



Bases de datos relacionales

El modelo entidad-relación

Introducción

- En el proceso de diseño de una BD se obtiene el esquema conceptual, en el que se definen todos los datos que intervienen en el problema y sus relaciones.
- El modelo conceptual más utilizado es el modelo entidad-relación (E/R).
- Este modelo fue propuesto por Peter Chen en 1976 para la representación gráfica de los problemas que forman parte del mundo real.
- Utiliza un conjunto de símbolos y reglas para representar los datos y las relaciones entre ellos.

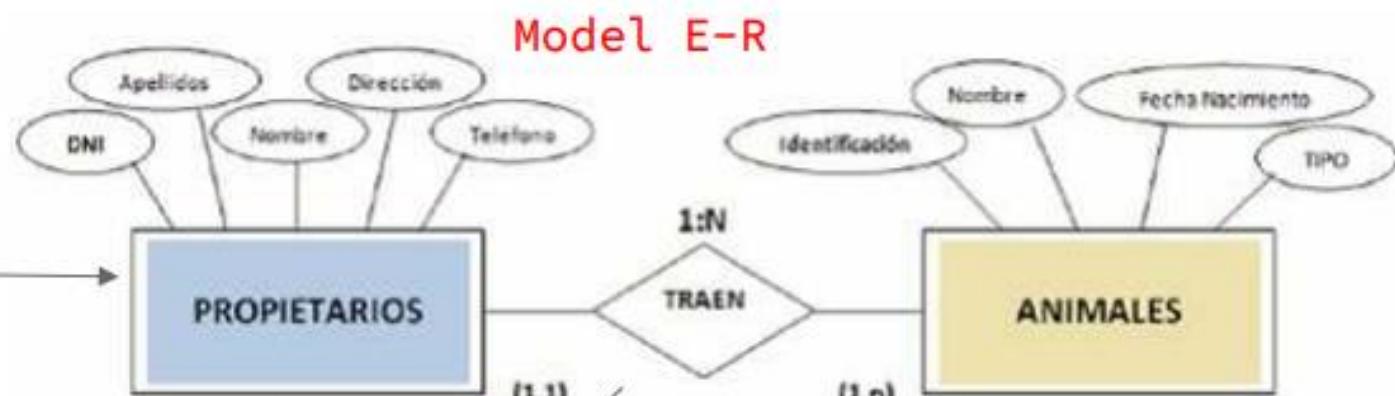


PROPIETARIOS

Món real
TRAEN



ANIMALES



PROPIETARIOS
DNI (PK)
Apellidos
Nombre
Direccion
Telefono

Model Relacional

ANIMALES
Identificación (PK)
Nombre
Fecha_nacimiento
TIPO
DNI (FK)

Introducción

ELEMENTOS DEL MODELO ENTIDAD - RELACIÓN

Es un modelo de datos que representa la realidad a través de **entidades**, que son objetos que existen y se distinguen de otros por sus características, que llamamos **atributos**.

Además, estas entidades podrán o no estar relacionadas unas con otras a través de lo que se conoce como **relación**.

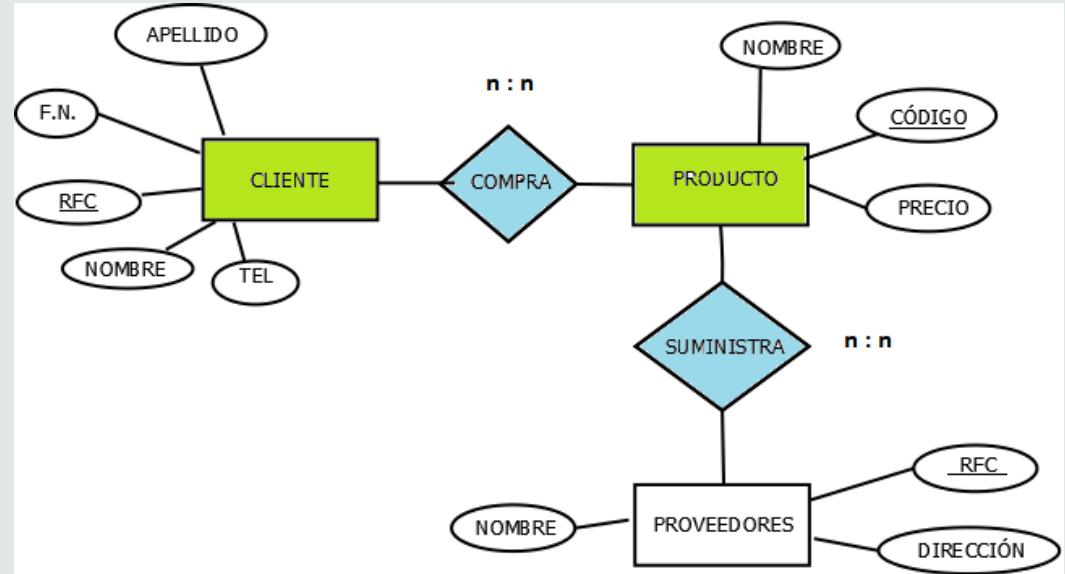
Hay que tener en cuenta que se trata solamente de un modelo de representación, por lo que no tiene correspondencia real con ningún sistema de almacenamiento.

Se utiliza en la etapa de Análisis y Diseño de una Base de Datos, por lo que habrá que convertirla a otro modelo antes de poder empezar a trabajar con ella.

ENTIDADES

Entidad: es un objeto acerca del cual se recoge información de interés para la base de datos. Por ejemplo: ALUMNO, se recogen los datos de los alumnos de un centro de estudios (DNI, nombre, dirección o población).

Las entidades **se representan** gráficamente por medio de **rectángulos**.



CLIENTE, PRODUCTO y
PROVEEDORES
son las entidades en este modelo
E/R

ENTIDADES

Existen entidades fuertes y entidades débiles. Las **entidades fuertes** son aquellas que **no dependen de otra entidad** para su existencia. Por ejemplo, la entidad EMPLEADO. Las **entidades débiles** son las que **dependen de otra entidad** para su existencia.

Ejemplos: Tren (fuerte) y vagón (débil) libro (fuerte) y ejemplar (débil), cuenta bancaria (fuerte) y transacción(débil).

Las entidades débiles se representan con un doble rectángulo.



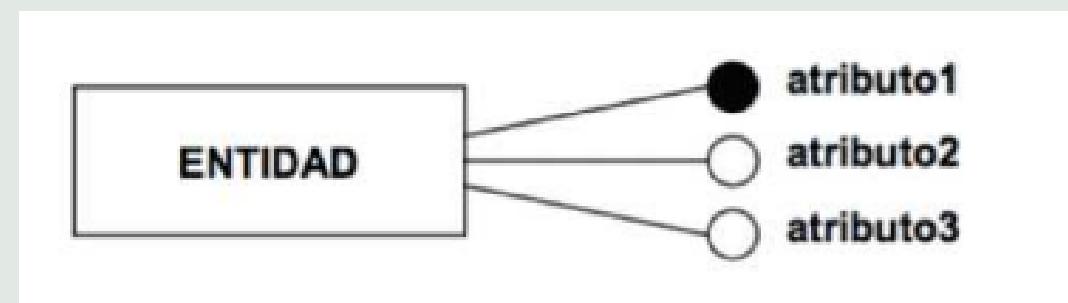
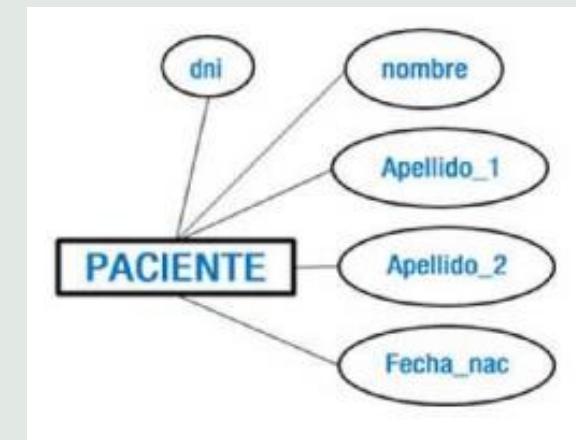
ATRIBUTOS

Atributo: es cada una de las propiedades o características de una entidad o una relación.

Cada atributo se identifica por un nombre y por todos los posibles valores que puede tener.

Se representa gráficamente con un **círculo unido a una línea** que parte de la entidad o relación de la que forma parte.

Otra forma de representar los atributos es con pequeños círculos con el nombre al lado.

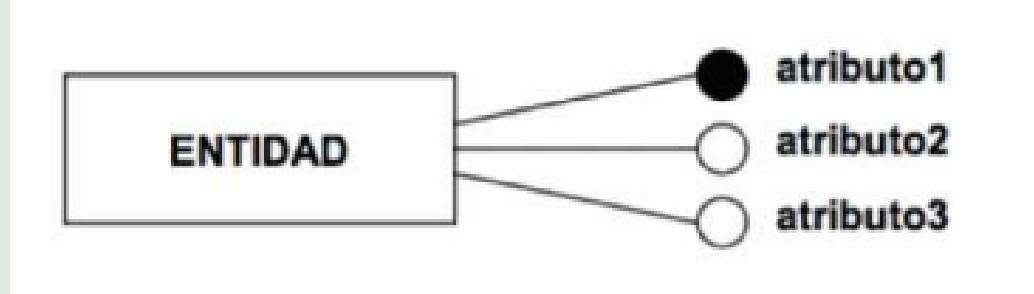


ATRIBUTOS

¿Todos los atributos son iguales?

El atributo o conjunto de atributos que identifican de forma univoca a la entidad se denomina **atributo identificador (clave primaria)**. Es **obligatorio** que exista al menos un atributo identificador.

El resto de los atributos son los descriptores y muestran características de la entidad.



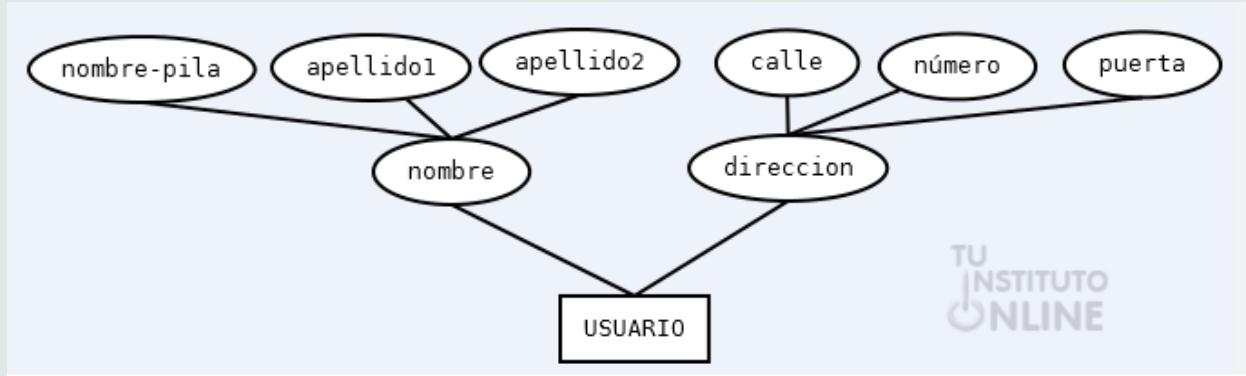
ATRIBUTOS

Tipos de atributos

Según su composición pueden ser:

Simples: son aquellos que tienen un solo componente y que no se pueden dividir en partes más pequeñas con significado propio.

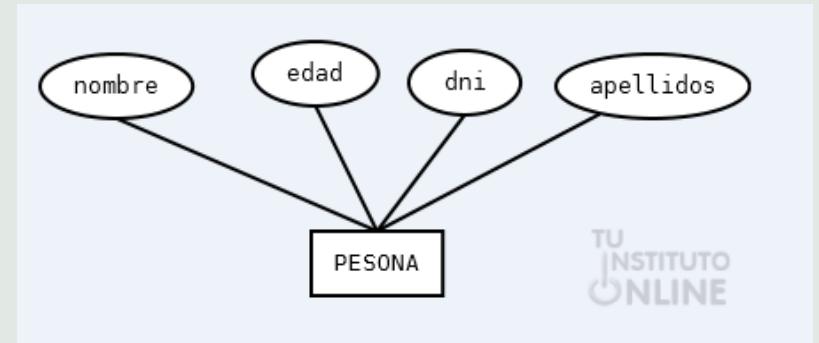
Compuestos: son los que están formados por varios componentes y que tienen afinidad en cuanto a su significado. Se representan, con círculos unidos a cada uno de los atributos de los que se compone.



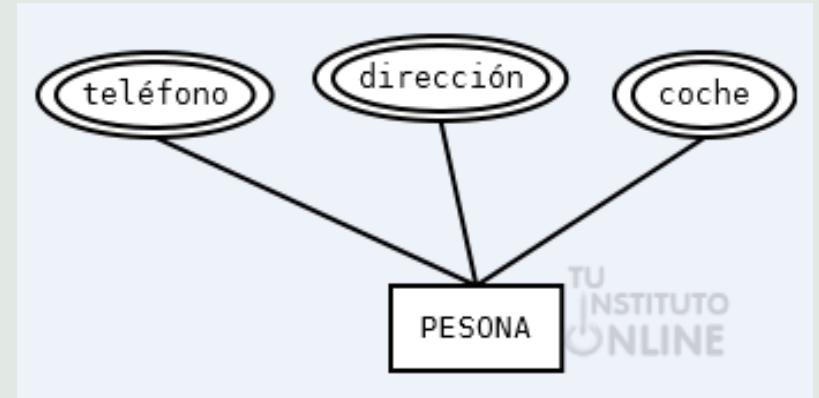
ATRIBUTOS

Atendiendo a los valores:

Monovaluados: es aquel que tiene un solo valor por cada ocurrencia de la entidad a la que pertenece.



Multivaluados: pueden tener varios valores por cada ocurrencia de la entidad. Se representan por un doble círculo.

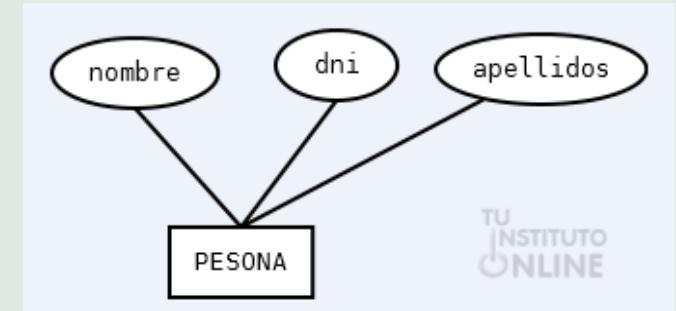


ATRIBUTOS

Atendiendo al origen:

Almacenados: cuyos datos se almacenan directamente en la base de datos sin necesidad de realizar ningún trámite intermedio.

Derivados: son aquellos que son obtenidos a partir del valor de uno o varios atributos existentes en la misma o en otras entidades. Se representa mediante círculos discontinuos.

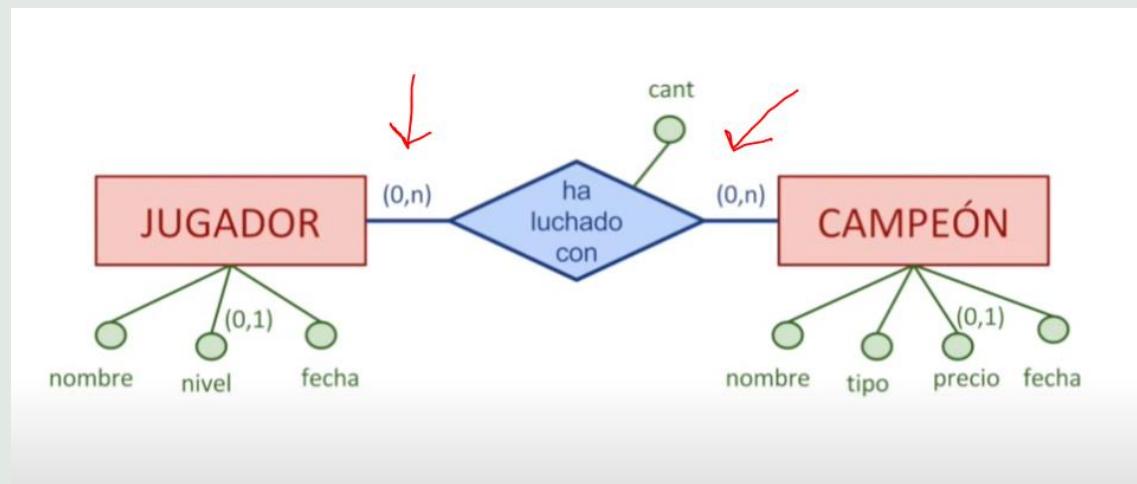


CARDINALIDAD

La **cardinalidad** es el número de tuplas que contiene una entidad o una relación. Ya que se van insertando y borrando tuplas a menudo, la cardinalidad varía constantemente.

Expresamos la cardinalidad definiendo el número **máximo** y el número **mínimo** de **ocurrencias** de cada tipo de entidad. Para representar la cardinalidad se encierra entre paréntesis los valores máximo y mínimo, de la siguiente manera: (**min, máx.**).

Los valores para la cardinalidad son: (0,1), (1,1), (0,N), (1,N) y (N,M).



DOMINIOS

Un **dominio** (Conjunto de Valores Legales) define la colección de valores de donde uno o más atributos obtienen sus valores reales.

Desde un punto de vista intuitivo, un dominio es equivalente a un **tipo de datos**. De este modo, se definen atributos (variables) que pertenezcan a un dominio (tipo de datos).

Atributo	Dominio
DNI	Cadena de Caracteres de longitud 10
Nombre	Cadena de Caracteres de longitud 50
Fecha_Nacimiento	Fecha
Dirección	Cadena de Caracteres de longitud 100
Sueldo	Números reales
Número de hijos	Números enteros
Departamento	Departamentos

DOMINIOS

El concepto de **dominio** es importante porque permite que el usuario defina, en un lugar común, el significado y la fuente de los valores que los atributos pueden tomar. Esto hace que haya más información disponible para el sistema cuando éste va a ejecutar una operación relacional, de modo que las operaciones que son semánticamente incorrectas se pueden evitar.

Por ejemplo, no tiene sentido comparar el nombre de una calle con un número de teléfono, aunque los dos atributos sean cadenas de caracteres. Sin embargo, el importe mensual del alquiler de un inmueble no estará definido sobre el mismo dominio que el número de meses que dura el alquiler, pero sí tiene sentido multiplicar los valores de ambos dominios para averiguar el importe total al que asciende el alquiler.

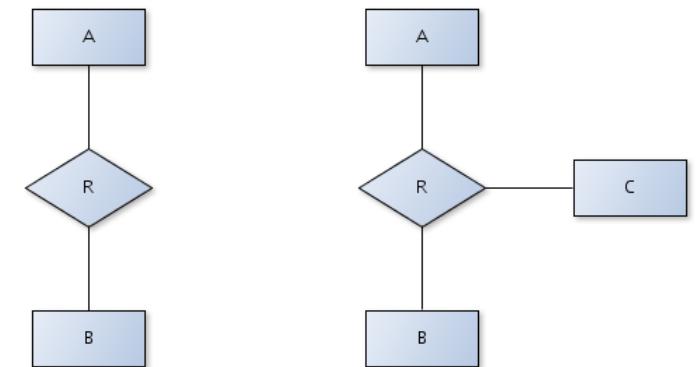
Los **SGBD relacionales no ofrecen soporte completo a los dominios** ya que su implementación es extremadamente compleja.

RELACIONES

Relación: es una asociación entre dos o más entidades. Suele tener un nombre que la identifica respecto al resto de las relaciones, ya que cada relación tiene un significado específico.

Las relaciones **se representan** gráficamente **con rombos**.

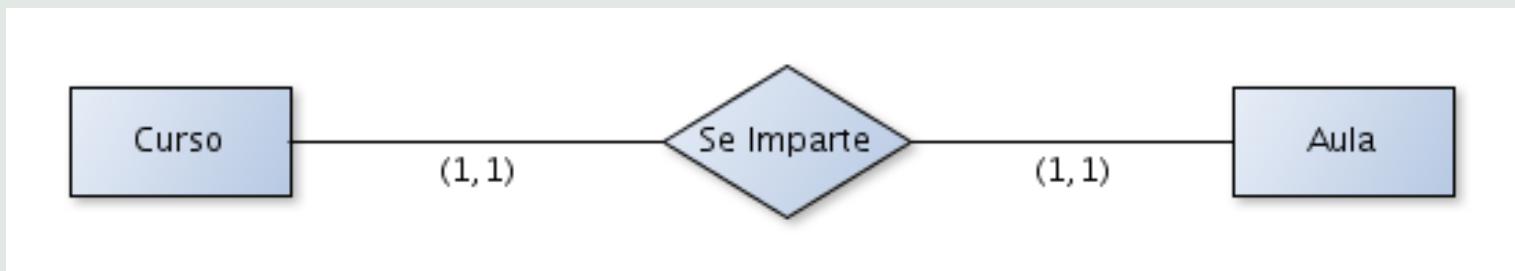
Si consideramos que dos entidades A y B están relacionadas a través de una relación R, deberemos determinar lo que se conoce como **cardinalidad de la relación**, que determina cuantas entidades de tipo A se relacionan, como máximo, con cuantas entidades de tipo B.



RELACIONES (1:1)

- Uno a uno: Se representa por 1:1. A cada ocurrencia de A le corresponde, como máximo, una ocurrencia de B, y viceversa.

En esta relación una entidad de tipo A sólo se puede relacionar con una entidad de tipo B, y viceversa. Por ejemplo, si suponemos dos entidades *Curso* y *Aula*, relacionadas a través de una relación *Se Imparte*, podremos suponer que un *Curso* se imparte en una *Aula* y en una *Aula* sólo se puede impartir un *Curso*. Se representaría como sigue:

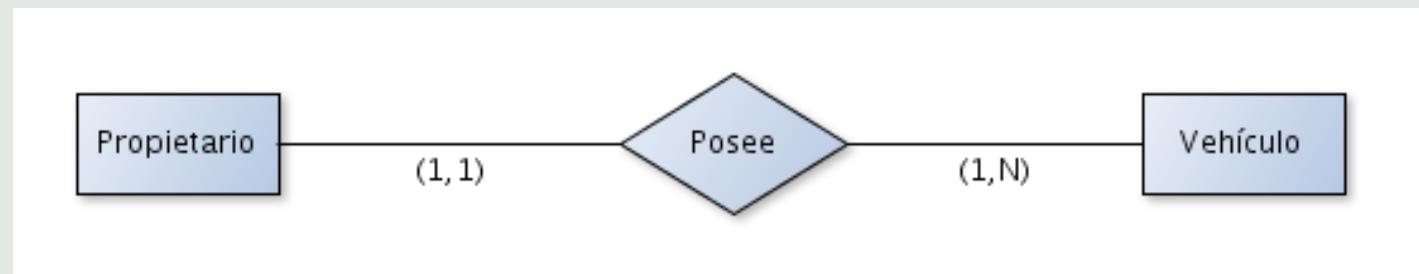


RELACIONES (1:N)

- Uno a muchos: Se representa por **1:N**. A cada ocurrencia de A le pueden corresponder varias de B, y a cada ocurrencia de B sólo le corresponde una de A. Si la relación es de B hacia A, la relación sería N:1.

Indica que una entidad de tipo A se puede relacionar con un número indeterminado de entidades de tipo B, pero a su vez una entidad de tipo B sólo puede relacionarse con una entidad de tipo A. Si suponemos una entidad Propietario y otra entidad Vehículo relacionadas a través de una relación Posee, podremos suponer que un Propietario puede poseer varios Vehículos, mientras que cada Vehículo sólo puede pertenecer a un Propietario.

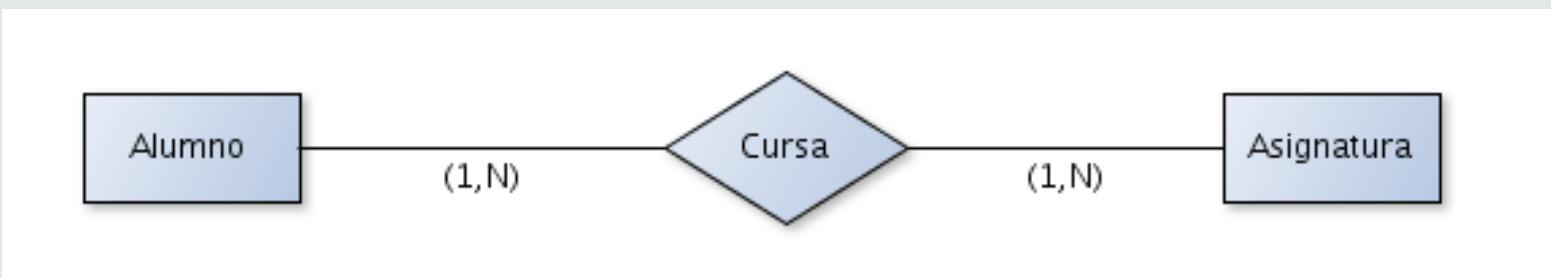
Quedaría representado de la siguiente manera:



RELACIONES (N:M)

- Muchos a muchos: Se representa por N:M. A cada ocurrencia de A le pueden corresponder varias de B, y viceversa.

En este caso, tanto las entidades de tipo A y B, pueden relacionarse con un número indeterminado de entidades del otro tipo. Por ejemplo, si suponemos las entidades Alumno y Asignatura y una relación Cursa, podremos suponer que un Alumno cursa varias asignaturas mientras que una Asignatura la cursan varios Alumnos. Quedaría representado de la siguiente manera:

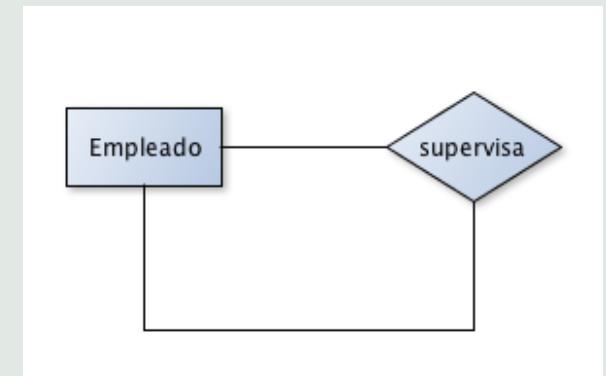


RELACIONES: REFLEXIVIDAD

Es posible que la misma entidad ocupe ambos lados de una relación. En ese caso estamos frente a lo que se conoce como relaciones reflexivas.

La cardinalidad de la relación indicará si todos los elementos de la relación están relacionados reflexivamente o bien sólo algunos están relacionados entre sí.

En el caso de la figura podríamos suponer una empresa en la que algunos empleados hacen de supervisor de otros empleados.



GRADO DE LAS RELACIONES

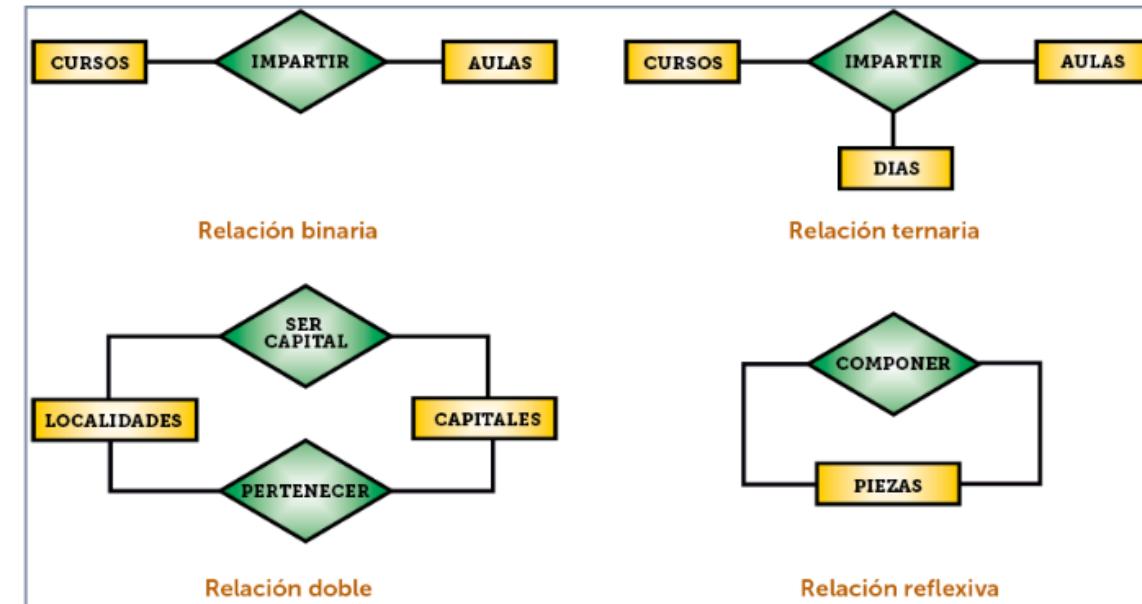
El **grado de la relación** es el número de entidades que participan en una relación.

Grado 1 o reflexivas: se relaciona una entidad consigo misma.

Grado 2 o binarias: dos entidades.

Grado 3 o ternarias: tres entidades.

Grado n o n-arias: n entidades.



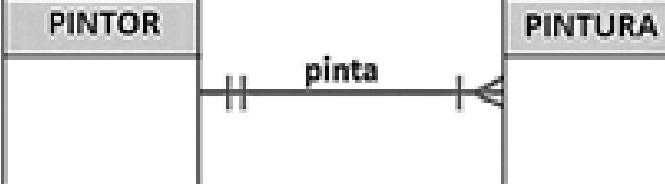
REPRESENTAR UNA RELACIÓN

NOTACIÓN CHEN

Relación uno a muchos (1:N). Un PINTOR puede pintar varias PINTURAS; cada PINTURA es pintada por un PINTOR



NOTACIÓN PATA DE GALLO



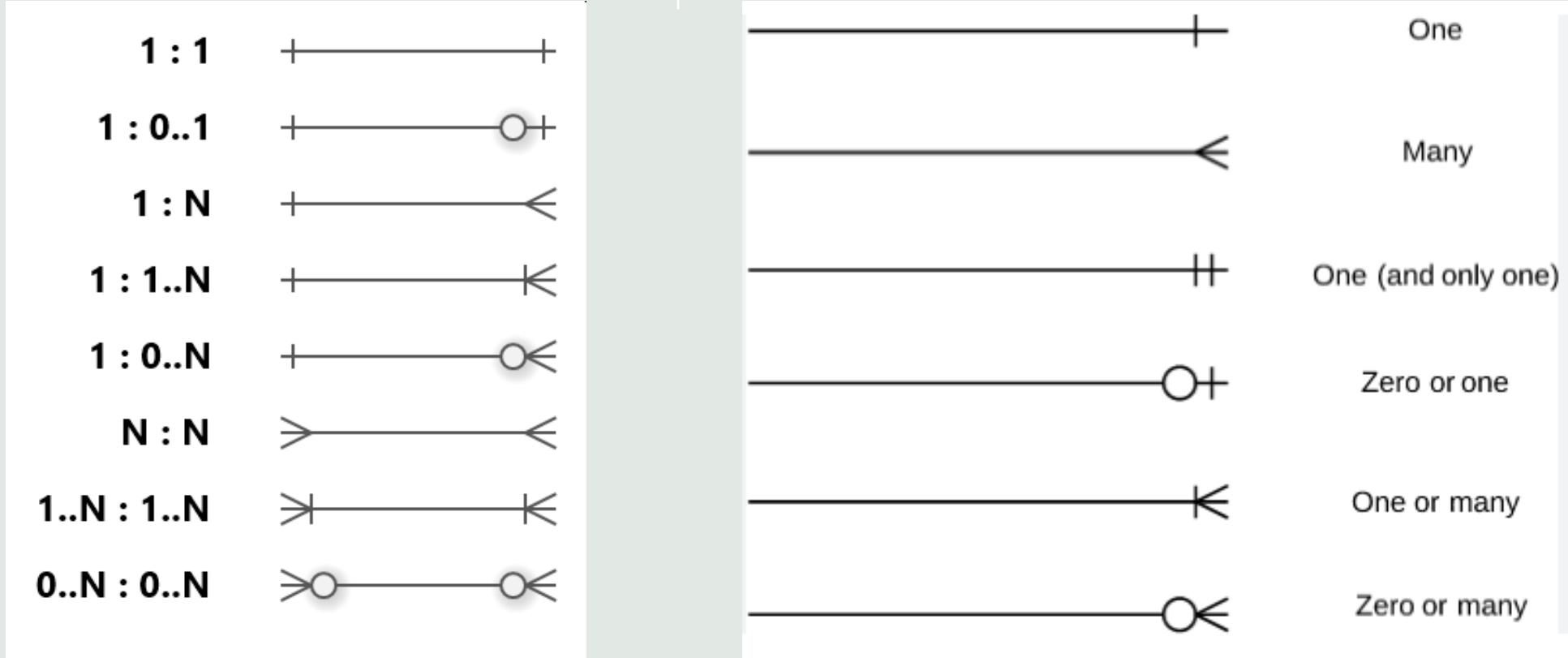
Relación muchos a muchos (N:N). Un EMPLEADO puede aprender muchas HABILIDADES, cada HABILIDAD puede ser aprendida por muchos EMPLEADOS



Relación uno a uno (1:1). Un EMPLEADO administra un ALMACÉN. Cada ALMACÉN es administrado por un EMPLEADO



REPRESENTAR UNA RELACIÓN



RELACIONES ESPECIALES

CLAVES

CLAVE CANDIDATA

Conjunto de atributos que identifican inequívocamente cada tupla de la relación. Es decir, columnas cuyos valores no se repiten en ninguna otra fila de esa tabla. Toda tabla, en el modelo relacional, debe tener al menos una clave candidata, pero puede haber muchas más.

CLAVE PRIMARIA

Clave candidata que se escoge como **identificador** de la tabla (para identificar cada fila de la misma).

Se debe dar prioridad a atributos cuyos dominios sean más eficientes de cara al tamaño que ocupan en disco y a la facilidad de proceso por parte de la máquina.

Por ejemplo, en una base de datos de una empresa de venta de productos, un campo con el *DNI* sería clave candidata de una tabla de clientes; si esa tabla tiene un campo de *código de cliente*, éste sería mejor candidato (y por lo tanto clave principal) porque es mejor identificador para ese contexto.

CLAVES

Las claves primarias son los únicos datos en el modelo relacional que provocan redundancia ya que se duplican en las claves secundarias para establecer las relaciones entre las tablas. Por ello hay que intentar que las claves primarias contengan datos pequeños, nunca textos largos y de tamaño variable (como nombres, títulos,...).

Buenos tipos de datos para claves primarias son las fechas, números enteros y textos, si son cortos y de tamaño fijo.

CLAVE ALTERNATIVA

Cualquier clave candidata que no es elegida como primaria. Hay que tenerlas en cuenta para, al menos, aplicar las restricciones correspondientes (se explican más adelante).

CLAVES

CLAVE AJENA, EXTERNA O SENCUNDARIA

Son los datos de atributos de una tabla cuyos valores están relacionados con atributos de otra tabla. Por ejemplo, en la tabla equipos tenemos estos datos:

Equipo	Nº Equipo
Real Madrid	1
F.C. Barcelona	2
Athletic Bilbao	3

En la tabla anterior la clave principal es el atributo nº equipo. En otra tabla tenemos:

Nº Jugador	Jugador	Nº Equipo
1	Muniain	3
2	Messi	2
3	Cristiano Ronaldo	1
4	Bale	1

CLAVES

El atributo *Nº Equipo* sirve para relacionar el Jugador con el equipo al que pertenece. Ese campo en la tabla de jugadores es una **clave ajena**. En el detalle siguiente se resalta la relación entre las claves primarias (en la base de las flechas) y las claves ajenas(en la punta de las flechas):



VALOR NULO

El nulo es considerado un valor fundamental ya que otorga de significado al atributo tanto como si tuviera un valor numérico o textual.

Por ejemplo, un valor null puesto en una clave ajena, indica que la fila que donde se ha puesto ese valor, no tiene ninguna clave principal asociada.

En un atributo para indicar el teléfono de una persona, significaría que esa persona no tiene teléfono.

Es importante indicar que el texto vacío ' ', no significa lo mismo que el valor nulo. Tampoco el valor numérico cero (0) significa nulo.

VALOR NULO

Puesto que los nulos se utilizan continuamente, resulta imprescindible saber cómo actúa cuando se emplean operaciones lógicas sobre ellos. Eso significa definir un tercer valor en la lógica booleana, además de los clásicos **verdadero** y **falso**. Un valor nulo no es ni verdadero ni falso (se suele interpretar como un quizás, o usando la aritmética clásica en valores lógicos, el 1 es verdadero, el 0 falso y el 0,5 nulo).

El uso de operadores lógicos con el nulo da lugar a que:

- **verdadero Y (AND) nulo** da como resultado, nulo. Siguiendo la aritmética planteada antes: $1 \cdot 0,5 = 0,5$
- **falso Y (AND) nulo** da como resultado, falso ($0 \cdot 0,5 = 0$)
- **verdadero O (OR) nulo** da como resultado, verdadero ($1 + 0,5 > 1$)
- **falso O nulo** da como resultado nulo ($0 + 0,5 = 0,5$)
- **la negación de nulo** da como resultado nulo

Se utiliza un operador en todas las bases relacionales llamado **es nulo** (*is null*) que devuelve verdadero si el atributo con el que se compara tiene valor nulo.

Reglas de integridad



Las reglas de integridad son restricciones que se deben cumplir en todas las bases de datos relacionales y en todos sus estados (las reglas se deben cumplir todo el tiempo). Estas reglas son:

- Regla de integridad de entidades
- Regla de integridad referencial

Regla de integridad de entidades

La primera regla de integridad se aplica a las claves primarias de las relaciones:

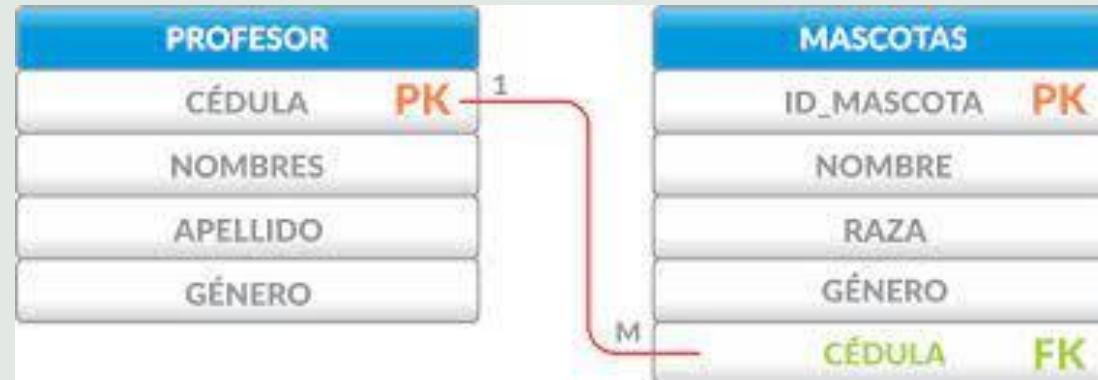
ninguno de los atributos que componen la clave primaria puede ser nulo.

Por definición, una clave primaria es una clave irreducible que se utiliza para identificar de modo único tuplas. Si se permitiera que parte de la clave primaria fuera nula, se estaría diciendo que no todos sus atributos son necesarios para distinguir las tuplas, con lo que se estaría contradiciendo la irreducibilidad.



Regla de integridad referencial

La segunda regla de integridad se aplica a las claves ajena: si en una relación hay alguna clave ajena, sus valores deben coincidir con valores de la clave primaria a la que hace referencia, o bien, deben ser completamente nulos.



Regla de integridad referencial

Para hacer respetar esta regla se debe contestar, para cada clave ajena, a las tres preguntas siguientes:

Regla de los nulos ¿Tiene sentido que acepte nulos?

Regla de borrado ¿Qué ocurre si se intenta borrar la tupla referenciada por la clave ajena?

Regla de modificación ¿Qué ocurre si se intenta modificar la tupla referenciada por la clave ajena?

Regla de integridad referencial

Ante el borrado y la modificación se pueden determinar los siguientes comportamientos:

- Restringir: no se permite borrar/modificar la tupla referenciada.
- Propagar: se borra/modifica la tupla referenciada y el cambio se propaga a las tuplas que la referencian mediante clave ajena.
- Anular: se borra/modifica la tupla referenciada y las tuplas que la referenciaban ponen a nulo la clave ajena (sólo si acepta nulos):
- Valor por defecto: se borra/modifica la tupla referenciada y las tuplas que la referenciaban ponen en la clave ajena un valor establecido por defecto.

PROBANDO . . .

Representa un diagrama E/R que contenga:

Una entidad EMPLEADO formada por los atributos: n° empleado, Apellido, Salario y Comisión.

Una entidad DEPARTAMENTO con los atributos: n° departamento, nombre y localidad.

Estas dos entidades se relacionan de forma que a un departamento le pertenecen cero o más empleados y un empleado pertenece a un departamento y sólo a uno.

EJEMPLO DEPARTAMENTO - EMPLEADOS

La relación "PERTENECE" entre las entidades EMPLEADOS y DEPARTAMENTO tiene la correspondencia 1:N.

Es decir, a un departamento le pertenecen cero o más empleados(0,N) y un empleado pertenece a un departamento y sólo a uno (1:1).

