



# DATABASES

**GS WEB APPLICATION DEVELOPMENT**

Teacher: Vanesa Maldonado Guerrero  
Curs 2025/26

# Unit 2. Conceptual Design: Entity–Relationship Model

Description of Conceptual Design

Before Getting Started

Entity–Relationship Model (ERM)

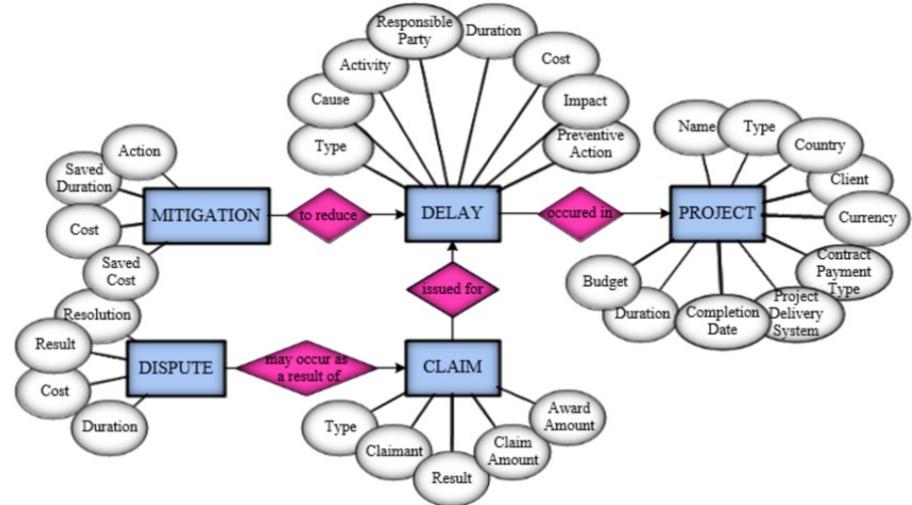
Entity, Relationship, and Attributes

Key, Cardinality

Types of Entities

Chen Notation Variants

Entity–Relationship Model Exercise



# Before Getting Started



- **GitMind**
- **Gliffy**
- **Visual Paradigm**
- **Draw.io**
- **Lucidchart**
- **SqlDBM**
- **DBDiagram.io**
- **QuickDBD**
- **Dia Diagram Editor**

# Conceptual Design

Describe the data that the user wants to collect for the development of a system.



Requires analysis



**Conceptual design** is the first stage in the database development process. Its purpose is to create an abstract representation of reality, aiming to capture all the information from the real world that should be represented in the computer system.



- ❖ Be independent of implementation models or languages
- ❖ Have a high semantic capacity
- ❖ Be as close as possible to human understanding

# Entity-Relationship Model (ERM)



Peter Chen



Massachusetts  
Institute of  
Technology

1976

“Entity–Relationship Model: Toward  
a Unified View of Data”

# Entity-Relationship Model (ERM)

## Entity

An entity represents an object, concept, or element from the real world that is stored in a database.

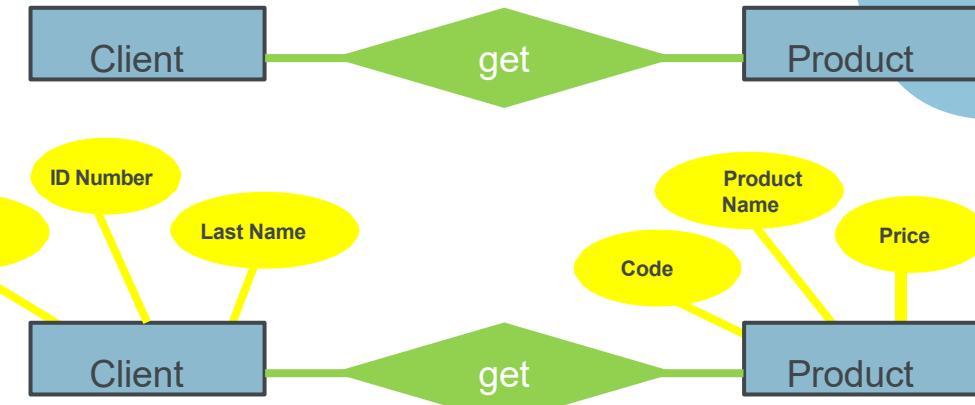
Entities are represented by a rectangle with their name inside.



## Relationship

Relationships represent the associations or connections between entities.

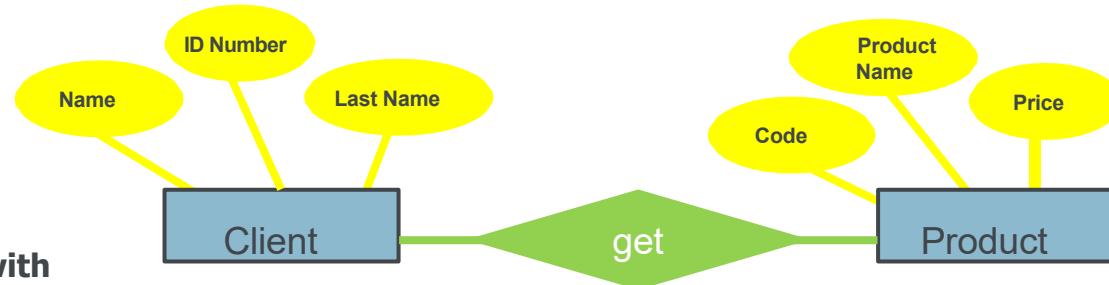
In the E-R model, they are depicted as diamonds and labeled with a descriptive name.



## Attribute

Attributes are the characteristics or properties of an entity.

They are represented as ovals and are connected to the entities they belong to with lines.



# Entity-Relationship Model (ERM)

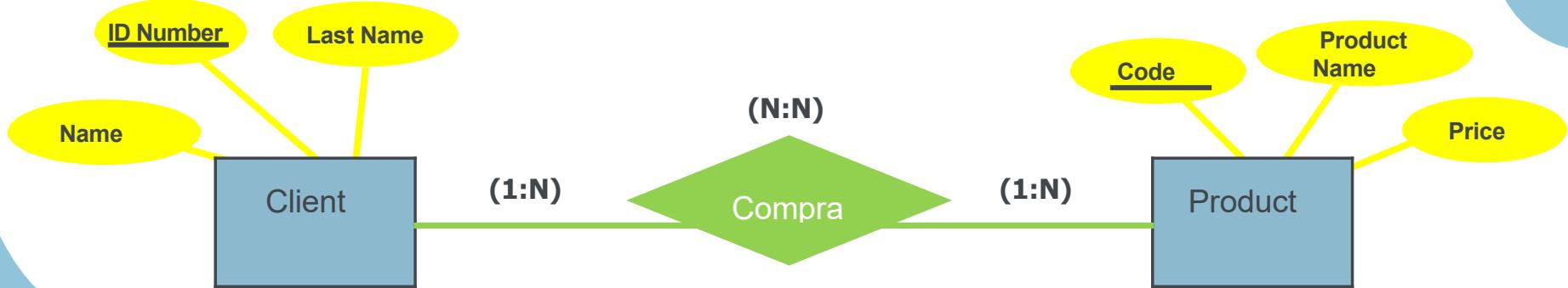
## Key

A key refers to a set of one or more attributes used to uniquely identify each instance or record in an entity within the database.

## Cardinality

The cardinality of a relationship describes how many instances of one entity can be associated with an instance of another entity through that relationship.

Common terms used to describe cardinality are one-to-one (1:1), one-to-many (1:N), and many-to-many (N:N).



# Entity-Relationship Model (ERM)

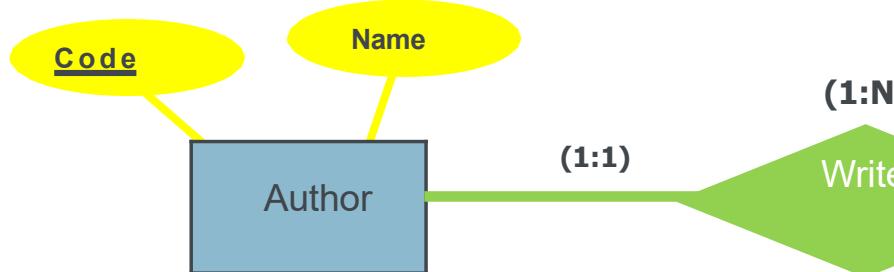
## Types of Entities

These terms are used to describe an entity's ability to exist independently in a database and its dependence on another entity for identification.

### Strong Entities

These are entities that do not require or depend on the existence of another entity to store data.

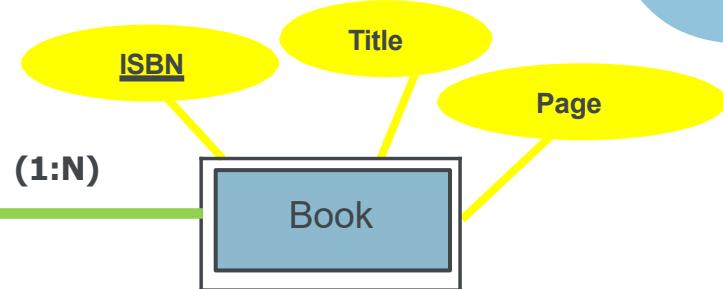
They are still represented with a rectangle.



### Weak Entities

These are entities that cannot exist independently and **DEPEND** on a strong entity for their identification.

They are usually represented with a double rectangle in the E-R model.



# Entity

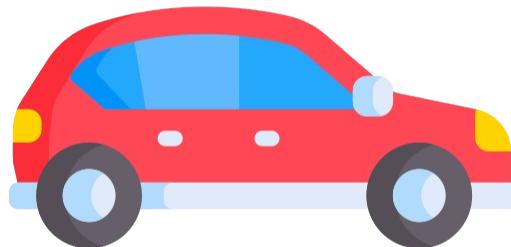
Person, Object, Concept, or Event



Think of entities as nouns



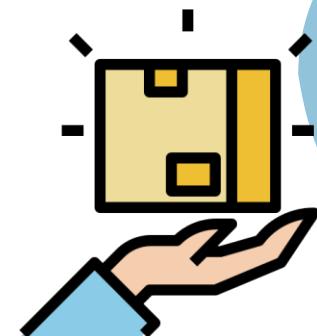
Client



Car



Student



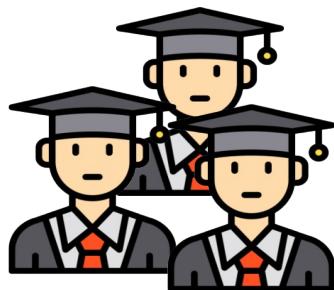
Product

# Entity

Person, Object, Concept, or Event

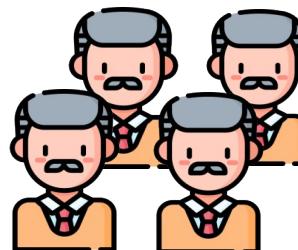


Think of entities as if they were nouns.



Student

Students



Teacher

Teacher

# Entity

**Person, Object, Concept, or Event**



**Think of entities as if they were nouns**



**The entity we identify is something about which we can store information.**



Student



Subject

# Entity

## Common Examples of Entities:

- ❖ **Person: Student, Passenger, Teacher, Customer**
- ❖ **Institutions: Bank, Company, University**
- ❖ **Organizational Units: Department, Branch, Plant, Line**
- ❖ **Classifications, Groups, and Hierarchies: Type, Class, Brand, Group, Category**
- ❖ **Documents: Invoice, Order, Purchase Order, Check**
- ❖ **Objects (physical or abstract): Material, Product, Subject, Skill**

# Entity

Hospital



Patient

Juan Jose

Av. Mayor 100

Doctor

Specialty

Cardiology

# Entity

Hospital



Patient

Juan Jose

Av. Mayor 100

Doctor

Specialty

Cardiology

Patient

Doctor

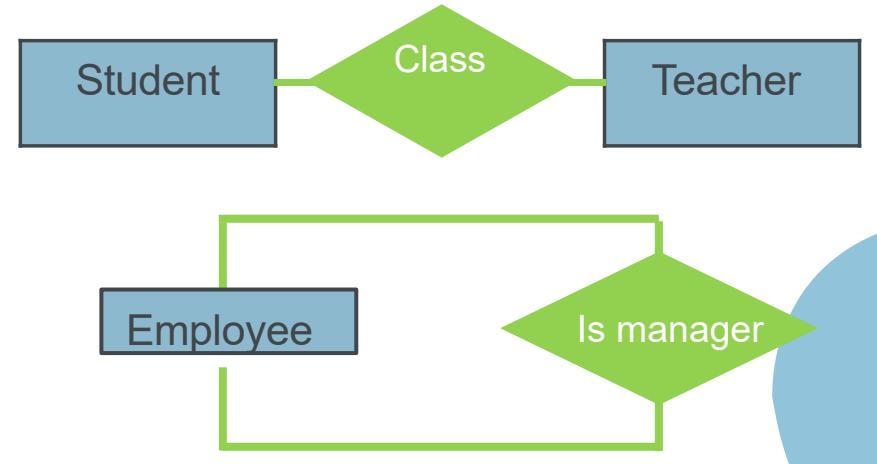
Specialty

# Relationship

A relationship represents an **association or connection between two or more entities**.

Relationships are used to model how entities interact and are related to each other in a database.

In other words, they describe how objects or concepts from the real world are **connected** in the information system being modeled.



A relationship has the following characteristics:

- ❖ Relationships are **graphically represented by diamonds**, with their name inside.
- ❖ The name of the diamond is usually a **verb or action** that connects the entities
- ❖ A relationship can associate an entity with **another entity** or with **itself**.

# Relationship

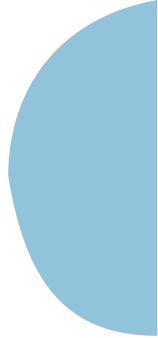
Hospital



Patient

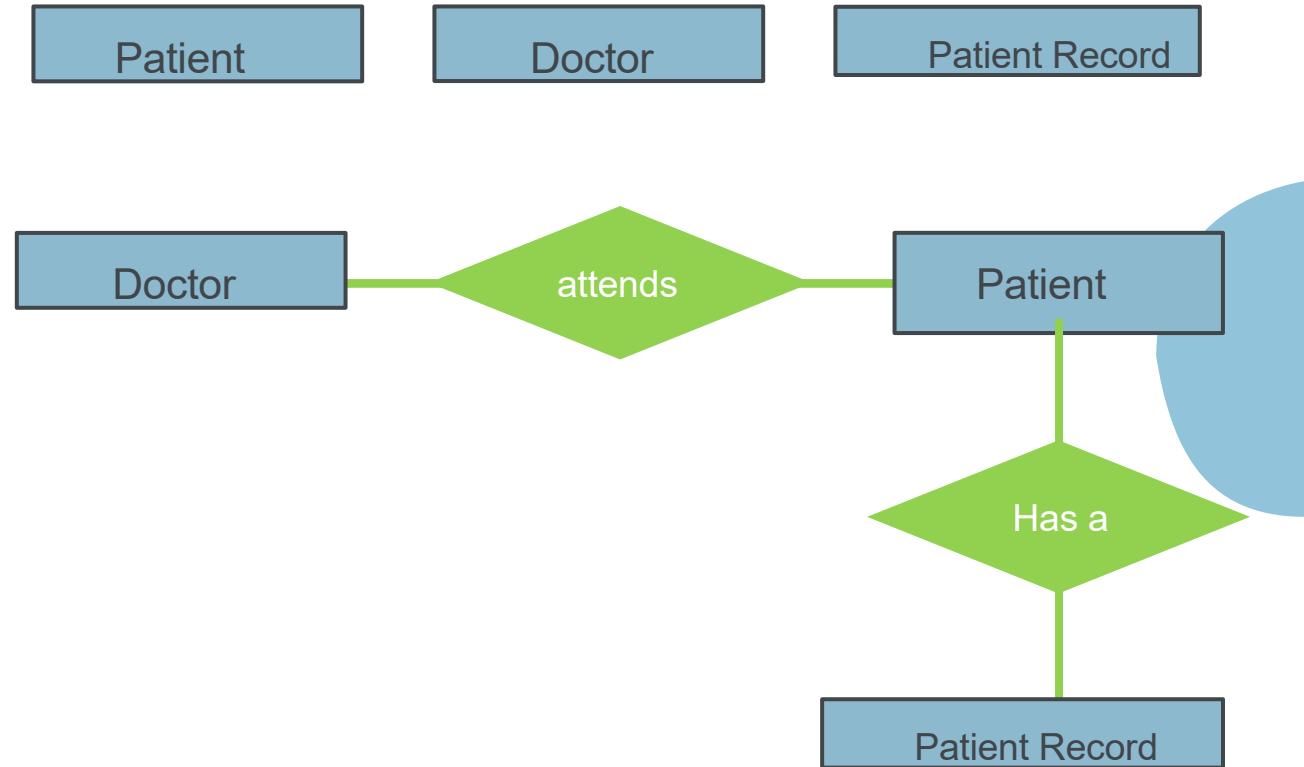
Doctor

Patient Record



# Relationship

# Hospital



# Relationship

Ecommerce



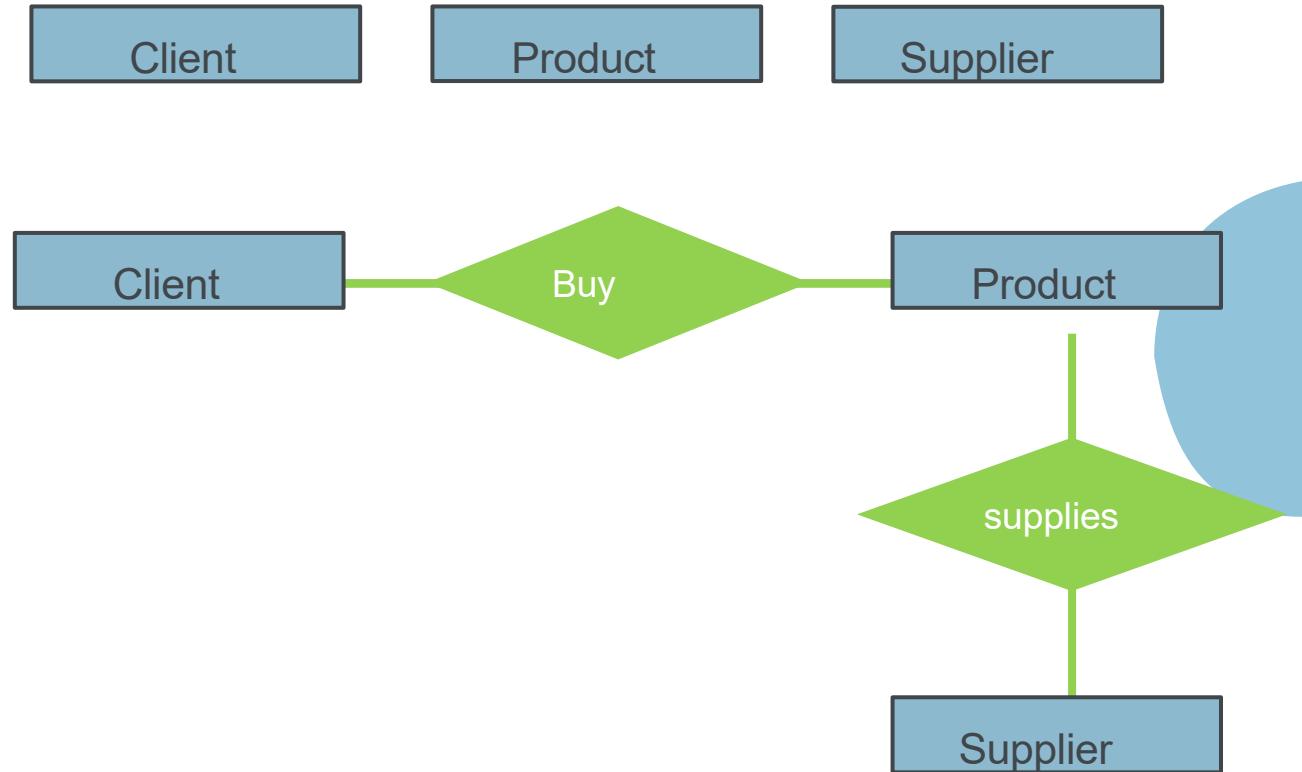
Client

Product

Supplier

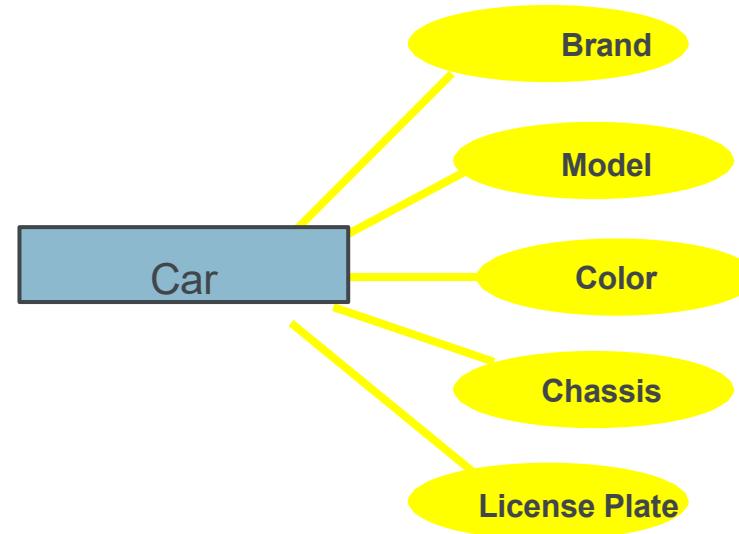
# Relationship

## Ecommerce



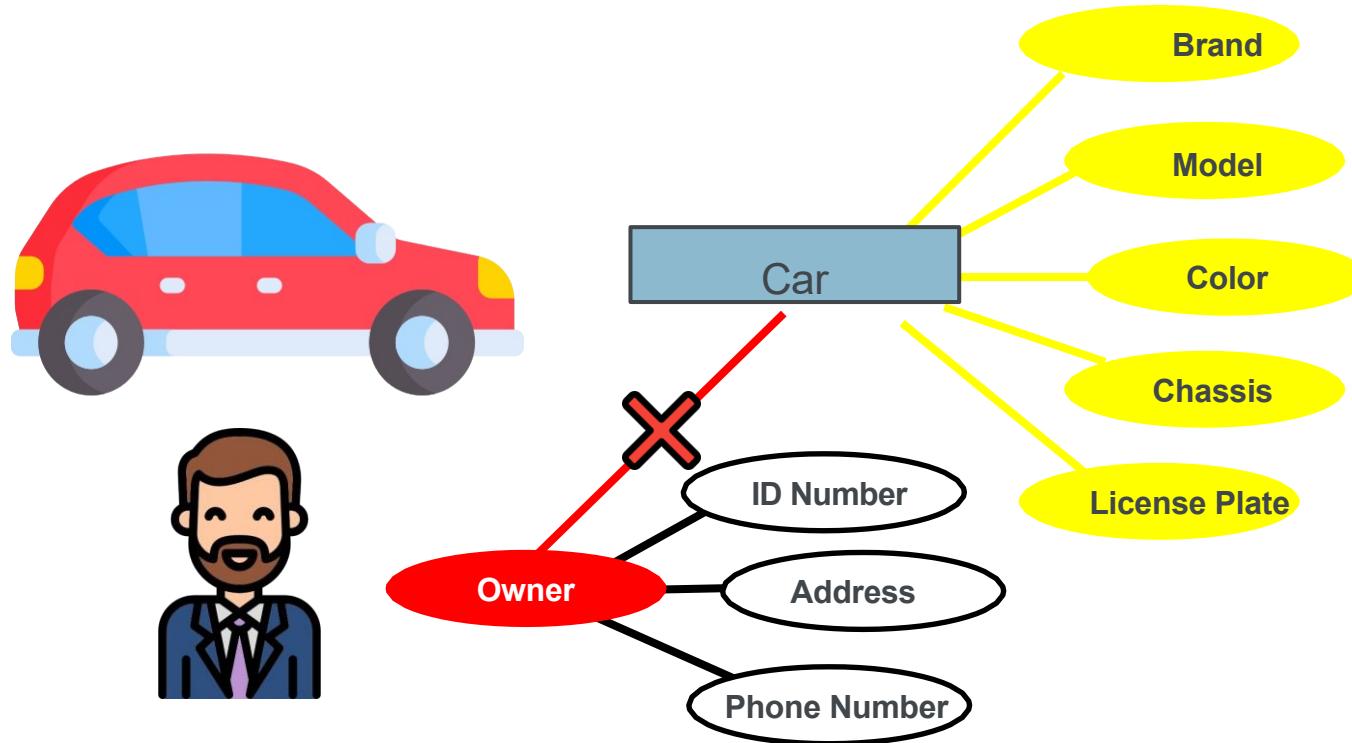
# Attributes

**They are the characteristics by which an entity can be described.**



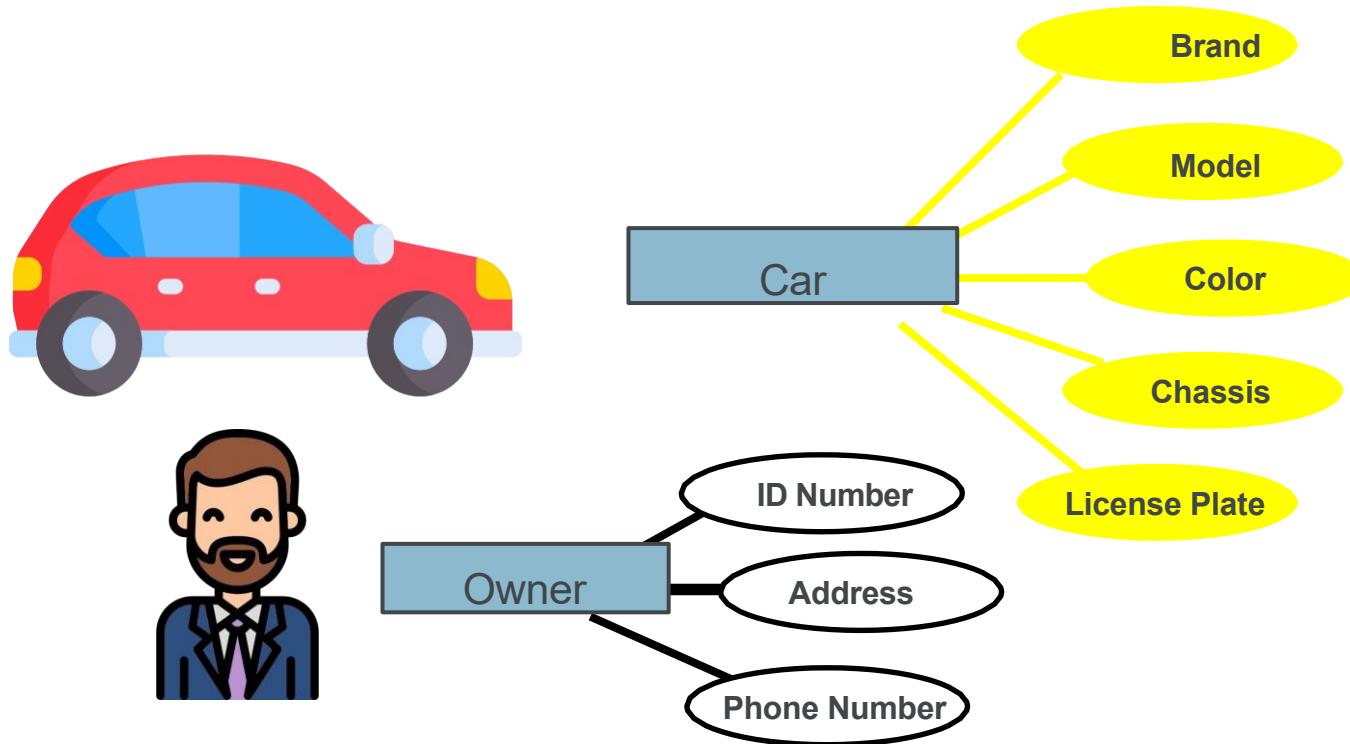
# Attributes

They are the characteristics by which an entity can be described.



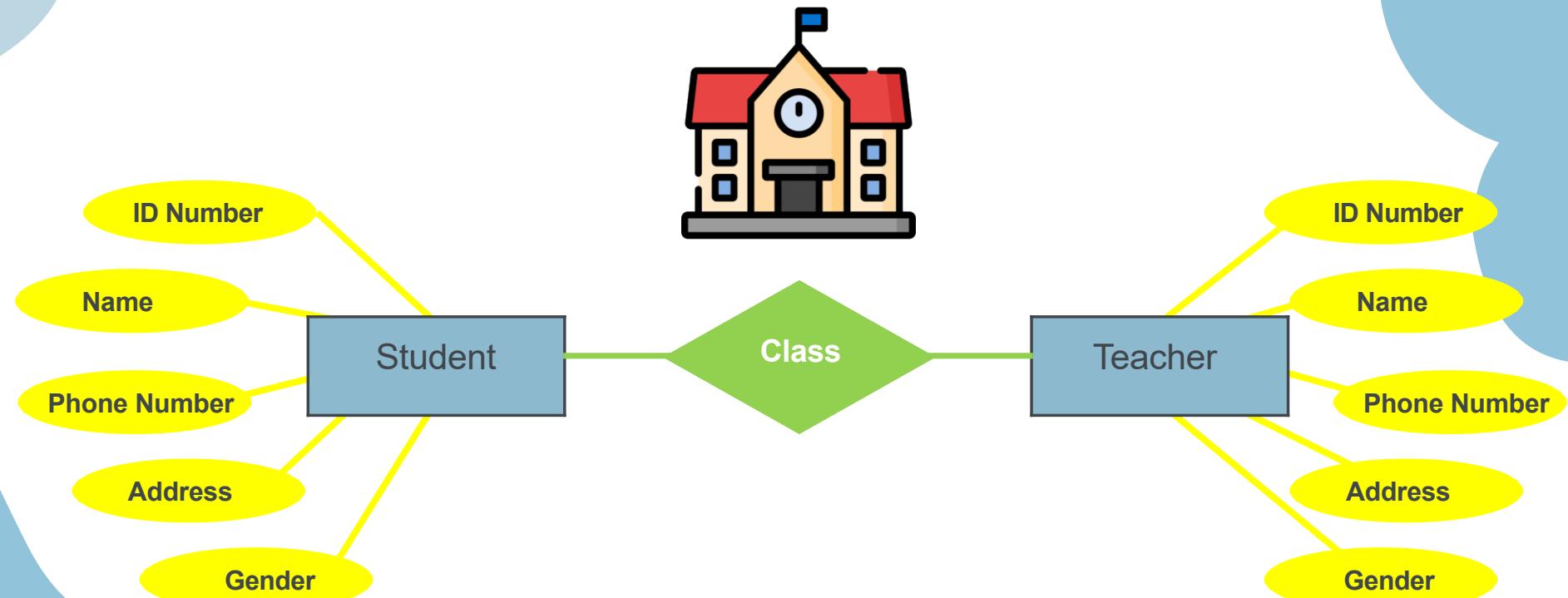
# Attributes

They are the characteristics by which an entity can be described.



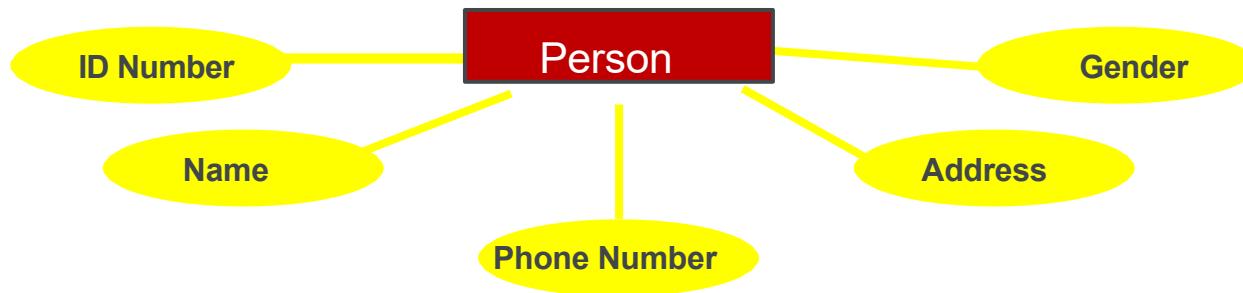
# Attributes

They are the characteristics by which an entity can be described.



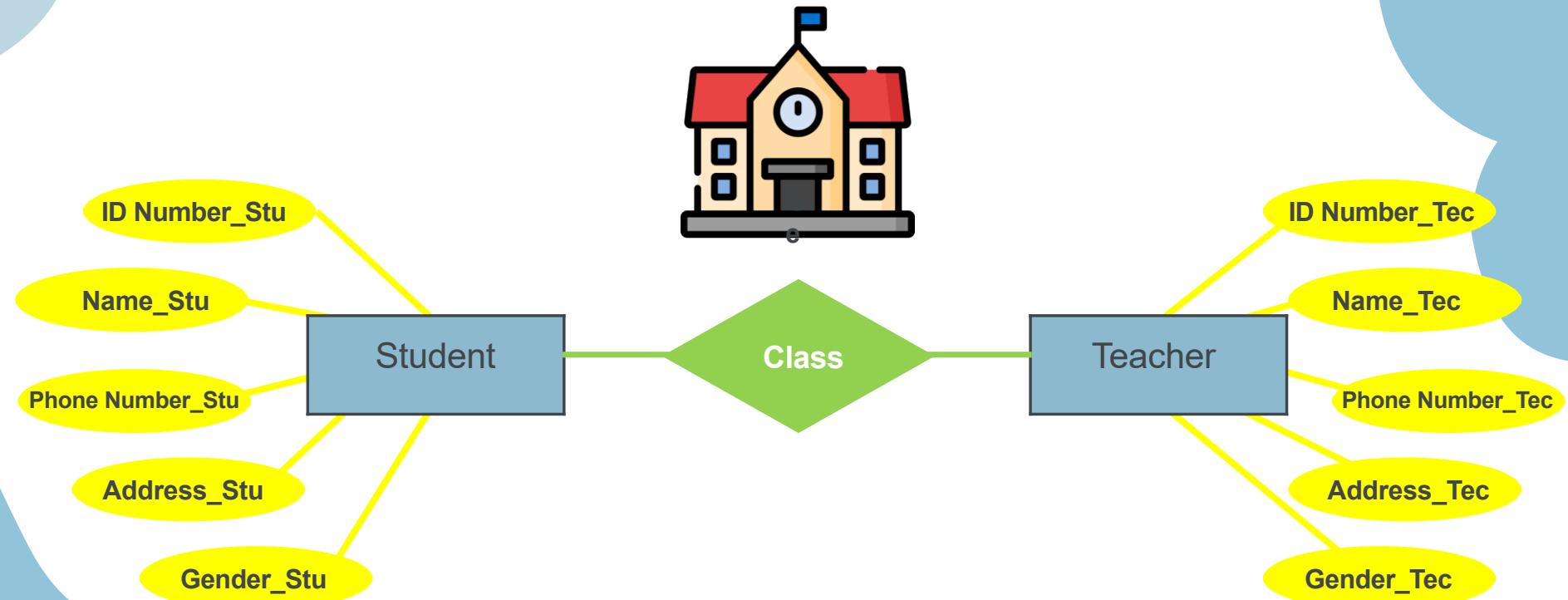
# Attributes

They are the characteristics by which an entity can be described.

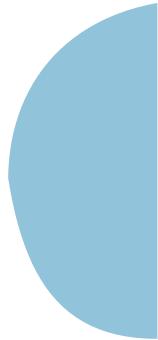


# Attributes

They are the characteristics by which an entity can be described.



# Let's practice



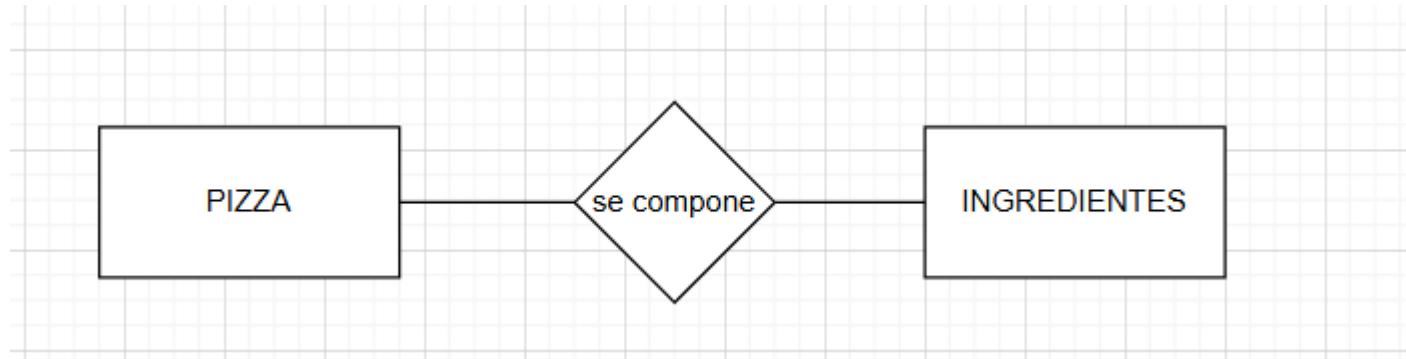
# Let's practice

## Ejercicio 1: Ingredientes de una pizza

Diseña un pequeño diagrama E-R para el siguiente fragmento del sistema: La cadena de pizzerías tiene una carta de pizzas donde cada pizza contiene varios ingredientes.

# Let's practice

## Ejercicio 1 Solución:



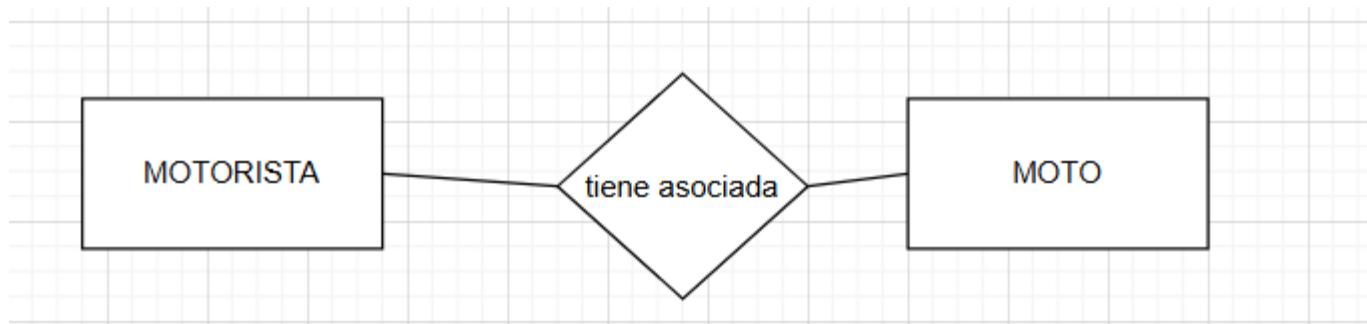
# Let's practice

## Ejercicio 2:

En una pizzería, cada empleado motorista tiene asociada una moto, pero una moto es compartida por varios motoristas de varios turnos.

# Let's practice

## Ejercicio 2 Solución:



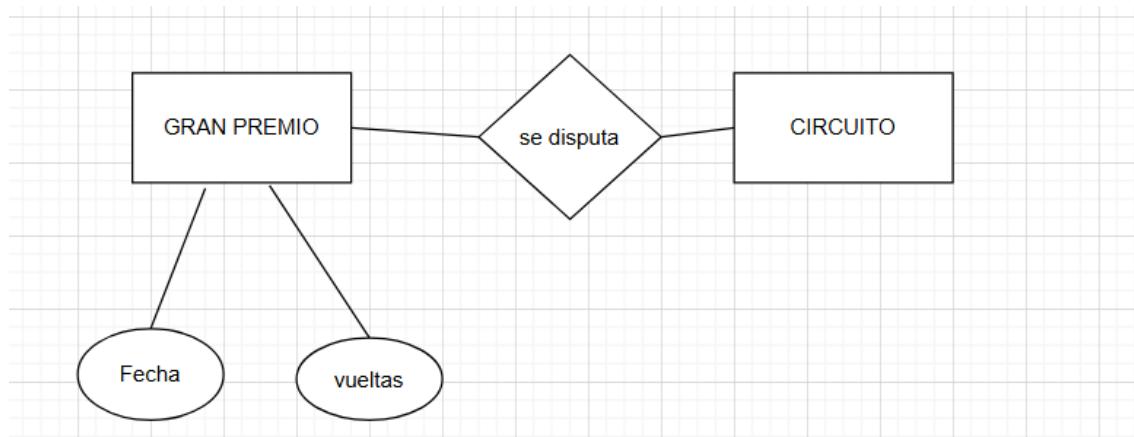
# Let's practice

## Ejercicio 3:

Cada Gran Premio de F1 tiene lugar en un Circuito concreto. Es necesario saber el número de vueltas a completar para terminar cada Gran Premio, así como la fecha en que éste tendrá lugar.

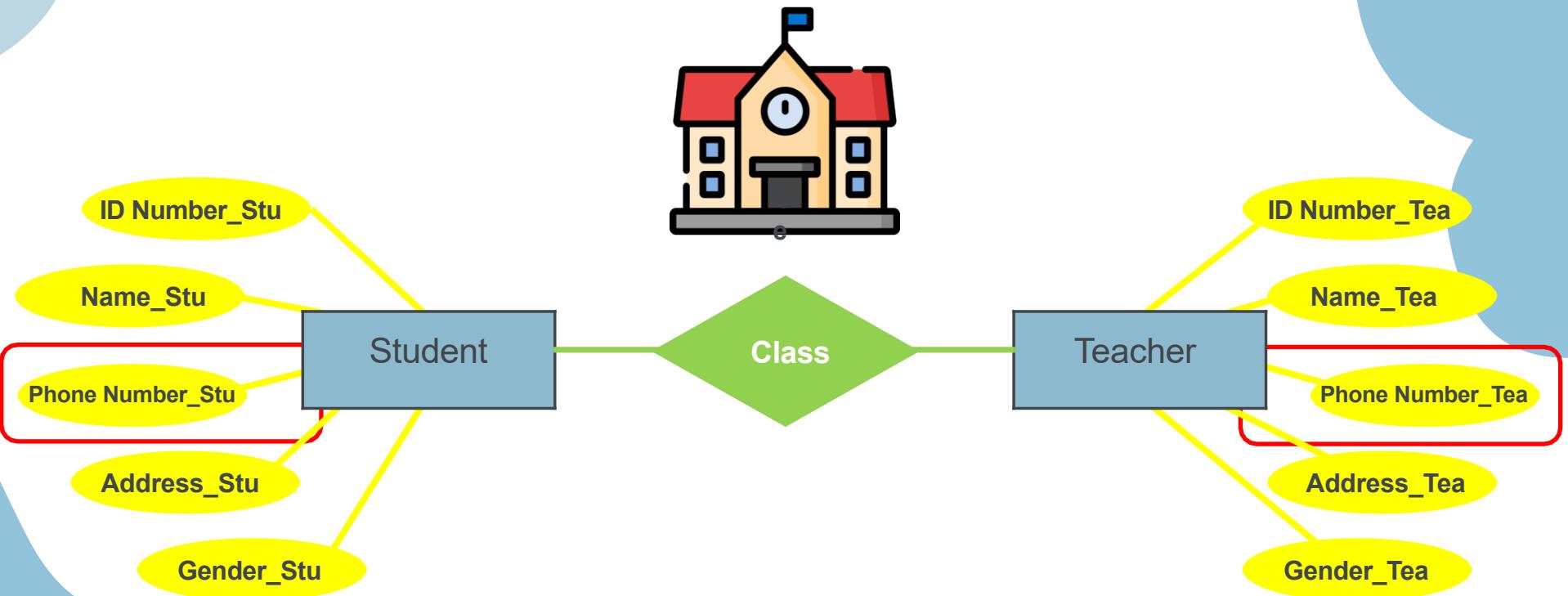
# Let's practice

## Ejercicio 3 Solución:



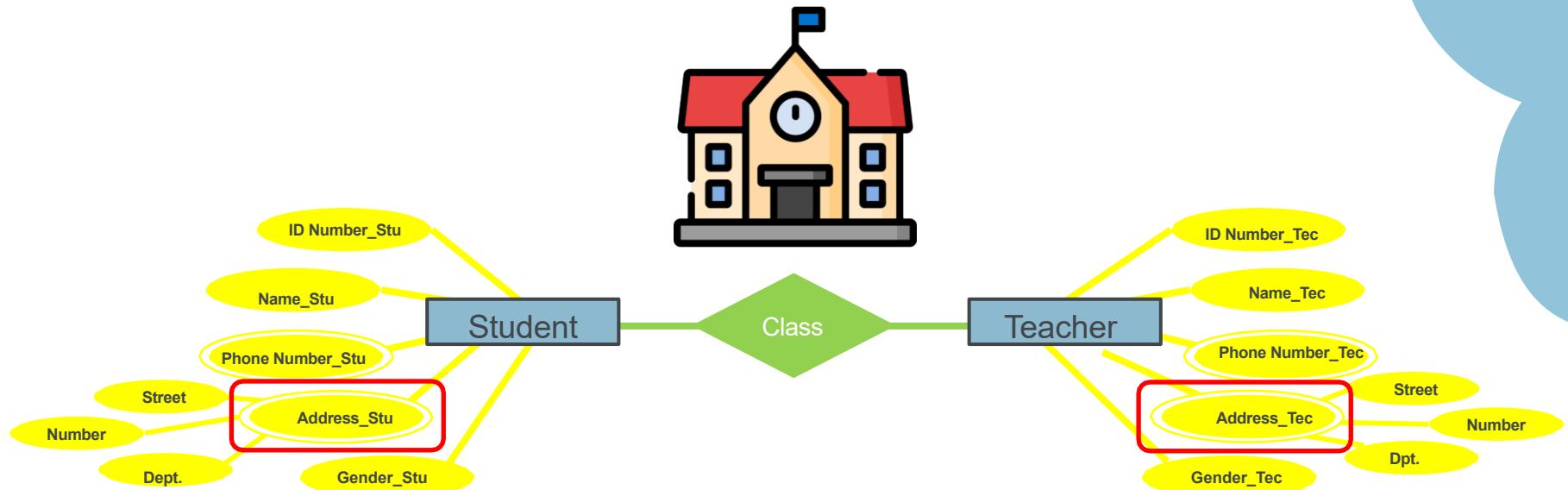
# Multivalued attributes

They are also often called 'multi-concurrent' and are those that can have more than a single value.



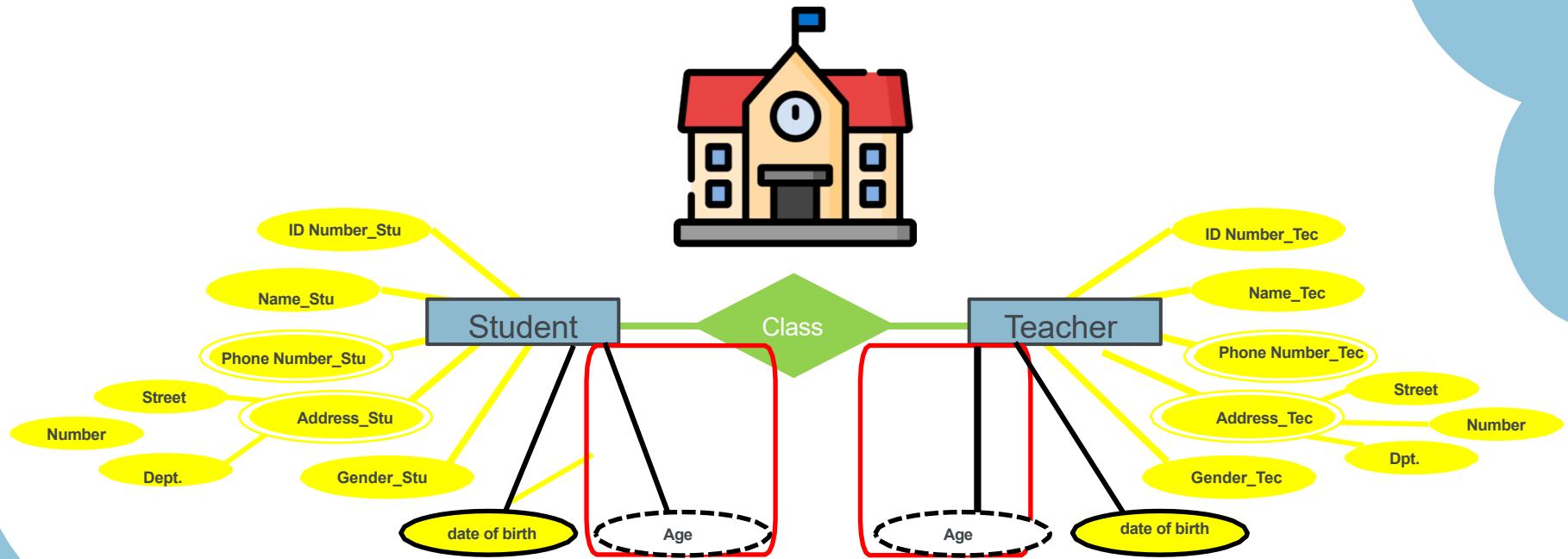
# Composite attributes

They are those that are derived from the combination of other attributes, can have multiple values, and are represented in the same way as multivalued attributes.



# Derived Attributes

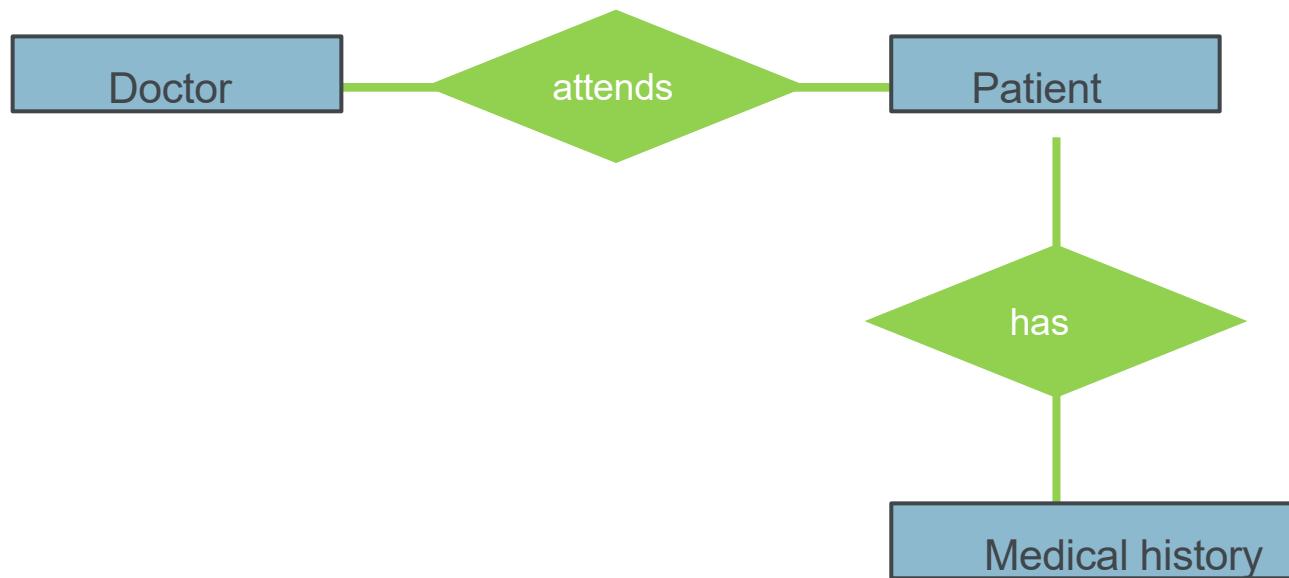
These are those that are obtained from other attributes of the Entity and are represented with a dotted ellipse.



# Hospital



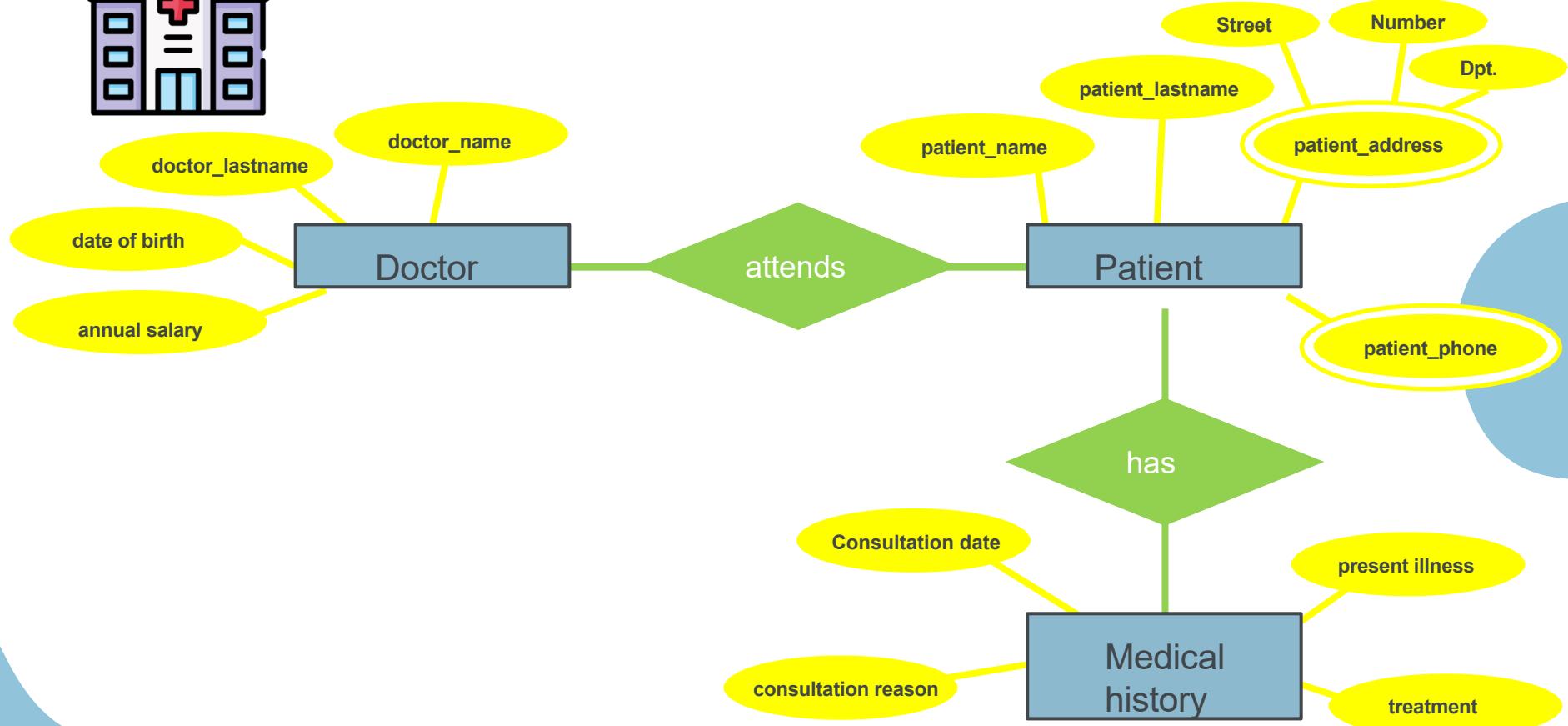
# Attributes



# Hospital



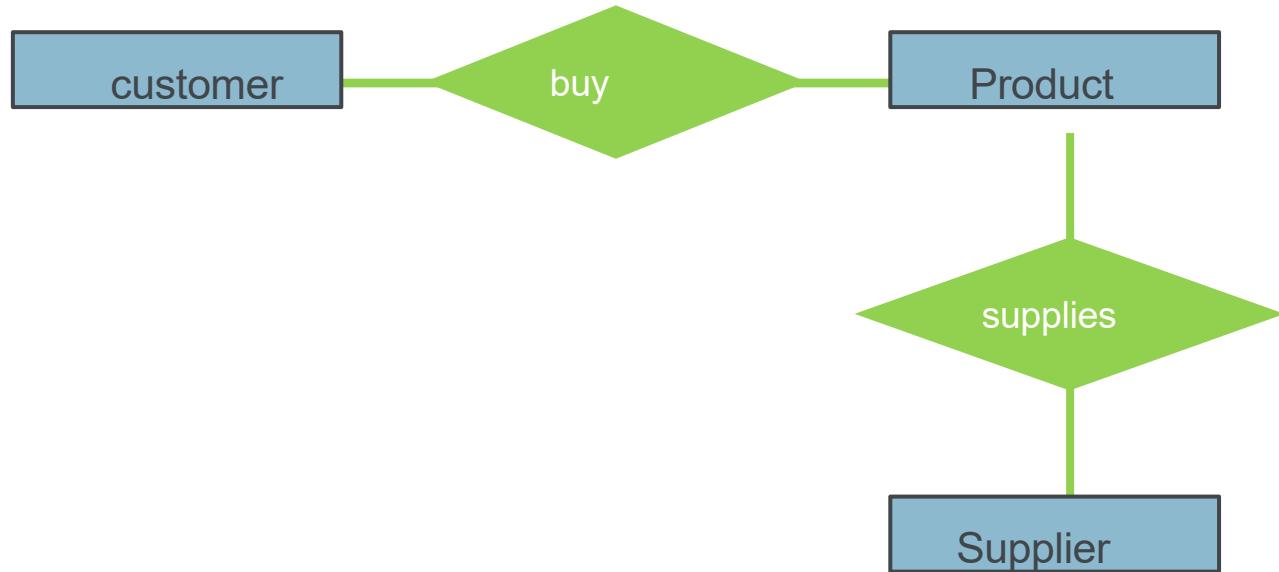
# Attributes



# Ecommerce

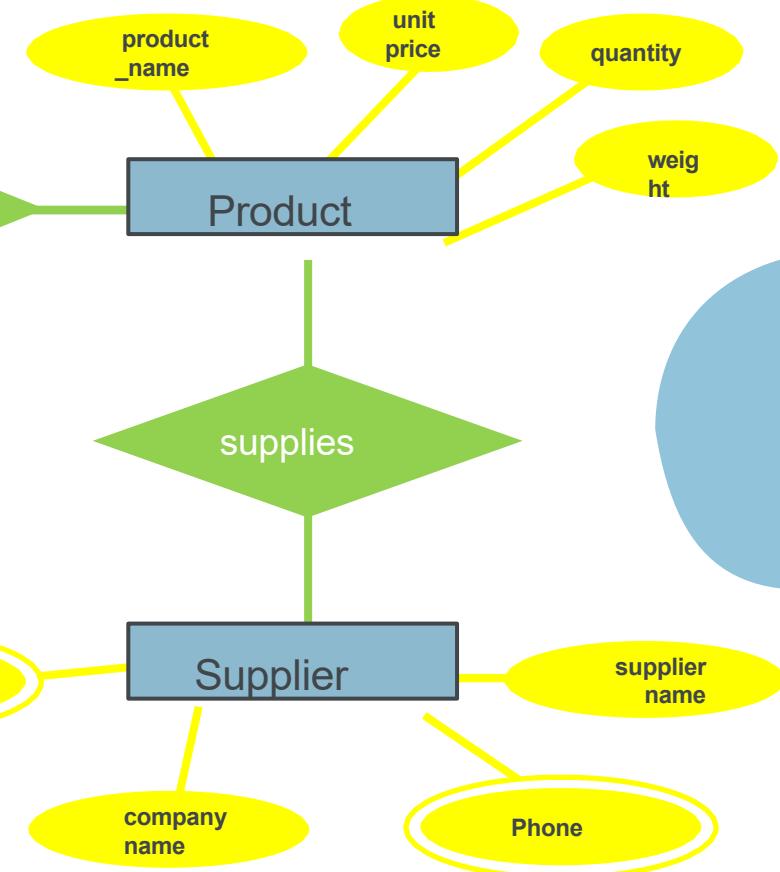
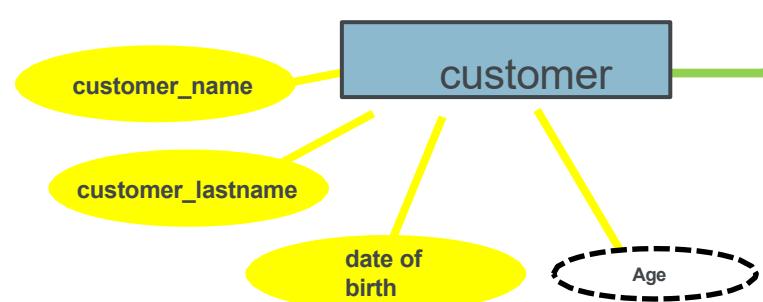


# Attributes



# Ecommerce

## Attributes



# key

Keys are the attribute or attributes that uniquely identify the values of an entity and have the following characteristics:



- ❖ The key value cannot be repeated
- ❖ The key value must not be null.

There are three types of keys:

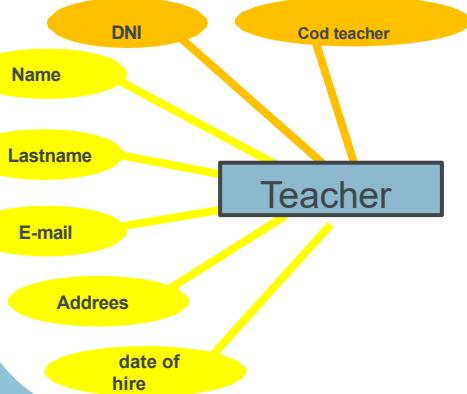
## Candidate Key



They are attributes or sets of attributes that have the property of uniquely identifying the entity.

### Teacher

- **Cod teacher**
- **DNI**
- **Name**
- **Lastname**
- **E-mail**
- **Address**
- **Date of hire**



TEACHER						
Cod teacher	DNI	Name	LastName	E-mail	Address	Date_hire
1234	99887766	Carla	Díaz	<a href="mailto:carla@example.com">carla@example.com</a>	av. Mexico 11	10/09/2023
2384	11223344	Hugo	Torres	<a href="mailto:hugo@example.com">hugo@example.com</a>	Jr. Historia 22	01/01/2020
3322	77223366	Jessy	Campos	<a href="mailto:jessy@example.com">jessy@example.com</a>	av. Brasil 99	02/05/2024



TEACHER						
Cod teacher	DNI	Name	LastName	E-mail	Address	Date_hire
1234	99887766	Carla	Díaz	<a href="mailto:carla@example.com">carla@example.com</a>	av. Mexico 11	10/09/2023
2384	11223344	Hugo	Torres	<a href="mailto:hugo@example.com">hugo@example.com</a>	Jr. Historia 22	01/01/2020
3322	77223366	Jessy	Campos	<a href="mailto:jessy@example.com">jessy@example.com</a>	av. Brasil 99	02/05/2024
1122	99112233	Carla	Ruiz	<a href="mailto:carlaR@example.com">carlaR@example.com</a>	av. Mexico 33	10/12/2023



# key

Keys are the attribute or attributes that uniquely identify the values of an entity and have the following characteristics:

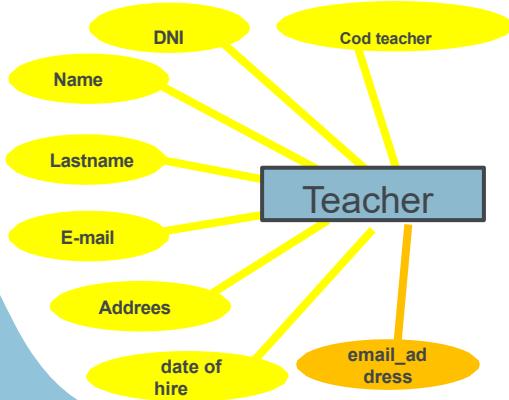


- ❖ The key value cannot be repeated
- ❖ The key value must not be null.

There are three types of keys:

## Composite Key

It is a key that is formed by two or more attributes instead of a single attribute.



## Teacher

- **Cod teacher**
- **DNI**
- **Name**
- **Lastname**
- **E-mail**
- **Address**
- **Date of hire**
- **address, E-mail**

TEACHER							
Email_addee ss	Cod teacher	DNI	Name	LastName	E-mail	address	Date_hire
Carla carla@example.co m	1234	99887766	Carla	Díaz	<a href="mailto:carla@example.com">carla@example.com</a>	av. Mexico 11	10/09/2023
Hugo hugo@example.co m	2384	11223344	Hugo	Torres	<a href="mailto:hugo@example.com">hugo@example.com</a>	Jr. Historia 22	01/01/2020
Jessy jessy@example.co m	3322	77223366	Jessy	Campos	<a href="mailto:jessy@example.com">jessy@example.com</a>	av. Brasil 99	02/05/2024



# key

Keys are the attribute or attributes that uniquely identify the values of an entity and have the following characteristics:



- ❖ The key value cannot be repeated
- ❖ The key value must not be null.

There are three types of keys:

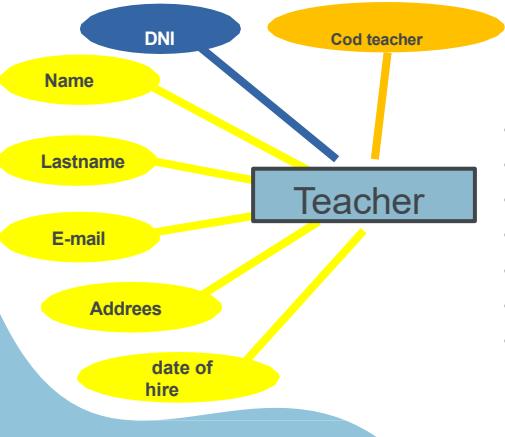
## Primary Key



The primary key is an attribute used to uniquely identify each record in an entity.

## Teacher

- **Cod teacher**
- **DNI**
- **Name**
- **Lastname**
- **E-mail**
- **Address**
- **Date of hire**



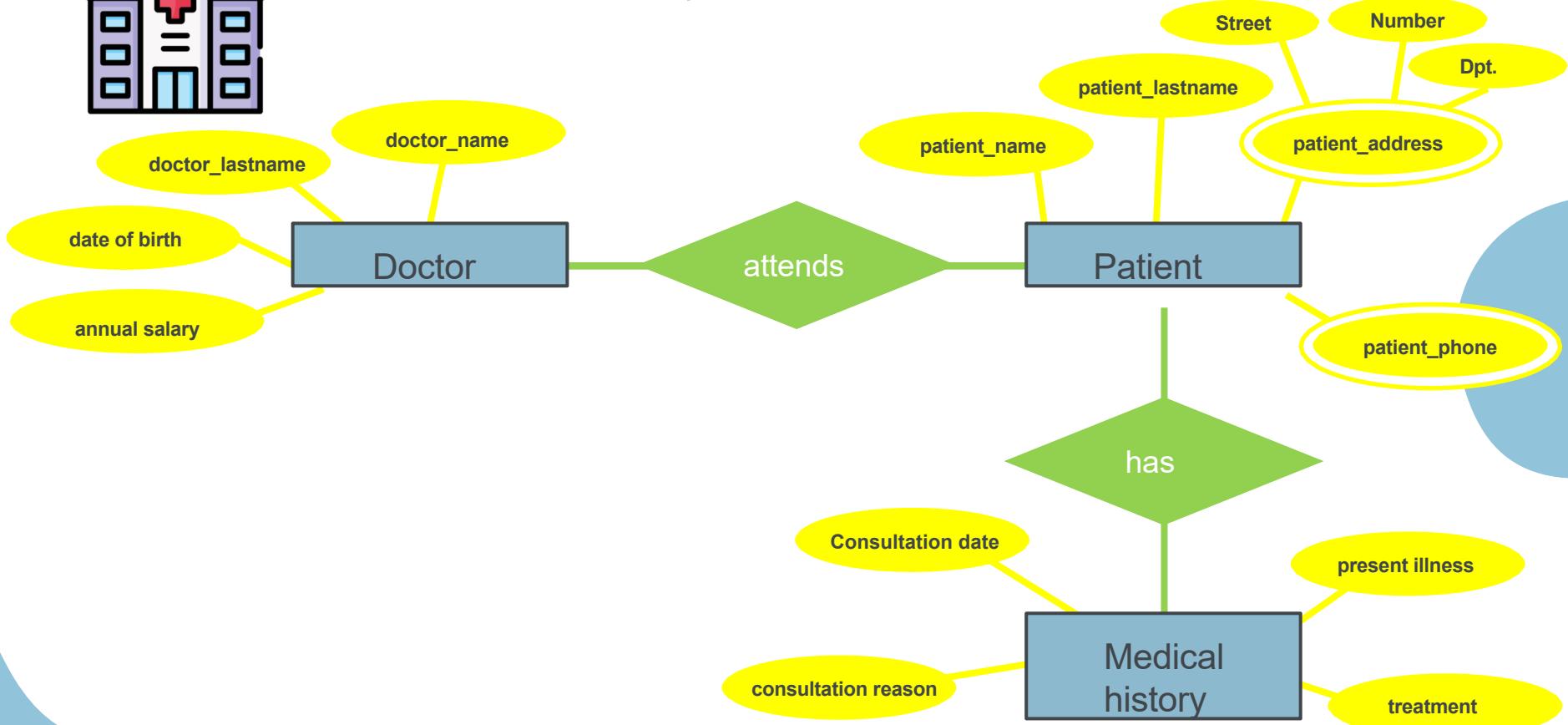
TEACHER						
Cod teacher	DNI	Name	LastName	E-mail	Address	Date_hire
1234	99887766	Carla	Díaz	<a href="mailto:carla@example.com">carla@example.com</a>	av. Mexico 11	10/09/2023
2384	11223344	Hugo	Torres	<a href="mailto:hugo@example.com">hugo@example.com</a>	Jr. Historia 22	01/01/2020
3322	77223366	Jessy	Campos	<a href="mailto:jessy@example.com">jessy@example.com</a>	av. Brasil 99	02/05/2024



# Hospital



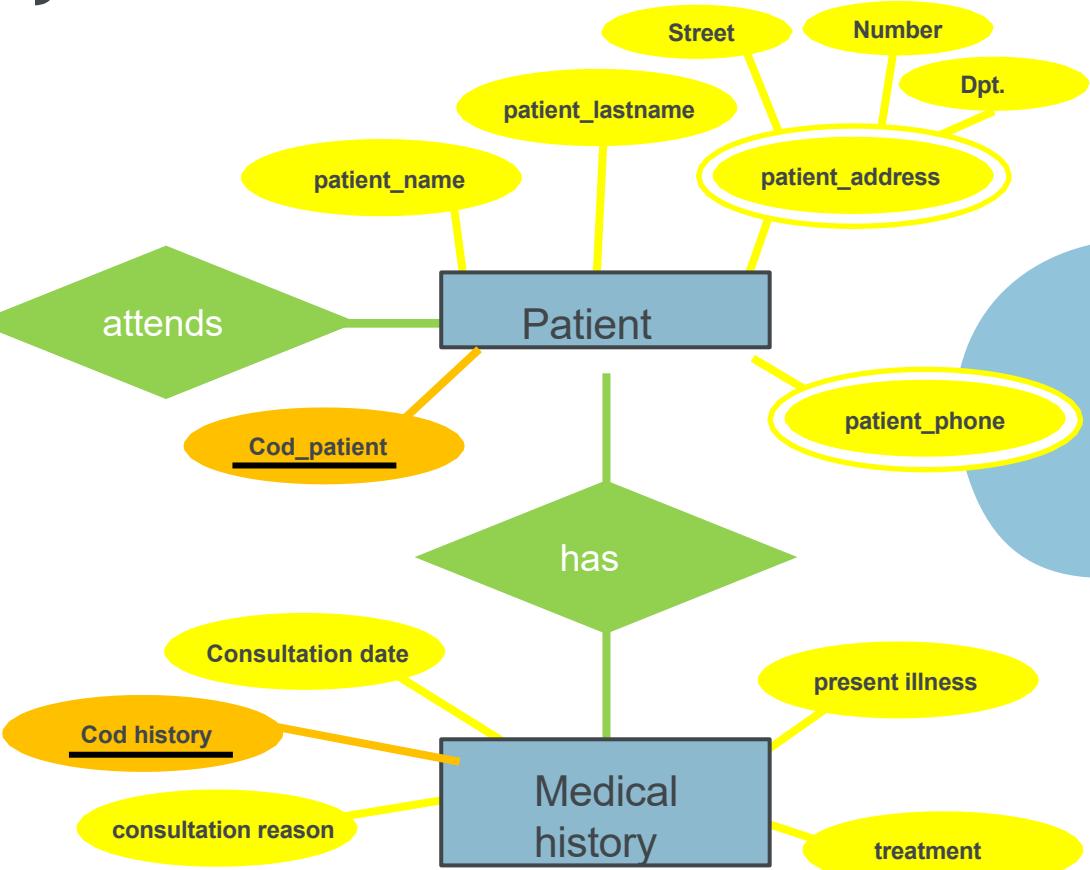
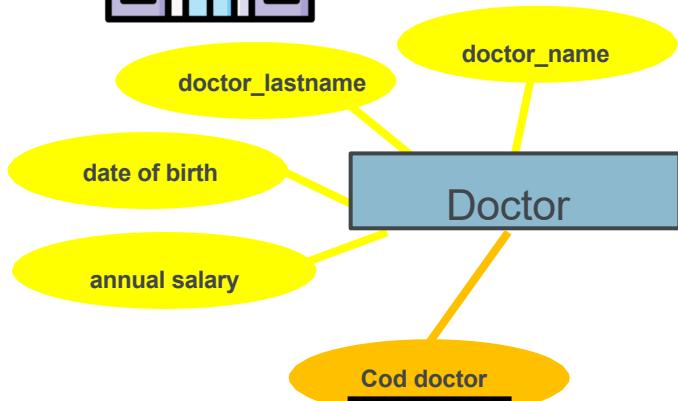
# Key



# Hospital

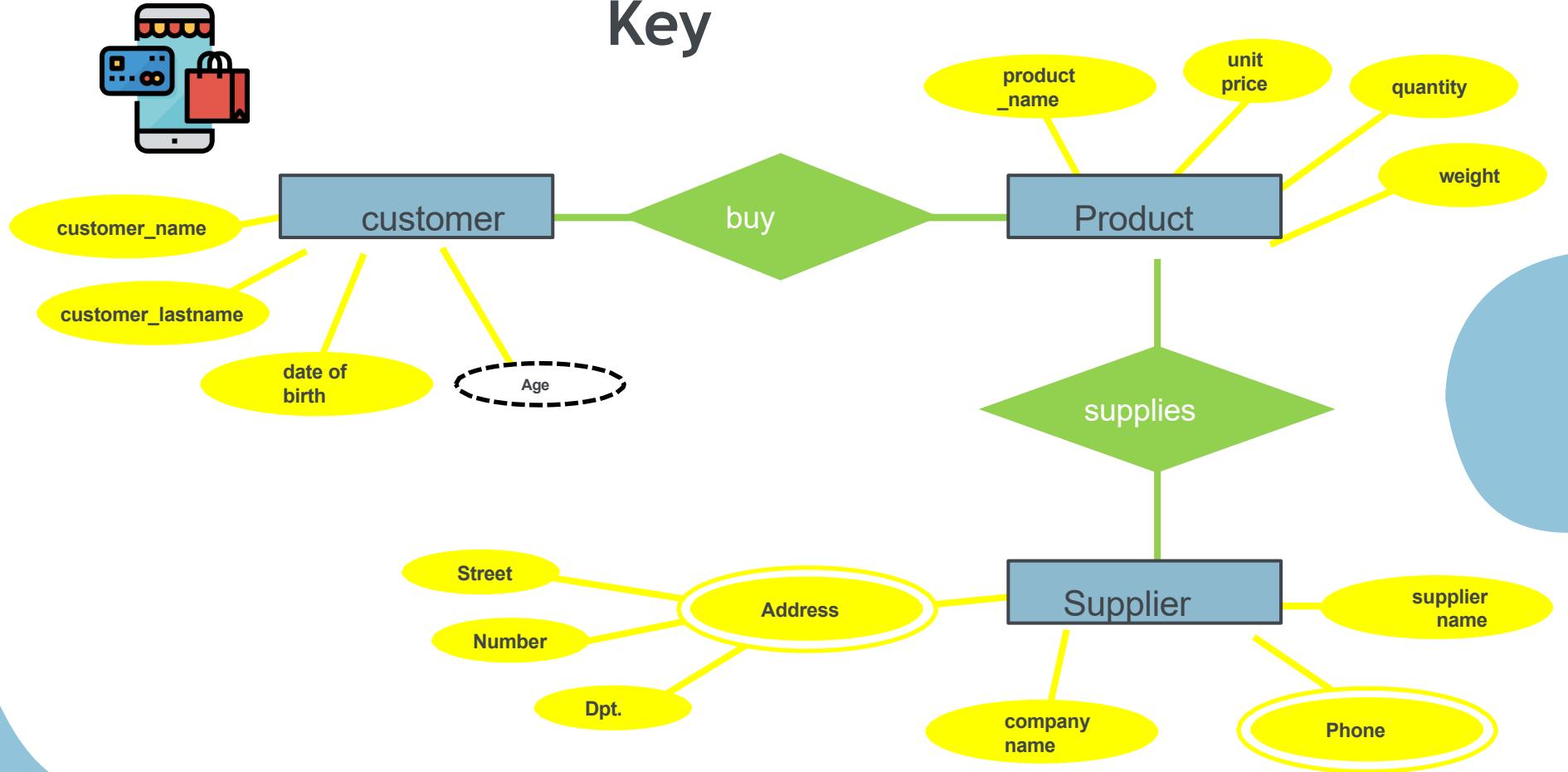


# Key



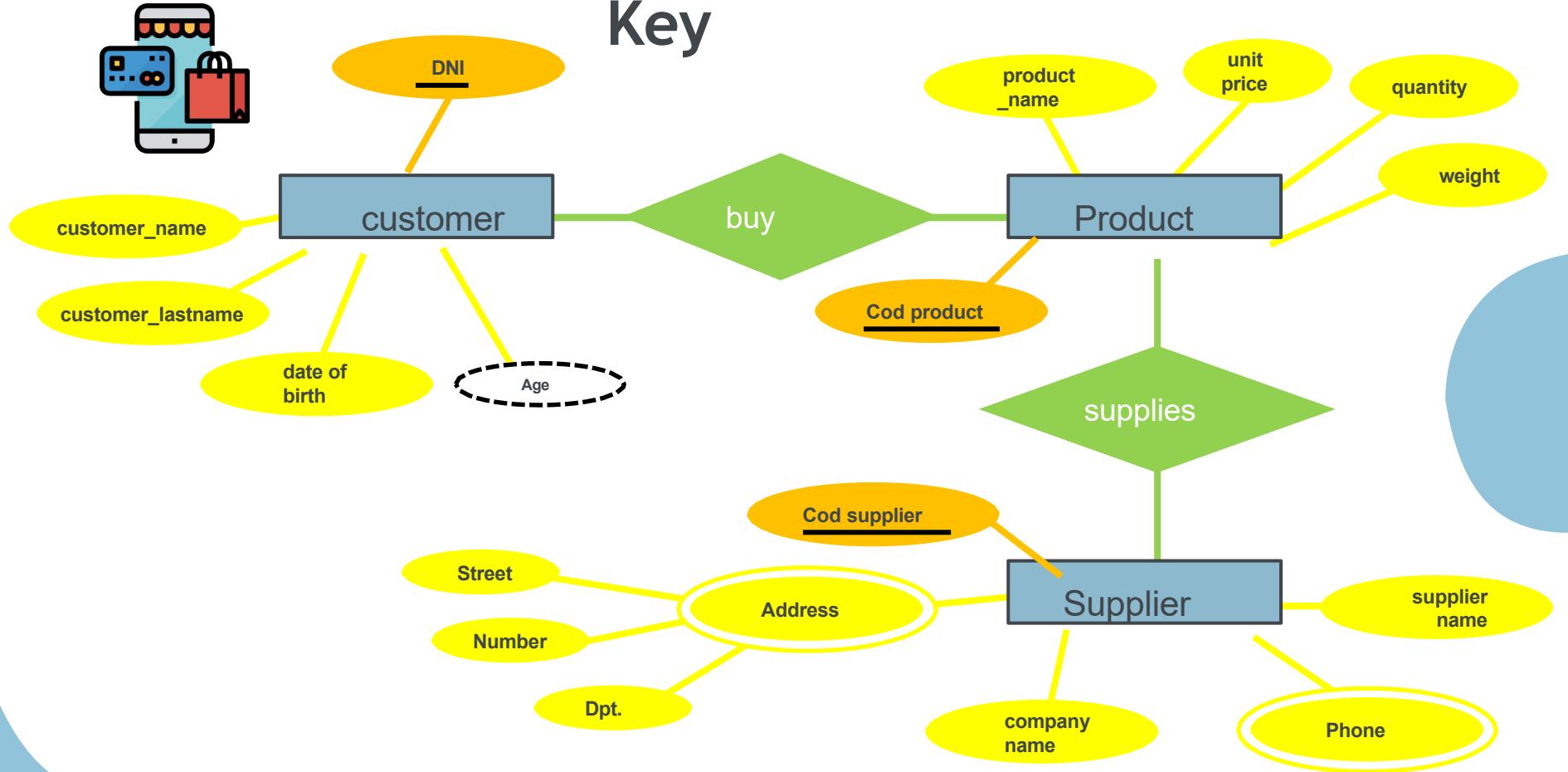
# Ecommerce

# Key



# Ecommerce

## Key



# Cardinality

Cardinality refers to the way entities are related to each other.



Cardinality is used to define the number of occurrences allowed in a relationship between entities.

We have the following types:

Relationship  
Cardinality



The cardinality of a relationship refers to the number of instances of one entity that can be related to one or more instances of another entity through that specific relationship.

1:1  
1:N  
N:N



# Cardinality

Cardinality refers to the way entities relate to each other.



Cardinality is used to define the number of occurrences allowed in a relationship between entities.

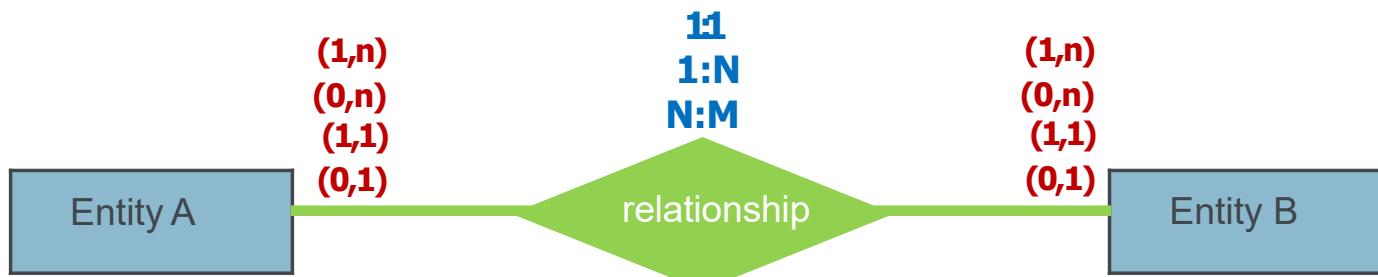
We have the following types:

Entity  
Cardinality



It indicates the minimum and maximum number of associations in which each instance of an entity can participate in a given relationship.

(Minimum cardinality, Maximum cardinality)



# Cardinality

Cardinality refers to the way entities relate to each other.



Cardinality is used to define the number of occurrences allowed in a relationship between entities.

Cardinality “1 : 1”

(Minimum cardinality, Maximum cardinality)



# Cardinality

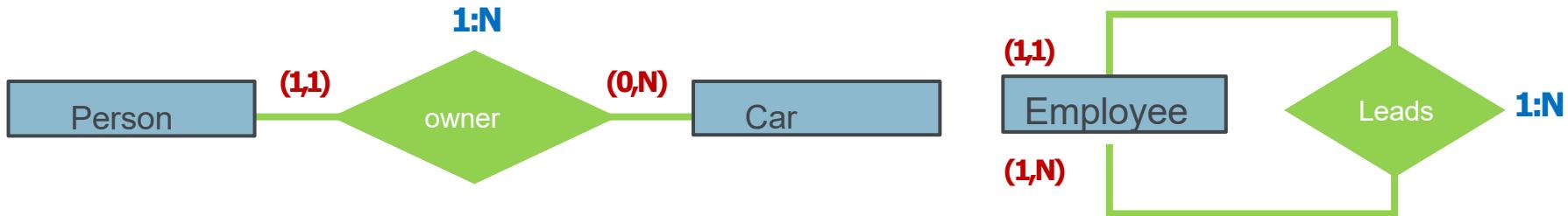
Cardinality refers to the way entities relate to each other.



Cardinality is used to define the number of occurrences allowed in a relationship between entities.

Cardinality "1 : N"

(Minimum cardinality, Maximum cardinality)



# Cardinality

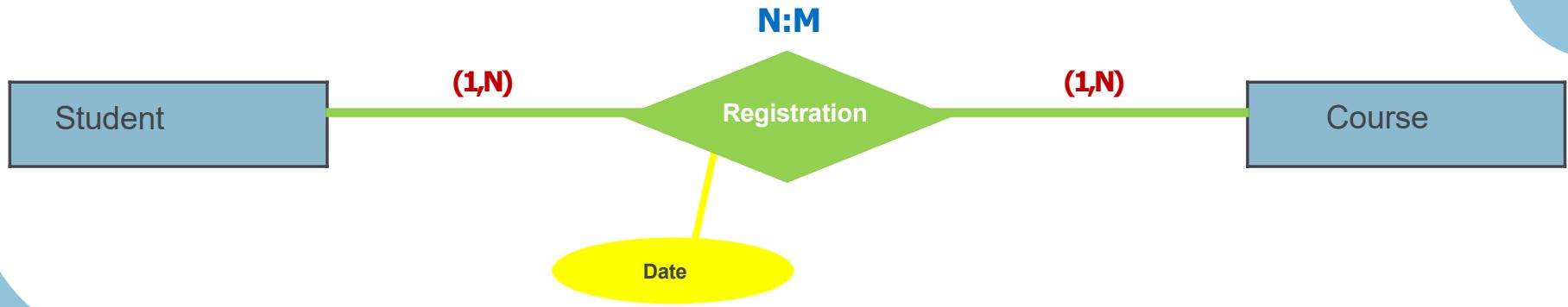
Cardinality refers to the way entities relate to each other.



Cardinality is used to define the number of occurrences allowed in a relationship between entities.

Cardinality "N : M"

(Minimum cardinality, Maximum cardinality)



# Example: Cardinality

For a veterinary clinic, model the entity-relationship (ER) diagram.



People can bring their pets, and we need to store the following data:

1. For the person: ID number (DNI), first name, last name, phone, address.
2. For the pet: name, birth date, type.
3. A person can bring one or more pets.
4. A pet can be brought by only one person

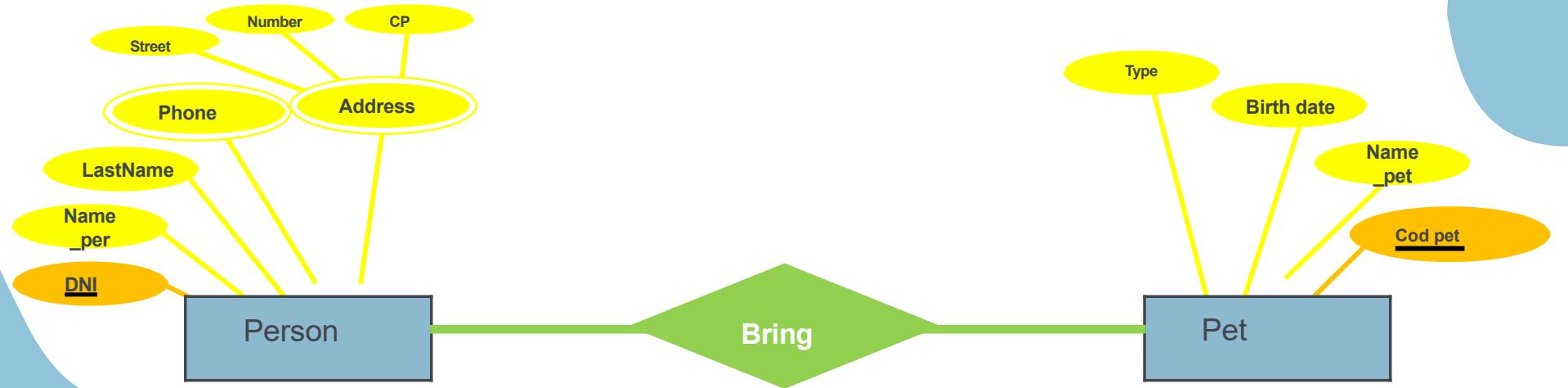
# Example: Cardinality

For a veterinary clinic, model the entity-relationship (ER) diagram.



People can bring their pets, and we need to store the following data:

1. For the person: ID number (DNI), first name, last name, phone, address.
2. For the pet: name, birth date, type.
3. A person can bring one or more pets.
4. A pet can be brought by only one person



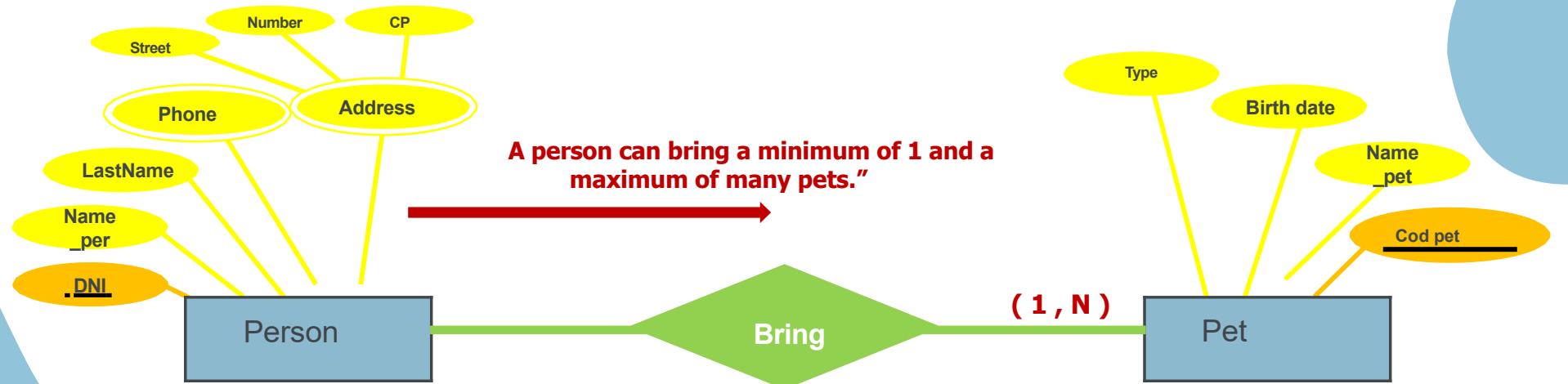
# Example: Cardinality

For a veterinary clinic, model the entity-relationship (ER) diagram.



People can bring their pets, and we need to store the following data:

1. For the **person**: ID number (DNI), first name, last name, phone, address.
2. For the **pet**: name, birth date, type.
3. A person can **bring** one or more pets.
4. A pet can be **brought** by only one person



"Cardinalities are placed on the opposite side of the relationship that connects the entities"

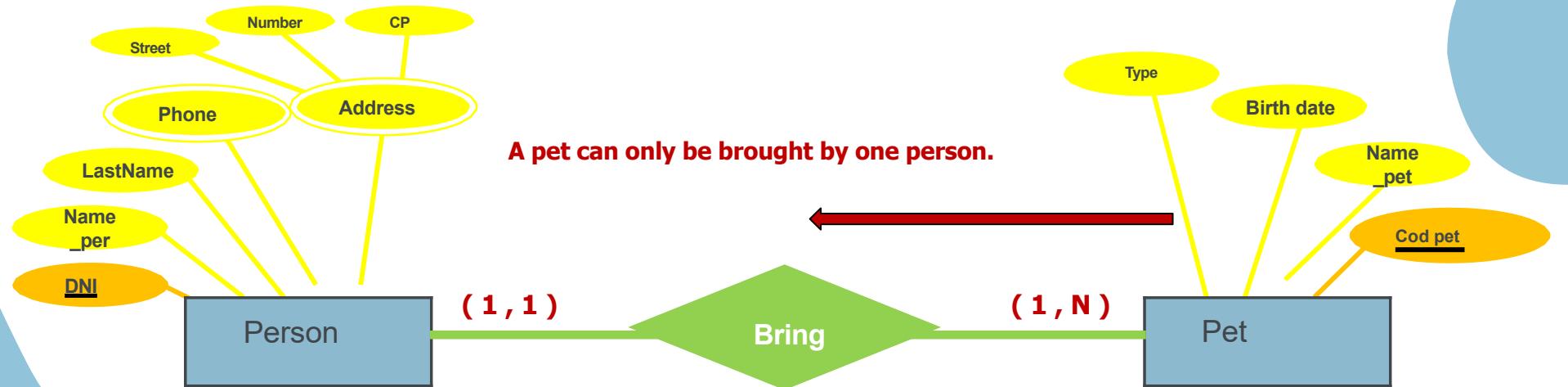
# Example: Cardinality

For a veterinary clinic, model the entity-relationship (ER) diagram.



People can bring their pets, and we need to store the following data:

1. For the **person**: ID number (DNI), first name, last name, phone, address.
2. For the **pet**: name, birth date, type.
3. A person can **bring** one or more pets.
4. A pet can be **brought** by only one person



"Cardinalities are placed on the opposite side of the relationship that connects the entities"

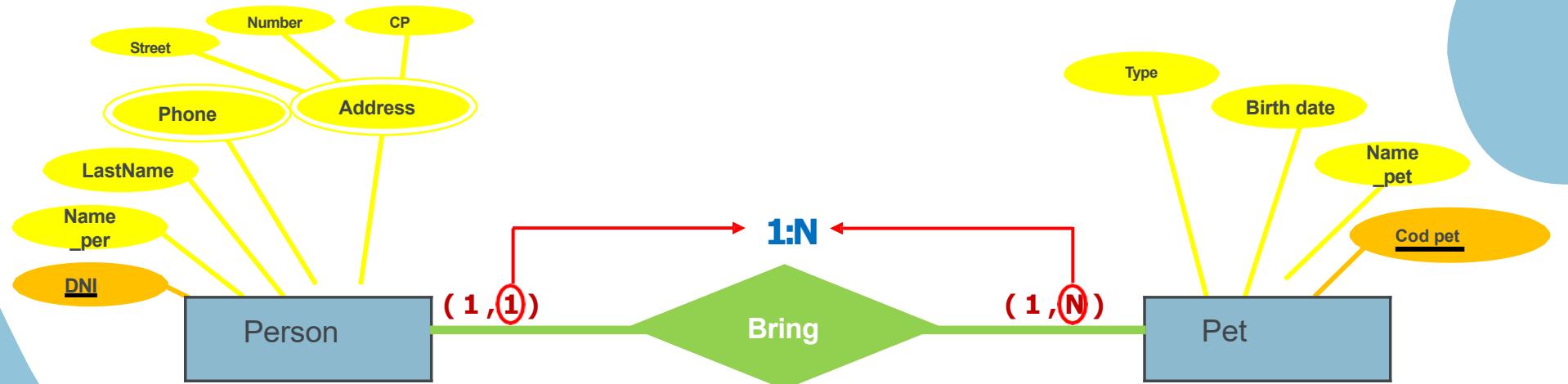
# Example: Cardinality

For a veterinary clinic, model the entity-relationship (ER) diagram.



People can bring their pets, and we need to store the following data:

1. For the **person**: ID number (DNI), first name, last name, phone, address.
2. For the **pet**: name, birth date, type.
3. A person can **bring** one or more pets.
4. A pet can be **brought** by only one person



"Cardinalities are placed on the opposite side of the relationship that connects the entities"

# Example: Cardinality

For a transportation company, create the entity-relationship (ER) model.



The transportation company delivers packages throughout Colombia.

The people in charge of delivering the packages are the truck drivers, and we want to store their ID number (DNI), name, phone, address, and salary.

For the transported packages, we are interested in the package code, description, and recipient's address.

A truck driver delivers many packages, and a package can only be delivered by one truck driver.

For the cities to which the packages are delivered, we want to store the postal code and name.

A package can only arrive at one city. However, many packages can arrive in the same city.

For the trucks driven by the truck drivers, we want to store the license plate, model, and type.

A truck driver can drive different trucks on different dates, and a truck can be driven by multiple truck drivers.

# For a transportation company, create the entity-relationship (ER) model.



The transportation company delivers packages throughout Colombia.

The people in charge of delivering the packages are the **truck drivers**, and we want to store their **ID number (DNI)**, name, phone, address, and salary.

For the transported **packages**, we are interested in the **package code, description, and recipient's address**.

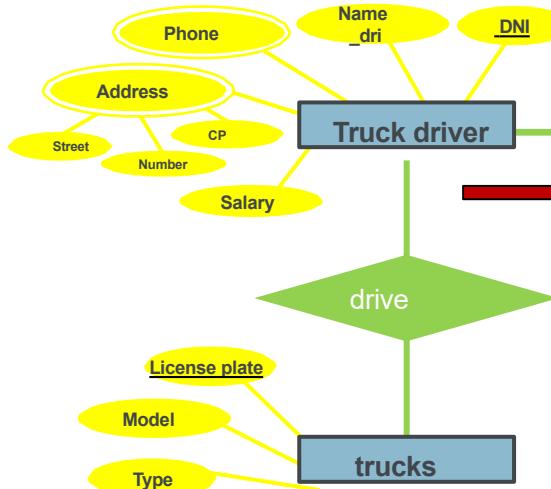
A truck driver **delivers** many packages, and a package can only be delivered by one truck driver.

For the **cities** to which the packages are delivered, we want to store the **postal code and name**.

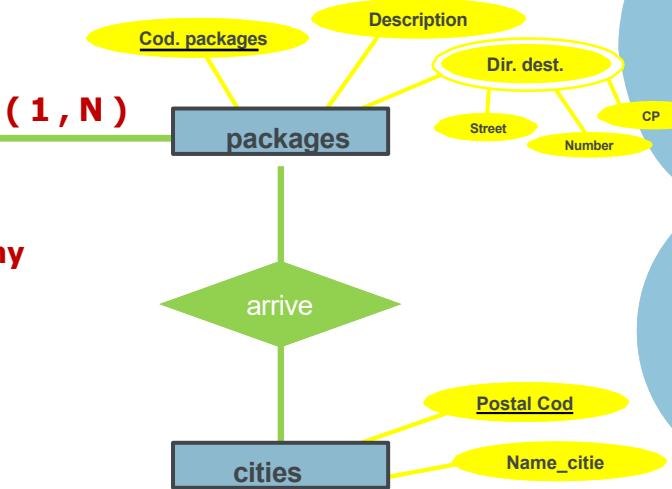
A package can only **arrive** at one city. However, many packages can arrive in the same city.

For the **trucks** driven by the truck drivers, we want to store the **license plate, model, and type**.

A truck driver can **drive** different trucks on different dates, and a truck can be driven by multiple truck drivers.



**" A truck driver delivers many packages."**



# For a transportation company, create the entity-relationship (ER) model.



The transportation company delivers packages throughout Colombia.

The people in charge of delivering the packages are the **truck drivers**, and we want to store their **ID number (DNI)**, name, phone, address, and salary.

For the transported **packages**, we are interested in the **package code, description, and recipient's address**.

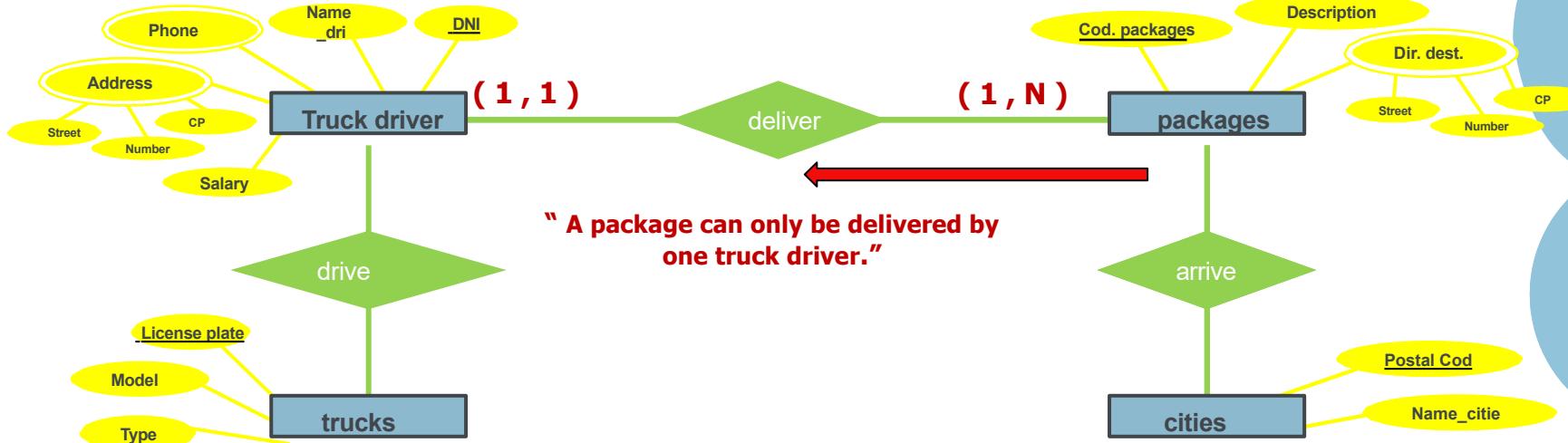
A truck driver **delivers** many packages, and a package can only be delivered by one truck driver.

For the **cities** to which the packages are delivered, we want to store the **postal code and name**.

A package can only **arrive** at one city. However, many packages can arrive in the same city.

For the **trucks** driven by the truck drivers, we want to store the **license plate, model, and type**.

A truck driver can **drive** different trucks on different dates, and a truck can be driven by multiple truck drivers.



# For a transportation company, create the entity-relationship (ER) model.



The transportation company delivers packages throughout Colombia.

The people in charge of delivering the packages are the **truck drivers**, and we want to store their **ID number (DNI)**, name, phone, address, and salary.

For the transported **packages**, we are interested in the **package code, description, and recipient's address**.

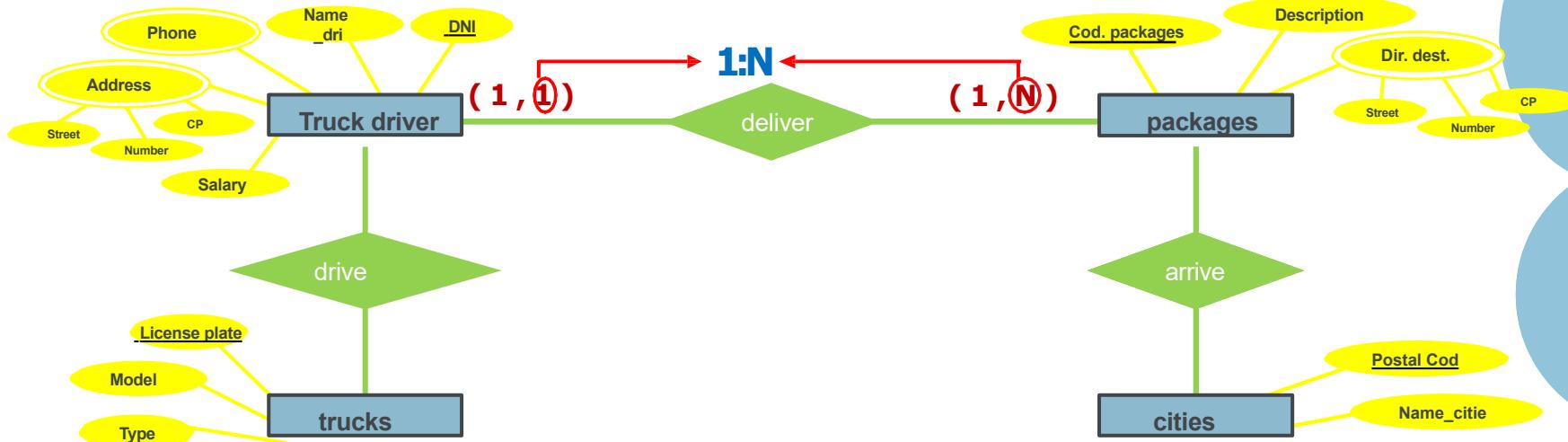
A truck driver **delivers** many packages, and a package can only be delivered by one truck driver.

For the **cities** to which the packages are delivered, we want to store the **postal code and name**.

A package can only **arrive** at one city. However, many packages can arrive in the same city.

For the **trucks** driven by the truck drivers, we want to store the **license plate, model, and type**.

A truck driver can **drive** different trucks on different dates, and a truck can be driven by multiple truck drivers.



# For a transportation company, create the entity-relationship (ER) model.



The transportation company delivers packages throughout Colombia.

The people in charge of delivering the packages are the **truck drivers**, and we want to store their **ID number (DNI)**, name, phone, address, and salary.

For the transported **packages**, we are interested in the **package code, description, and recipient's address**.

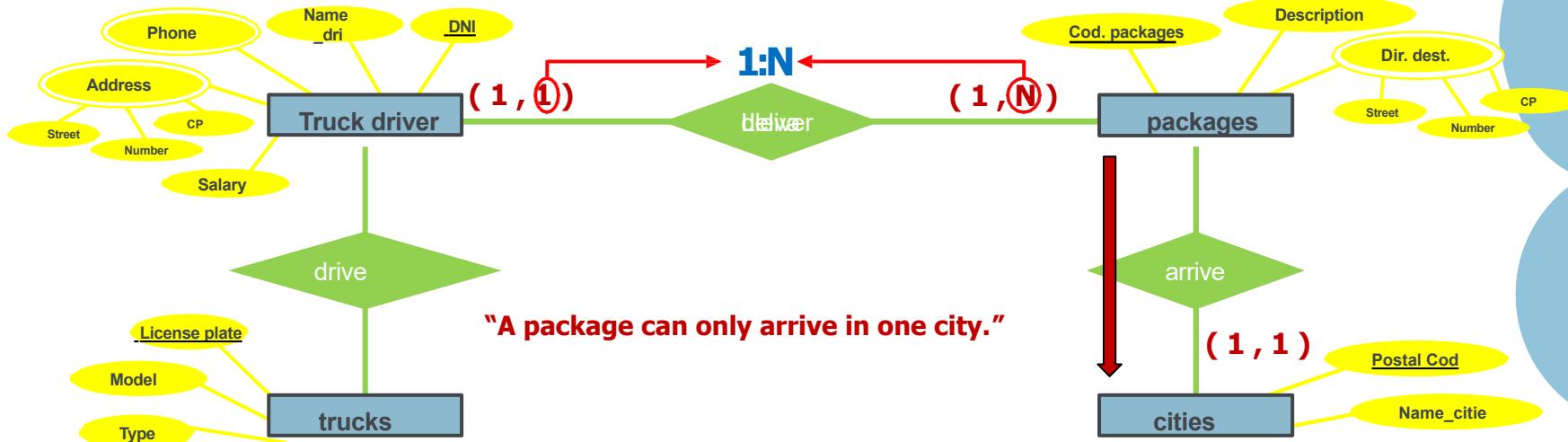
A truck driver **delivers** many packages, and a package can only be delivered by one truck driver.

For the **cities** to which the packages are delivered, we want to store the **postal code and name**.

A package can only **arrive** at one city. However, many packages can arrive in the same city.

For the **trucks** driven by the truck drivers, we want to store the **license plate, model, and type**.

A truck driver can **drive** different trucks on different dates, and a truck can be driven by multiple truck drivers.



# For a transportation company, create the entity-relationship (ER) model.



The transportation company delivers packages throughout Colombia.

The people in charge of delivering the packages are the **truck drivers**, and we want to store their **ID number (DNI)**, name, phone, address, and salary.

For the transported **packages**, we are interested in the **package code, description, and recipient's address**.

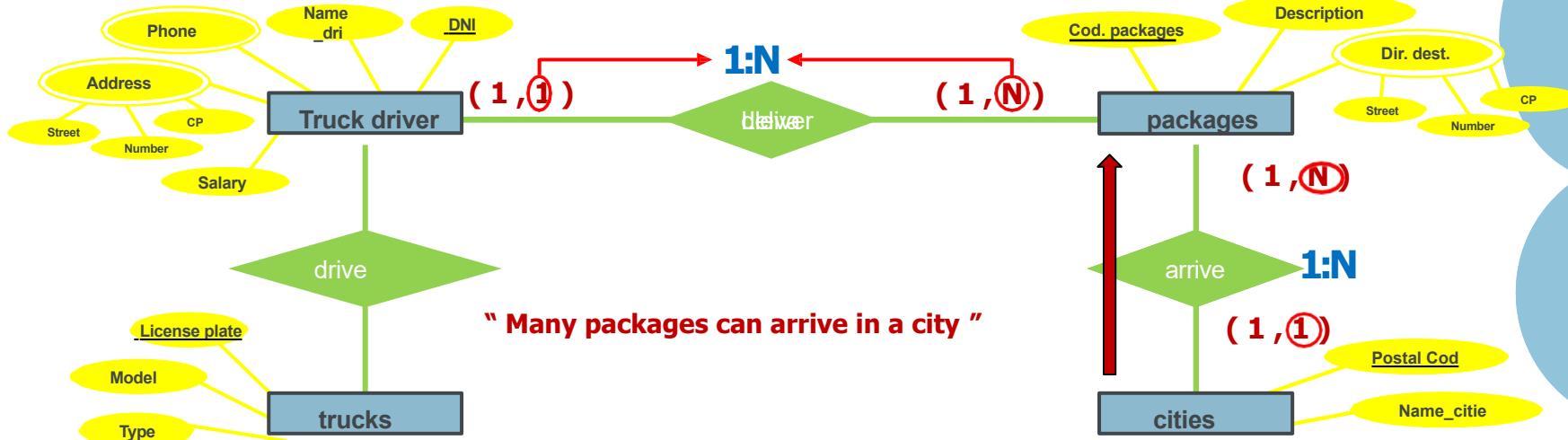
A truck driver **delivers** many packages, and a package can only be delivered by one truck driver.

For the **cities** to which the packages are delivered, we want to store the **postal code and name**.

A package can only **arrive** at one city. However, many packages can arrive in the same city.

For the **trucks** driven by the truck drivers, we want to store the **license plate, model, and type**.

A truck driver can **drive** different trucks on different dates, and a truck can be driven by multiple truck drivers.



# For a transportation company, create the entity-relationship (ER) model.



The transportation company delivers packages throughout Colombia.

The people in charge of delivering the packages are the **truck drivers**, and we want to store their **ID number (DNI)**, name, phone, address, and salary.

For the transported **packages**, we are interested in the **package code, description, and recipient's address**.

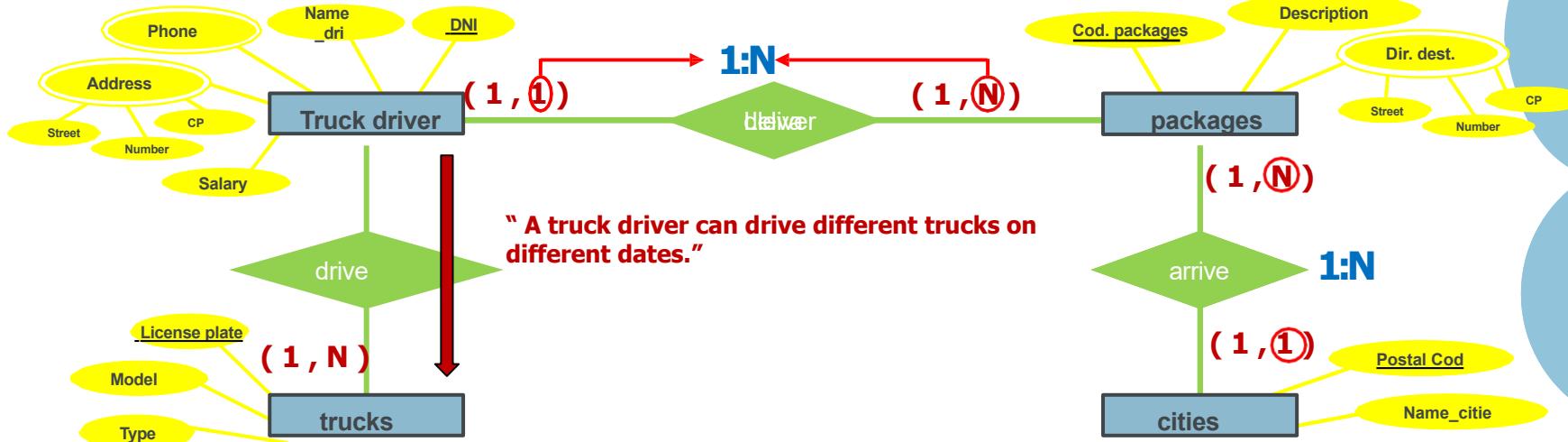
A truck driver **delivers** many packages, and a package can only be delivered by one truck driver.

For the **cities** to which the packages are delivered, we want to store the **postal code and name**.

A package can only **arrive** at one city. However, many packages can arrive in the same city.

For the **trucks** driven by the truck drivers, we want to store the **license plate, model, and type**.

A truck driver can **drive** different trucks on different dates, and a truck can be driven by multiple truck drivers.



# For a transportation company, create the entity-relationship (ER) model.



The transportation company delivers packages throughout Colombia.

The people in charge of delivering the packages are the **truck drivers**, and we want to store their **ID number (DNI)**, name, phone, address, and salary.

For the transported **packages**, we are interested in the **package code, description, and recipient's address**.

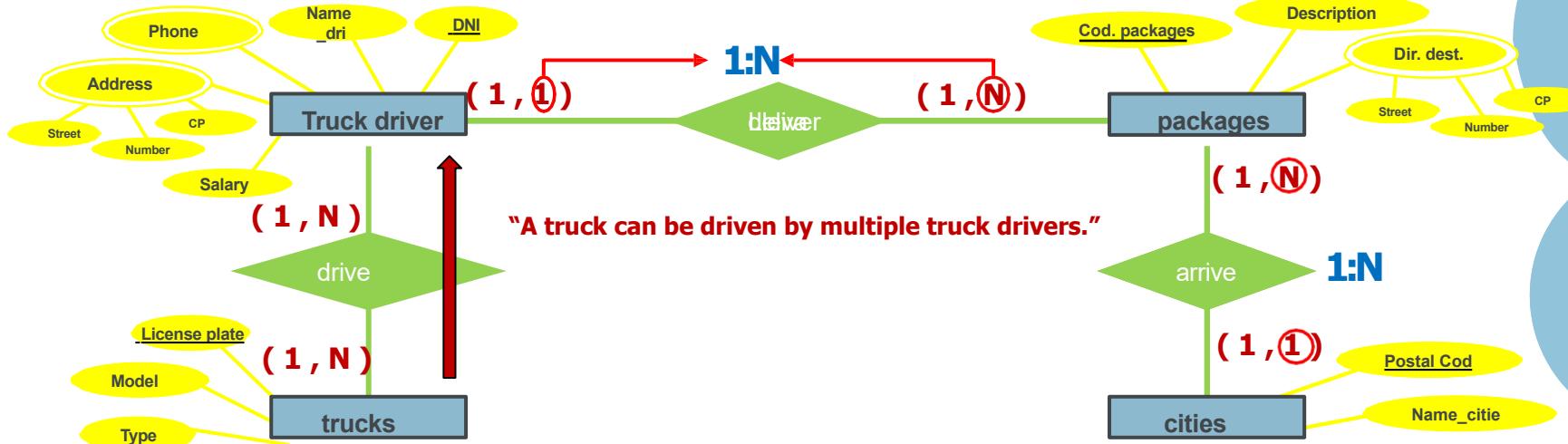
A truck driver **delivers** many packages, and a package can only be delivered by one truck driver.

For the **cities** to which the packages are delivered, we want to store the **postal code and name**.

A package can only **arrive** at one city. However, many packages can arrive in the same city.

For the **trucks** driven by the truck drivers, we want to store the **license plate, model, and type**.

A truck driver can **drive** different trucks on different dates, and a truck can be driven by multiple truck drivers.



# For a transportation company, create the entity-relationship (ER) model.



The transportation company delivers packages throughout Colombia.

The people in charge of delivering the packages are the **truck drivers**, and we want to store their **ID number (DNI)**, name, phone, address, and salary.

For the transported **packages**, we are interested in the **package code, description, and recipient's address**.

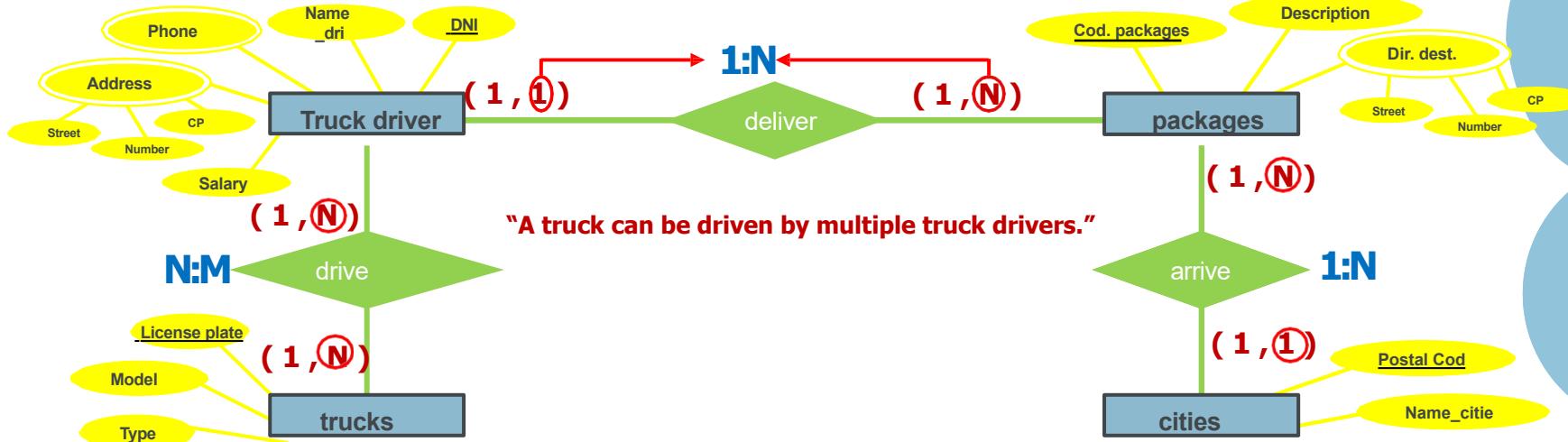
A truck driver **delivers** many packages, and a package can only be delivered by one truck driver.

For the **cities** to which the packages are delivered, we want to store the **postal code and name**.

A package can only **arrive** at one city. However, many packages can arrive in the same city.

For the **trucks** driven by the truck drivers, we want to store the **license plate, model, and type**.

A truck driver can **drive** different trucks on different dates, and a truck can be driven by multiple truck drivers.

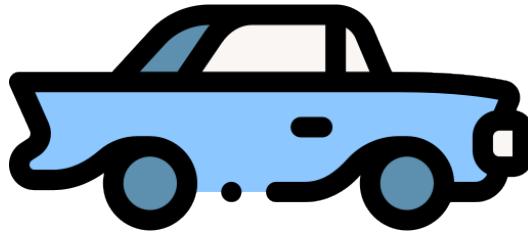


# Types of Entity

An entity is something that exists in the real world, distinguishable from other things, and about which we are interested in certain properties.



Entities can have a physical existence, such as a person, a car, or a book, but they can also consist of more abstract concepts, such as an insurance policy.



# Example: Types of Entity

An entity is something that exists in the real world, distinguishable from other things, and about which we are interested in certain properties .



Entities can have a physical existence, such as a person, a car, or a book, but they can also consist of more abstract concepts, such as an insurance policy.

There are two types of entities:

**Strong Entity:** These are entities that have an existence by themselves.



The CLIENT entity does not depend on other entities to exist..

**Weak Entity:** These are entities whose existence depends on another entity or other types of entities.



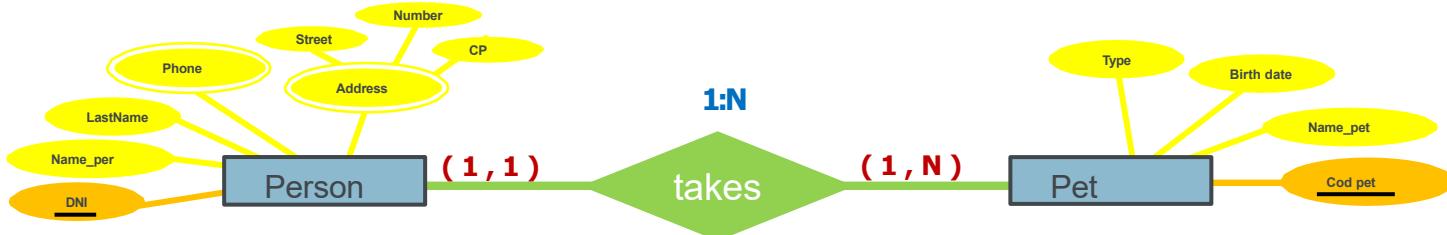
The INVOICE entity will be weak with respect to the CLIENT entity Because there are no invoices that do not correspond to a client.

## New Requirements for the Veterinary Clinic



1. For the person: ID number (DNI), first name, last name, phone, address.
2. For the pet: name, birth date, type.
3. A person can bring one or more pets.
4. A pet can be brought by only one person.
5. For the veterinarian, we are interested in: name, ID number (DNI), specialty.
6. Each person has an emergency contact family member; the family member is a contact for only one person.
7. For the family member, we are interested in: ID number (DNI), name, phone.
8. Pets have consultations with the veterinarian. A veterinarian can consult with none or several pets, and the same pet attending the clinic can have one or more consultations with various veterinarians on different dates.
9. If the person leaves the veterinary clinic, the emergency contact family member is no longer relevant.

### Entity-Relationship Model:



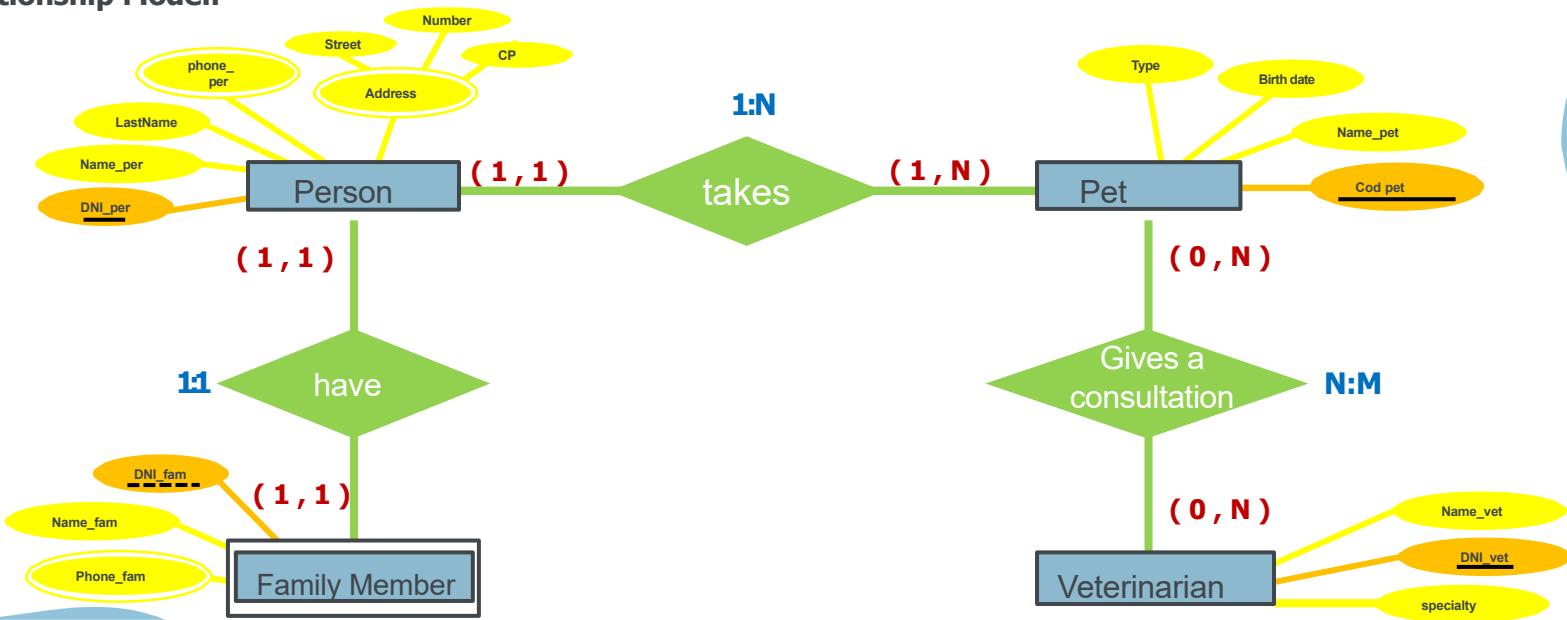
# New Requirements for the Veterinary Clinic



1. For the person: ID number (DNI), first name, last name, phone, address.
2. For the pet: name, birth date, type.
3. A person can bring one or more pets.
4. A pet can be brought by only one person.

5. For the veterinarian, we are interested in: name, ID number (DNI), specialty.
6. Each person has an emergency contact family member; the family member is a contact for only one person.
7. For the family member, we are interested in: ID number (DNI), name, phone.
8. Pets have consultations with the veterinarian. A veterinarian can consult with none or several pets, and the same pet attending the clinic can have one or more consultations with various veterinarians on different dates.
9. If the person leaves the veterinary clinic, the emergency contact family member is no longer relevant.

## Entity-Relationship Model:

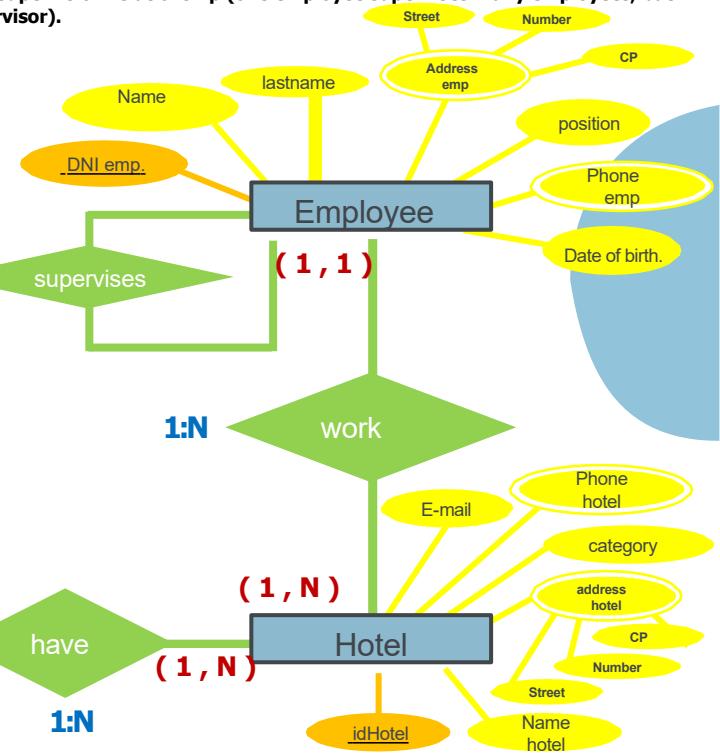
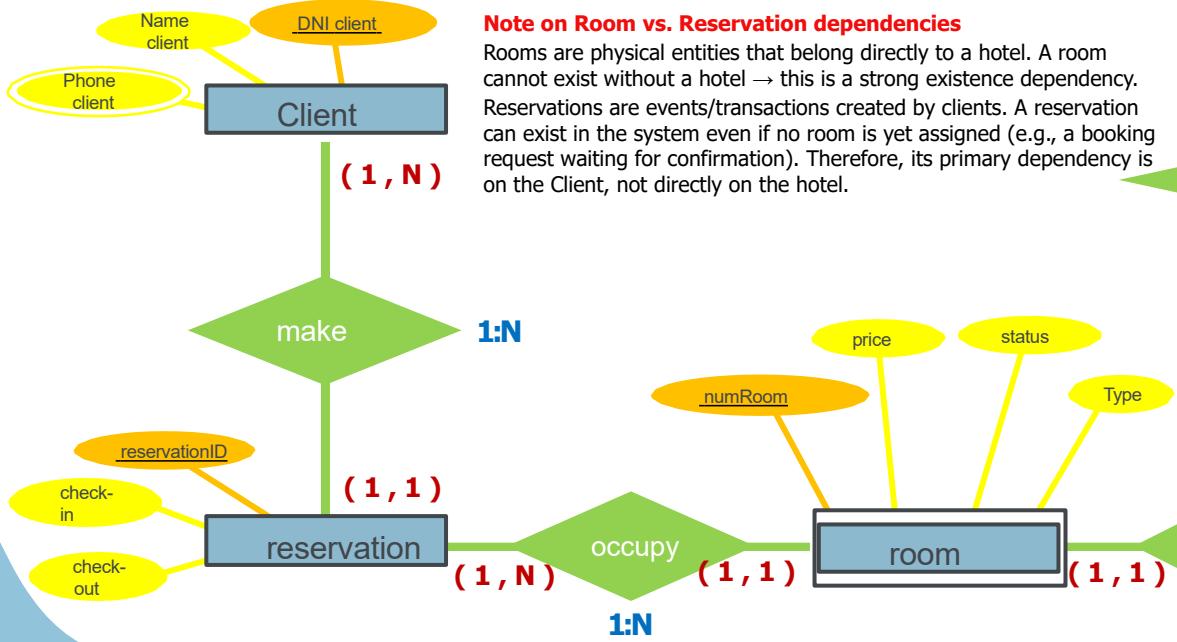


# Exercise: Hotel

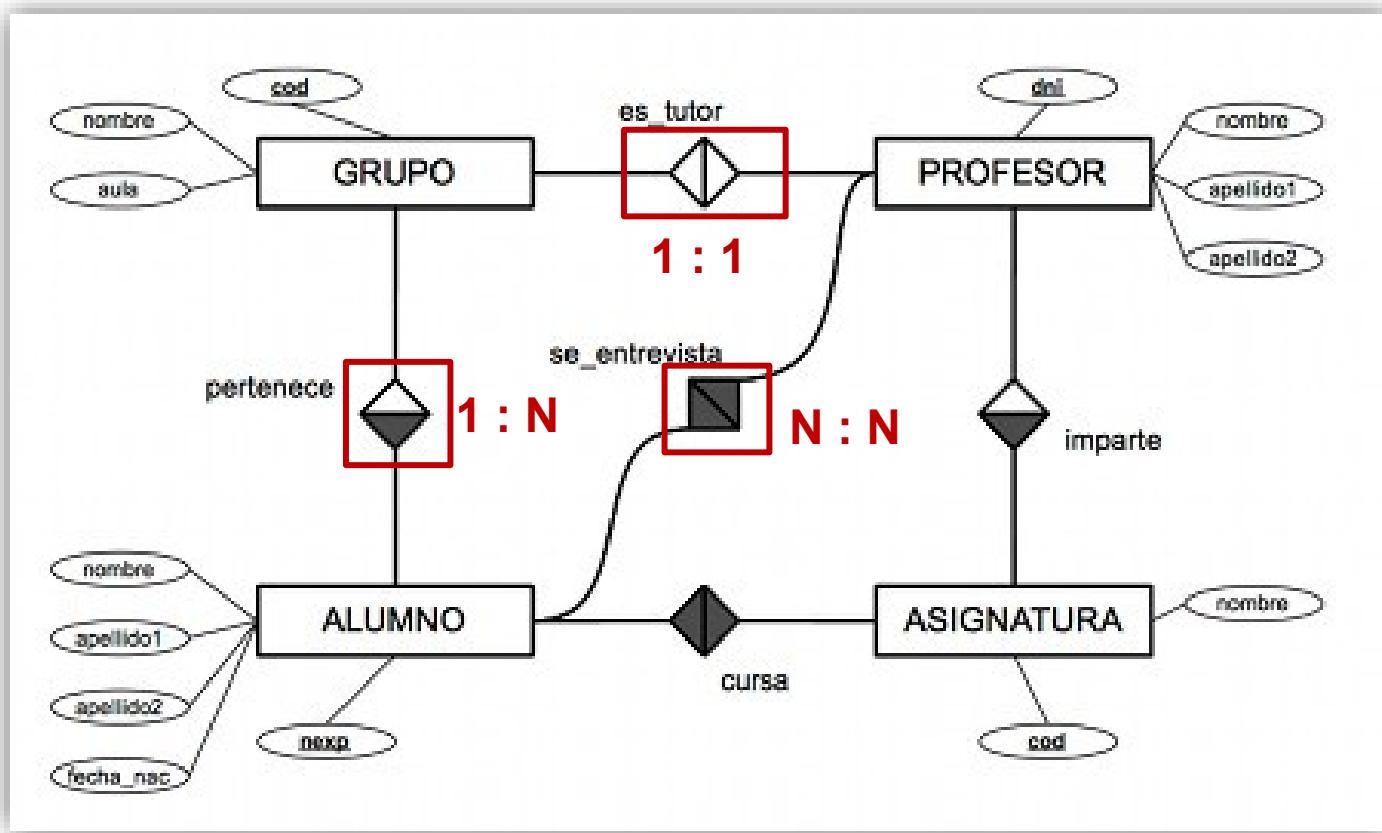
1. From the client: ID number (DNI), name, phone.
2. From the reservation: Reservation\_ID, check-in date, check-out date.
3. From the room: Room\_number, price, status, type.
4. From the hotel: Hotel\_ID, hotel name, address, rating (stars), phone, email.
5. From the employee: First name, last name, address, ID number (DNI), position, phone, date of birth.
6. A client can make one or many reservations, and a reservation is made by a single client.
7. A reservation can include one or many rooms, and a room corresponds to a single reservation.
8. A hotel has several rooms for accommodating clients.
9. Many employees work in the hotel, and there is also a supervisor for the employees.

# Exercise: Hotel

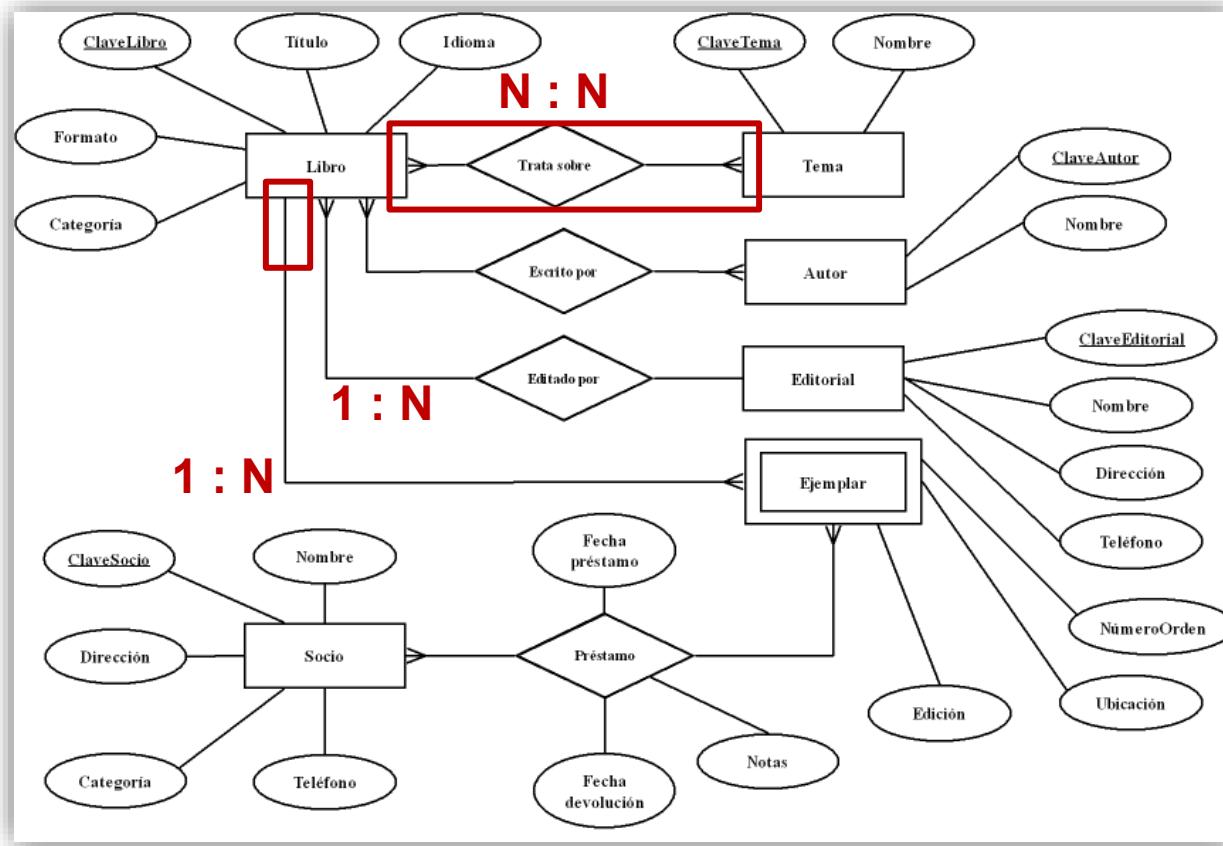
1. From Client: ID, name, phone.
2. From Reservation: ReservationID, check-in date, check-out date.
3. From Room: Room number, price, status, type.
4. From Hotel: HotelID, hotel name, address, category (stars), phone, email.
5. From Employee: First name, last name, address, ID, position, phone, date of birth.
6. A client can make one or many reservations, and a reservation is made by only one client. → 1:N relationship between Client and Reservation.
7. A reservation can occupy one or many rooms, and a room belongs to only one reservation. → 1:N relationship between Reservation and Room.
8. A hotel has many rooms for hosting clients. → 1:N relationship between Hotel and Room.
9. A hotel employs many employees, and there is also a supervisor of the employees. → 1:N relationship between Hotel and Employee + a supervision relationship (one employee supervises many employees, but each employee has only one supervisor).



# Chen-style Variants



# Chen-style Variants



# Chen-style Variants

