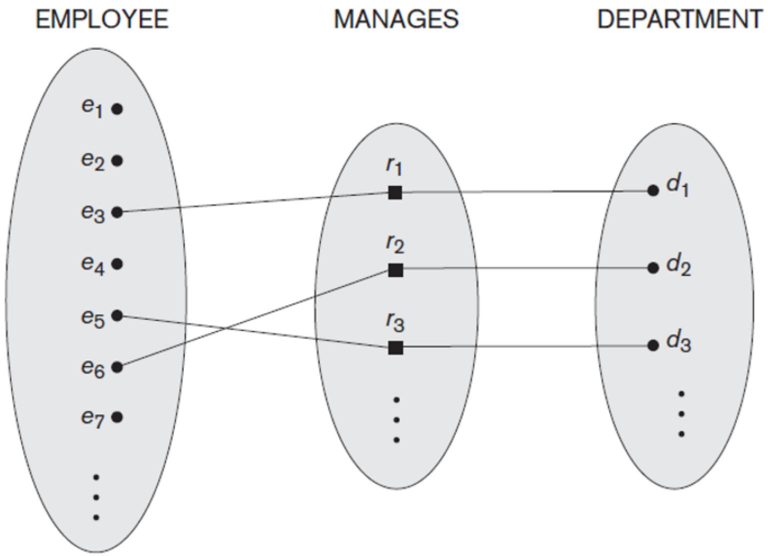
Participation Constraints and Existence Dependencies

- ➔ Specifies when the existence of an entity depends on being related to another entity.

- ➔ There are two types of participation constraints:



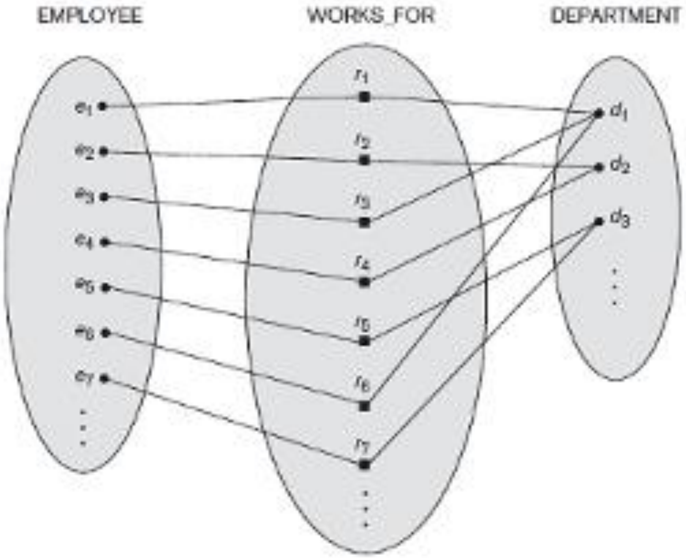
Total



Partial

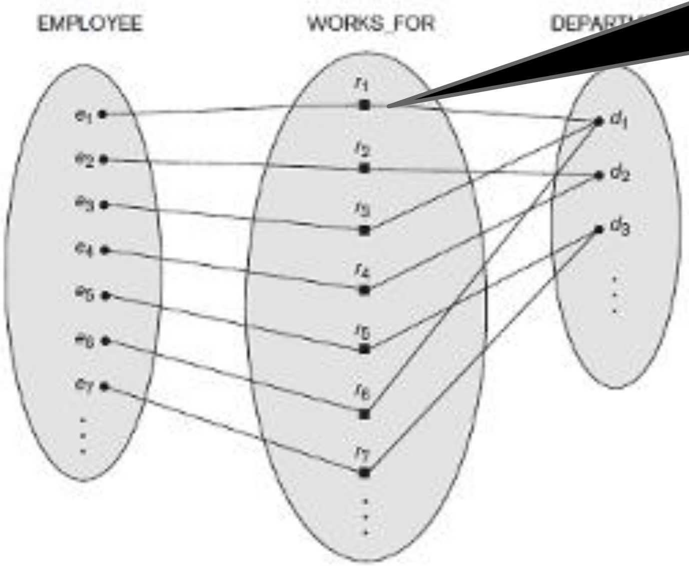
Attributes of Relationship Types

➔ Relationship types can also have attributes, similar to those of entity types.



Attributes of Relationship Types

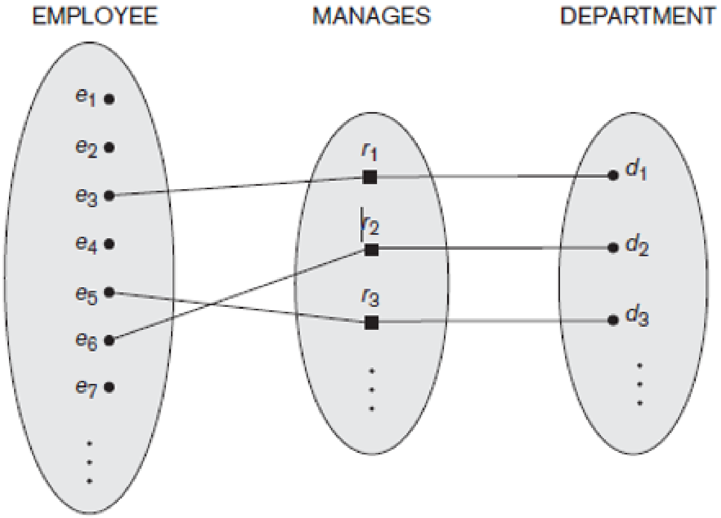
➔ Relationship types can also have attributes, similar to those of entity types.



Example: Number of hours per week that an employee works on a particular project

Attributes of Relationship Types

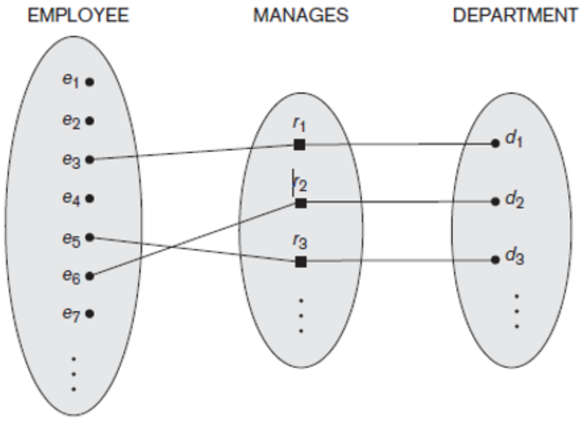
➔ Relationship types can also have attributes, similar to those of entity types.



Notice that attributes of 1:1 or 1:N relationship types can be migrated to one of the participating entity types.

Attributes of Relationship Types

➔ Relationship types can also have attributes.

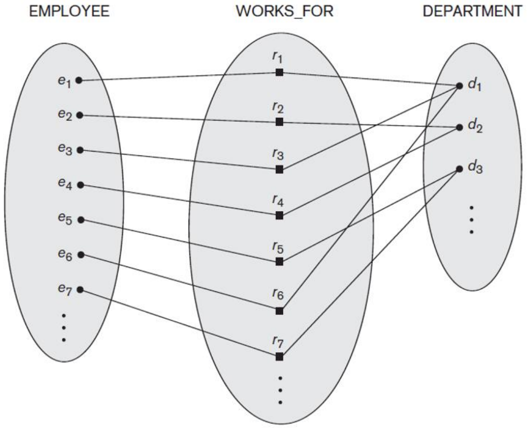


StartDate?

1:1 RELATIONSHIP	1:N RELATIONSHIP
➔ StartDate for MANAGES relationship can be an attribute of EMPLOYEE or DEPARTMENT	

Attributes of Relationship Types

➔ Relationship types can also have attributes.



StartDate?

1:1 RELATIONSHIP

- ➔ StartDate for MANAGES relationship can be an attribute of EMPLOYEE or DEPARTMENT.

1:N RELATIONSHIP

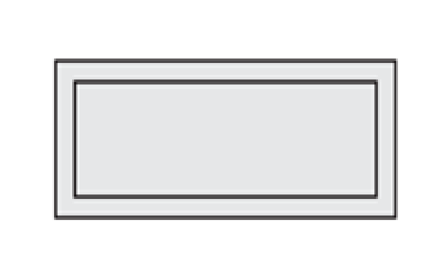
- ➔ Attribute can be migrated only to the entity type on the N-side of the relationship.

Weak Entity Types

DEFINITION

- ➔ Entity types that do not have key attributes of their own.
- ➔ A weak entity type always has a total participation constraint (existence dependency) with respect to its identifying relationship because a weak entity cannot be identified without an owner entity.
- ➔ Both a weak entity type and its identifying relationship are distinguished by surrounding their boxes and diamonds with double lines.

SYMBOL



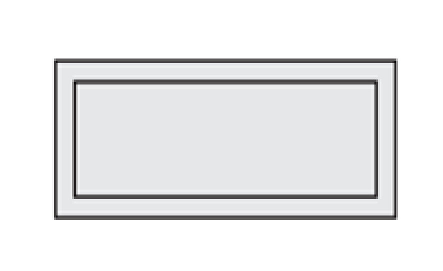
Weak Entity Types

Weak entity types can sometimes be represented as complex (composite, multivalued) attributes.

DEFINITION

- ➔ Entity types that do not have key attributes of their own.
- ➔ A weak entity type always has a total participation constraint (existence dependency) with respect to its identifying relationship because a weak entity cannot be identified without an owner entity.
- ➔ Both a weak entity type and its identifying relationship are distinguished by surrounding their boxes and diamonds with double lines.

SYMBOL



Proper Naming of Schema Constructs

- We choose to use singular names for entity types, rather than plural ones, because the entity type name applies to each individual entity belonging to that entity type.

- In our ER diagrams, we will use the convention that entity type and relationship type names are uppercase letters, attribute names have their initial letter capitalized.

Proper Naming of Schema Constructs

- ➔ As a general practice, given a narrative description of the database requirements:
  - ◆ The nouns appearing in the narrative tend to give rise to entity type names.
  - ◆ The verbs tend to indicate names of relationship types.
  - ◆ Attribute names generally arise from additional nouns that describe the nouns corresponding to entity types.

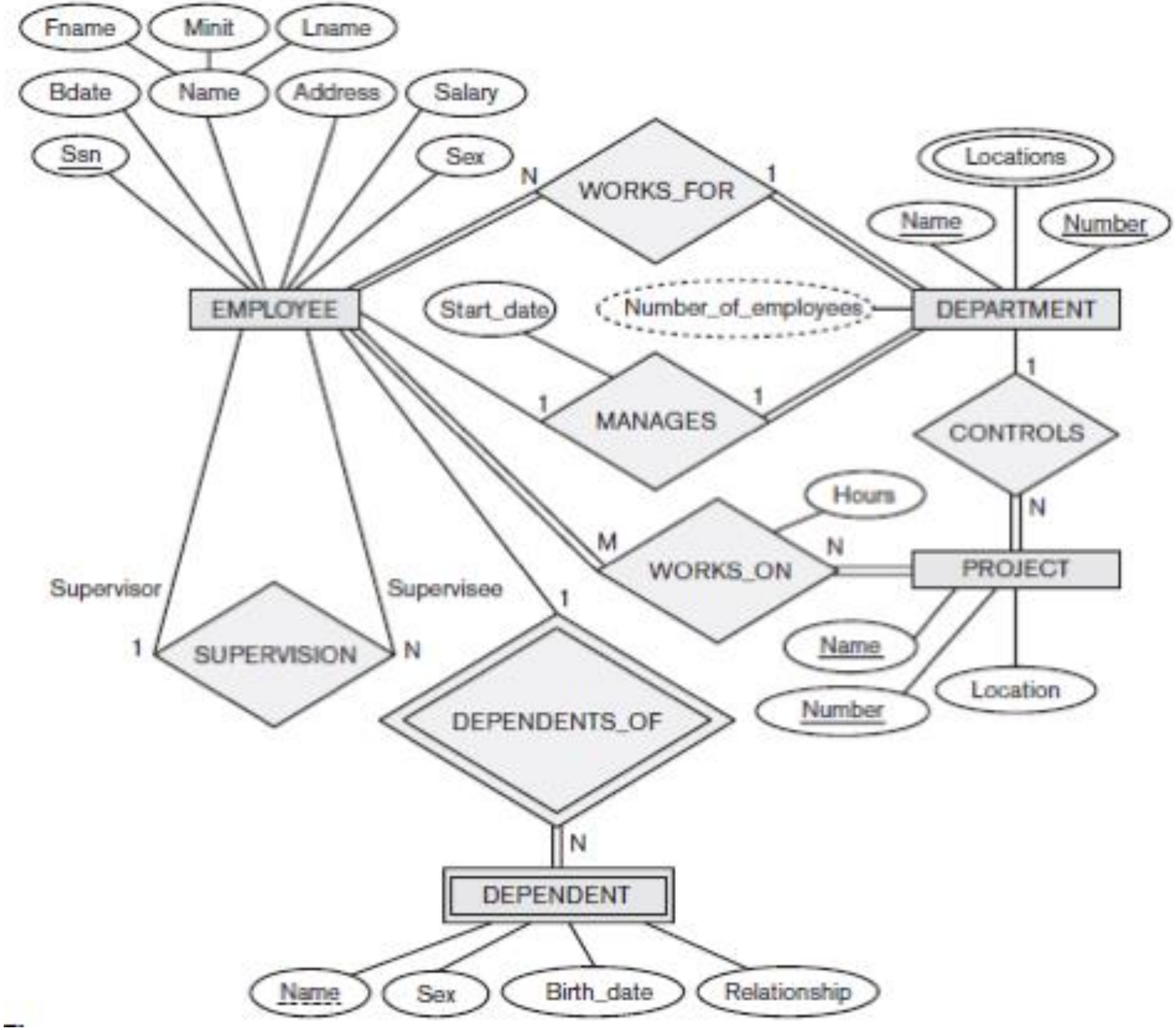
Respuesta a pregunta inicial

Aparte de las huellas dactilares, el aroma y la lengua de cada persona también es una marca de identidad, de hecho este ultimo tiene huellas únicas e irrepetibles

Study Case

- ➔ The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.
- ➔ A department controls a number of projects, each of which has a unique name, a unique number, and a single location.
- ➔ We store each employee’s name, Social Security number,2 address, salary, sex(gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. We keep track of the current number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee (who is another employee).
- ➔ We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent’s first name, sex, birth date, and relationship to the employee.

Study Case


Fname
Minit
Lname
Bdate
Name
Address
Salary
Ssn
Sex
N
WORKS_FOR
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
1
MANAGES
1
CONTROLS
N
Hours
M
N
WORKS_ON
PROJECT
Supervisor
Supervisee
1
1
SUPERVISION
N
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship

Study Case (Taller #1)

Conceptual Model

CE3101 - Databases