

BD

Bases de Datos

UD 2

Diseño conceptual – Diagramas ER

ÍNDICE

1. Diseño de bases de datos
2. Los diagramas E-R
3. Relaciones
4. Paso a paso – Cómo hacer un diagrama E-R
5. Diagrama E-R extendido (generalización y especialización)

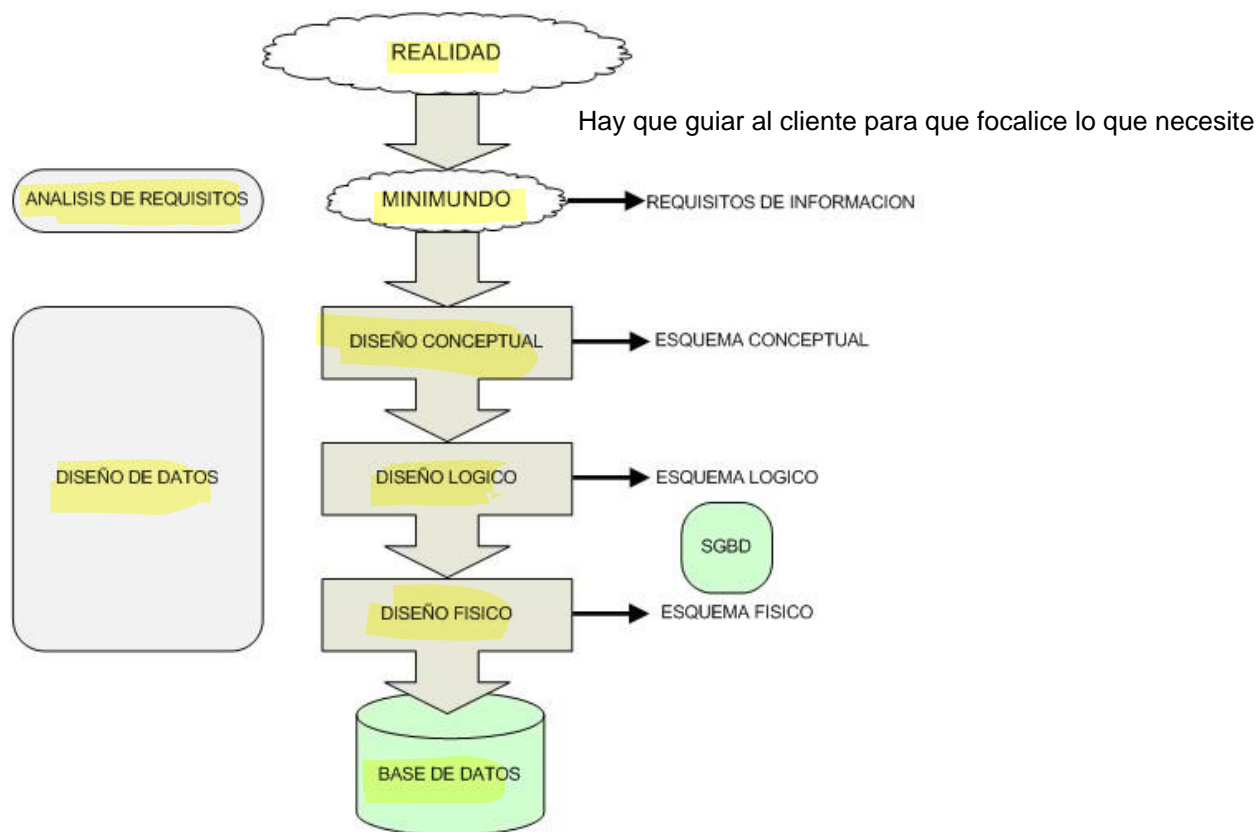
1. Diseño de Bases de Datos

El **Diseño de Bases de Datos** es el proceso por el que se determina la organización de una Base de Datos, incluidas su estructura, contenido y las aplicaciones que se han de desarrollar.

El diseño de Base de Datos desempeña un papel central en el empleo de los recursos de información en la mayoría de las organizaciones. El diseño de Base de Datos ha pasado a constituir parte de la formación general de los informáticos, en el mismo nivel que la capacidad de construir algoritmos usando un lenguaje de programación convencional.

Según ha ido avanzando la tecnología de Bases de Datos, se han desarrollado y adaptado las metodologías y técnicas de diseño. Ahora existe un consenso, por ejemplo, sobre la descomposición del proceso en fases, sobre los principales objetivos de cada fase y sobre las técnicas para conseguir estos objetivos. Así, el diseño de una Base de Datos se descompone en:

- **diseño conceptual (UD2)**
- **diseño lógico (UD3)**
- **diseño físico (UD4)**



El **diseño conceptual** parte de las especificaciones de requisitos de usuario y su resultado es el esquema conceptual de la Base de Datos.

El modelo más utilizado actualmente es el **Entidad-Relación** y será el que trabajaremos durante el curso. Más adelante explicaremos en qué consiste.

El **diseño lógico** es el proceso de construir un esquema de la información que utiliza la empresa, basándose en un modelo conceptual de base de datos específico, independiente del SGBD concreto que se vaya a utilizar y de cualquier otra consideración física.

En esta etapa, se transforma el esquema conceptual en un esquema lógico que utilizará las estructuras de datos del modelo de base de datos en el que se basa el SGBD que se vaya a utilizar, como puede ser el modelo relacional, el modelo de red, el modelo jerárquico o el modelo orientado a objetos.

La **normalización** es una técnica que se utiliza para comprobar la validez de los esquemas lógicos basados en el modelo relacional, ya que asegura que las tablas obtenidas, conociendo las entidades, sus atributos y las relaciones, no tienen datos redundantes, lo cual permite corregir errores de diseño antes de realizar la implementación en un SGBD.

Tanto el diseño conceptual, como el diseño lógico, son procesos iterativos, tienen un punto de inicio y se van refinando continuamente.

Ambos se deben ver como un proceso de aprendizaje en el que el diseñador va comprendiendo el funcionamiento de la empresa/organización y el significado de los datos que maneja.

El diseño conceptual y el diseño lógico son etapas clave para conseguir un sistema que funcione correctamente. Si el esquema no es una representación fiel de la realidad, será difícil o imposible, definir todas las vistas de usuario (esquemas externos) o mantener la integridad de la base de datos. También puede ser difícil definir la implementación física o el mantener unas prestaciones aceptables del sistema.

Además, hay que tener en cuenta que la capacidad de ajustarse a futuros cambios es un sello que identifica a los buenos diseños de bases de datos.

El **diseño físico** es el proceso de producir la descripción de la implementación de la base de datos en memoria secundaria: estructuras de almacenamiento y métodos de acceso que garanticen un acceso eficiente a los datos.

Para llevar a cabo esta etapa, **se debe haber decidido cuál es el SGBD** que se va a utilizar, ya que el esquema físico se adapta a él.

Entre el diseño físico y el diseño lógico hay una realimentación, ya que algunas de las decisiones que se tomen durante el diseño físico para mejorar las prestaciones, pueden afectar a la estructura del esquema lógico.

En general, el propósito del diseño físico es describir cómo se va a implementar físicamente el esquema lógico obtenido en la fase anterior.

Concretamente, en el **modelo relacional**, esto consiste en:

- Obtener un **conjunto de relaciones (tablas)** y las **restricciones** que se deben cumplir sobre ellas.
- Determinar las **estructuras de almacenamiento** y los métodos de acceso que se van a utilizar para conseguir unas prestaciones óptimas.
- Diseñar el **modelo de seguridad** del sistema.

2. Diagramas E-R (Entidad-Relación)

Un **modelo** es una representación de la realidad que contempla sólo los detalles relevantes.

Ejemplo

Consideremos una transacción bancaria tal como un depósito en una cuenta corriente, es decir, el ingreso de efectivo de una persona en su cuenta corriente.

El departamento de contabilidad desea conservar ciertos detalles (número de cuenta, importe del depósito, fecha, número del cajero) e ignorar otros (como la conversación mantenida con el personal de caja durante la transacción, la longitud de la cola, la temperatura ambiental dentro del banco, ...). La realidad involucra un sin número de detalles, pero el departamento de contabilidad considerará la mayoría de ellos irrelevantes para la transacción.

De este modo, un modelo sólo considerará aquellos detalles que este consideren relevantes. Por supuesto, algunos detalles considerados irrelevantes para un usuario o grupo de usuarios pueden ser relevantes para otros.

Ejemplo

La longitud de la cola puede ser interesante para el director del banco en el sentido de contratar a más personas para atender al público. Por tanto, diferentes usuarios o grupos de usuarios pueden tener distintos modelos de la realidad.

Una Base de Datos incorpora un modelo de la realidad. El SGBD gestiona la Base de Datos de modo que cada usuario pueda registrar, acceder y manipular los datos que constituyen su modelo de la realidad. Manipulando los datos, los usuarios pueden obtener la información que les sea útil. Por tanto, los modelos son herramientas poderosas para eliminar los detalles irrelevantes y comprender la realidad de los usuarios individuales.

Para modelar debemos asociar e identificar elementos de la realidad con elementos del modelo. Si esta asociación se hace correctamente, entonces el modelo se puede usar para resolver el problema. De lo contrario, el modelo probablemente conducirá a una solución incompleta o incorrecta.

El modelo Entidad-Relación es un modelo conceptual de datos orientado a entidades. Se basa en una técnica de representación gráfica que incorpora información relativa a los datos y las relaciones existentes entre ellos, para darnos una visión de mundo real, eliminando los detalles irrelevantes.

El modelo Entidad-Relación (ER) fue propuesto por **Peter Chen en 1976** en un artículo muy famoso: "*The Entity-Relationship Model: Toward a Unified View of Data*".

Según Chen: “*El Modelo Entidad-Interrelación puede ser usado como una base para una vista unificada de los datos*”, adoptando “*el enfoque más natural del mundo real que consiste en entidades y relaciones (interrelaciones)*”.

Posteriormente, otros autores han ampliado el modelo (modelo Entidad-Relación extendido), con importantes aportaciones, formándose en realidad una familia de modelos.

Este tema describe el Modelo Entidad-Relación, sin discriminar de manera detallada los elementos originales y los extendidos. El objetivo es disponer de un buen modelo para representar datos de cara a diseñar bases de datos.

Características del modelo

- Refleja tan solo la existencia de los datos, **no lo que se hace con ellos**.
- Se incluyen todos los datos relevantes del sistema en estudio.
- No está orientado a aplicaciones específicas.
- Es independiente de los SGBD.
- No tiene en cuenta restricciones de espacio, almacenamiento, ni tiempo de ejecución.
- Está **abierto a la evolución** del sistema.
- Es el modelo conceptual más utilizado.

Elementos del modelo

Los elementos básicos del modelo E-R original son:

- **ENTIDAD** (entity) El más importante
- **ATRIBUTO** (attribute) Qué hace esa tabla (Ejemplo: Persona - Edad, nombre...)
- **DOMINIO** (domain) Tipo de dato asociado a cada atributo
- **RELACIÓN** (relationship) La relación entre entidades

A lo largo de este tema describiremos esos elementos básicos.

2.1 Entidades

Entidad es cualquier **objeto** (real o abstracto) que existe en la realidad y acerca del cual queremos almacenar información en la BD. Están formadas por datos comunes a ella (**atributos**) que tienen un sentido particular.

Por ejemplo: Una entidad sería “Persona”, que es la entidad en la que representaremos a todas las personas almacenadas en nuestra base de datos, mientras que cada persona: Silvia, Agustin, Robert... son instancias de la entidad “Persona”.

Estructura genérica → Entidad Cada uno de los elementos → Instancias

Una entidad se representa mediante un **rectángulo** donde aparece el nombre correspondiente para su identificación. El nombre se muestra **en singular sin abreviaturas**. Debe representar un tipo o clase de elemento, no una instancia.

Reglas para definir una entidad (importante)

- UNA INSTANCIA SOLO PUEDE SER REPRESENTADA POR UNA ENTIDAD.
- CADA ENTIDAD DEBE ESTAR IDENTIFICADA DE FORMA ÚNICA. Se debe identificar claramente de otras entidades (es decir, evitar que existan varias entidades que representen lo mismo o datos muy similares de otras).

La instancia sería como la "persona única" dentro de la tabla "personas"

Ejemplos de entidades:

Entidad: PERSON

PERSON

Instancia1: 'Alvaro' Instancia2: 'María' Instancia3: 'Carlos' ...

Entidad: SUBJECT

SUBJECT

Instancia1: 'Maths' Instancia2: 'Databases' ...

¿Se te ocurren más ejemplos de entidades?

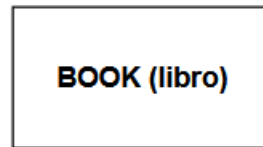
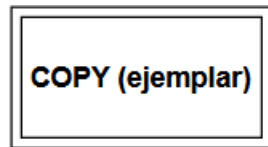
RECOMENDACIÓN: La entidad siempre en singular

Entidades Fuertes y Débiles

Existen dos tipos de entidades:

- **Fuertes:** existen por sí mismas y no dependen de otras entidades.
- **Débiles:** dependen para identificarse de otras entidades fuertes.
Se representa con un doble rectaculo

Por ejemplo: EJEMPLAR (entidad débil) y LIBRO (entidad fuerte)



Un libro que pretas, no sabes el estado. Creas el libro como base, y las copias para controlar su estado y cuantas copias tienes, así sabes cual dejas y cual te quedas y su estado de salida/entrada

Otro Ejemplo:

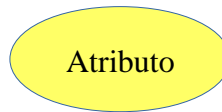
Paciente - Tratamiento (No hay tratamiento si no hay paciente)

PAIS - CIUDAD (Ejemplo: Cartagena es de Alicante y de Argentina, ergo hay que tener algo que las diferencie en este caso, el país)

2.2 Atributos

Los **atributos** son cada una de las propiedades o características que tiene una entidad.

Los atributos se representan mediante un óvalo con el nombre del atributo dentro.

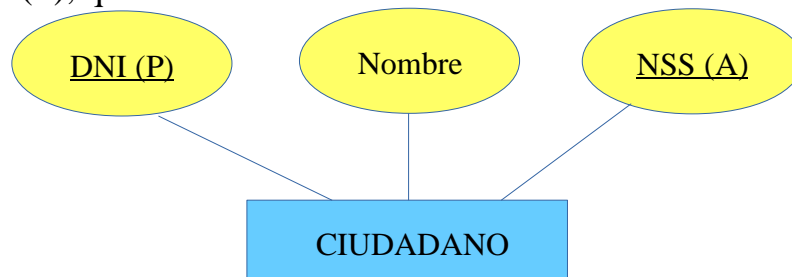


Pueden clasificarse según:

- **Identificadores:** son atributos que identifican de manera unívoca cada ocurrencia de una entidad. Toda entidad debe tener al menos un atributo identificador. Los atributos se representan subrayando el nombre del atributo.



- **Identificador primario e identificadores alternativos:** Una entidad puede tener más de un atributo identificador; en ese caso, elegimos un atributo como primario (P), quedando el resto como identificadores alternativos (A).

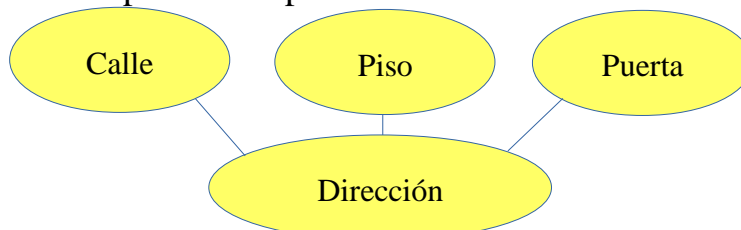


- **Simple y compuestos**

Simple: son atributos que no están formados por otros atributos.



Compuestos: son atributos que están formados por otros atributos que a su vez pueden ser simples o compuestos.



- **Monovaluados y multivaluados**

Monovaluados: son atributos que representan un solo valor para una determinada ocurrencia de una entidad en un momento determinado. Pueden ser simples o compuestos.



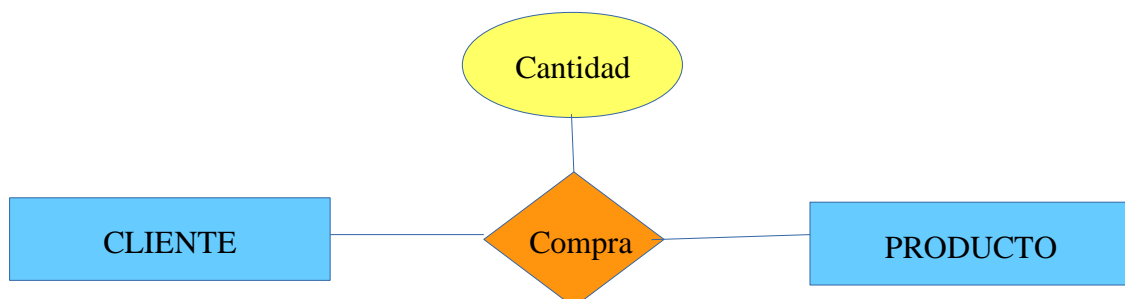
Multivaluados: son atributos que pueden representar varios valores simultáneamente para una misma ocurrencia de una entidad. Se representan mediante un doble óvalo.



- **Calculados (o derivados)**: son atributos cuyo valor se obtiene aplicando una fórmula (normalmente a partir del valor de otros atributos). Son atributos que NO se almacenarán como tal en la base de datos, sino que su columna tendrá asociada una fórmula para calcularlos en tiempo real. Por ejemplo, la edad de una persona se calcula a partir de la fecha de nacimiento (Fecha Actual – Fecha de Nacimiento).



- **Atributos de las relaciones**: se representan unidos al rombo de la relación. Son aquellos que sólo tiene sentido incluir en la relación.



2.3 Dominios

Cada atributo simple de una entidad está asociado a un dominio, que **representa el conjunto de valores que puede tomar dicho atributo**.

Ejemplos en una tabla de **EMPLEADOS**:

Atributo	Dominio	Descripción
FECHA DE ALTA	Día del Calendario	Indica cuando se introdujo el registro en la BD
TELEFONO	Conjunto de números	Puedes ser tanto un número fijo como un número móvil, pero será siempre de España, es decir, no es necesario almacenar el prefijo del país porque todos tendrán el +34
COBRO DE INCENTIVOS	SI/NO	Indica si esta persona cobra incentivos
ALTURA	0 - 300	Indica los cm de altura de la persona

Los dominios se especificarán en el diccionario de datos.

Es necesario también especificar el tipo, formato, la unidad o los valores correspondientes. El formato se especificará acorde a la siguiente notación de expresiones regulares:

Atributo	Tipo	Formato	Unidad	Valores o fórmulas
DNI	Cadena(9)	[0-9] ₈ + [A-Z] ₁		
NOMBRE	Cadena(30)	[A-Za-z] _{1,30}		
APELLIDO	Cadena(40)	[A-Za-z] _{1,40}		
PESO	Número	[0-9] _{1,3}	Kg	
ALTURA	Número	[0-9] _{1,3}	cm	
TIPO_TELF	Cadena(5)	[A-Z] _{3,5}		‘FIJO’, ‘MÓVIL’, ‘FAX’
TELEFONO	Número	[0-9] ₉		
...
EDAD	Número	[0-9] _{1,3}	Años	Fecha hoy – Fecha Nacimiento

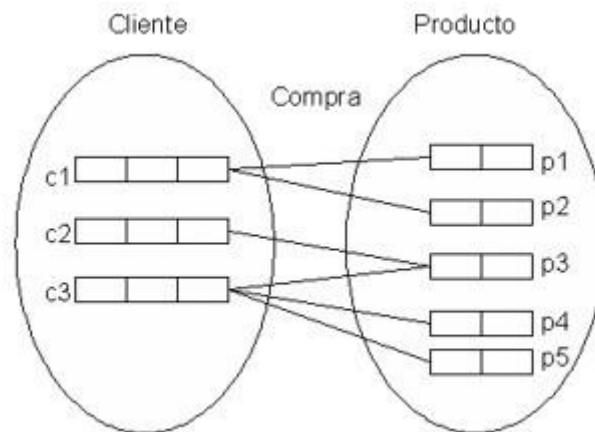
3. Relaciones

Relación (interrelación, vínculo) es una correspondencia o asociación entre 2 o más entidades.

Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior. Normalmente son verbos o formas verbales.



El esquema de tablas y registros podría representarse como sigue:



¡Punto importante para recordar!

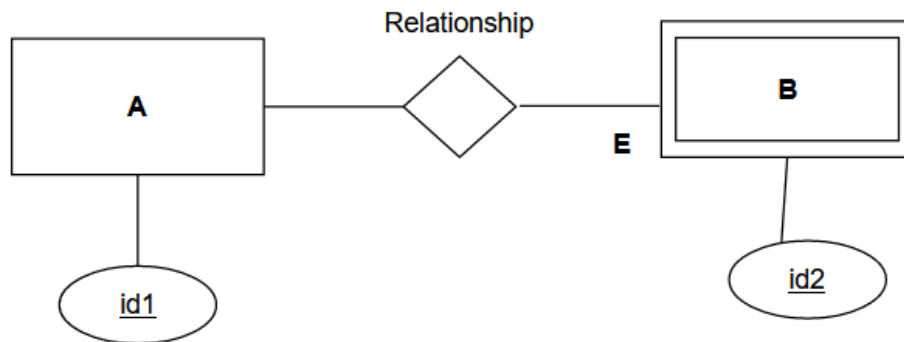
Cualquier entidad debe tener un atributo identificador o CLAVE. Los atributos clave deben aparecer subrayados. Además, esta clave puede estar compuesta por varios atributos, por lo que en ese caso se llama clave compuesta.



3.1 Cardinalidad de las relaciones

La **cardinalidad** de una relación es el número de ocurrencias de una entidad asociadas a una ocurrencia de la otra entidad.

A) **Restricción de cardinalidad mínima:** Son **existenciales**; se utiliza un doble rectángulo para indicar que la entidad es débil y además se pone una **E** en la línea de la relación.

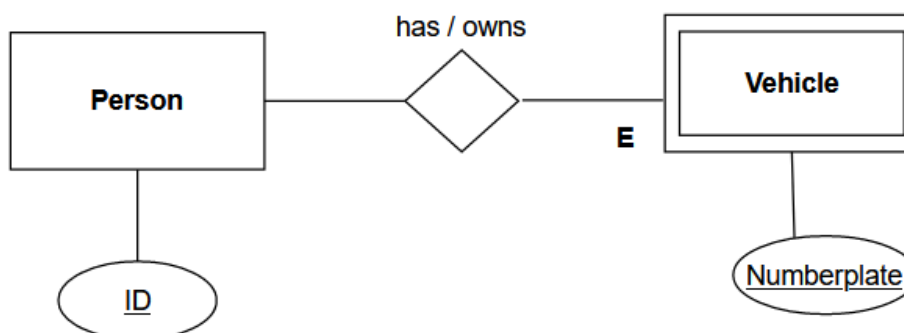


"Cada ocurrencia de B debe interactuar con al menos una de A", por lo tanto, la cardinalidad mínima de A es 1. Para la Entidad A, algunas de sus ocurrencias no pueden relacionarse con ninguna ocurrencia de B, y la cardinalidad mínima de B es 0. Por esta razón, la entidad A no sufre una restricción de existencia.

A Cardinalidad (1, 1)

B Cardinalidad (0, 1)

Ejemplo.



Una persona puede tener 0, 1 o más vehículos y un vehículo debe pertenecer necesariamente a una sola persona. Puede haber personas que no dispongan de vehículo propio, pero un vehículo debe ser siempre de una persona y sólo de una.

B) Restricción de cardinalidad máxima.

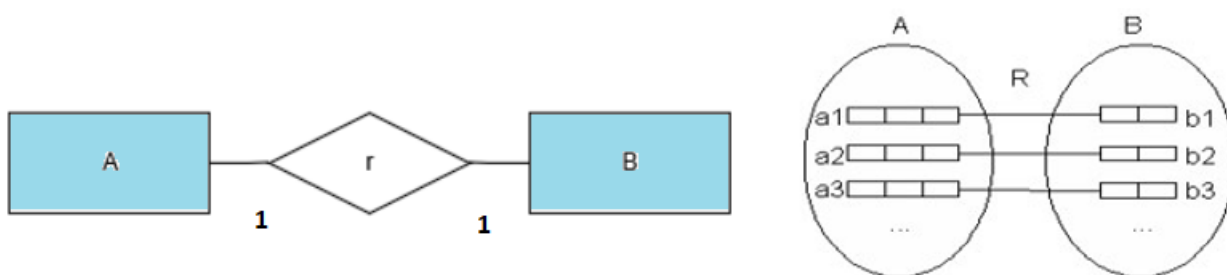
Existen tres tipos: Uno a uno (1:1), Uno a Muchos (1:N), y Muchos a Muchos (N:N).

Para obtener la cardinalidad de una relación: **se debe fijar una ocurrencia particular de una entidad y averiguar cuántas ocurrencias de la otra entidad le corresponden como máximo. Después, haremos lo mismo en la otra dirección.**

Supongamos 2 entidades A y B unidas por la relación R. La cardinalidad se coloca sobre la línea de la relación.

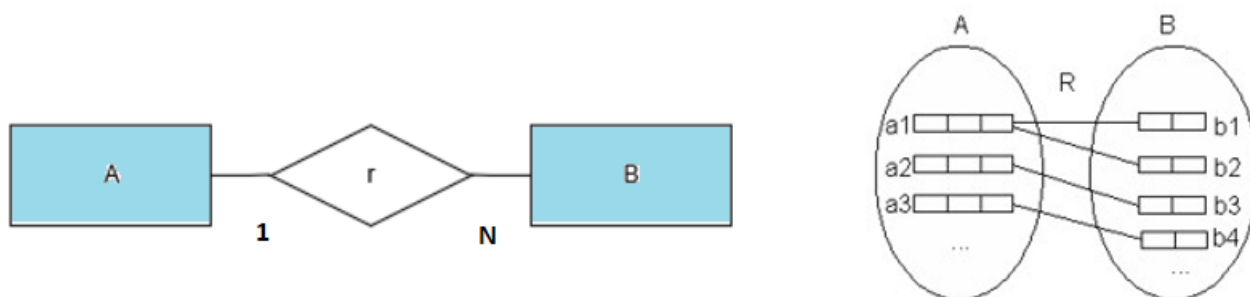
Uno a Uno (1:1)

Cada ocurrencia de la entidad A corresponde a una ocurrencia de B, y viceversa.



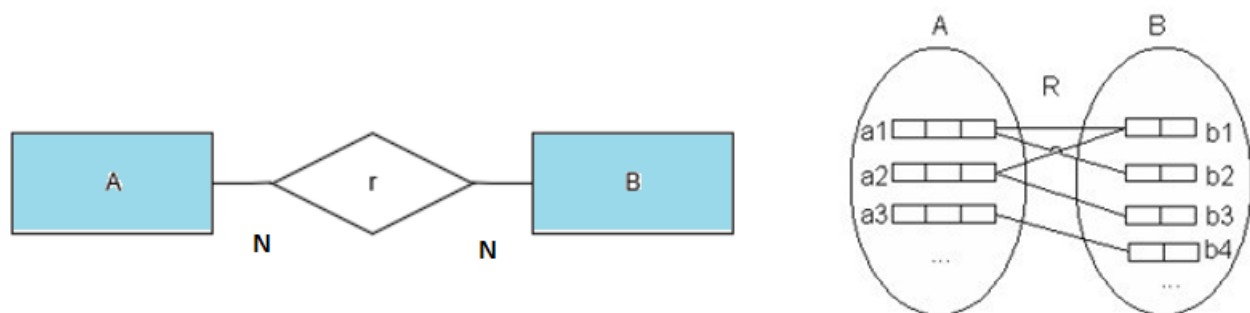
Uno a Muchos (1:N)

Cada ocurrencia de la entidad A puede corresponder a varias ocurrencias de la entidad B, pero cada ocurrencia de B corresponde a una sola ocurrencia de la entidad A.



Muchos a Muchos (N:N)

Cada ocurrencia de la entidad A puede corresponder a varias ocurrencias de la entidad B, y cada ocurrencia de B puede corresponder a varias ocurrencias de la entidad A.



3.2 Participación de las Entidades en las Relaciones

Cada entidad podrá **participar** en la relación con un mínimo y un máximo de ocurrencias.

Para obtener las participaciones fijamos una ocurrencia en una entidad A y calculamos con cuantas ocurrencias de la entidad B se puede relacionar como mínimo y cómo máximo; posteriormente, hacemos lo mismo al revés.

Ejemplo

Un profesor puede ser o no tutor de un grupo de alumnos y un grupo de alumnos tiene necesariamente y exactamente 1 tutor.



Un profesor puede o no aparecer en la relación, mientras que un grupo de alumnos siempre aparecerá.

Ejemplo

Un profesor puede ser o no tutor de un muchos alumnos y un alumno tiene necesariamente y exactamente 1 tutor.



- Participación mínima cero: significa que puede haber ocurrencias de una entidad que no estén asociadas a ninguna ocurrencia de la otra entidad.
- Participación mínima uno: significa que toda ocurrencia de una entidad debe estar asociada a una ocurrencia de la otra entidad.

Ejemplo

Si lo expresamos como lo vemos en el esquema, un cliente puede comprar de 0 a N productos y un producto puede ser comprado desde 0/1 a N clientes.



Para obtener el tipo de correspondencia y las cardinalidades de la relación, se miran los máximos de las participaciones.

Aunque nosotros usaremos la notación de Peter Chen, existen otras notaciones para expresar estas relaciones como se muestra a continuación:

Notación	Uno a Uno	Uno a Muchos	Muchos a Muchos
Ross			
Bachman			
Martin (*)			
Chen			
IDEFX1			

(*) Crow's Foot Notation

3.3 Grado de una relación

Es el número de entidades que participan en la relación.

Las relaciones pueden ser REFLEXIVAS, BINARIAS o TERNARIAS según su grado.

Reflexivas (Grado 1)

Son relaciones donde participa sólo 1 entidad. Se relacionan ocurrencias de la entidad con otras ocurrencias de la propia entidad.

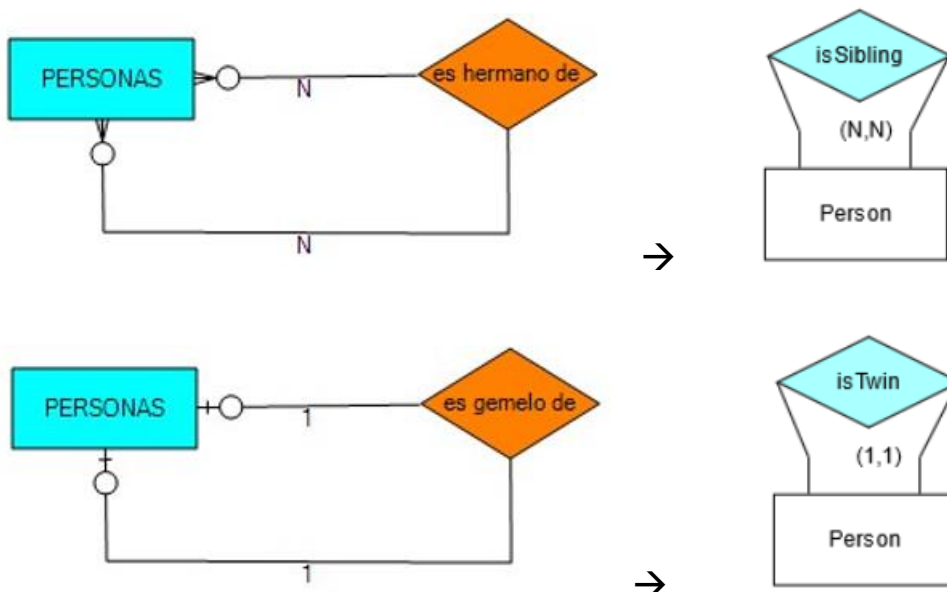
Ejemplo – Reflexiva con jerarquía

En este tipo de relaciones reflexivas se suelen especificar roles. Un rol es el papel que desempeña una ocurrencia de una entidad participante en una relación.



Ejemplo – Reflexiva sin jerarquía

En este tipo de relaciones reflexivas sin jerarquía no es necesario indicar los roles porque la relación es igual en ambos sentidos.



Binarias (Grado 2)

Son relaciones donde participan 2 entidades. **Este tipo de relación es MUY importante, por lo que asegúrate de entenderlas bien.**

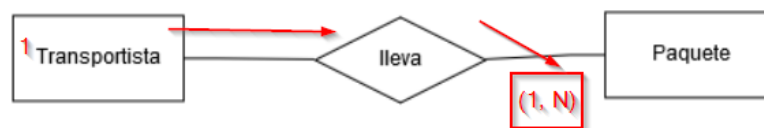


“Supongamos una empresa de transporte que reparte paquetes por toda España. Hay transportistas que se encargan de llevar los paquetes a las diferentes provincias. Un transportista reparte muchos paquetes, y un paquete solo lo puede repartir un transportista”.

Detectamos que **Transportista** (entidad) <entrega> **Paquetes** (entidad). Colocamos la relación con la notación de Chen:



- **Un Transportista (entidad) distribuye (relación), ¿cuántos (Paquetes)?**
Min: 1; Max: N, por lo que lo escribimos en el lado de Paquete



- **Un Paquete (entidad) es distribuido por, ¿cuántos Transportistas (entidad)?**
Min: 1; Max: 1, por lo que lo escribimos al lado de Transportista

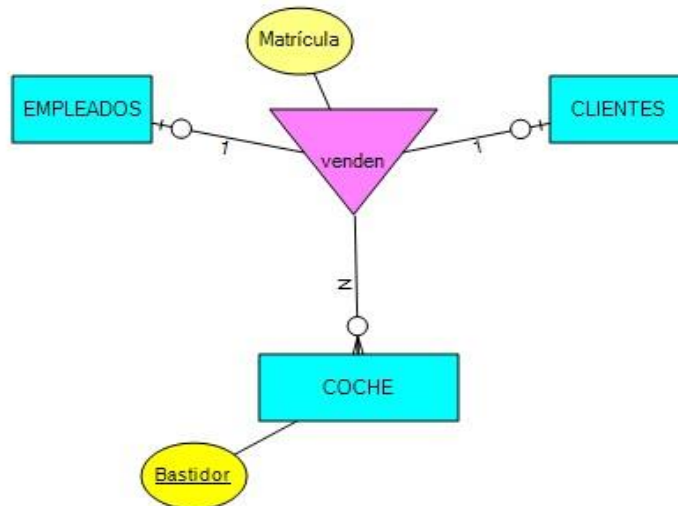


Ternarias (Grado 3)

Son relaciones donde participan 3 entidades. Para calcular las participaciones mínimas y máximas se compara un par de ocurrencias (a,b) de las entidades A y B con una ocurrencia c de la entidad C (y así con las otras 2 combinaciones).

Ejemplo

Empleados de un concesionario que venden coches a clientes.



Para obtener las cardinalidades:

1. UN Empleado VENDE UN Coche a... → ¿Cuántos Clientes? **1**
2. UN Empleado VENDE a UN Cliente... → ¿Cuántos Coches? **N**
3. UN Coche ES VENDIDO a UN CLIENTE por... ¿Cuántos Vendedores? **1**

Este procedimiento es el mismo para cualquier relación ternaria.

3.4 Dependencia entre entidades

Las ocurrencias de una entidad pueden **depender** de una ocurrencia de otra entidad a través de una relación.

Las relaciones pueden ser FUERTES o DÉBILES según su dependencia.

Entidad **fuerte** es aquella que no depende de otra para existan sus ocurrencias o instancias, y además dispone de una clave primaria que identifica cada una de esas instancias.

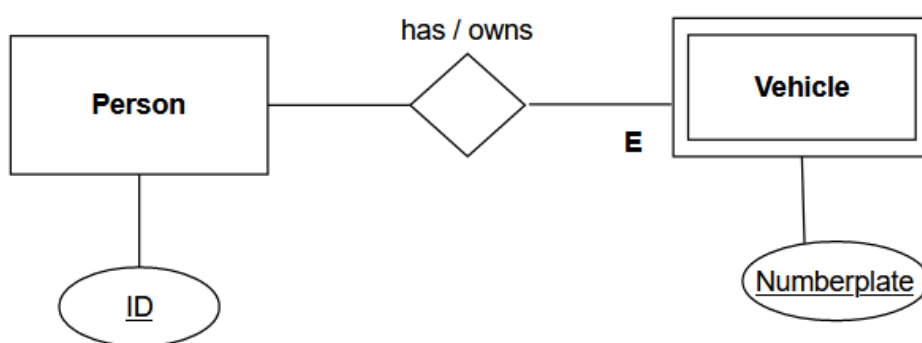
Entidad **débil** es aquella que depende de otra para existan sus ocurrencias o elementos.

La dependencia en las entidades débiles puede ser de existencia o por identificación.

Dependencia de Existencia (E)

Una entidad débil queda definida siempre a través de una relación especial que representa la dependencia de esta entidad de otra de orden superior (que puede ser a su vez una entidad fuerte o débil). Toda entidad débil tiene una dependencia en existencia de la entidad de orden superior, definiéndose entre ellas una jerarquía de dos niveles.

Una instancia de la entidad débil está vinculada a una instancia de la entidad de orden superior, de modo que no puede existir sin ella; es decir para existir la débil, debe existir previamente la de orden superior y si desaparece la instancia de orden superior, entonces deben desaparecer todas las instancias de la entidad débil que están vinculadas.



Una persona puede tener 0, 1 o más vehículos y un vehículo debe pertenecer necesariamente a una sola persona. Puede haber personas que no dispongan de vehículo propio, pero un vehículo debe pertenecer siempre a una persona y sólo a una.

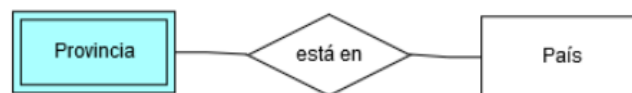
Dependencia de Identificación (ID)

Existen algunas entidades débiles que no tienen suficientes atributos para garantizar la identificación o distinción de sus ocurrencias. En estos casos es necesario forzar el mecanismo de identificación de dicha entidad débil con la composición de atributos primarios de la entidad de orden superior y algunos atributos de la entidad débil. Una dependencia en identificación implica también dependencia en existencia.

La dependencia en identificación se representa mediante una relación débil (rombo con línea doble) y una entidad débil (rectángulo con línea doble).

Ejemplo. “Una Provincia se encuentra en un País”

En este caso, Provincia debería ser una entidad débil dependiente de País, ya que necesita el País para dar sentido a dónde se encuentra la Provincia (no es lo mismo Valencia de España que Valencia de Venezuela).



Si eliminásemos una ocurrencia de la entidad País, todas las Provincias de ese País desaparecerían (dejarían de tener sentido).

4. Paso a paso – Cómo hacer un diagrama E-R

Veamos con un ejemplo los pasos a seguir para obtener un esquema conceptual del modelo Entidad-Relación.

4.1 PASOS PARA DISEÑAR UN DIAGRAMA E-R

1. Hacer una lectura comprensiva del enunciado (hasta entenderlo perfectamente y no tener dudas)
2. Obtener una lista inicial de candidatos a entidades, relaciones y atributos
 - **Identificar las entidades:** suelen ser nombres comunes que son importantes para el desarrollo del problema. Poner las entidades en singular
 - No obsesionarnos con las entidades débiles, si hay dudas, las dejamos como fuertes
 - **Extraer los atributos de cada entidad, identificando aquellos que puedan ser clave.** Se suelen distinguir por ser adjetivos asociados a las entidades detectadas. Si es compuesto se indicará su composición, y si es derivado se indicará cómo se calcula.
 - Es posible que los nombres comunes contengan muy poca información y que no sea posible incluirlas como entidades. En ese caso, intentaremos incorporarla como atributo de otra entidad.
Ej. El autor de un libro puede ser una entidad, pero si solamente vamos a almacenar el nombre de este, podemos incluirlo como atributo dentro de la entidad Libro.
 - **Identificar las relaciones:** se pueden detectar extrayendo los verbos del texto del problema. Las entidades relacionadas serán el sujeto y el predicado de la oración.
 - **Identificar los atributos de cada relación (si los hubiera).** Se detectan por ser aplicables únicamente a la relación.
Ej. Cliente <compra> Producto → “Fecha-Compra” es atributo relación
 - Tras identificar las relaciones, tendríamos que afinar cómo afecta la relación a las entidades implicadas. Es el momento de distinguir las fuertes de las débiles:
 - i. ¿Tiene sentido esta ocurrencia de entidad si quito una ocurrencia de la otra entidad? (*ocurrencia sería una instancia, Ej. Libro → “Harry Potter”*)
SI: las entidades son fuertes
NO: alguna entidad puede ser débil
 - ii. ¿se pueden identificar por sí mismas las ocurrencias de esta entidad?
SI: sólo tendrán una dependencia de existencia
NO: alguna de las dos depende en identificación de la otra
(*Ej. Pasajero – Niño*) (*Ej. Provincia - Municipio*)

3. Averiguar las cardinalidades de las relaciones. Generalmente se extraen del propio enunciado del problema. Si no vienen especificadas, se elegirá la que almacene mayor cantidad de información en la base de datos. Es decir, es salvo que esté muy claro que la cardinalidad mínima sea 0 (Ej. Cliente <compra> Producto) pondremos siempre 1.
4. Representar el diagrama y comprobar en base al enunciado que las decisiones de diseño tomadas son correctas. Si hubiera algo que es susceptible de mejora, refinar el diagrama hasta dejar la versión definitiva del mismo.

4.2 ENTIDADES Y ATRIBUTOS

1) ¿Este dato del enunciado, es una entidad, un atributo o una relación?

He detectado que muchas veces confundís estos tres conceptos.

Será una entidad: cuando el problema indique que hay que almacenar datos de algo.

Será un atributo de entidad: cuando hayamos detectado una entidad y nos den información de ella.

Será un atributo de la relación: cuando el atributo sólo tenga sentido en la relación.

Será una relación: cuando se relacionen dos o más entidades por medio de una acción común.

Ejemplo 1.

“Su base de datos debe ser capaz de gestionar los datos de los diversos **viajes** que se efectúan, **identificándolos con un número de viaje, se desea saber el destino, la fecha de salida y la fecha de vuelta**”.

Viaje debe ser una entidad y además nos está diciendo cosas que hay que guardar de los viajes, por lo que serán los atributos y además nos dice cuál es la clave principal (identificándolos con un número de viaje).

Ejemplo 2.

“Los barcos que efectúan los viajes se conocen por su matrícula, su nombre, el tonelaje, el número máximo de pasajeros, fecha de entrada en servicio y fecha prevista de retirada.”

Barco debe ser una entidad que además está relacionada con la entidad Viaje (la hemos identificado antes). Por lo que quedaría como:



Ejemplo 3.

“Los vehículos son revisados por los mecánicos en una fecha concreta”.

La fecha de la revisión sólo tendría sentido en la relación revisar:



2) Aunque un dato aparezca en un enunciado, ¿cuándo NO sería una entidad?

Será atributo si éste estuviera relacionado con una entidad descrita anteriormente, pero si NO nos pide el problema que almacenemos datos él, no sería necesario incluirlo en el diagrama.

Ejemplo 1.

“Un aficionado al cine quiere mantener una BD de las películas que tiene grabadas en DVD (nada pirata).”

Puesto que no tenemos que almacenar ningún dato del aficionado, Aficionado NO sería entidad.

Ejemplo 2.

“A un concesionario de coches llegan clientes para comprar automóviles. De cada coche interesa saber la matrícula, modelo, marca y color. Un cliente puede comprar varios coches en el concesionario. Cuando un cliente compra un coche, se le hace una ficha en el concesionario con la siguiente información: DNI, nombre, apellidos, dirección y teléfono.”

Puesto que NO nos dice que almacenemos ningún dato del concesionario, Concesionario NO sería entidad.

3) ¿Cuándo es entidad y cuándo es relación?

Como hemos comentado anteriormente, será relación binaria o ternaria cuando dos o más entidades tengan una acción que realicen en común simultáneamente.

Ejemplo 1.

“Cuando un cliente compra un coche, se le hace una ficha en el concesionario con la siguiente información: DNI, nombre, apellidos, dirección y teléfono.”

En este caso, cuando el Cliente (entidad), compra (relación) un Coche (entidad), se le hace una ficha. Esto no quiere decir que la Ficha sea una entidad diferente, sino que la Ficha se refiere a los datos que se van a guardar acerca del cliente en la entidad Cliente, por lo que Ficha no sería entidad.

Ejemplo 2.

“El concesionario dispone de un taller en el que los mecánicos reparan los coches que llevan los clientes”

Tenemos que, **Mecánico** (entidad), **repara** (relación) **Coche** y que los coches son llevados por los clientes.

Que los coches sean llevados por los clientes o por otra persona es indiferente al modelo, lo importante es que los **Coches** (entidad) son **reparados** (relación) por los **Mecánicos** (entidad).

Ejemplo 3.

“Los particulares pueden realizar reservas de las habitaciones...”

“Las agencias de viaje pueden realizar reservas de las habitaciones...”

En este caso tendríamos una relación **BINARIA** entre la entidad **Particular** <reserva> **Habitación** y **Agencia** <reserva> **Habitación**. En ningún caso tendríamos una relación ternaria, puesto que la acción de reservar se realiza de forma independiente por los particulares y por las agencias.

4) ¿Qué hago si detecto una entidad pero el enunciado no me indica ninguna posible clave principal para ella?

Todas las entidades deben tener obligatoriamente un atributo que sea clave principal. Esto quiere decir, que si no nos lo dicen, tendremos que crear nosotros una.

NOTA: Generalmente, utilizar como clave atributos como el nombre no es una buena decisión de diseño, porque la BD pierde mucha eficacia gestionando la tabla. Por ello, suele utilizarse un atributo numérico que se irá incrementando según se inserten registros en la tabla desde 1 hasta “infinito”.

Ejemplo 1.

“De las raquetas de tenis almacenaremos: su nombre y precio”.

Crearemos la clave “**cód-raqueta**”, “**id-raqueta**” o similar

Ejemplo 2.

“De los productos almacenaremos: nombre, descripción, tipo de producto y precio”

Crearemos la clave “**cód-producto**”, “**id-producto**” o similar.

5) ¿Una entidad puede tener varias claves principales?

NO, una entidad sólo puede tener una clave principal. En caso de que podamos identificar los registros de dos modos diferentes, decidiremos cuál de los dos atributos actuará como clave principal (P) y cuál como clave alternativa (A), y así lo indicaremos en el diseño.

Ejemplo.

“Los alumnos se identificarán por su NIA y por un código interno de alumno. Además almacenaremos...”

La clave principal podría ser el “NIA” y la alternativa “cód-alumno”, o viceversa.

6) ¿Puedo añadir cosas que me invente relacionadas con el enunciado pero que no lo diga expresamente?

NO, deberemos ceñirnos a lo que dice el enunciado. A nivel de relaciones y entidades NO podemos añadir nada nuevo, salvo algún atributo en alguna entidad de forma muy justificada.

7) ¿Cómo puedo detectar entidades débiles?

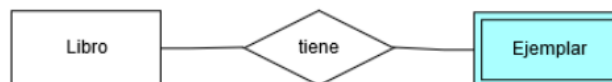
Las entidades débiles son aquellas que deben su existencia a una entidad fuerte, y que, por tanto, no deben existir si la fuerte no existe.

Deberán colocarse en casos justificados, como:

Ejemplo 1.

“Una biblioteca tiene libros identificados por un ISBN. Los ejemplares de los libros están identificados con un código de barras de la biblioteca particular a cada libro y éstos son prestados a los socios”.

Libro sería la entidad fuerte y Ejemplar sería entidad débil de Libro.



Ejemplo 2.

“Un coche está compuesto por cuatro ruedas”.

En este caso, tanto Coche como Rueda serían entidades fuertes. Coche <tiene> Rueda

Puesto que las ruedas existen aunque el coche no exista.

Ejemplo 3.

“Una provincia está en un país”.

En este caso, Provincia sería una entidad débil de País, porque necesita de País para dar sentido a dónde está la Provincia (no es lo mismo Valencia de España que Valencia de Venezuela).



En este caso, tendríamos una **dependencia de identificación**, necesitamos saber el país al que pertenece la provincia para darle sentido.

8) ¿Cómo identifico de qué tipo es un atributo según lo que me indique el enunciado?

1. Atributos simples

Aquellos que únicamente pueden almacenar un valor (dentro del dominio del propio atributo). Se representan con un óvalo sencillo.

Ejemplos. Nombre, DNI, precio, todas las claves principales y alternativas, etc.

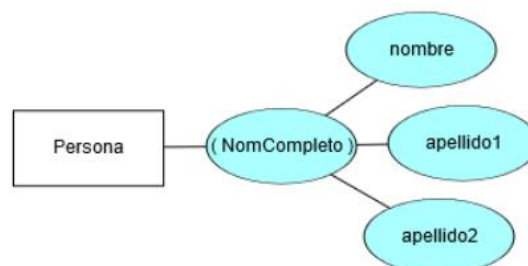
2. Atributos compuestos

Son aquellos que se utilizan para indicar que un atributo se divide en más atributos.

Ejemplos.

NombreCompleto = Nombre + Apellido1 + Apellido2

Dirección = Siglas + Calle + NombreCalle + Número + Puerta

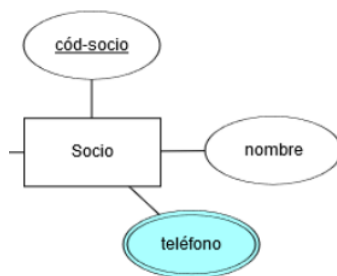


3. Atributos multivaluados

Cuando de un mismo atributo debemos almacenar varios valores. Normalmente suele ponerse el “teléfono” como multivaluado (teléfono fijo de casa, teléfono móvil, teléfono del trabajo).

Se representa de dos formas:

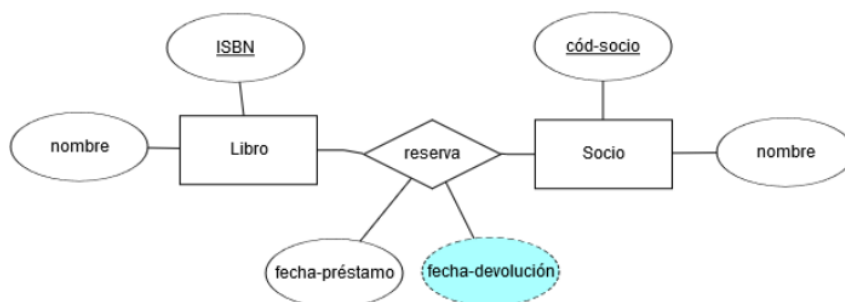
- Con una **n** en el arco que une el atributo a la entidad
- Con un doble óvalo



4. Atributos derivados

Son aquellos cuyo valor deriva de otros atributos. Se representan mediante un óvalo discontinuo.

Ejemplo. Si nos indican que la fecha de devolución de la reserva son 10 días después de realizarla, “fecha-devolución” será un atributo derivado.



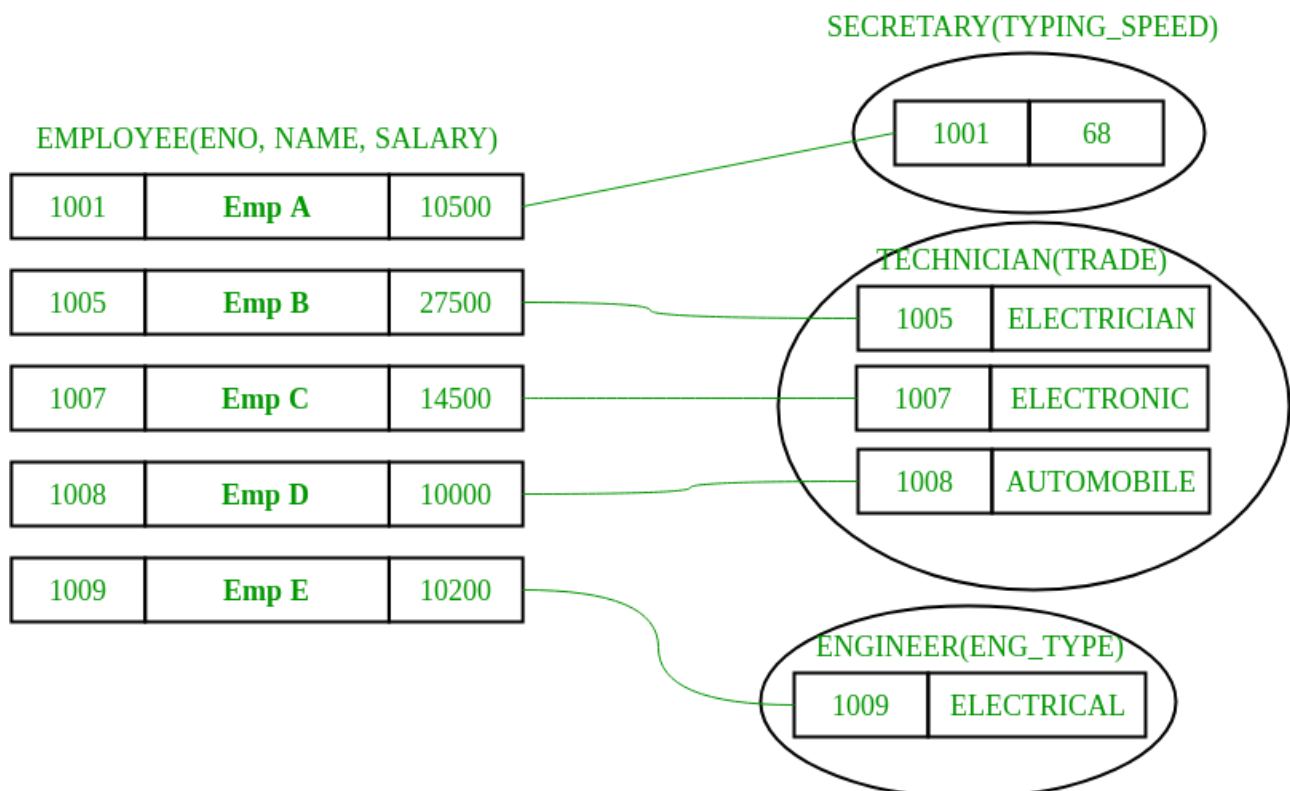
5. Modelo ER Extendido

Además de los elementos que hemos estudiado, existen extensiones del modelo Entidad-Relación que incorporan determinados conceptos o mecanismos de abstracción para facilitar la representación de ciertas estructuras del mundo real.

Estas son relaciones muy comunes que existen en entidades reales. Sin embargo, este tipo de relación se agregó más tarde como una **extensión mejorada del modelo ER clásico**. Las clases **especializadas** a menudo se denominan **subclases**, mientras que una clase **generalizada** se denomina **superclase**, inspirada en la programación orientada a objetos. Una subclase se entiende mejor por "**análisis IS-A**". Es obligatorio para poderla utilizar que las declaraciones tengan sentido "**Técnico ES-UN empleado**", "**Portátil ES-UN Ordenador**".

Una entidad es un tipo/clase especializada de otra entidad. Por ejemplo, un Técnico es una clase especial de Empleado en una universidad y un "Profesor" es un tipo especial de Empleado también. Llamamos a este fenómeno generalización/especialización. En el ejemplo, "Empleado" es una clase de entidad generalizada, mientras que "Técnico" y "Profesor" son clases especializadas de "Empleado".

Ejemplo – Este es un ejemplo de relaciones de "subclase" o "especialización". Aquí tenemos cuatro conjuntos de empleados: Secretario/a, Técnico/a e Ingeniero/a. El empleado es una superclase del resto de tres conjuntos de subclases individuales que son un subconjunto del conjunto de empleados.



Total es que no hay nada más por representar, y Parcial es que aún quedan cosas por declarar. EJEMPLO: Persona (principal), (herencia) Alumno, Profesor ----- Falta Conserje, ergo es parcial.

Restricciones – Hay dos tipos de restricciones en la relación de subclase.

- **Total (T) o Parcial (P)** – Una relación de subclase es total **si cada entidad de superclase se va a asociar con alguna entidad de subclase**; de lo contrario, es parcial.

La subclase "categoría de empleado basada en el tipo de trabajo" es una subclase parcial; no es necesario que cada empleado sea uno de (secretario, ingeniero y técnico) es decir, la unión de estos tres tipos es un subconjunto propio de todos los empleados. Mientras que otras subclases "Empleado asalariado Y Empleado por hora" son totales; la unión de entidades de subclases es igual al conjunto total de empleados, es decir, todo empleado tiene que ser necesariamente uno de ellos.

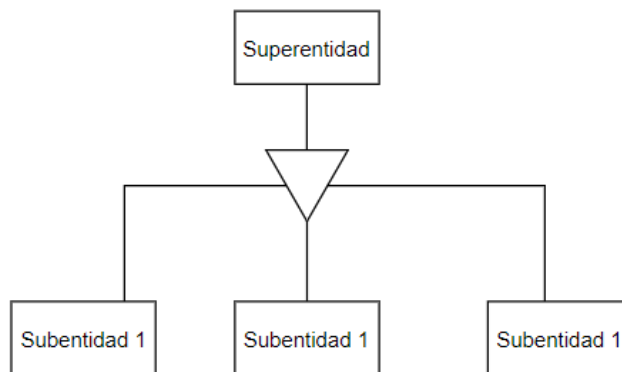
- **Solapada (S) o Disjunta (D)** – Si una entidad de un superconjunto se puede relacionar (puede ocurrir) en múltiples conjuntos de subclases, entonces será solapada; de lo contrario, será disjunta.



Solapado = un alumno puede ser profesor, y un profe puede ser alumno.
Suelen ser disjuntas

<p style="text-align: center;">CASO 1 (P, S)</p> <p>Animal [es un]</p> <ul style="list-style-type: none"> - Mamífero - Gato <ul style="list-style-type: none"> - Parcial: no tenemos representados todos los casos (faltarían los Perros, por ejemplo) - Solapada: un gato también es un mamífero 	<p style="text-align: center;">CASO 2 (T, D)</p> <p>Coche [es un]</p> <ul style="list-style-type: none"> - New-Car - Old-Car <ul style="list-style-type: none"> - Total: hemos representado todos los casos (Km0 estarían dentro de los nuevos) - Disjunta: Los coches nuevos no pueden ser viejos y viceversa.
<p style="text-align: center;">CASO 3 (T, S)</p> <p>Animal [es un]</p> <ul style="list-style-type: none"> - Carnívoro - Herbívoro - Omnívoro <p>Es un caso de:</p> <ul style="list-style-type: none"> - Total: se han representado todos los tipos de alimentación (carne, verduras o ambos) - Solapada: un omnívoro es carnívoro y herbívoro. 	<p style="text-align: center;">CASO 4 (D, S)</p> <p>Persona [es un]</p> <ul style="list-style-type: none"> - Empleado - Afiliado al sindicato <p>Es un caso de:</p> <ul style="list-style-type: none"> - Parcial: no están representados todos ellos, faltarían los no afiliados y los desempleados. - Solapada: un empleado puede estar afiliado y también puede estar desempleado.

¿Cómo las representamos?

Se representan conectando la entidad generalizada por un triángulo invertido que conecta a las entidades especializadas.



<p>Cuando sea TOTAL:</p> <ul style="list-style-type: none">- Un pequeño círculo encima del triángulo 	<p>Cuando sea DISJOINT:</p> <ul style="list-style-type: none">- Un arco debajo del triángulo. 
<p>Cuando sea PARCIAL:</p> <ul style="list-style-type: none">- SIN CIRCULO	<p>Cuando sea SOLAPADA:</p> <ul style="list-style-type: none">- SIN ARCO

Puntos clave para recordar

1) ¿Cuándo utilizamos generalización y cuándo especialización? ¿Son lo mismo?

Son muy similares, pero hay una gran diferencia entre ellas

- **Especialización** debe hacerse cuando pasamos de una superentidad a varias subentidades.
- **Generalización** debe hacerse cuando pasamos de varias subentidades a una superentidad.

Recuerda, sólo podemos utilizar este mecanismo cuando podemos decir que:

Una **super-entidad** [es un/a]

1. Sub-entidad 1

2. Sub-entidad 2
3. Sub-entidad 3
4. Etc.

Ejemplo de especialización.

“Tenemos una serie de vehículos que se ponen a subasta. Según el tipo de vehículo (ciclomotor, coche, tractor, camión, etc.) pagas un determinado impuesto que tiene un coste asociado”.

Un enfoque que podemos tomar es ver la entidad "Vehículo" como la superentidad y que se especializa en varias subentidades ("Ciclomotor", "Coche", "Tractor", "Camión").

Vehículo [es un]

1. Ciclomotor
2. Coche
3. Tractor
4. Camión

Ejemplo de generalización.

“En una empresa tienen contratado personal que realiza las funciones de Administración, Informática y Gestión. Todo el personal debe darse de alta en la empresa al inicio de su jornada laboral y al final de la misma.”

Podríamos plantearlo haciendo una generalización a Empleado, que une los tres tipos de trabajo y Empleado será quien realizará la función de <dar de alta>.

Employee [is a/is a]

1. Administrativo/a
2. Informático/a
3. Gestión

IMPORTANTE, ambos ejemplos podrían haberse hecho con especialización o por generalización. Todo depende de cómo se enfoque el problema.

2) ¿Cómo obtengo si una generalización/especialización es Total/Parcial o Disjunta/Solapada?

Son excluyentes entre sí:

- **TOTAL:** toda ocurrencia en la Entidad Generalizada se encuentra al menos una vez en la especializada (están representados todos los casos posibles).
- **PARCIAL:** puede existir alguna ocurrencia generalizada que no esté asociada a ninguna ocurrencia especializada (NO están representados todos los casos posibles)

Son excluyentes entre sí:

- **DISJUNTA:** Cuando una ocurrencia generalizada puede estar en 1 y sólo 1 especializada (Sólo puede estar asociada a una ocurrencia de uno de los objetos especializados).

Ej. Subentidad1 sólo puede ser Subentidad1, Subentidad2 sólo puede ser Subentidad2, etc.

- **SOLAPADA:** Una ocurrencia de la entidad Generalizada puede estar asociada a una ocurrencia de más de un objeto especializado.

Ej. Subentidad1 puede ser Subentidad1 y además Subentidad2.