

# BASES DE DATOS PARA DATA SCIENCE



Lopez, Yoel  
Pelli, Nahuel

# **BASES DE DATOS RELACIONALES**

Modelo lógico,  
relaciones,  
relaciones y más  
relaciones



# AGENDA

Relaciones

Llaves

Diagramas de entidad relación

Modelo Relacional

Introducción a SQL

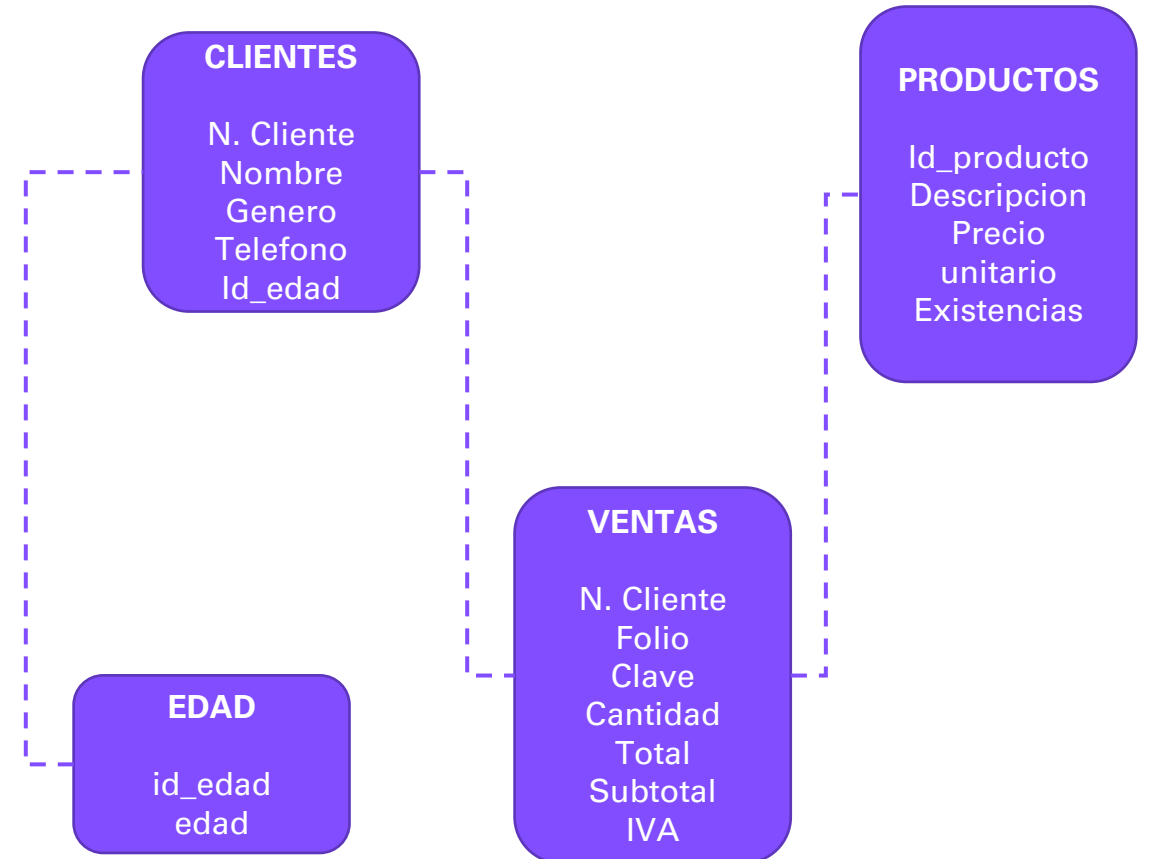
# RELACIONES



# 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.

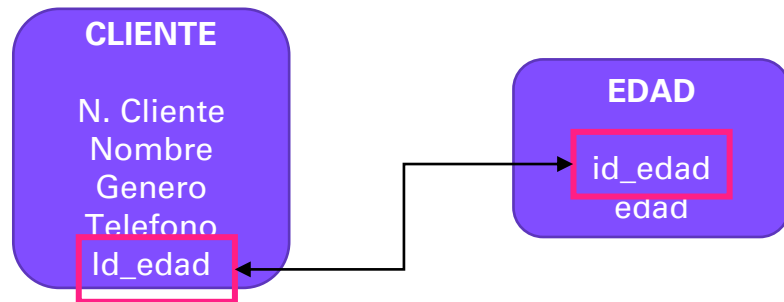


# Relaciones

## Uno a uno

### *One-to-one*

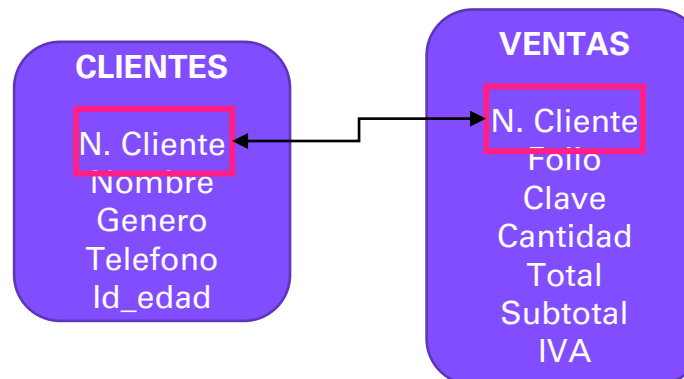
- Las relaciones uno a uno son aquellas que para cada registro de ambas tablas existe una relación univoca.
- Esto significa que cada valor en las tablas aparece solo una vez en la tabla.



## Uno a muchos

### *One-to-many*

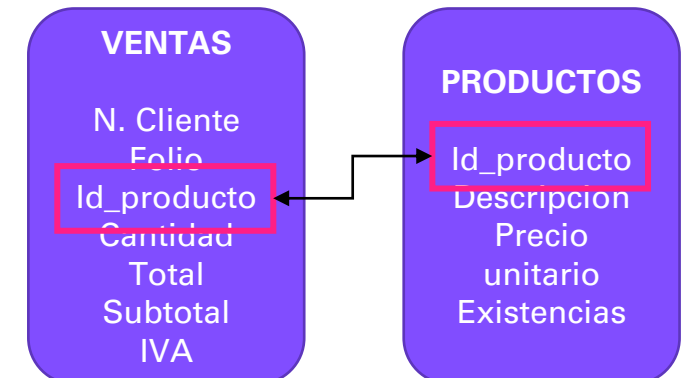
- En estos casos, consideramos que los datos de una tabla pueden aparecer múltiples veces en la siguiente.
- La restricción es que en una de ellas el registro debe ser único



## Muchos a muchos

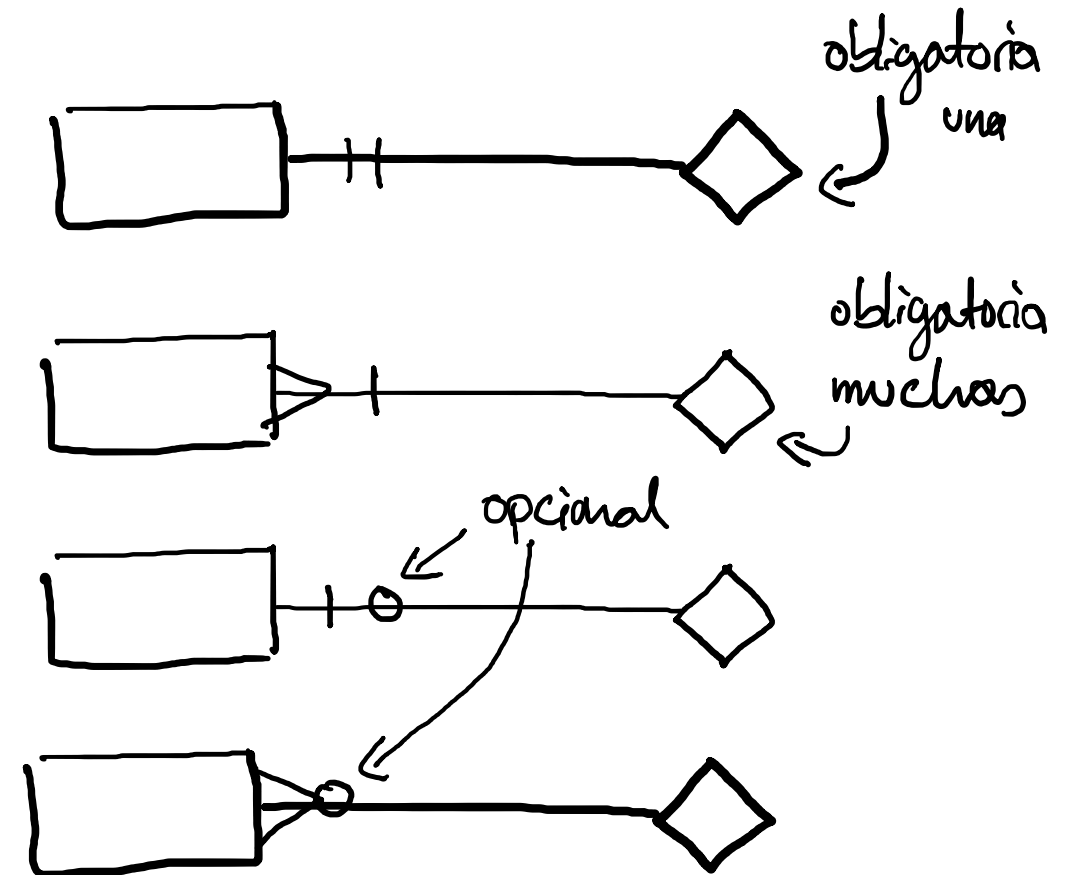
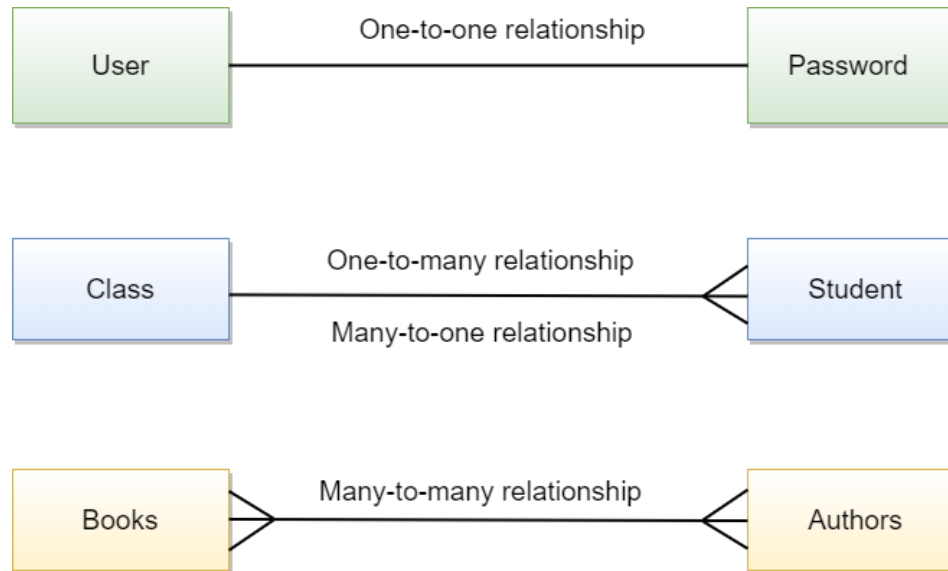
### *Many-to-Many*

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



# Relaciones

## representación



**LLAVES**



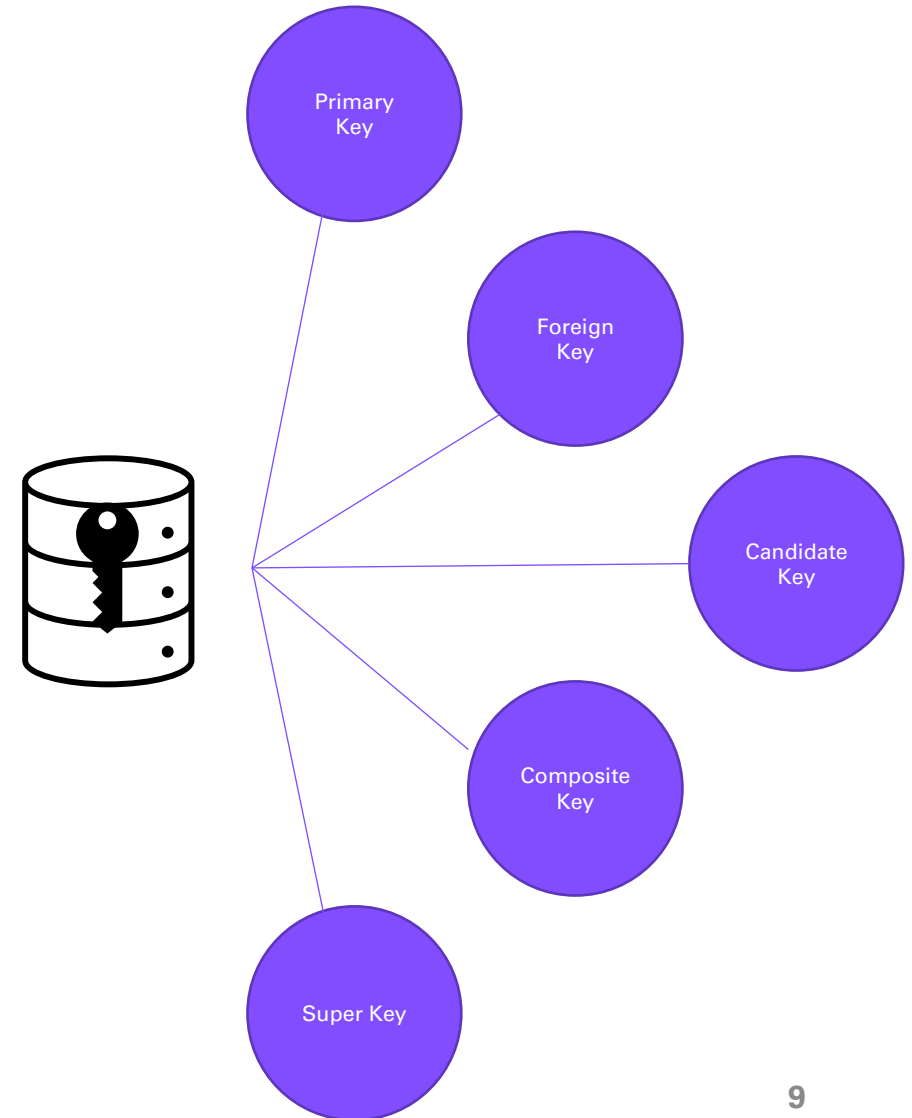


# Claves (Keys)

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*)

## 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

## 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

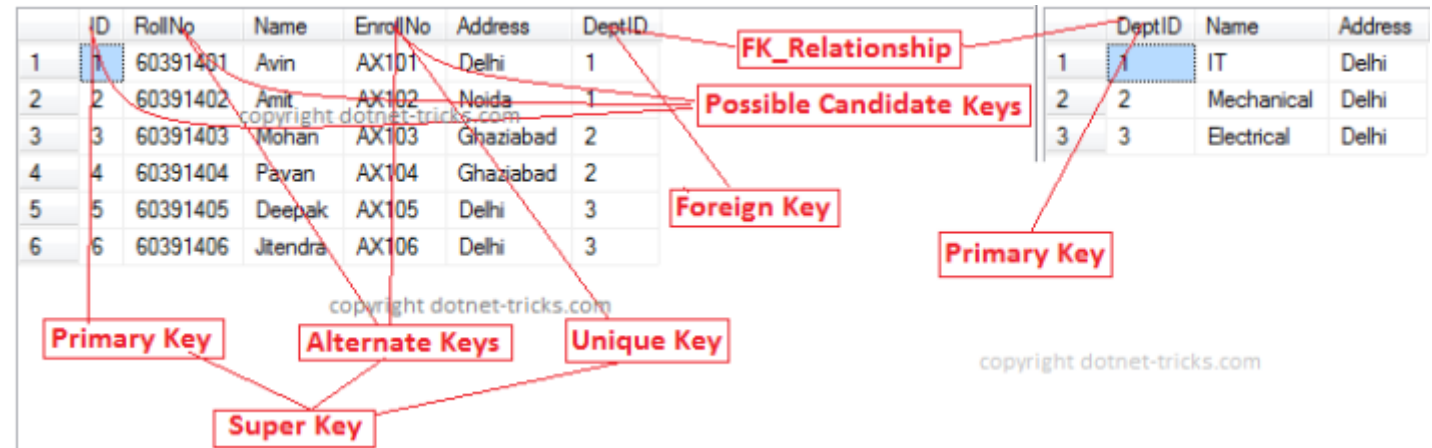
## Foreign Key

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

Acepta nulos

Acepta duplicados

# Keys (*keys*)





# DER



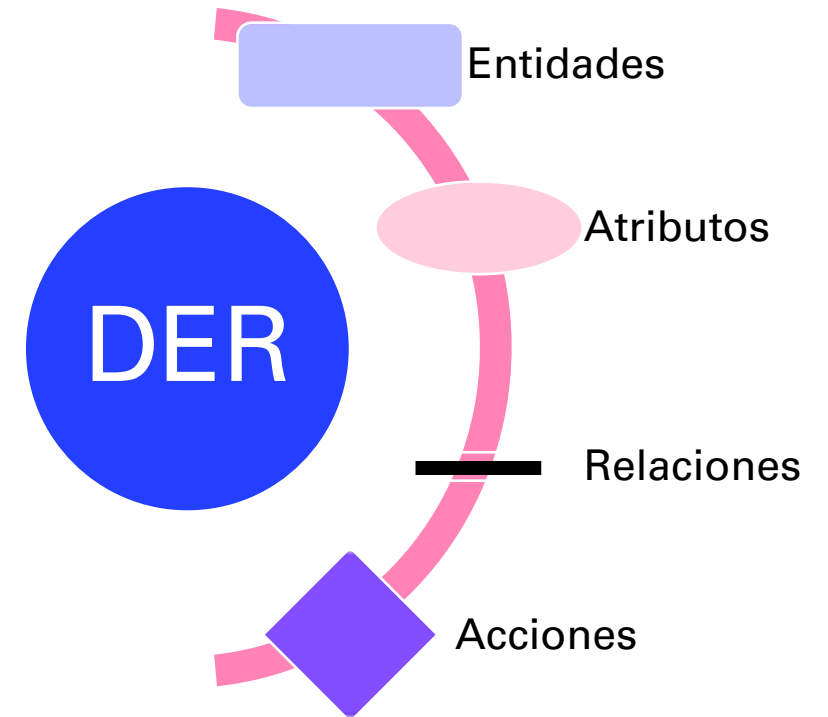
# Conceptos básicos

## 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**



Entidad: el concepto en el q' baso el modelo.

**Entidad:** Todo objeto o concepto del cual queremos registrar información constituye una entidad.

ent. fuerte 

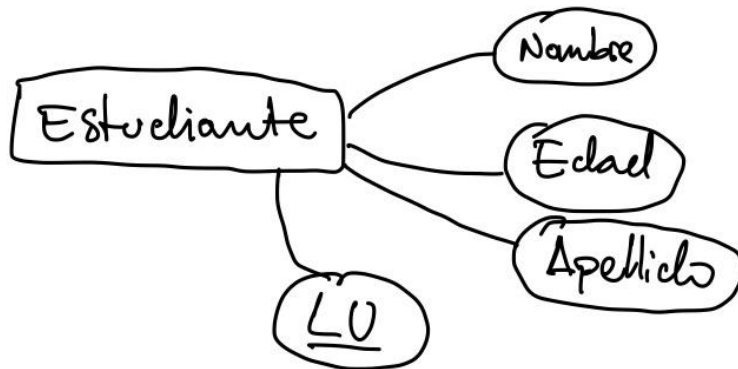
→ Entidades fuertes → existe por sí misma

↳ Ent. débiles → depende de una fuerte

ent. debil 

Atributos: son las comp. de mi entidad.

**Atributo:** Las propiedades que componen mi entidad



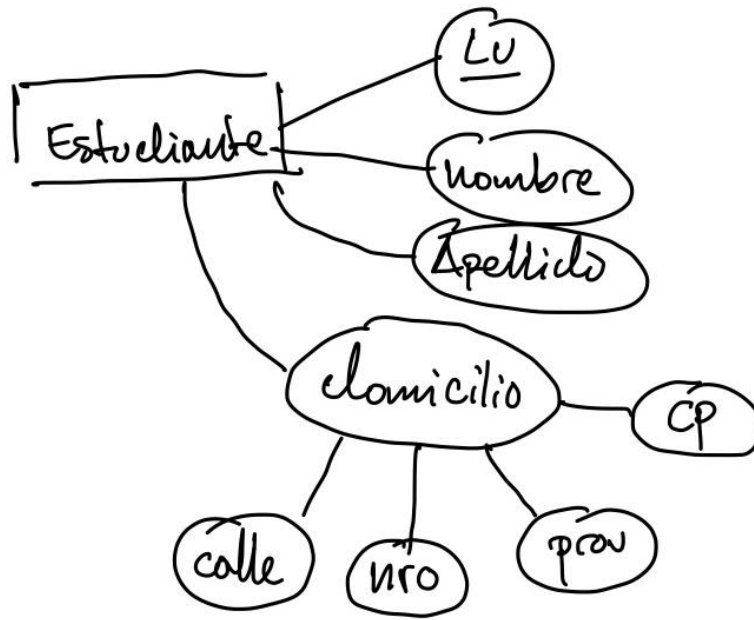
- Multivaluados → puede tener un set de valores

- compuestos

+ Jerarquias

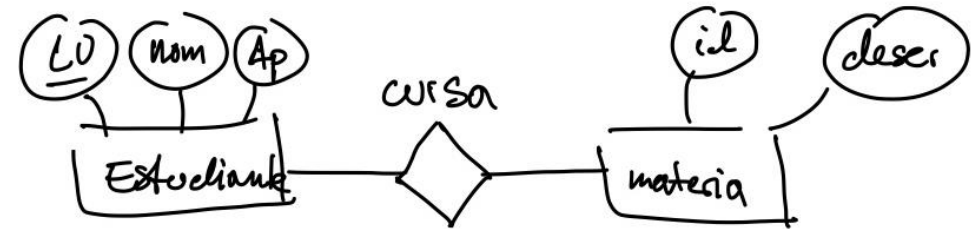
↳ Existen uno o más attr que forman este attr

↳ mi attr princ. puede dos o más formas.



**Relaciones:** son las acciones que interconectan las entidades. Las relaciones pueden tener una o mas entidades participantes

• interacciones y las acciones: es el fenómeno que conecta entidades



LU	nombre	Apellido
1	Carlos	Patito



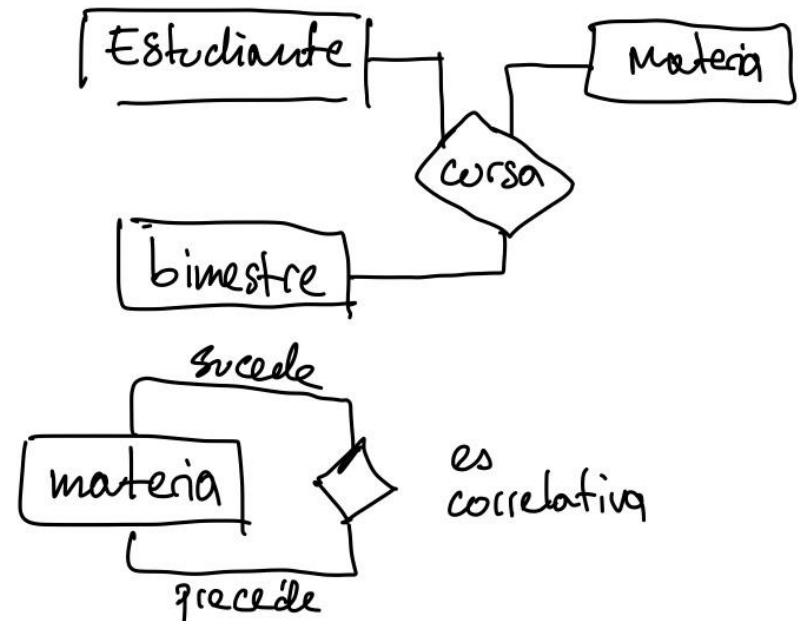
id	nombre	descri
6	bdia	cosas

LU	nombre	Apellido	curso
1	carlos	patito	6

en interrelaciones tenemos 3 características → Grado  
↳ Cardinalidad  
↳ Participación

Grado: es la cantidad de entidades a relacionar

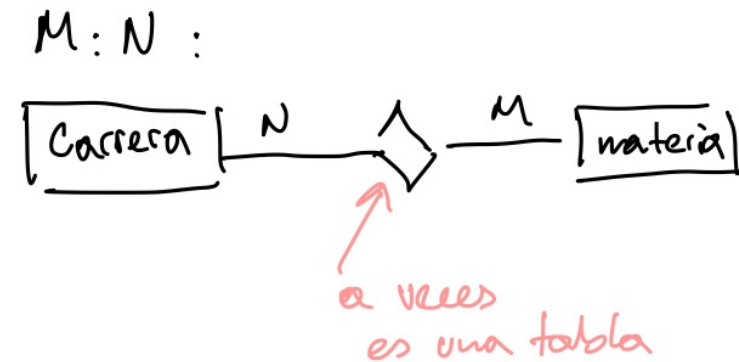
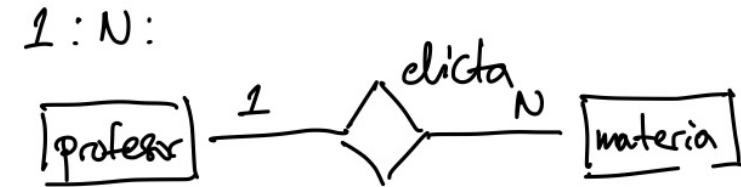
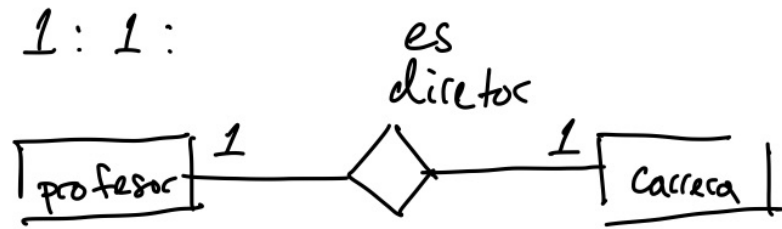
- relación unaria (consigo misma)
- " binaria (entre 2 ent.)
- " ternaria (3 ent.)
- " n-aria





**Cardinalidad:** refiere a la cant. de registros que conectan la interrelación.

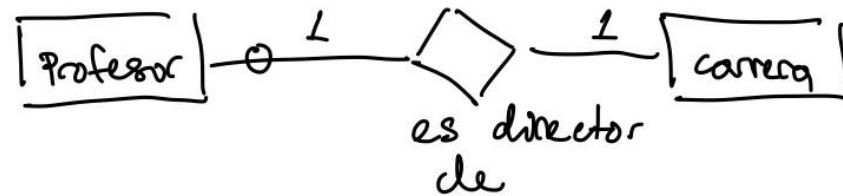
- 1 : 1
- 1 : N
- M : N



Participación: la necesidad de una entidad en una relación

→ Participación parcial (opcional)

→ Participación Total (obligatoria):



# A PRACTICAR!!

---

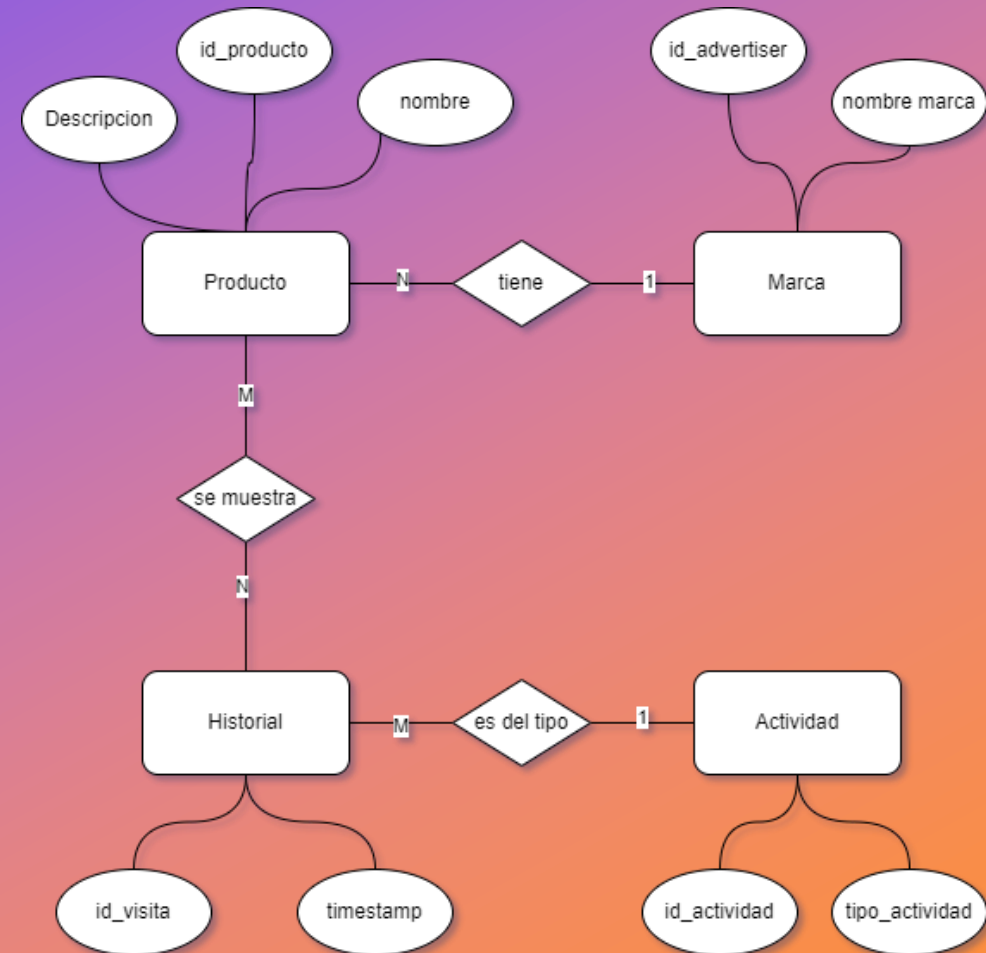


# Ejemplo de análisis

Vamos a crear un DER a partir del desarrollo de un problema de advertising. Supongamos que nuestra empresa *patitos*<sup>TM</sup> 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.

Idx	Marca	Producto	interacción	Fecha
1	5E325T5HYL61QSABVR 5V	9trbal	impression	4/1/2022
2	03KNVBO915KY2ZPGA 57J	qd5esu	impression	4/1/2022
3	HC26ZE93SA4WWA0B RFM6	99watc	impression	4/1/2022
...				

# SOLUCIÓN PROPUESTA (CASI UNÁNIME!)



# SQL PRIMEROS PASOS

+

•

○

+

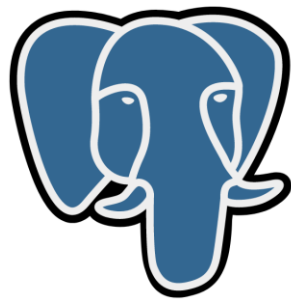
○

•

# Primeros pasos



Amazon RDS



PostgreSQL

- Para esta parte vamos a crear nuestra base directamente en AWS.
- Por simplicidad vamos a utilizar PostgreSQL
- Tutorial:  
<https://aws.amazon.com/getting-started/hands-on/create-connect-postgresql-db/>

**NOTA:** No es necesario activar Enhanced Monitoring ni logs. Lo más importante es **exponer públicamente** la base de datos para poder conectarnos desde Dbeaver, pgAdmin, o el connector de su gusto

BASE DE DATOS

+



o



.



# DUDAS?