

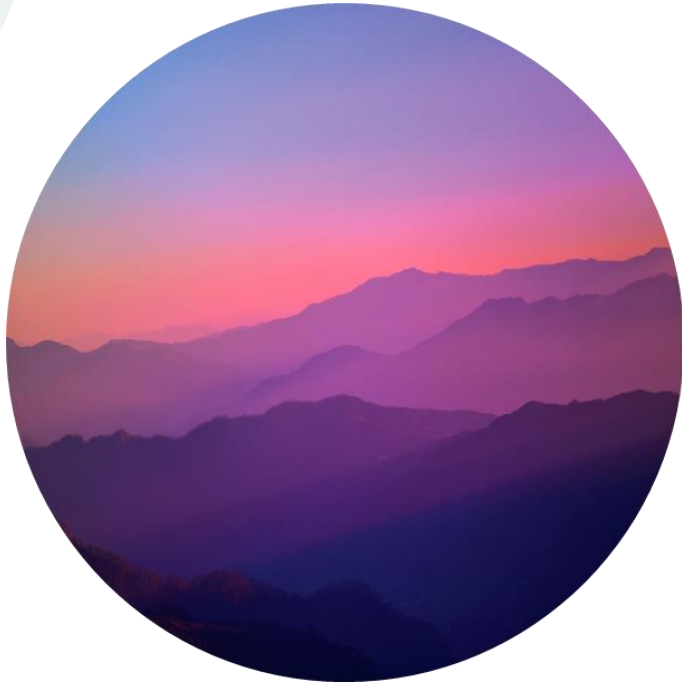
# BASES DE DATOS PARA IA/DATA SCIENCE

PABLO OCTAVIANO



# BASES DE DATOS RELACIONALES

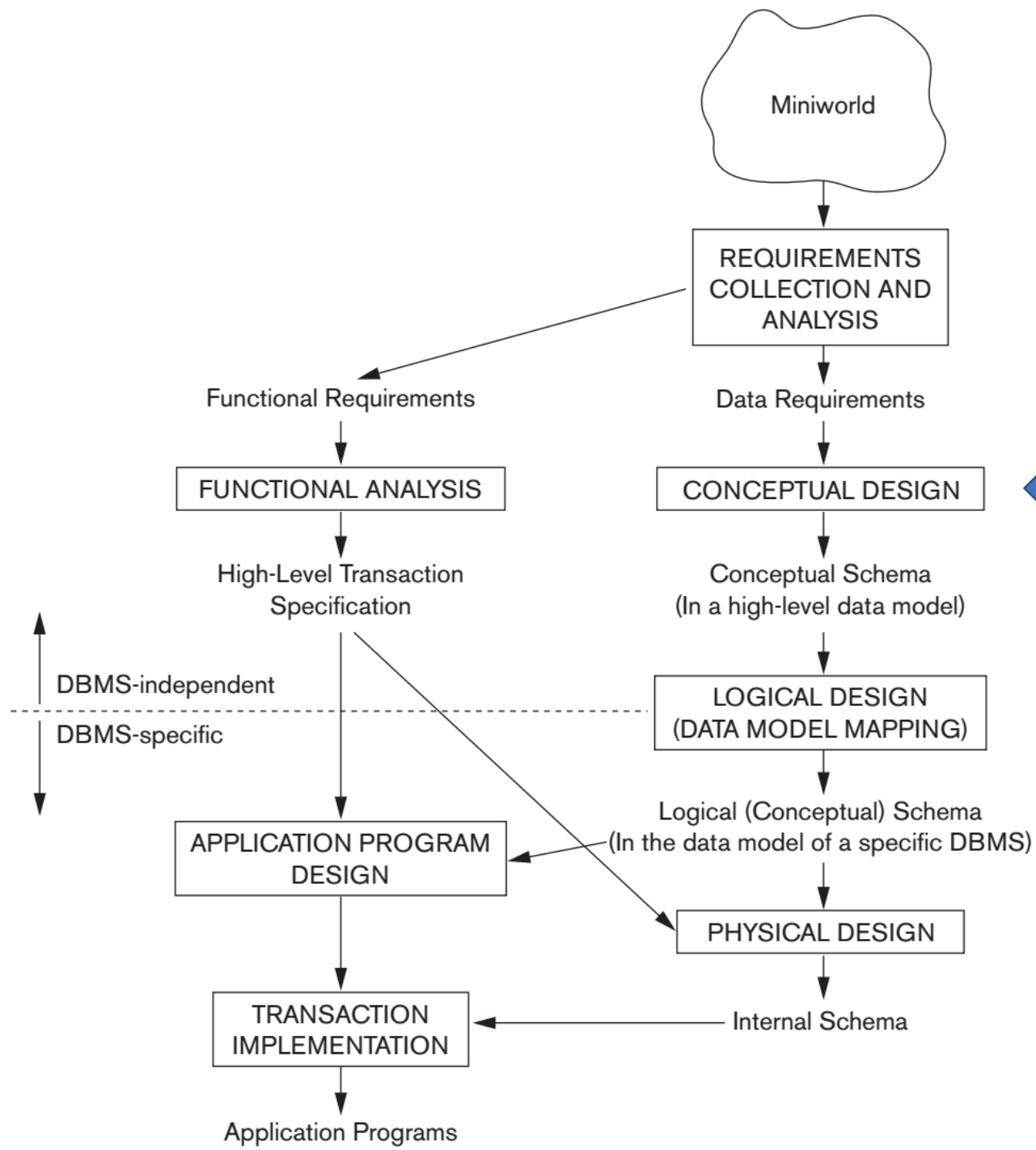
Modelo lógico,  
relaciones, relaciones  
y más relaciones



- Relaciones
- Diagramas de entidad relación
- Llaves
- Modelo Relacional
- Postgres

# AGENDA

# Proceso de Creación de Base de datos



Usted está aquí



RELACIONES

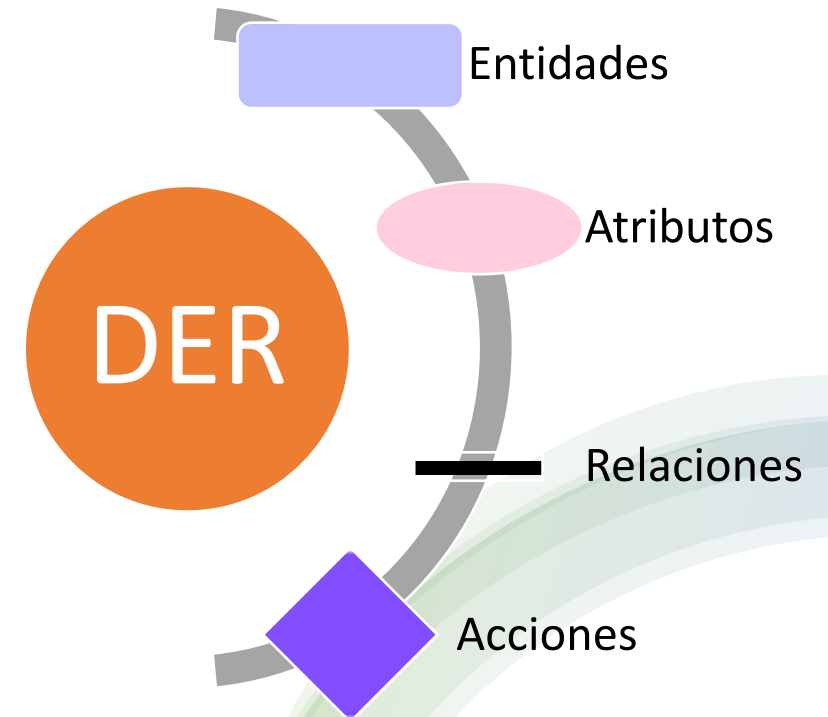
# Conceptos básicos

## Diagrama Entidad Relación (DER/ERD)

El modelo Entidad-Relación (E-R) es una manera de representar nuestra percepción del sistema que vamos a modelar.

Este consiste en un conjunto de objetos básicos:

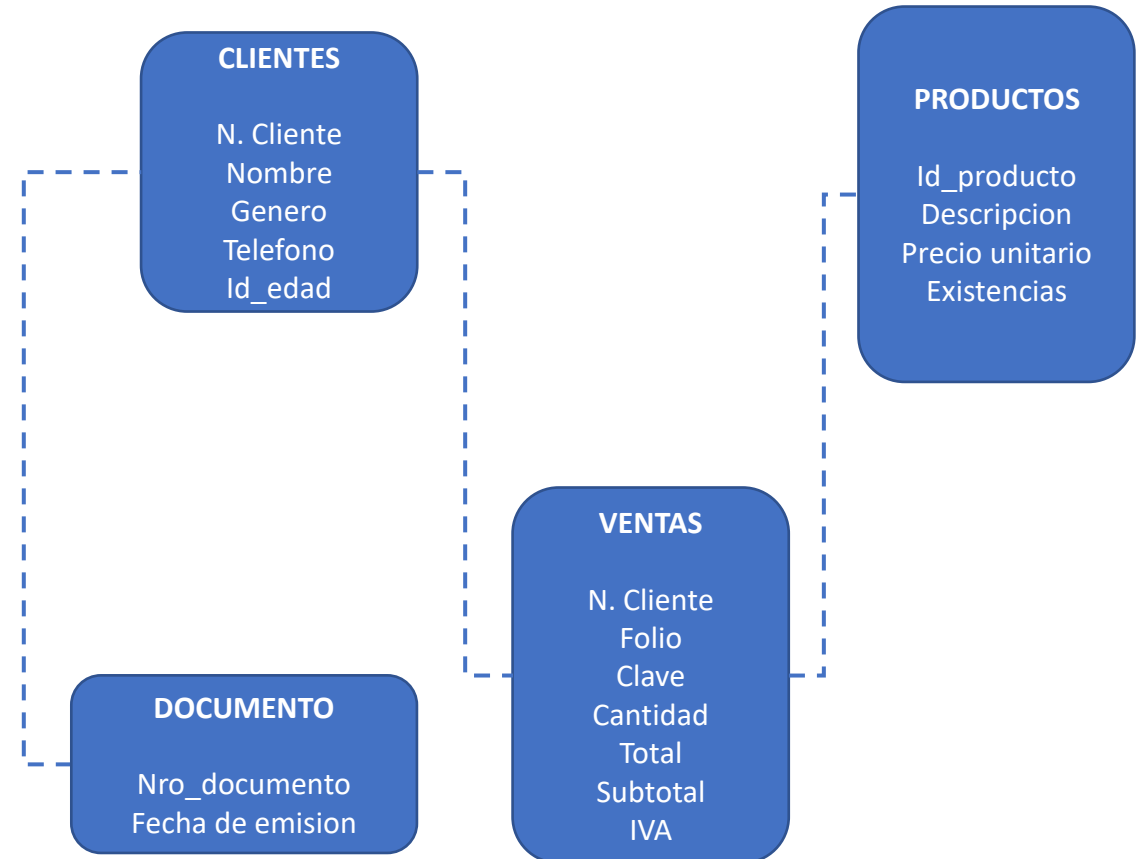
- **Entidades**
- **Atributos**
- **Interrelaciones**
- **Acciones**



# Repasamos!

Recordamos que a una tabla también la podemos llamar **relación**. Sin embargo, este término es mucho más utilizado para describir las interrelaciones entre tablas.

Una relación entre tablas supone que existen referencias entre llaves de una tabla.

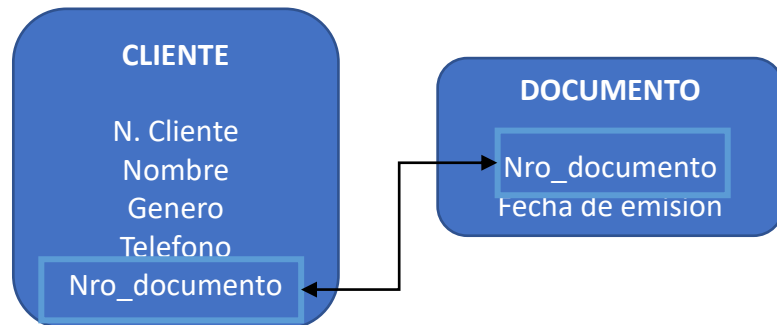


# Relaciones

## Uno a uno

### *One-to-one*

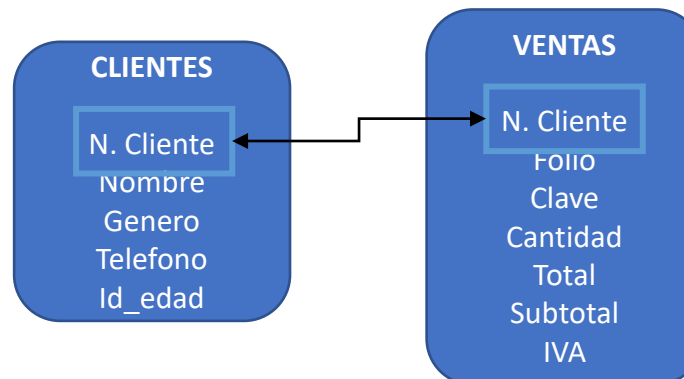
- Las relaciones uno a uno son aquellas que para cada registro de ambas tablas existe una relación univoca.
- Esto significa que cada valor en las tablas aparece solo una vez en la tabla.



## Uno a muchos

### *One-to-many*

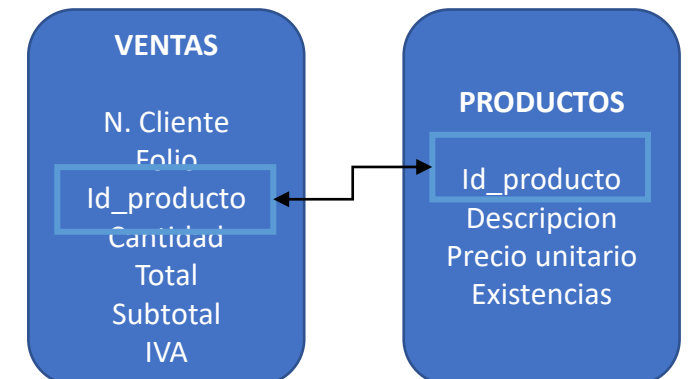
- En estos casos, consideramos que los datos de una tabla pueden aparecer múltiples veces en la siguiente.
- La restricción es que en una de ellas el registro debe ser único



## Muchos a muchos

### *Many-to-Many*

- Es un tipo de relación que ocurre cuando múltiples registros de una tabla se relacionan con muchos elementos de la otra tabla

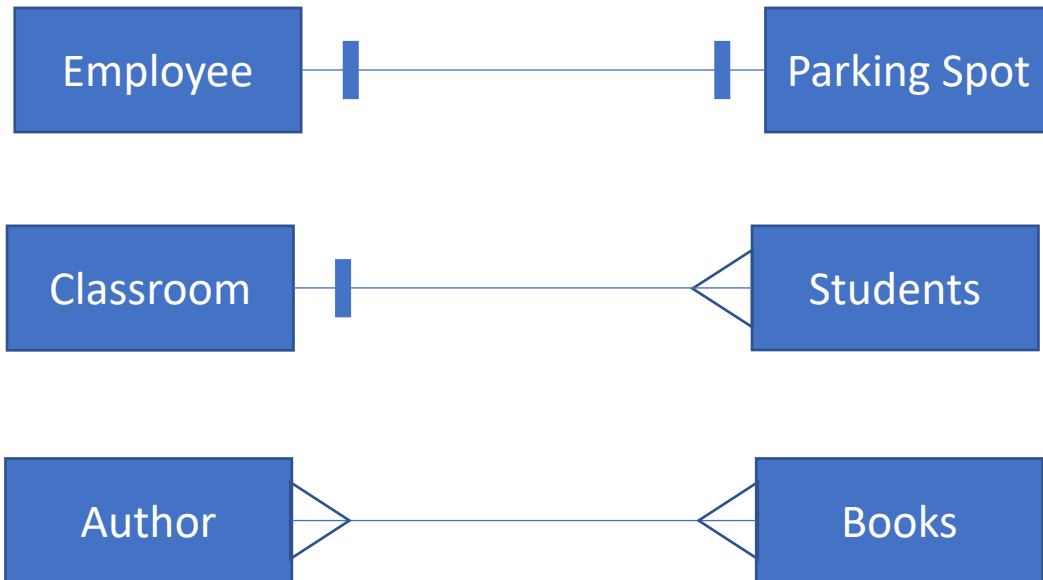




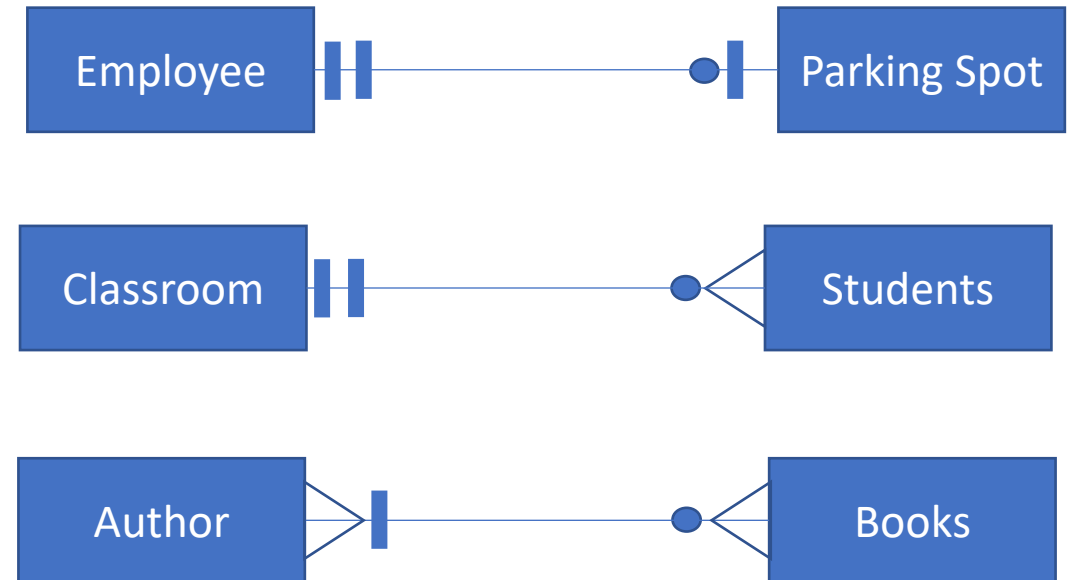
# Relaciones

representación – notación Crow

## Cardinalidad



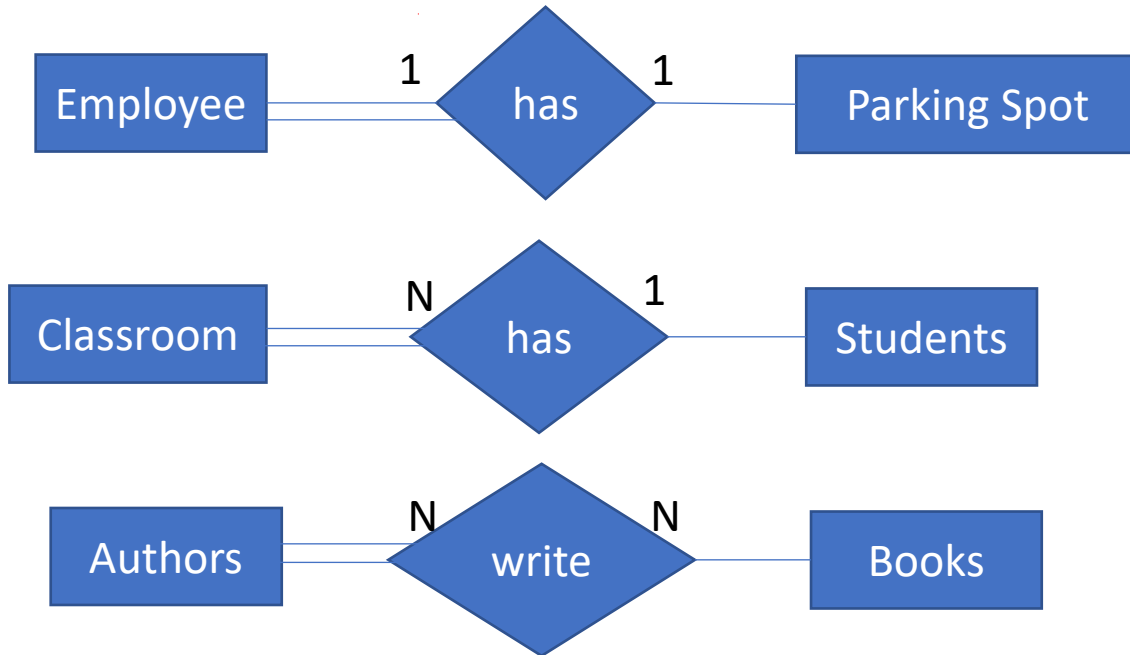
## Participación



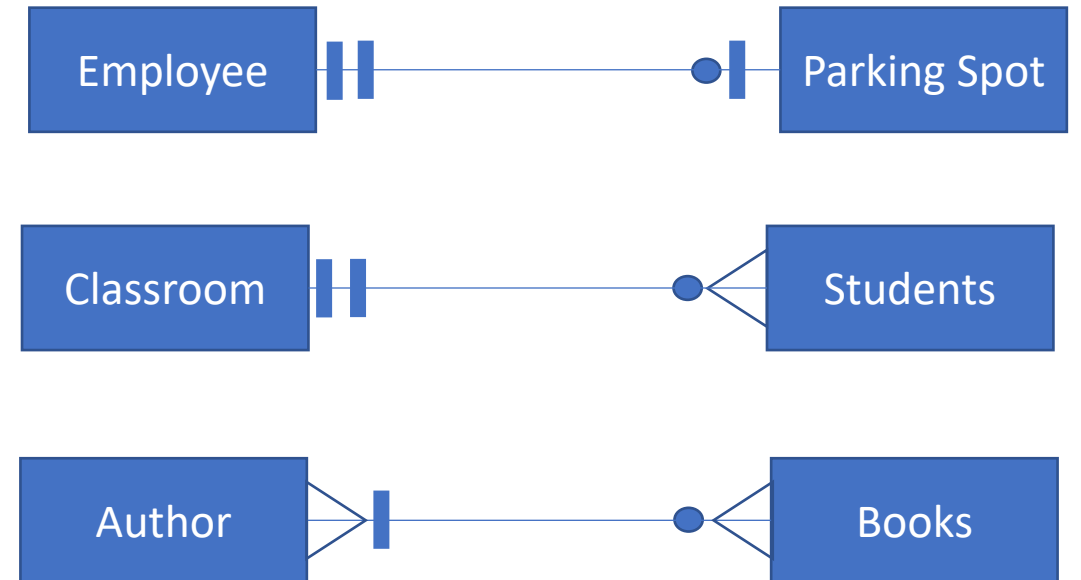
# Relaciones

representación – notación de Chen

Chen



Crow





DER



## BASES DE DATOS PARA INTELIGENCIA ARTIFICIAL

Base de datos corporativa

- 1) Una compañía se organiza en **departamentos**, cada departamento **tiene** nombre único, y **un empleado en particular** que maneja el departamento
- 2) Un **departamento controla** cierto número de **proyectos**, donde **cada proyecto** no cuenta con un **nombre unívoco** y un lugar determinado



## BASES DE DATOS PARA INTELIGENCIA ARTIFICIAL

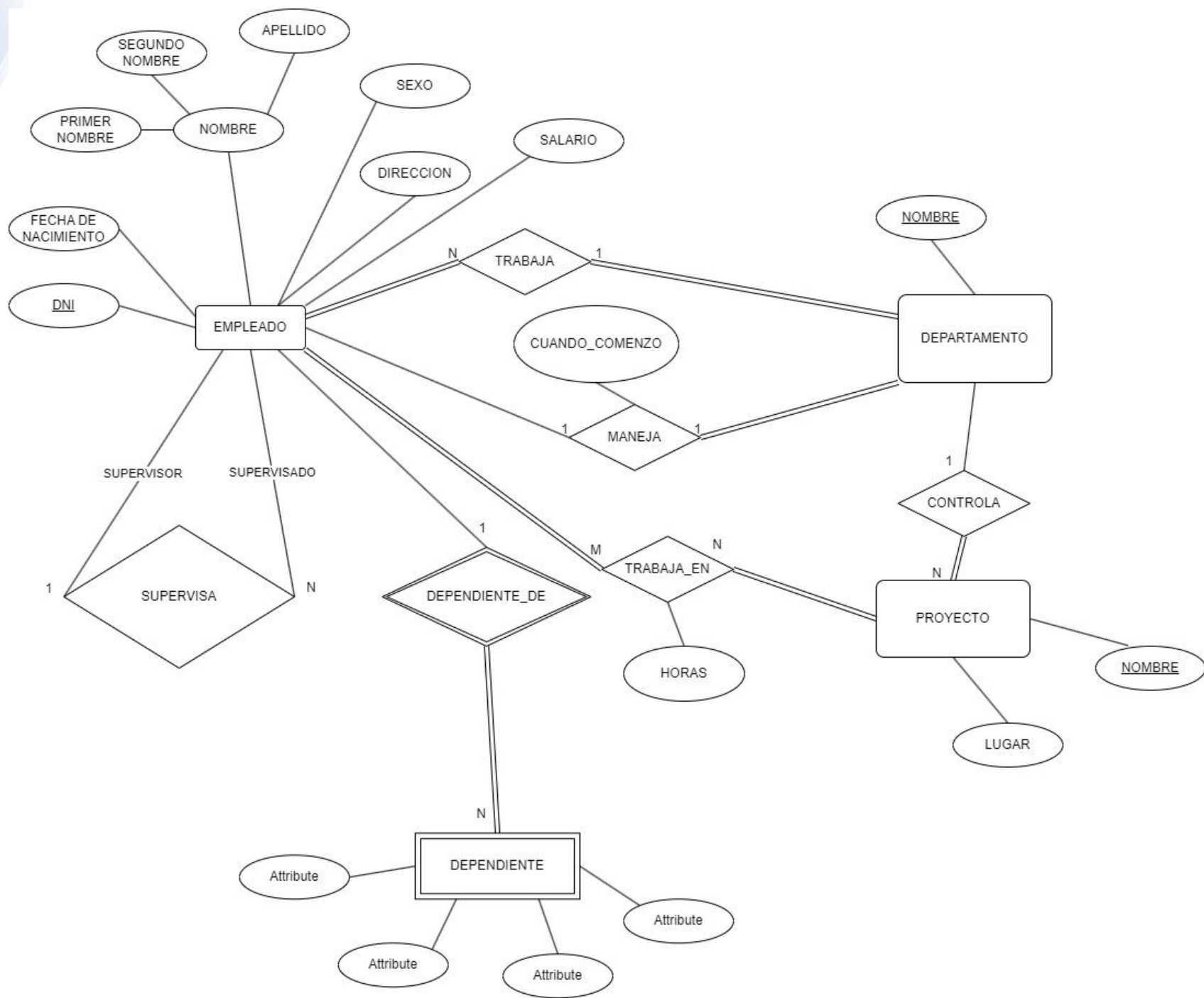
Base de datos corporativa

3) La base de datos guarda **el nombre** de cada empleado, el dni, domicilio, salario, sexo, y fecha de nacimiento. El **empleado** es asignado a un departamento, pero puede trabajar en **múltiples proyectos**.

4) **Se requiere llevar un track del número de horas por semana que cada empleado trabaja por proyecto.**

5) La base de datos también debería guardar información acerca de **cada persona que dependa** directamente del **empleado**, por temas de seguro. Deberá contar con el primer nombre, el sexo, la fecha de nacimiento y la relación que lleva con el empleado.

# Solución planteada

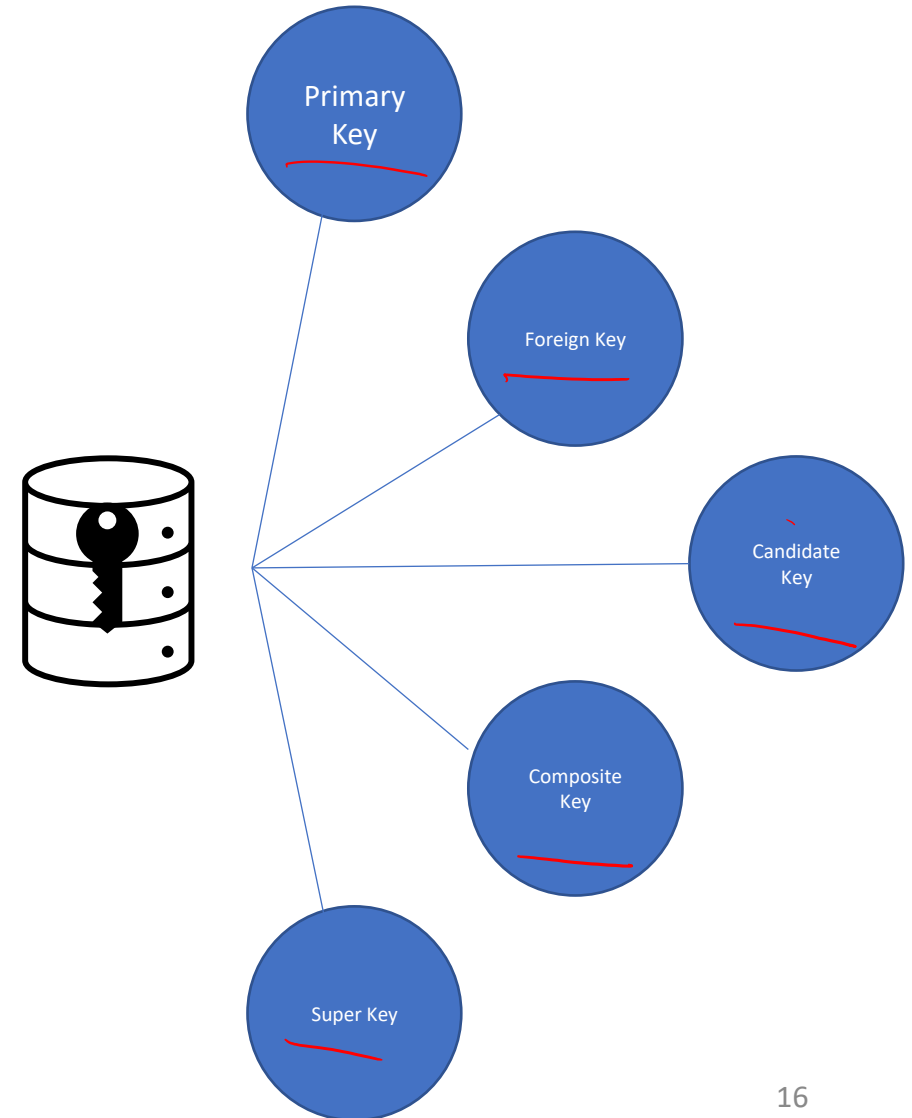




LLAVES

# CLAVES (KEYS).

- Dentro del mundo de RDBs vamos a encontrar que siempre se habla de distintos tipos de llaves (keys) que gobiernan nuestros datos.
- Estas llaves no son más que uno o más atributos que nos permiten identificar de manera univoca un dato en nuestra tabla.
- Además permiten establecer las relaciones del modelo.





# Llaves (*keys*)

RESTRICCION



Primary  
Key

Nos permite identificar unívocamente TODAS las tuplas.

Obligatoria

No Nula

Unique  
Key

Sirven para identificar unívocamente una tupla.

Uno o mas registros pueden ser UK

Puede ser Nula

Se utilizan para mejorar performance

Candidate  
Key

Son registros que se proponen como *unique keys*

Toda tabla tiene al menos una CK

Cada CK puede comportarse como PK en ciertos casos

# Llaves (*keys*)

RESTRICCION

INTEGRIDAD REFERENCIAL:  
TODAS LAS FOREIGN KEYS DEBEN EXISTIR COMO  
PRIMARY KEYS EN OTRAS TABLAS

## Alternate Key

Es un registro que se propone como alternativa a una FK si es necesario

Es una posibilidad como FK pero en un diseño implementado no lo es.

## Composite Key

También conocida como *compound* o *concatenated key*.

Refiere a un grupo de registros que pueden identificar a una tupla en la base

Se utilizan cuando el grupo identifica tuplas pero cada componente por separado no lo hace

## Super Key

Es una combinación de una o mas *keys*

Identifican unívocamente un registro

PK, UK, AK son un subset de super keys

## Foreign Key

Es una *key* que en otra tabla de nuestro modelo es una PK

Acepta nulos

Acepta duplicados

# Algoritmo de mapeo DER -> MR

Para llegar al modelo relacional final, se necesitan una secuencia de pasos que definirán las formas de las tablas finales, al igual que los vínculos y llaves entre ellas.

- 1. Mapear entidades fuertes**
- 2. Mapear entidades débiles**

# Algoritmo de mapeo DER -> MR

## 3. Mapear Relaciones del tipo 1:1

- Opción 1. Foreign keys, preferiblemente en entidades con participación *total*
- Se une a la entidad en una sola tabla, válido cuando la participación de ambos es *total*
- Referencias cruzadas (tabla de relaciones)

# Algoritmo de mapeo DER -> MR

## 4. Mapear Relaciones del tipo 1:N

- Opción 1. Foreign keys, preferiblemente sobre la entidad que tenga cardinalidad N
- Referencias cruzadas (tabla de relaciones)

## 5. Mapear Relaciones del tipo M:N

- Referencias cruzadas (tabla de relaciones)

## 6. Mapear Atributos



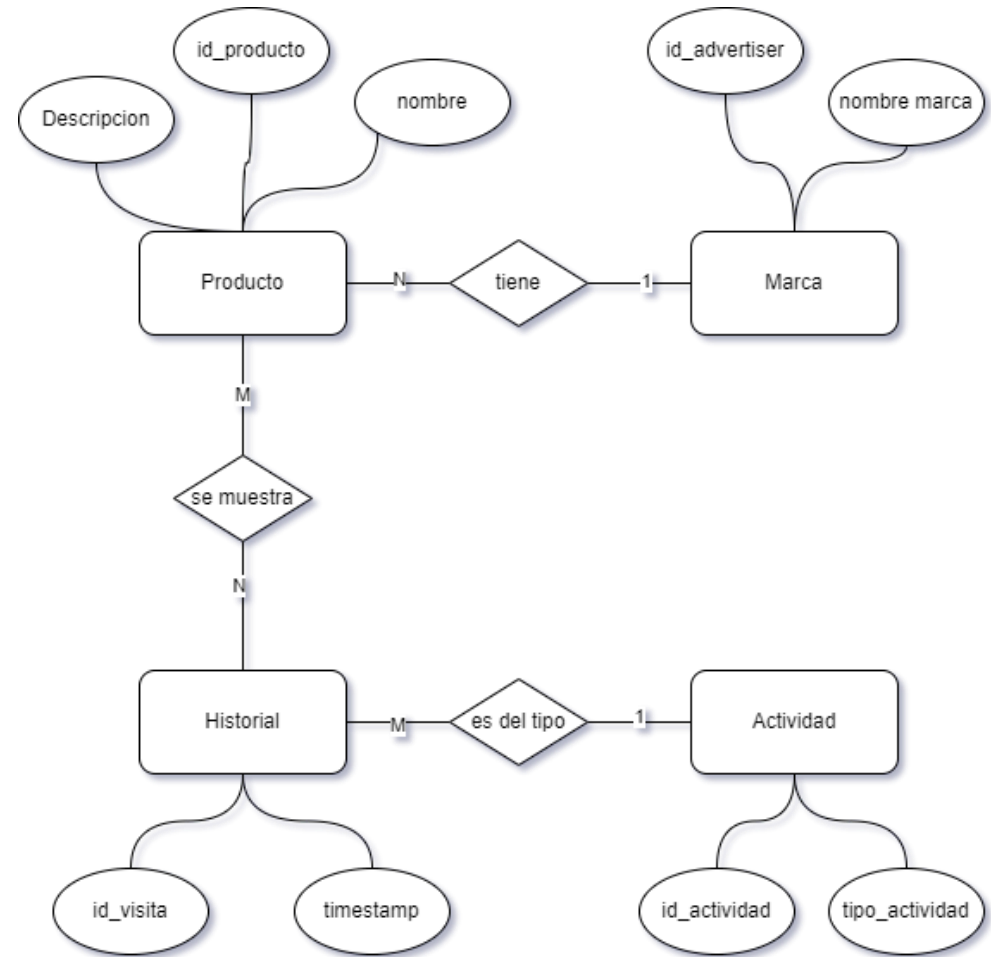
A PRACTICAR

## Ejemplo de análisis

Idx	Marca	Producto	interacción	Fecha
1	5E325T5HYL61QSABVR5V	9trbal	impression	4/1/2022
2	03KNVBO915KY2ZPGA57J	qd5esu	impression	4/1/2022
3	HC26ZE93SA4WWA0BRFM 6	99watc	impression	4/1/2022
...				

Vamos a crear un DER a partir del desarrollo de un problema de advertising. Supongamos que nuestra empresa *patitos™* debemos crear una base relacional para hacer storage de todas las transacciones que hay de publicidades (si fueron presentadas, si el usuario clickeo). Nos dan una tabla de ejemplo.

# SOLUCIÓN PROPUESTA (CASI UNÁNIME!)





# DUDAS?

