

# UD 02 – BASES DE DATOS RELACIONALES

## 1. Fases de análisis y de diseño de bases de datos relacionales. El modelo E/R en el esquema conceptual ubicado en la fase de análisis.

**Modelo de datos:** Conjunto de herramientas conceptuales que permiten describir los datos, sus relaciones, su semántica (significado) y las restricciones aplicables.

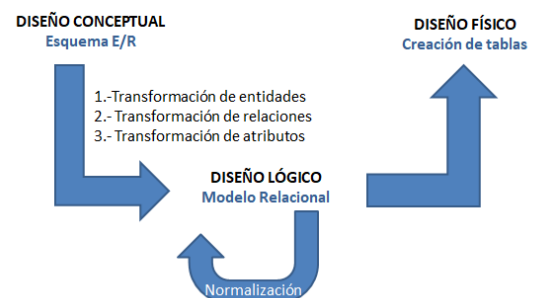
Los SGBD proponen una **arquitectura** que simplifica el modelo de datos a sus usuarios. Se proponen tres niveles de abstracción:

- Interno o físico
- **Lógico o conceptual**
- Externo o de visión de usuario

El **nivel lógico o conceptual** describe la estructura completa de la base de datos a través del **esquema conceptual**: Representa la información de forma independiente al SGBD.

Fase de análisis	Fase de diseño
<b>Análisis de entidades:</b> Localizar y definir entidades y atributos	<b>Diseño de tablas</b>
<b>Análisis de relaciones</b> entre entidades	<b>Normalización</b>
<b>Esquema conceptual</b> a través del <b>modelo E-R</b>	<b>Retrodiseño</b> si se necesita (modificación del esquema conceptual para mantener la equivalencia entre entidades y tablas)
<b>Fusión de vistas:</b> Reunir en un único esquema todos los existentes según las vistas de cada perfil de usuario	Diseño de <b>transacciones</b> : Localizar transacciones que operan en el esquema conceptual
Aplicación de <b>enfoque de datos relacional</b>	Diseño de <b>sendas de acceso</b> : Métodos de acceso a la estructura de datos

¡Suprimir el esquema conceptual puede conllevar pérdida de información respecto al problema real a solucionar! En él deben estar todos los aspectos del mundo real que se va a modelar.



## Modelo E/R en el diseño del esquema conceptual

Peter Chen describe el **modelo Entidad/Relación (ER)** (1976-1977). Con el tiempo este modelo ha sido mejorado al **modelo de Entidad/Relación extendido (EER)**.

El **modelo de Entidad/Relación**:

- Modelo semántico, Representa el significado de los datos

- No está orientado a ningún sistema físico concreto ni tiene ámbito informático puro de aplicación (puede describir procesos de producción, estructuras...)
- Permitiría representar cualquier tipo de sistema y a cualquier nivel de abstracción

## 2. Entidades

**Entidad:** Es un objeto real o abstracto con características que lo diferencia de otros.

(Simil: Instancia)

**Conjunto de entidades:** Entidades con las mismas características o propiedades (Simil: Clase)

Por lo general se usa “entidad” para identificar “conjuntos de entidades”.

A cada elemento del conjunto de entidades (“entidad”) se le llama “ocurrencia de entidad”, “instancia de entidad”

**Las entidades se nombran con sustantivos en singular.**

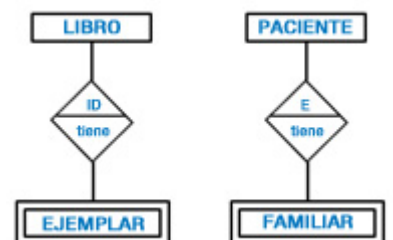
### 2.1. Tipos de entidades: Fuertes y débiles

- **Entidades fuertes o regulares:** Existen por sí mismas, no dependen de otras entidades. Ej.: DOCTOR no depende de la existencia de PACIENTE. Se presenta con el nombre de la entidad (en mayúscula) encerrado dentro de un rectángulo.
- **Entidades débiles:** Su existencia depende de una entidad fuerte con la que se relaciona. Ej.: EDIFICIO Y AULA. El código del aula no la identifica completamente. Necesitamos saber en qué edificio está localizada. Se representa como las entidades fuertes pero rodeadas de un rectángulo doble.



**Dependencias de las entidades débiles:**

- Dependencia en existencia: Si desaparece una entidad fuerte, desaparecen las entidades débiles que dependan de la primera. Se incluye “E” en el interior de la relación.
- Dependencia en identificación. Se da dependencia en existencia y además la entidad débil no puede identificarse por sí misma, debe hacerse mediante la clave de la entidad fuerte asociada. Se incluye “ID” en el interior de la relación.



## 3. Atributos

**Atributo:** Cada propiedad o característica que tiene una entidad o a una relación. Toman valores de uno o varios dominios.

Se representa con el nombre del atributo (en minúscula) rodeado por una elipse y conectándose con la entidad mediante una línea recta.

**Dominio:** Conjunto de valores permitidos para un atributo. Varios atributos pueden definirse dentro del mismo dominio (Ej.: Nombre, apellido primero, apellido segundo definidos en el dominio de cadenas de caracteres de determinada longitud). Deben establecerse límites adecuados para que el SGBD pueda realizar buenas verificaciones y garantice la integridad de los datos.



### 3.1. Clasificaciones de tipos de atributos

#### SEGÚN A QUIEN PERTENECEN:

- **Atributos de entidad:** Pertenecen a una entidad.
- **Atributos de relación:** Pertenecen a una relación.  
Las entidades ALUMNO y ASIGNATURA pueden estar unidas por la relación CURSA. Esta puede tener un atributo nota que indique la nota que un alumno ha obtenido en determinada asignatura.

También sirven para **históricos**. Son relaciones que tienen datos como fecha y hora. Ej.: Al emitir una factura se registra el momento en el que se realiza esa acción. FACTURA EMITIDA CLIENTE. La relación EMITIDA posee el atributo fecha\_emision.

#### OBLIGATORIEDAD

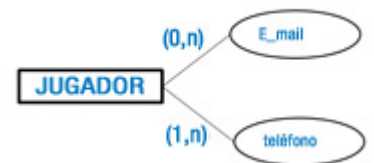
- **Atributo obligatorio:** Siempre debe estar definido. Las claves son atributos obligatorios. Ej.: dni
- **Atributo opcional:** Puede estar definido o no. Ej.: color\_pelo. El opcional tiene la línea con círculo discontinuo.

#### COMPOSICIÓN

- **Atributo simple o atómico:** No puede dividirse en otras partes o atributos. Es un único elemento. No pueden extraerse partes más pequeñas... Ej.: dni
- **Atributo compuesto:** Pueden dividirse en subpartes que serían otros atributos con significado propio. Ej.: dirección que se divide en calle, numero, localidad

#### VALORES

- **Atributo monovaluado:** Solo un valor para cada entidad
- **Atributo multivaluado:** Diferentes valores para cada entidad. Ej.: Un EMPLEADO puede tener varios email. En ese caso debe considerarse la **cardinalidad** (número mínimo y máximo de valores que puede tomar para cada entidad o relación a la que pertenece). Cardinalidad mínima: Valores que deben existir para que la entidad sea válida (Suele ser 0 -puede no tener ningún valor- o 1 -al menos un valor-). Cardinalidad máxima: Cantidad máxima de valores (Suele ser 1 -debe tener como mucho 1- o n -puede tener múltiples valores-). La cardinalidad se indica entre paréntesis (mínima,máxima) sobre la flecha que une el atributo a la entidad.



#### FORMA DE OBTENCIÓN

- **Atributo derivado:** Se puede obtener del valor de otros atributos almacenados. El derivado tiene una D en la flechas de unión o está con círculo discontinuo. Ej.: Edad
- **Atributo almacenado:** Debe almacenarse.

En cualquier caso, los valores de los atributos deben ser tales que permitan identificar unívocamente a la entidad.

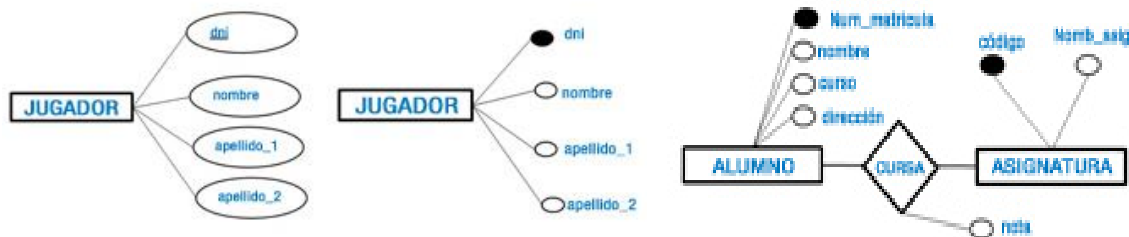
### 3.2. Claves

- **Superclave:** Conjunto de atributos que permite identificar de forma única a una entidad. Puede tener atributos no obligatorios (que por sí solos no identifican una entidad).
- **Clave candidata:** Superclave más pequeña que puede obtenerse en una entidad. (Cuando de una superclave no puede obtenerse otro subconjunto que sea superclave).
- **Clave primaria o principal (Primary Key):** Clave escogida de entre todas las claves candidatas. (Atributo o conjunto de atributos que toman valores únicos y distintos para cada entidad, identificándola unívocamente. No puede tener valores nulos). La clave puede ser simple o compuesta de varios atributos.
- **Claves alternativas:** El resto de claves candidatas.

#### Representación

Si se usan elipses los atributos que forman la clave primaria aparecen subrayados. (Y si hay claves alternativas se podrían distinguir con IP -identificador principal-)

Si se usan círculos para representar los atributos, los atributos que forman la clave primaria tienen un círculo negro. Las claves alternativas tienen un hemicírculo negro.



## 4. Relaciones

La **relación** es una asociación entre diferentes entidades. No pueden aparecer dos veces relacionada en las mismas entidades.

Se representa con un rombo en cuyo interior está inscrito el nombre de la relación. Está conectado por líneas rectas y puede tener o no punta de flecha según el tipo de relación.



A veces habrá que indicar el papel o rol en las líneas que conectan las entidades con la relación. (Útil en relaciones con entidades reflexivas).

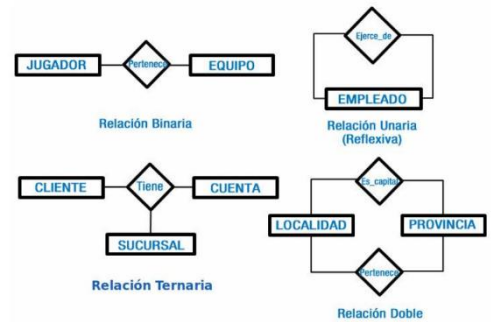
Debemos hablar del:

- Grado de una relación
- Cardinalidad de entidades
- Cardinalidad de relaciones

## 4.1. Grado de una relación

Número de entidades que participan en una relación. Puede ser:

- **Relación unaria** (Grado 1) (**Reflexivas o recursivas**): 1 entidad
- **Relación binaria** (Grado 2): 2 entidades. En general se busca tener esto en el esquema conceptual.
- **Relación ternaria** (Grado 3): 3 entidades.
- **Relación N-aria** (grado n): n entidades. No son usuales. Deben ser simplificadas a menor grado
- **Relación doble**: Dos entidades están relacionadas a través de dos relaciones. Son complejas de manejar.



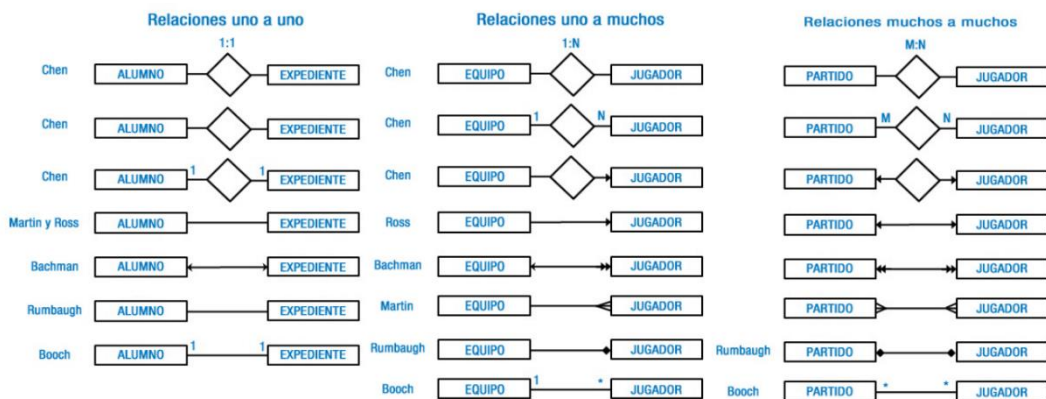
## 4.2. Cardinalidad de relaciones

**Cardinalidad de relación:** Número de entidades que intervienen en una relación. (Número máximo de ocurrencias de cada entidad que pueden intervenir en una ocurrencia de relación.)

- **Uno a uno (1:1):** Un A se relaciona solo con un B y viceversa. Ejemplo: ALUMNO posee EXPEDIENTE
- **Uno a muchos (1:N):** Un A se relaciona con muchos B. Pero un B solo se relaciona con un A. Ejemplo: DOCENTE imparte muchas ASIGNATURA. ASIGNATURA solo tiene un DOCENTE.
- **Muchos a uno (N:1):** Un A se relaciona con un B. Un B se relaciona con muchos A. Ejemplo: JUGADOR pertenece a un EQUIPO. Pero EQUIPO puede tener muchos JUGADOR.
- **Muchos a muchos (M:N):** Un A tiene muchos B y viceversa. Ejemplo: ALUMNO matriculado en muchas ASIGNATURA. ASIGNATURA tiene muchos ALUMNO.

La representación:

Nota: Cuando se representa con número, este se pone en el lado de la entidad a la que se está mencionando. (JUGADOR N:1 EQUIPO) . Corresponde a las cardinalidades máximas de las entidades (dadas la vuelta). Para representarla en el gráfico se pone, bajo el rombo



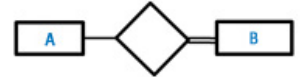
“cardinalidad\_maxima\_izquierda”: “cardinalidad\_maxima\_derecha”. Hay muchas más en verdad:

### 4.3. Cardinalidad de entidades

Número de relaciones en las que una entidad puede aparecer.

La participación de una entidad puede ser:

- Obligatoria (total): Cada ocurrencia necesita como mínimo de una ocurrencia de B. Se representaría con doble línea.
- Opcional (parcial): No es necesaria una ocurrencia de B.



Se representa con el número mínimo y máximo de correspondencias en las que puede tomar parte cada ejemplar de la entidad, entre paréntesis.  
(cardinalidad\_minima:cardinalidad\_maxima).

Cardinalidad mínima: Número mínimo de asociaciones en las que aparece.  
(Cero(opcional) o uno. Si tiene cardinalidad mínima de más de uno, se indica solo un uno.)

Cardinalidad máxima: Número máximo de asociaciones en las que aparece. Puede ser uno, otro valor concreto mayor que uno (3,4,5,...) o muchos (n).

### Ejemplos

Entidades fuertes JUGADOR y EQUIPO.

JUGADOR pertenece como mínimo a ningún equipo y como máximo a un equipo. (0,1).

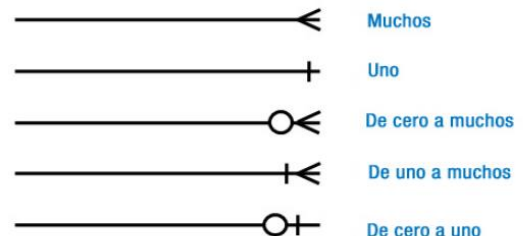
EQUIPO como mínimo tiene un jugador y como máximo muchos (1, n).

**¡Se colocan al revés!**

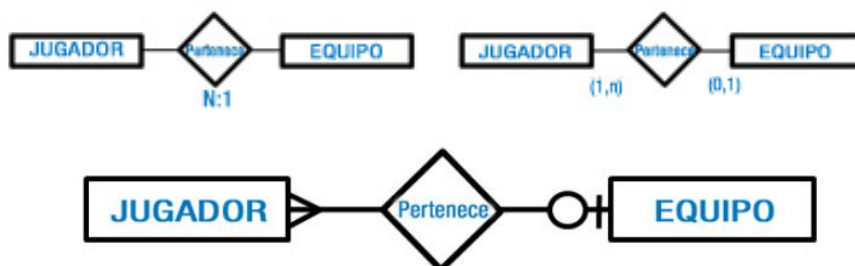
Para la cardinalidad de EQUIPO se coloca la cardinalidad junto a la entidad JUGADOR para expresar que el equipo debe tener mínimo un jugador y máximo varios.

Para la cardinalidad de JUGADOR se coloca la cardinalidad junto a EQUIPO para mostrar que el jugador puede pertenecer a ningún equipo o a uno.

Notación alternativa para representar cardinalidad de entidades



**La cardinalidad máxima de las entidades equivale a la cardinalidad de la relación.**



Supongamos que seguimos diseñando una base de datos para un sitio de juegos online. En un punto del proceso de diseño se ha de modelar el siguiente requisito: cada usuario registrado podrá crear las partidas que desee (a las que otros usuarios pueden unirse), pero una partida solo podrá estar creada por un único usuario. Un usuario podrá o no crear partidas. ¿Cuáles serían las etiquetas del tipo (cardinalidad mínima, cardinalidad máxima) que deberían ponerse junto a las entidades USUARIO

y PARTIDA respectivamente, si éstas están asociadas por la relación CREAR (partida)?  
 $(1,N)$  y  $(0,N)$     $(1,1)$  y  $(1,N)$     $(1,1)$  y  $(0,N)$


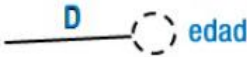


## 5. Simbología del modelo E/R

Notaciones del modelo Entidad/Relación		
	Entidad	
	Entidad Débil	
	Atributo simple o atómico	
	Atributo compuesto	
	Atributo multivaluado	

Notaciones del modelo Entidad/Relación		
	Relación	
	Relación uno a uno	
	Relación uno a muchos	
	Relación muchos a muchos	
	Cardinalidad de entidad	



## Notaciones del modelo Entidad/Relación

	Atributo Derivado	
	Atributo opcional	
	Clave primaria	
	Clave alternativa	

## 6. El modelo E/R extendido

Algunos requisitos tienen dificultad para ser representados con la simbología tradicional del modelo E/R.

En el modelo E/R extendido se incorporan nuevas extensiones para circunstancias especiales en elementos de difícil representación como relaciones con cardinalidad N:M o no identificación clara de entidades.

### 6.1 Restricciones en las relaciones

**Restricción de exclusividad:** O se produce una relación u otra pero nunca ambas, ni a la vez, ni por separado. Una entidad participa en dos o más relaciones y cada ocurrencia de entidad solo puede pertenecer a una de las relaciones únicamente.

**Se usa un arco que engloba a todas aquellas relaciones que son exclusivas.**

Ej.: EMPLEADOS o son diseñadores (DISEÑAN) o son operarios (FABRICAN) PRODUCTOS. Pero no pueden diseñar y fabricar, ni a la vez, ni por separado.

**Restricción de exclusión:** Las ocurrencias de las entidades solo pueden asociarse usando una única relación. O se produce una u otra pero nunca ambas simultáneamente. La entidad puede actuar en todas las relaciones pero siempre por separado.

**Se representa con línea discontinua entre las relaciones**

Ej: Un PROFESOR no puede RECIBIR e IMPARTIR el mismo curso. Pero si imparte un curso, puede estar recibiendo uno.

**Restricción de inclusividad:** Para que dos ocurrencias de entidad se asocien a través de una relación, deben haberlo estado antes a través de otra relación.

Se indica la cardinalidad mínima y máxima de la restricción.



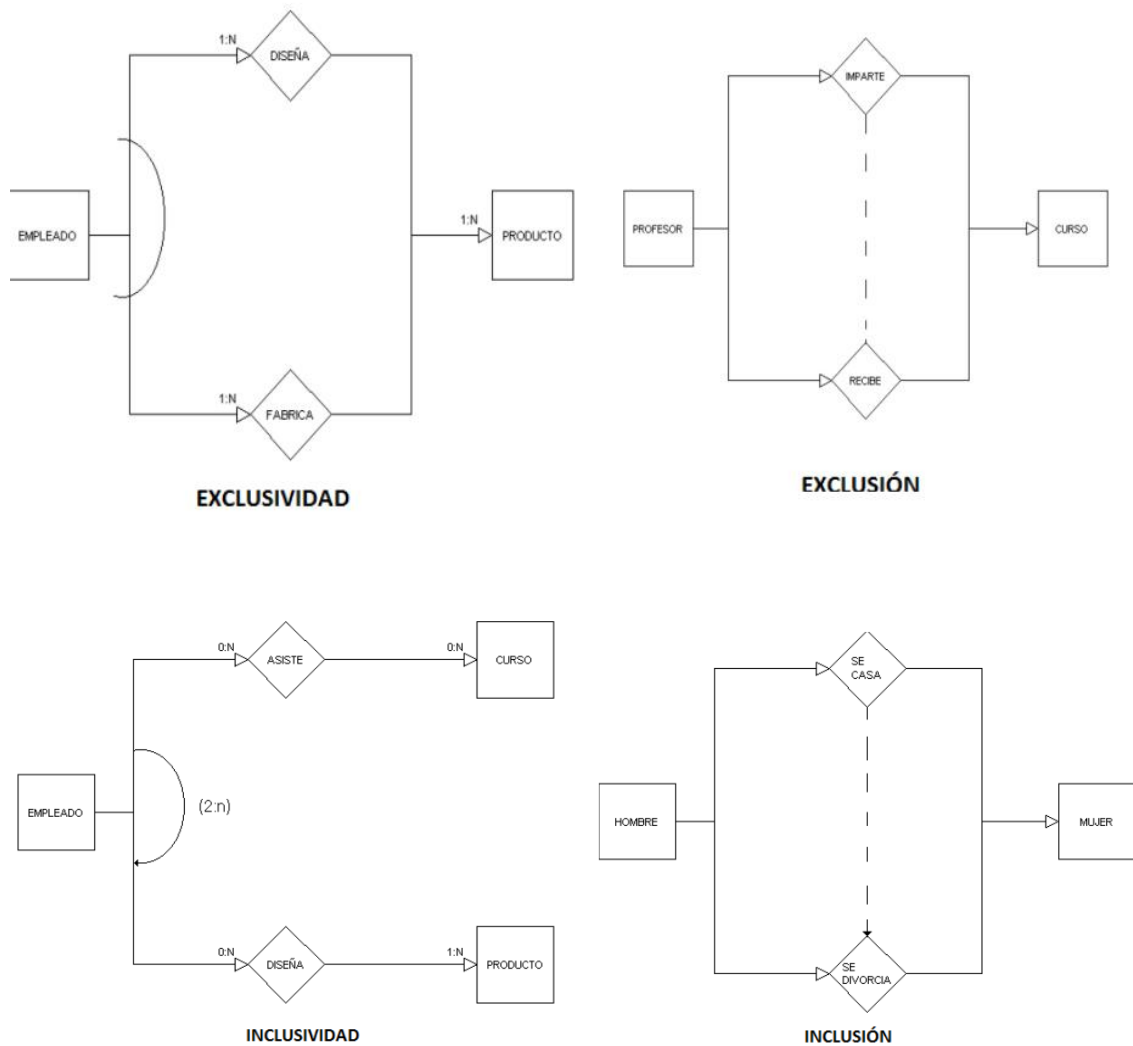
Se representa con un **arco acabado en flecha que parte de la relación que debe cumplirse primero hacia la otra relación.**

Ejemplo.: Para que EMPLEADO pueda DISEÑAR debe haber ASISTIDO a dos cursos. (2,n)

**Restricción de inclusión:** Para participar en la asociación con otro elemento de entidad mediante interrelación, ambos deben estar asociados por una interrelación por naturaleza.

**Se usa una flecha discontinua entre las dos relaciones.**

Ejemplo.: Para que un PERSONA se DIVORCIE de PERSONA debe haber una interrelación previa, haberse casado con ella.



## 6.2. Generalización y especialización

Se basan en el uso de **JERARQUÍAS**. Al modelar la base de datos podemos encontrarnos con entidades que tienen características comunes. Puede crearse un tipo de entidad de nivel más alto que englobe esas características y dividirse la entidad en diferentes subgrupos de entidades por sus características diferenciadoras.

Entidades de nivel superior → Subclase, supertipo

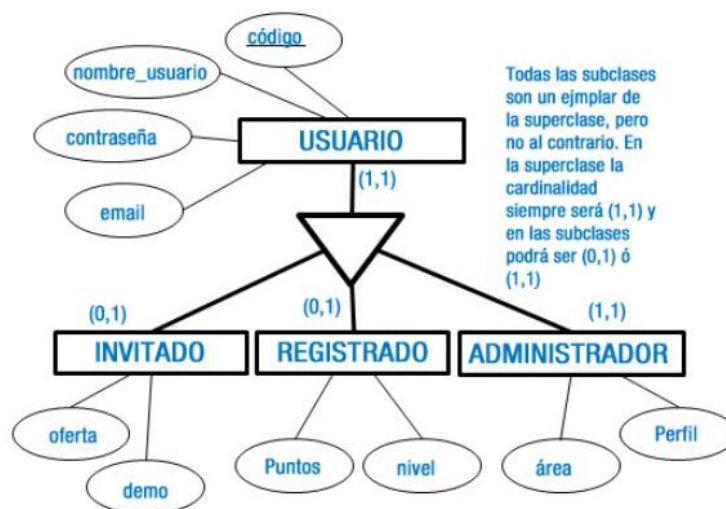
Entidades de nivel inferior → Subclase, subtipo

Se puede hacer **especialización de superclase en subclases** y **generalización de subclases en superclases**.

Las jerarquías se caracterizan por la **herencia**. Los atributos de la superclase son heredados por la subclases. Si la superclase intervinene en una relación, las subclases también lo harán.

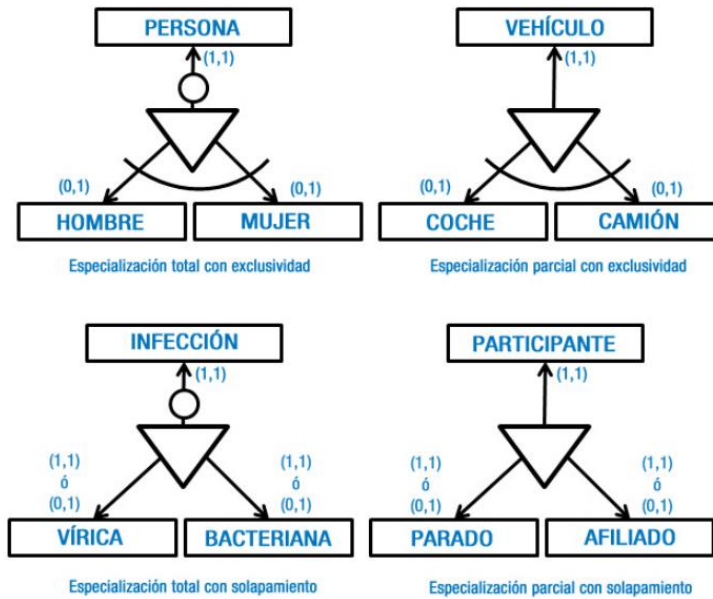
Se representan las relaciones entre superclase y subtipos con “ES UN” “IS A” (un tipo de). La jerarquía se representa mediante un triángulo invertido sobre el que queda la superclase y, conectadas a él a través de líneas rectas, las subclases.

Ejemplo: Superclase USUARIO. Subclases INVITADO, REGISTRADO, ADMINISTRADOR. Con sus propias características y heredan las de la superclase.



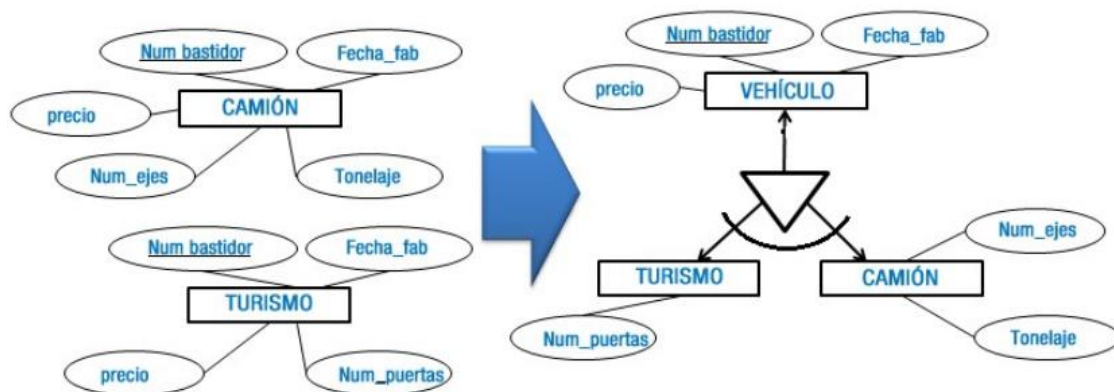
La generalización/especialización puede tener las siguientes restricciones semánticas:

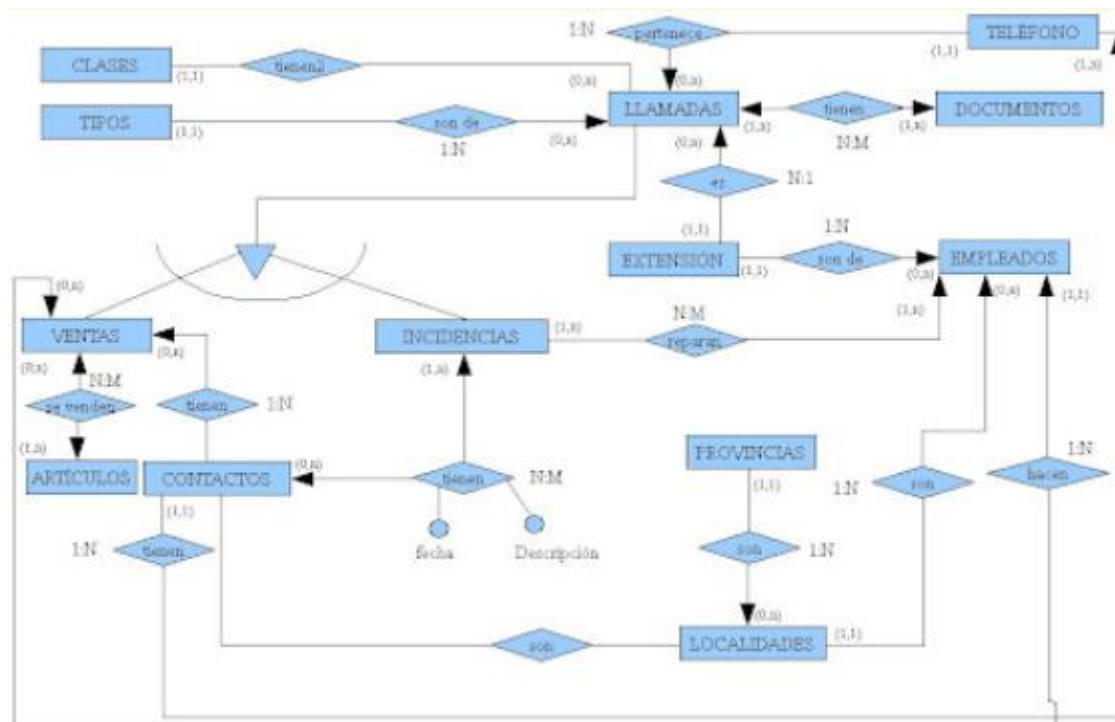
- **TOTALIDAD:** Todo ejemplar de superclase pertenece a alguna de las subclases
- **PARCIALIDAD:** No todos los ejemplares de la superclase pertenecen a alguna de las subclases.
- **SOLAPAMIENTO:** Un ejemplar de la superclase puede pertenecer a más de una subclase.
- **EXCLUSIVIDAD:** Un ejemplar de superclase pertenece a solo una subclase.



	SOLAPAMIENTO	EXCLUSIVIDAD
TOTAL	<b>INFECCIÓN</b> (VIRICA / BACTERIANA) (debe ser de uno de ellos y podría ser de los dos)	<b>PERSONA</b> (HOMBRE / MUJER) (debe ser uno de ellos y solo uno)
PARCIAL	<b>PARTICIPANTE</b> (PARADO / AFILIADO) (puede ser otro y podría ser los dos a la vez)	<b>VEHICULO</b> (COCHE / CAMION) (puede ser otro, pero no los dos a la vez)

Veamos como las entidades CAMION y TURISMO pasan a ser VEHICULO (CAMION / TURISMO) en una jerarquía parcial con exclusividad.





### 6.3. Agregación

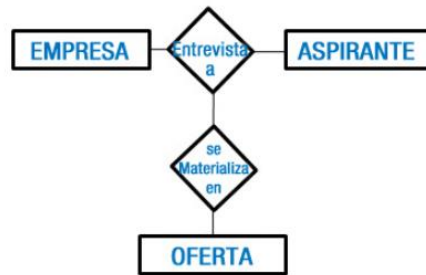
Permite representar las relaciones entre relaciones. Se engloban como un rectángulo las entidades y la relación a abstraer, creándose una nueva entidad agregada que puede participar en otras relaciones con otras entidades. La cardinalidad máxima y mínima de la entidad agregada siempre es (1,1) no indicándose por ello en el esquema.

Es una abstracción en la que las relaciones se tratan como entidades de niveles más alto, usándose para expresar relaciones entre relaciones o entre entidades y relaciones.

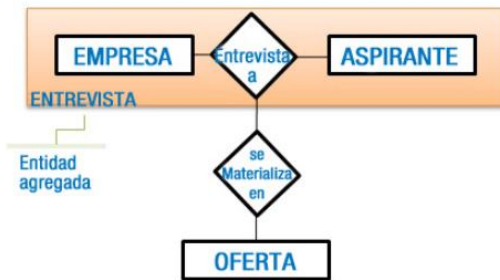
Supongamos un ejemplo en el que hemos de modelar la siguiente situación: una empresa de selección de personal realiza entrevistas a diferentes aspirantes. Puede ser que, de algunas de estas entrevistas a aspirantes, se derive una oferta de empleo, o no. En el siguiente gráfico se representan tres soluciones, las dos primeras erróneas y una tercera correcta, utilizando una agregación.



**Solución 1:** Errónea, ya que estaríamos representando que, por cada entrevista realizada por una empresa a un aspirante, se genera una oferta de empleo



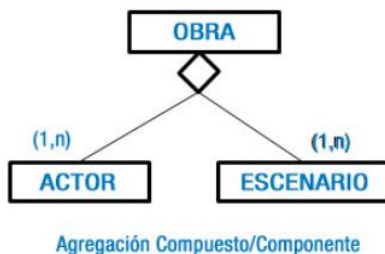
**Solución 2:** Errónea, porque en el modelo E/R no pueden establecerse relaciones entre varias relaciones



**Solución 3:** En el modelo E/R Extendido, puede crearse una entidad agregada llamada ENTREVISTA, compuesta por la relación "Entrevista a" que existe entre EMPRESA y ASPIRANTE. Entre esta nueva entidad y OFERTA si puede establecerse una relación "se materializa en"

Existen dos clases de agregaciones:

- **Compuesto/Componente:** Un todo por la unión de las partes, que pueden ser objetos distintos y desempeñar papeles distintos.
- **Miembro/Colección:** Un todo se obtiene por unión de las partes y desempeñan el mismo papel o rol. Puede incluir una restricción de orden (indicando el atributo de ordenación) entre los miembros.



## 7. Diseño conceptual: Elaboración de diagramas E/R

Primero se parte del documento de especificación de requerimientos (conjunto de requerimientos, requisitos o condiciones que la base de datos debe cumplir), enunciado del problema a modelar que, cuanto más completo sea, mejor.

1. **Identificación de entidades.** Localizar elementos que serán entidades. Buscar nombres o sustantivos sin fijarnos en características o propiedades. También identificar objetos que existen por sí mismos. El número de entidades debe ser manejable y se les otorgarán nombres representativos en mayúsculas y singular. Recomiendan que para que sea entidad debe cumplir:
  - o Existencia propia
  - o Cada ejemplar debe diferenciarse del resto de ejemplares

- Todos los ejemplares deben tener las mismas propiedades
- 2. **Identificación de relaciones.** Buscar verbos o expresiones que conecten unas con otras. Mayormente habrá relaciones binarias aunque puede haber recursivas o unarias también. Se dará nombre representativo en minúscula.
- 3. **Representación de la cardinalidad de entidades y relaciones**
- 4. **Identificación de atributos de entidad y relaciones.** Buscamos nombres de características, propiedades... ¿qué debe tenerse en cuenta en esa entidad o relación?. No todos estarán directamente en el documento de especificación de requerimientos, el diseñador podrá establecerlos aplicando el sentido común o indagando en el problema. Debe tenerse en cuenta si los atributos son simples/compuestos, derivados/calculados o si el atributo se repite en varias entidades. En este caso debe plantearse la posibilidad de una jerarquía de especialización. Debería recopilarse del atributo: Nombre y descripción; Atributos simples que lo componen si es compuesto; Método de cálculo si es derivado; En atributos con cardinalidad uno a muchos, valorar asignar ese atributo o atributos a la entidad con mayor cardinalidad participante en la relación.
- 5. **Identificación de claves.** Establecer una o varias claves candidatas y escoger una de ellas como clave primaria de la entidad que la describan de manera unívoca. La identificación de claves permite determinar la fortaleza (al menos una clave candidata) o la debilidad (ninguna clave candidata) de las entidades encontradas. Se representa la existencia de la clave primaria y las entidades fuertes y débiles.
- 6. **Determinación de jerarquías.** Si hay entidades con características comunes que puedan ser generalizadas o si es necesario expresar en el esquema las particularidades de un tipo de entidad por lo que deben ser especializadas, debe analizarse con detenimiento el documento de especificación y si se identifica alguna jerarquía, representarla adecuadamente.

## 7.1. Metodologías

Partiendo de una versión preliminar del esquema conceptual o diagrama E/R que, tras sucesivos refinamientos, será modificado para llegar al E/R definitivo se podrían tener las siguientes metodologías:

- **Metodología descendente (top-down):** Partir del esquema generar e ir descomponiendo este en niveles, cada uno con mayor número de detalles. Objetos abstractos se refinan paso a paso hasta llegar al esquema final.
- **Metodología ascendente (bottom-up):** Se parte de los atributos. Se agrupan en entidades para crear relaciones entre estas y las posibles jerarquías hasta obtener un diagrama completo. Se parte de objetos atómicos y se obtienen abstracciones de mayor nivel a partir de ahí.
- **Metodologías dentro-fuera (inside-out):** El esquema se desarrolla en una parte del papel y a medida que se analiza la especificación de requerimientos se va completando con entidades y relaciones hasta ocupar todo el documento.
- **Metodología mixta:** En problemas complejos. Requerimientos divididos en subconjuntos que serán analizados independientemente. Se crea un esquema que servirá como estructura en la que se irán interconectando los conceptos importantes con el resultado del análisis de los subconjuntos creados. Se aplica

técnica descendente para dividir los requerimientos y en cada subconjunto se emplea técnica ascendente.

## 7.2. Redundancia en diagramas de E/R

**Redundancia** es la reproducción, reiteración, insistencia.... Almacenamiento de los mismos datos varias veces en varios lugares.

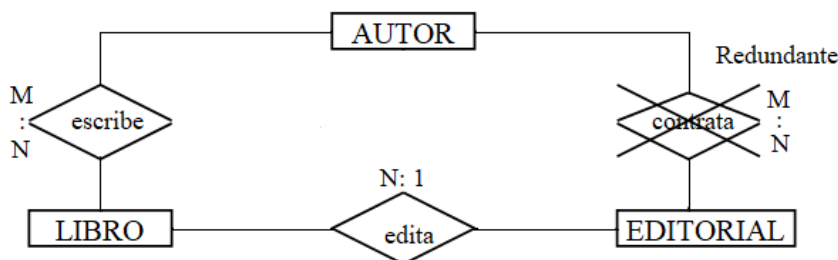
Esto provoca problemas como:

- Aumento de la carga de trabajo. Operaciones de grabación o actualización necesitan hacerse en varias ocasiones
- Gasto extra de espacio de almacenamiento
- Inconsistencia: Los datos repetidos podrían no contener los mismo valores (valor actualizado en un lugar y en otro no).

Es imprescindible el control de la redundancia mediante un buen diseño, analizando el esquema y valorando qué elementos pueden incorporar redundancia.

- Atributos redundantes pueden ser elementos cuyo contenido se calcula en función de otro.
- Existencia de ciclos. Varias entidades unidas de forma circular o cíclica no tienen por qué implicar una redundancia. Para confirmar que puede eliminarse debemos asegurarnos de que
  - o El significado de las relaciones que componen el ciclo es el mismo
  - o Al eliminar la relación redundante el significado del resto de relaciones en el mismo
  - o Si la relación eliminada tiene atributos asociados, estos pueden ser asignados a alguna entidad participante en el esquema sin que se pierda su significado.
- No toda redundancia es perjudicial. Si por ejemplo el método de cálculo es complejo puede ser conveniente definirlo desde el principio en lugar de calcularlo a cada vez. Depende de la elección que haga el diseñador.

Si lo analizamos vemos que si se conocen los libros de un autor y las editoriales que los han editado, se puede deducir qué editoriales ha contratado dicho autor; de forma similar, si sabemos qué libros ha editado una editorial concreta, podemos deducir qué autores han escrito para ella (la han contratado), por lo que la interrelación contrata entre las entidades AUTOR y EDITORIAL es redundante. Ahora bien, si la relación considerada como redundante presentara cardinalidad M:N y tuviese atributos propios (por ejemplo fecha de contratación, en nuestro caso) no podría ser eliminada.





### 7.3. Propiedades deseables en diagrama E/R

Las propiedades deseables serían:

- **Compleitud:** Cada representación del diagrama tiene un equivalente en los requerimientos y viceversa.
- **Corrección:** Emplea todos los elementos del modelo E-R.
  - o **Corrección sintáctica:** No hay representaciones erróneas
  - o **Corrección semántica (SIGNIFICAR):** La representación significa exactamente lo que se estipula en los requerimientos. Posibles errores: Usos de atributo en lugar de entidad, uso de entidad en lugar de relación, uso de mismo identificador para dos entidades o relaciones, indicar erróneamente alguna cardinalidad u omitirla...
- **Minimalidad:** Evitar que sea redundante. ¿Al eliminar cualquier elemento, se pierde información? -> Es mínimo.
- **Legibilidad:** Fácil de interpretar. Depende de cómo se dispongan los elementos y conexiones (aspectos estéticos).
- **Escalabilidad:** Es capaz de incorporar cambios dados por nuevos requerimientos.

## 8. Comenzando el diseño lógico: Transformación de diagramas E/R al modelo relacional

Cuando el esquema E/R cumple los requerimientos del problema a modelar se transforma al esquema lógico basado en el modelo de datos (modelo de datos relacional) en el que se basa el SGDB con el que se implementarán las tablas.

La transformación tendrá tres fases:

Transformación de las entidades  
Transformación de las relaciones  
Transformación de los atributos

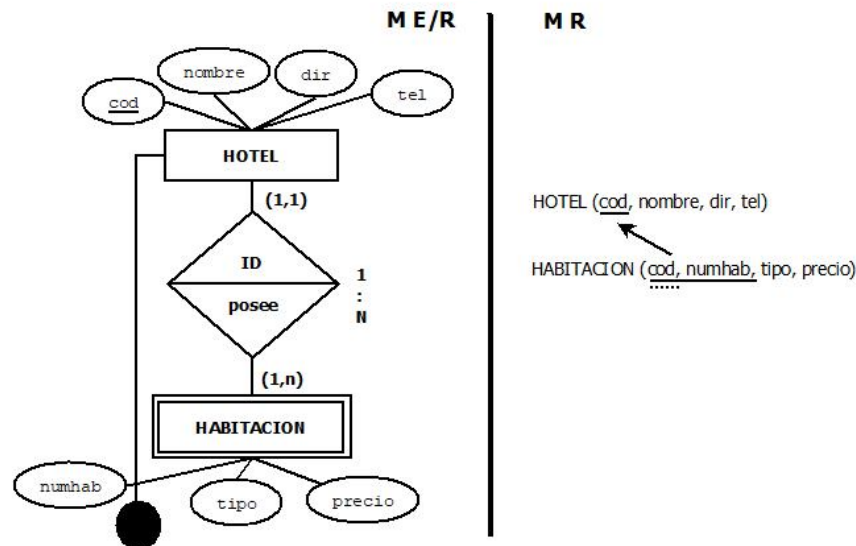
Cada tabla se representa con su nombre y, a continuación, y entre paréntesis, su lista de atributos. La clave principal de cada tabla se representa con subrayado continuo. La clave o claves ajenas con subrayado discontinuo. Las clases alternativas no se representan.

### 8.1. Transformación de entidades

Norma general: Cada entidad se transforma en una tabla nombrándose igual que la entidad de la que proviene.

Casos especiales

**Dependencia en identificación:** Si una entidad débil depende en identificación de una entidad fuerte, la clave primaria de la entidad fuerte se exporta a la tabla de la entidad débil donde formará parte de su clave primaria siendo a la vez una clave ajena que referenciará a la tabla de la entidad fuerte.

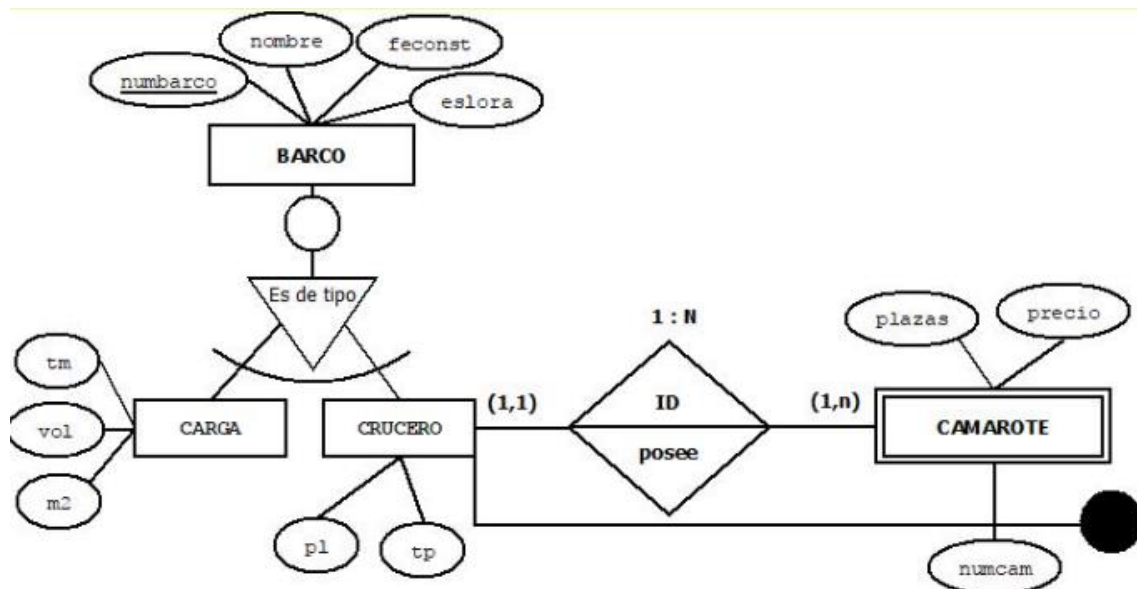
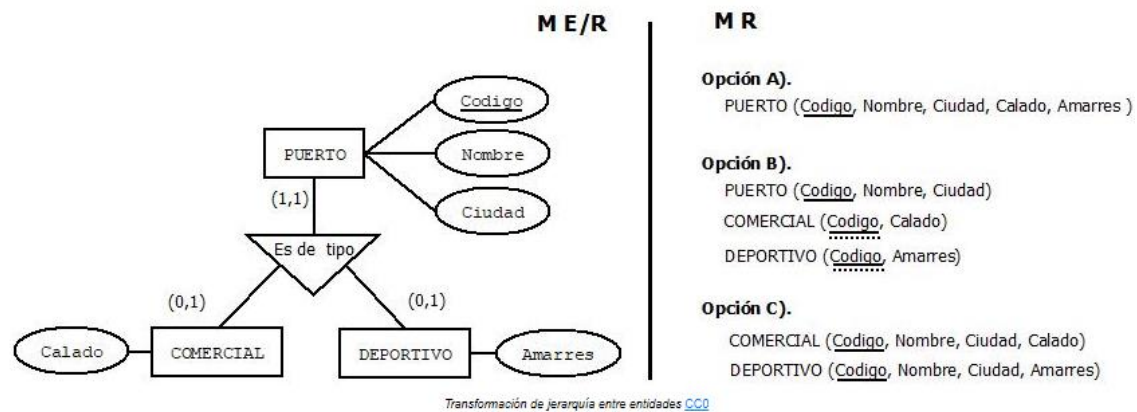


**Generalización y especialización:** Hay varias opciones.

Opción A) **Crear una única tabla para el supertipo y los subtipos que recoja todos los atributos.** Si los subtipos no tienen atributos diferenciadores, añadir un atributo indicando el subtipo al que se refiere. Es la solución más simple pero puede provocar valores nulos en los atributos propios de cada subtipo (Solo usar en casos en los que se diferencien en muy pocos atributos y las relaciones sean las mismas para los subtipos o no existan).

Opción B) **Crear una tabla para el supertipo con los atributos generales y una para cada uno de los subtipos con sus correspondientes atributos específicos.** (Aconsejable cuando los subtipos tienen atributos distintos y/o sus relaciones con otras entidades sean diferentes. Permite mantener agrupados en una tabla los atributos comunes).

Opción C) **Crear una tabla para cada subtipo que además de los atributos específicos contenga los atributos comunes del supertipo.** Suele aplicarse a jerarquías totales con exclusividad en las que se dan subtipos con atributos dispares y relaciones diferentes con otras entidades.



**CARGA** (numbarco, nombre, feconst, eslora, tm, vol, m2)

**CRUCERO** (numbarco, nombre, feconst, eslora, pl, tp)

↑  
**CAMAROTE** (numbarco, numcam, plazas, precio)

**BARCO** (numbarco, nombre, feconst, eslora)

**CARGA** (numbarco, tm, vol, m2)

**PASAJEROS** (numbarco, pl, tp)

**CAMAROTE** (numbarco, numcam, plazas, precio)

## 8.2. Transformación de relaciones

### Cardinalidad Muchos a muchos N:M

**Siempre se transforma en una tabla nueva.**

Clave primaria compuesta la concatenación de las claves primarias de las entidades que relaciona. Cada una de esas claves será parte de la clave primaria de la nueva tabla y clave foránea que referenciará a la tabla de procedencia donde es clave primaria.

Si la relación contiene atributos, pasan a formar parte de la nueva tabla. Debe comprobarse que la clave primaria compuesta de la nueva tabla es suficiente (quizás deba completarse con algún atributo de la relación, especialmente si es de tipo fecha).

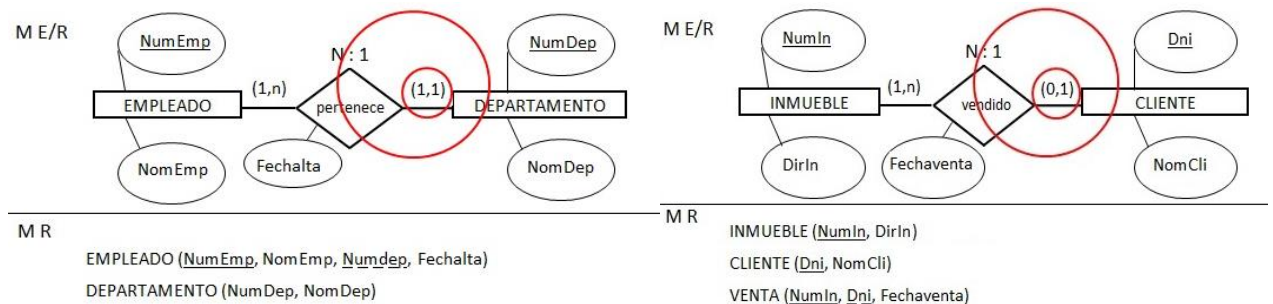
## Cardinalidad 1:N

Dos opciones:

Cardinalidad de la tabla de cardinalidad 1 es (1,1): La clave primaria de la tabla de cardinalidad 1, ponerla en la tabla de cardinalidad N como clave foránea. Los atributos de la relación pasan a la entidad con tipo N.

Cardinalidad de la tabla de cardinalidad 1 es (0,1): Transformar la relación en una nueva tabla. Esta tendrá como clave primaria la clave primaria de la entidad con cardinalidad N y como clave ajena la clave primaria de la entidad con cardinalidad 1. La nueva tabla tendrá los atributos de la relación.

Si en este caso se propagase la clave, habrá algunos NULL en los campos foráneos. Esto TAMBIÉN se hace muchos casos.



## Cardinalidad 1:1

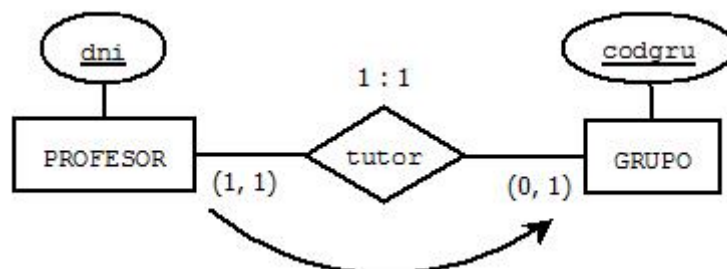
Dos opciones:

Las dos tablas tienen cardinalidad (0,1). Crear una nueva tabla. En ambas tablas habrá ocurrencias no relacionadas con la otra entidad.

Al menos una tiene cardinalidad (1,1). Propagar la clave primaria de una de las entidades como clave ajena de la otra entidad

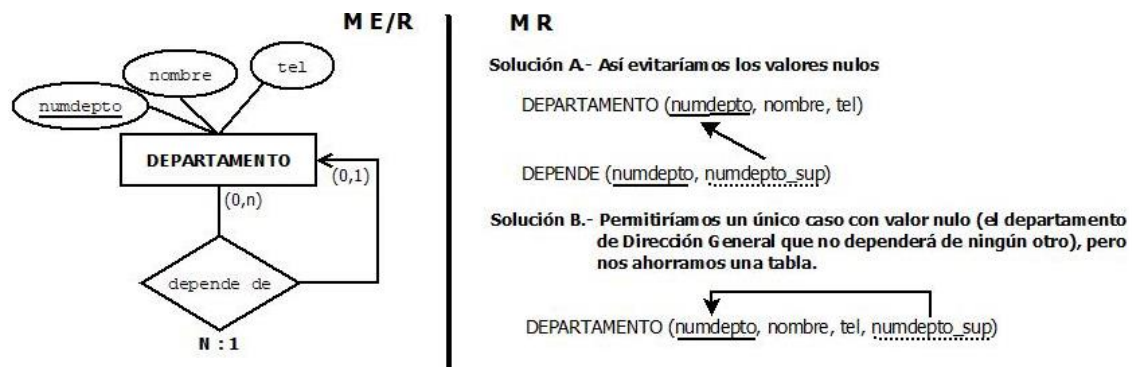
Si es (0,1) y (1,1) se hará de la tabla con (1,1) a la que tiene (0,1).

Si las dos tienen (1,1) entonces da igual.



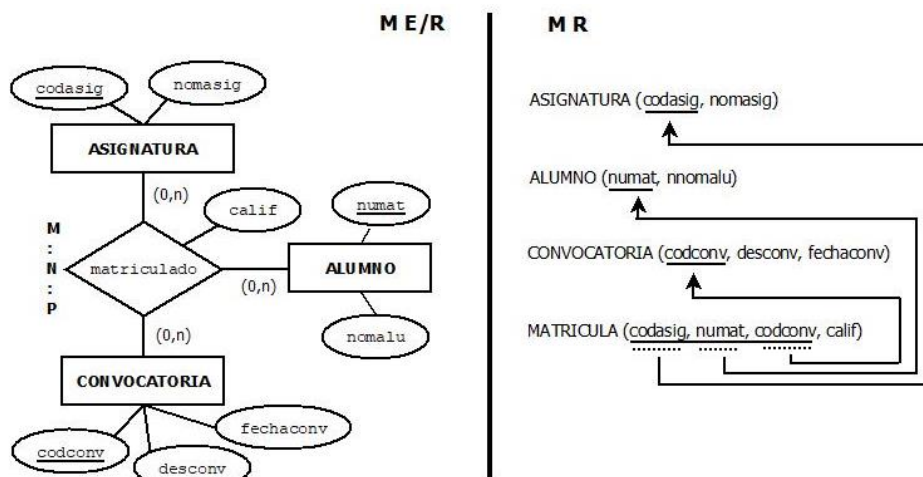
## Relaciones de grado 1 o reflexivas

Se siguen las mismas pautas dadas, solo que con una sola entidad.



## Relaciones de grado superior a 2

Recomendable descomponerlas en varias de grado 2. En este caso se suelen transformar en una nueva tabla que contendrá como claves ajenas las tres claves primarias de las entidades que relaciona. La elección de la clave primaria dependerá de la cardinalidad.



Cardinalidad de la relación	Clave primaria de la nueva tabla
M:N:P	Compuesta por las distintas claves primarias de las entidades relacionadas.
M:N:1	Compuesta por las dos claves primarias de las entidades etiquetadas con cardinalidad M y N.
N:1:1	Compuesta por la clave primaria de la entidad etiquetada con cardinalidad N y cualquiera de las claves primarias de las entidades etiquetadas con cardinalidad 1.
1:1:1	Compuesta por las claves primarias de dos entidades cualesquiera.

## Relaciones con restricciones de exclusividad, exclusión, inclusividad, inclusión

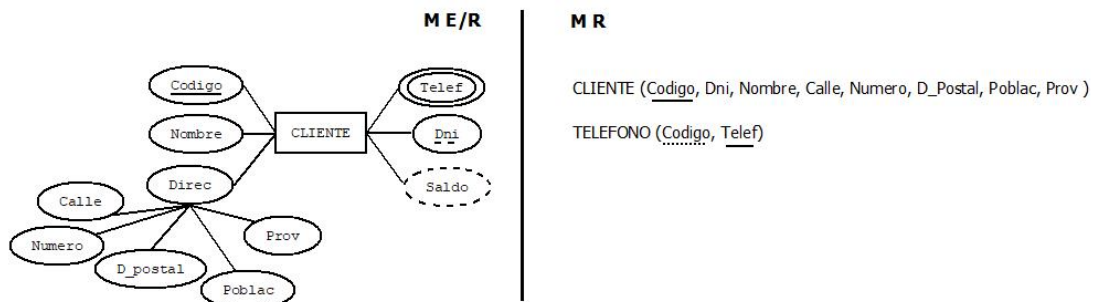
No se consideran en el modelo relacional. Se harán con disjuntadores.

## 8.3. Transformación de atributos

### Atributos de entidades

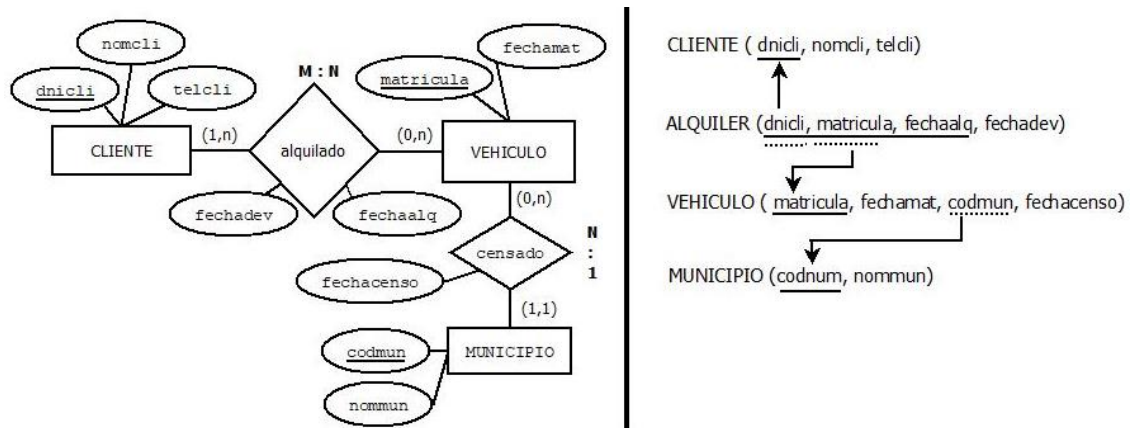
Como norma general cada atributo de entidad se transforma en una columna de la tabla a la que ha dado lugar la entidad. Se distinguen según el atributo sea:

- Clave primaria: Pasa a ser clave primaria de la tabla
- Claves alternativas: Pasan a ser columnas de la tabla
- Atributos no identificadores: Pasan a ser columnas de la tabla
- Atributos compuestos: Sus atributos elementales pasan a ser columnas de la tabla y desaparece el atributo compuesto
- Atributos calculados: Deben eliminarse de la tabla o razonarse por qué se mantienen
- Atributos multivaluados: Origina nueva tabla con atributo multivaluado y la clave primaria de la entidad que también será clave ajena. La clave de la nueva tabla estará compuesta por todos sus atributos.



### Atributos de relaciones

Si la relación se transforma en tabla todos los atributos pasan a ser columnas de la tabla. Si se transforma mediante propagación de clave, los atributos migran junto a la clave a la tabla que corresponda.



## 9. Ejemplillos

Para la **representación del diagrama E/R** previamente deberás:

1. Identificar las entidades (fuertes y débiles), las relaciones entre las entidades y los atributos (tanto de entidades como posiblemente en algún caso de relaciones).
2. Identificar las cardinalidades de las entidades y después las cardinalidades de las relaciones.



3. Identificar las claves (candidatas y primarias) y los atributos calculados, compuestos y multivaluados.
4. Estudiar si existen otras características del modelo E/R que no se puedan representar con el modelo ER.

#### Para la **representación del Modelo Relacional**:

1. Transformar las entidades y sus atributos, con especial atención a los casos de dependencia en identificación y de jerarquías.
2. Pasar a tablas las relaciones y sus posibles atributos.
3. Identificar las claves primarias de las tablas (subrayado continuo) y las claves ajenas (subrayado discontinuo). El uso de flechas puede ayudarte para representar más gráficamente las relaciones entre tablas.

- Las cardinalidades de entidad se componen de una cardinalidad mínima y otra cardinalidad máxima y se representa así:  
( **mínima** , **máxima** )
- La cardinalidad **mínima** nos da la información de obligar o no a que un registro de una entidad participe en la relación:  
Tomará el valor **0** cuando **NO** sea obligatorio.  
Tomará el valor **1** cuando **SI** sea obligatorio.
- La cardinalidad **máxima** nos da la información de la cantidad de registros de la otra relación con las que se puede relacionar un mismo registro.  
Tomará el valor **1** cuando sólo se pueda relacionar **con un único registro**.  
Tomará el valor **n** cuando se pueda relacionar **con varios registros**.
- Por tanto los posibles valores de las cardinalidades de entidad serán::  
(**0,1**) , (**1,1**) , (**0,n**) , (**1,n**)

Una empresa desea guardar la información referente a la comercialización de sus productos. Para ello nos pide que le gestionemos una base de datos donde guardemos entre otros los datos de sus comerciales como el DNI, nombre, apellidos, salario, número de hijos que tiene, y fecha de nacimiento.

Por otro lado también interesa saber de los vehículos que conducen los comerciales la matrícula, la marca, el modelo, la última fecha de revisión y los kilómetros que tiene. Cada comercial sólo puede conducir un único vehículo que tiene asignado aunque podemos tener vehículos sin ningún comercial asignado.

La empresa tiene una serie de oficinas repartidas en varias ciudades y de ellas interesa conocer el código, la dirección, la localidad y la provincia. En cada oficina pueden trabajar varios comerciales pero un comercial sólo podrá pertenecer a una única oficina.




Los productos que comercializa la empresa son muy variados y de ellos se guarda un número de referencia único, un nombre, una descripción, el precio por unidad y el descuento que puede tener. Cuando un comercial vende los productos se desea almacenar la fecha de venta y la cantidad vendida de cada producto

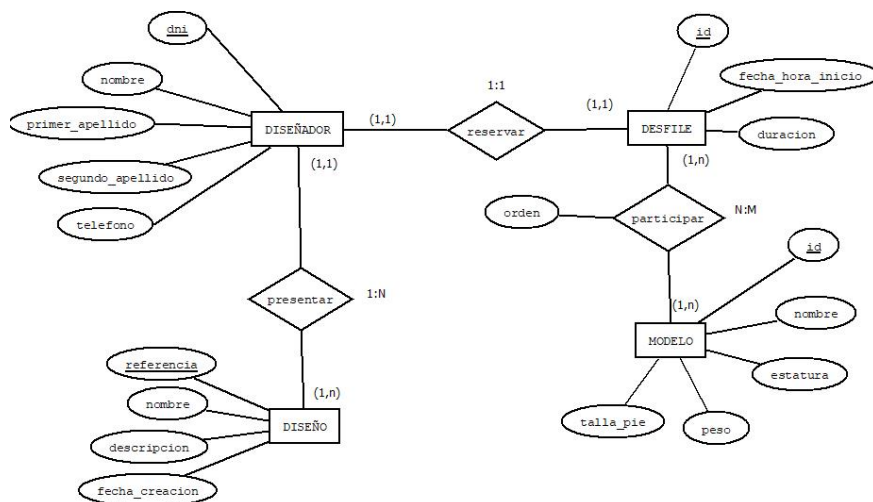
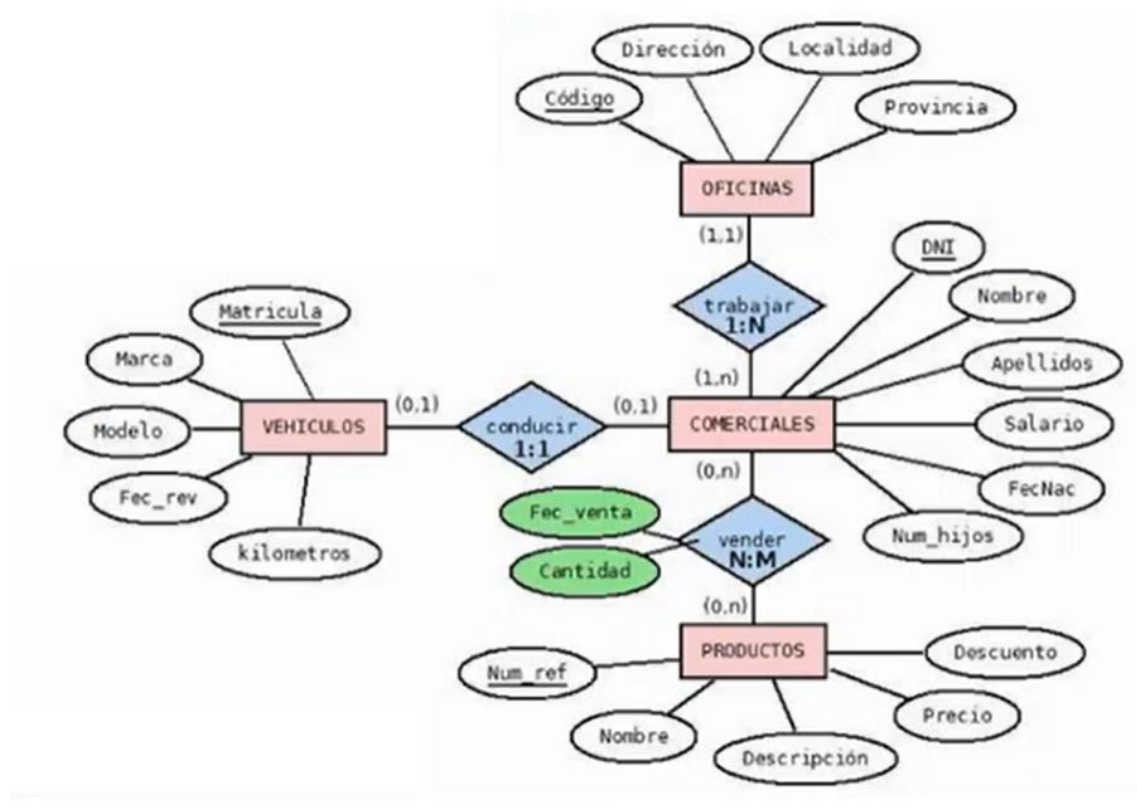


ENTIDADES	Atributos de entidad
COMERCIAL	<u>DNI</u> , nombre, apellidos, salario, número_hijos y fecha_nacimiento
VEHICULO	<u>Matrícula</u> , marca, modelo, fecha de revisión y kilómetros
OFICINA	<u>Código</u> , dirección, localidad y provincia
PRODUCTO	<u>Núm_referencia</u> , nombre, descripción, precio y descuento

RELACIONES	Atributos de relación
conducir	----
trabajar	----
vender	fecha_venta y cantidad

La clave. Lo que esté fuera de la pregunta en singular y lo de dentro en plural.

<p><b>RELACIÓN: Conducir</b></p> <p>«Cada comercial sólo puede conducir un único vehículo que tiene asignado aunque podemos tener vehículos sin ningún comercial asignado.»</p> <ul style="list-style-type: none"> <li>➤ Un <b>comercial</b>, ¿cuántos <b>vehículos</b> puede conducir? (0,1)</li> <li>➤ Un <b>vehículo</b>, ¿por cuántos <b>comerciales</b> puede ser conducido? (0,1)</li> </ul>	 <p><b>1:1</b></p>
<p><b>RELACIÓN: Trabajar</b></p> <p>«En cada oficina pueden trabajar varios comerciales pero un comercial sólo podrá pertenecer a una única oficina.»</p> <ul style="list-style-type: none"> <li>➤ En una <b>oficina</b>, ¿cuántos <b>comerciales</b> pueden trabajar? (1,n)</li> <li>➤ Un <b>comercial</b>, ¿a cuántas <b>oficinas</b> puede pertenecer? (1,1)</li> </ul>	 <p><b>1:N</b></p>
<p><b>RELACIÓN: Vender</b></p> <p>«Cuando un comercial vende los productos se desea almacenar la fecha de venta y la cantidad vendida de cada producto.»</p> <ul style="list-style-type: none"> <li>➤ Un <b>comercial</b>, ¿cuántos <b>productos</b> puede vender? (0,n)</li> <li>➤ Un <b>producto</b>, ¿por cuántos <b>comerciales</b> puede ser vendido? (0,n)</li> </ul>	 <p><b>N:M</b></p>



**DISEÑO** (referencia, dni, nombre, descripción, fecha\_creación)

**DISEÑADOR** (dni, nombre, primer\_apellido, segundo\_apellido, telefono)

**DESFILE** (id, dni, fecha\_hora\_inicio, duracion)

**PARTICIPAR** (id\_desfile, id\_modelo, orden)

**MODELO** (id, nombre, estatura, peso, talla\_pie)

## 10. Normalización de modelos relacionales

**Normalización:** Es imponer a las tablas del modelo relacional una serie de restricciones a través de un conjunto de transformaciones consecutivas. Este proceso garantizará que las tablas contienen los atributos necesarios y suficientes para describir la realidad de la entidad que representan, separando los atributos que por su contenido podrían generar la creación de otra tabla.

Codd establece una técnica para llevar a cabo el diseño de la estructura lógica de los datos representados en el modelo relacional. Esta técnica debe usarse como refinamiento que conseguirá:

- Suprimir dependencias erróneas entre atributos
- Optimizar procesos inserción, modificación y borrado de la BBDD.

La normalización:

- Se basa en el análisis de dependencias entre atributos.
- Se realiza en varias etapas consecutivas asociadas a una **forma normal** que establece unos requisitos a cumplir por la tabla sobre la que se aplica. Si no se satisface una forma normal no puede pasarse al análisis de la siguiente.
- A cada avance los criterios son más restrictivos y por tanto el modelo relacional más robusto.

### 10.1 Tipos de dependencias

El análisis de dependencias entre atributos debe tener en cuenta los siguientes conceptos:

- **Dependencia funcional (dependencia funcional simple):** Dados A y B (siendo cada uno de ellos un atributo o un conjunto de ellos), B depende funcionalmente de A, sí y solo sí, para cada valor de A solo puede existir un valor de B. Se da siempre entre atributos de una misma tabla. El atributo A se llama “determinante” porque A determina el valor de B. Se representa como  $A \rightarrow B$
- **Dependencia funcional completa (dependencia funcional plena):** Dados los atributos  $A_1, A_2, \dots, A_k$  y B se dice que B depende funcionalmente de forma completa de  $A_1, A_2, \dots, A_k$  si y solo sí B depende del conjunto de atributos  $A_1, A_2, \dots, A_k$  pero no de ninguno de sus subconjuntos.
- **Dependencia funcional transitiva:** Dados A, B y C (siendo cada uno de ellos un atributo o un conjunto de ellos) se dice que hay una dependencia transitiva entre A y C, si B depende funcionalmente de A y C depende funcionalmente de B.

**Ejemplito:**

**EMPLEADO**(dni, nombre, direccion, localidad, cod\_localidad, nombre\_hijo, edad\_hijo)

**LIBRO**(titulo\_libro, num\_ejemplar, autor, editorial, precio)

*¿Qué atributos presentan una **dependencia transitiva** en la tabla EMPLEADO?*

Cod\_localidad y localidad dependen funcionalmente de DNI pero cod\_localidad y localidad tienen otra dependencia funcional.

DNI -> Cod\_localidad -> Localidad. Hay dependencia transitiva entre localidad y DNI.

¿Qué atributos presentan **dependencia funcional de la clave primaria** de la tabla EMPLEADO?

Asumiendo que todos los empleados tienen nombres distintos, vemos una dependencia de nombre y dirección. DNI -> Nombre -> Dirección

¿Qué atributos presentan una **dependencia funcional completa** en la tabla LIBRO?

Editorial y Precio depende del conjunto de atributos de la clave primaria de la tabla, pero no de los atributos por separado.

Autor **depende funcionalmente** solo de titulo\_libro por lo que no es dependencia funcional completa.

## 10.2. Formas normales

Requisitos a cumplir por las tablas según la forma normal que se aplique.

- **Primera forma normal (FN1):**
- **Segunda forma normal (FN2):**
- **Tercera forma normal (FN3):**
- **Forma normal de Boyce Codd:**
- **Cuarta forma normal (FN4):** Basada en dependencias multivaluadas.
- **Quinta forma normal (FN5):** Basada en dependencias de Join o de reunión
- **Forma normal de dominio clave (DKFN):** Basada en restricciones impuestas sobre dominios y claves

Es recomendable para garantizar que no hay problemas en la actualización de datos aplicar el proceso de normalización hasta Tercera Forma Normal o incluso hasta Forma Normal de Boyce-Codd.

Sea

FACTURA(NºFactura,Producto,Fecha,RUT Cliente, Cliente, Valor, Cantidad, Descuento)



Descuento y Cantidad dependen funcionalmente de la clave primaria Producto, Nº Factura  
Valor del producto depende funcionalmente del producto

Fecha, Cliente y RutClient dependen funcionalmente de Nº Factura

Existe una dependencia funcional de cliente y RutCliente.

**Aficion\_Persona** (**codPersona**, **codAficion**, nombrePersona, apellidosPersona, teléfonos, localidad, provincia, nombreAvision)

<b>codPersona</b>	<b>codAficion</b>	nombrePersona	apellidosPersona	teléfonos	localidad	provincia	nombreAficion
A1	BOX	Antonio	López Sánchez	666123456 955555555	Dos Hermanas	Sevilla	Boxeo
A1	FUT	Antonio	López Sánchez	666123456 955555555	Dos Hermanas	Sevilla	Fútbol
B2	BOX	José	Ruiz García	666654321 955000000 600111111	Alcalá de Guadaira.	Sevilla	Boxeo
B2	LEC	José	Ruiz García	666654321 955000000 600111111	Alcalá de Guadaira.	Sevilla	Lectura

### Primera Forma Normal

- **Cualquier atributo que no pertenece a la clave toma valores atómicos**, es decir, los valores de los atributos son indivisibles. (No multivaluados)
- **Cualquier atributo que no pertenece a la clave tiene una dependencia funcional de la clave.**

Podríamos separar por apellidos (si lo necesitamos). Pero claramente está mal porque una persona tiene varios teléfonos.

**Aficion\_Persona** ( **codPersona**, **codAficion**, nombrePersona, apellidosPersona, localidad, provincia, nombreAvision).

**Teléfono** ( **codPersona**, **teléfono** ).

1 FN

teléfonos

<b>CodPersona</b>	<b>teléfono</b>
A1	666123456
A1	955555555
B2	666654321
B2	955000000
B2	600111111

Afición-persona

<b>codPersona</b>	<b>codAficion</b>	nombrePersona	apellidosPersona	localidad	provincia	nombreAficion
A1	BOX	Antonio	López Sánchez	Dos Hermanas	Sevilla	Boxeo
A1	FUT	Antonio	López Sánchez	Dos Hermanas	Sevilla	Fútbol
B2	BOX	José	Ruiz García	Alcalá de Guadaira.	Sevilla	Boxeo
B2	LEC	José	Ruiz García	Alcalá de Guadaira.	Sevilla	Lectura

(Se crea a partir de la tabla inicial una nueva tabla con atributos los que presentan dependencia funcional de la clave primaria (atómicos). La clave de esta es la

misma clave primaria de la tabla. Después con los atributos no atómicos se crea otra tabla y se elige entre ellos uno que será la clave primaria de la tabla junto a la clave primaria de la tabla inicial. Se comprueba que la segunda tabla esté en 1FN. Si no está en 1FN se toma la segunda tabla como tabla inicial y se repite el proceso).

## Segunda Forma Normal

Está en primera forma normal y...

- **Todo atributo que no pertenece a la clave tiene una dependencia funcional completa de la clave (se obtiene a partir de toda la clave y no de una parte de la clave).**

Afición ( codAfición, nombreAfición)

Persona ( codPersona, nombrePersona, apellidosPersona, localidad, provincia)

Afición\_Persona ( codPersona, codAfición ).

Teléfono ( codPersona, teléfono ).

### 2 FN

<u>CodAfición</u>	nombreAfición
BOX	Boxeo
FUT	Fútbol
LEC	Lectura

<u>codPersona</u>	<u>codAfición</u>
A1	BOX
A1	FUT
B2	BOX
B2	LEC

<u>CodPersona</u>	<u>teléfono</u>
A1	666123456
A1	955555555
B2	666654321
B2	955000000
B2	600111111

<u>CodPersona</u>	nombre	apellidos	localidad	provincia
A1	Antonio	López Sánchez	Dos Hermanas	Sevilla
B2	José	Ruiz García	Alcalá de Guadaira	Sevilla

(Se crea una nueva tabla con los atributos que dependen funcionalmente de forma completa de la clave. La clave será la misma clave primaria que la tabla inicial. Con los restantes, se crea otra tabla que tendrá por clave el subconjunto de atributos de la clave inicial de los que dependen de forma completa. Si la tabla está en 2FN el proceso termina y, si no, se toma la segunda tabla como inicial y se repite el proceso).

## Tercera Forma Normal

Está en segunda forma normal y...

- Todo atributo que no pertenece a la clave, no depende transitivamente de la clave.

La provincia se podría determinar a partir de la localidad que a su vez se determina de persona. (Dependencia funcional transitiva de provincia respecto a la clave).

Si hay un único atributo que no pertenece a la clave ya podemos asegurar que está en tercera forma normal.

CodPersona -> Localidad -> Provincia

Generamos una nueva relación:

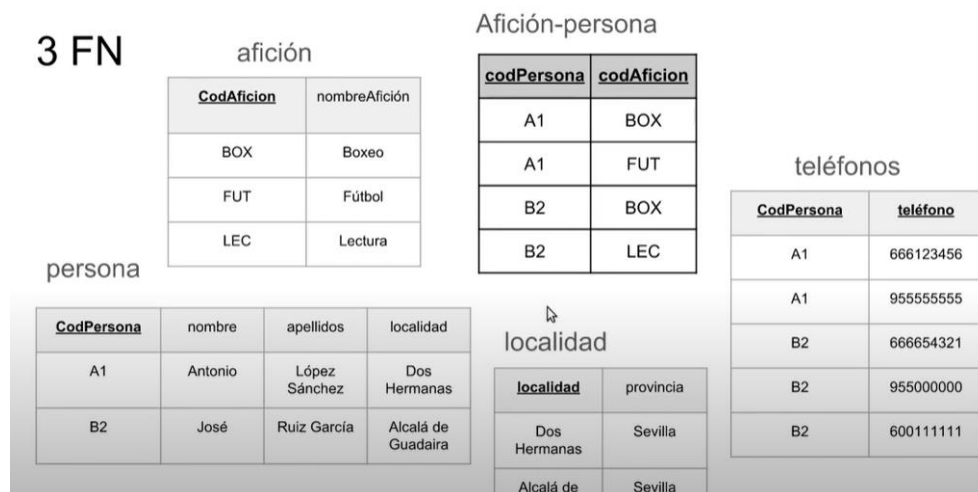
**Afición** ( codAficion, nombreAficion)

**Persona** ( codPersona, nombrePersona, apellidosPersona, localidad)

**Localidad** ( localidad, provincia )

**Aficion\_Persona** ( codPersona, codAficion ).

**Teléfono** ( codPersona, teléfono ).



(Se crea una nueva tabla con los atributos que no dependen transitivamente de forma completa de la clave. La clave será la misma clave primaria que la tabla inicial. Con los restantes, se crea otra tabla con los atributos no clave que intervienen en la dependencia transitiva y se elige uno de ellos como clave primaria, si cumple los requisitos para ellos. Si la tabla está en 3FN el proceso termina y, si no, se toma la segunda tabla como inicial y se repite el proceso).

## Forma Normal de Boyce-Codd

**Está en forma normal de Boyce-Codd si está en 3FN y...**

- Todo determinante es una clave candidata. (Un determinante es todo atributo simple o compuesto del que depende funcionalmente de forma completa algún otro atributo de la tabla.

Aquellas tablas en las que todos sus atributos forman parte de la clave primaria están en FNBC.



Si se encuentra un determinante que no es clave primaria no está en FNBC. Esta redundancia suele ocurrir por una mala elección de la clave. Para normalizarlo debe descomponerse la tabla inicial en dos, siendo cuidadosos para evitar la pérdida de información en la descomposición.

---

**COMPRAS (cod\_compra, cod\_prod, nomb\_prod, fecha, cantidad, precio, fecha\_rec, cod\_prov, nomb\_prov, tfno).**

**Se pide normalizarla hasta FNBC.**

**Comprobamos 1FN:**

La tabla COMPRAS está en 1FN ya que todos sus atributos son atómicos y todos los atributos no clave dependen funcionalmente de la clave.

**Comprobamos 2FN:**

Nos preguntaremos ¿Todo atributo depende de todo el conjunto de atributos que forman la clave primaria, o sólo de parte?. Como vemos, existen atributos que dependen sólo de una parte de la clave, por lo que esta tabla no está en 2FN.

Veamos las dependencias:

cod\_prod → nomb\_prod, y cod\_prod es parte de la clave primaria.

Al no estar en 2FN, hemos de descomponer la tabla COMPRAS en:

COMPRA1 (cod\_compra, cod\_prod, fecha, cantidad, precio, fecha\_rec, cod\_prov, nomb\_prov, tfno).  
PRODUCTO (cod\_prod, nomb\_prod).

Una vez hecha esta descomposición, ambas tablas están en 2FN. Todos los atributos no clave dependen de toda la clave primaria.

**Comprobamos 3FN:**

PRODUCTO está en 3FN, ya que por el número de atributos que tiene no puede tener dependencias transitivas. ¿COMPRA1 está en 3FN? Hemos de preguntarnos si existen dependencias transitivas entre atributos no clave.

Veamos las dependencias:

cod\_prov → nomb\_prov  
cod\_prov → tfno (siendo cod\_prov el código del proveedor y nomb\_prov el nombre del proveedor)

COMPRA1 no está en 3FN porque existen dependencias transitivas entre atributos no clave, por tanto hemos de descomponer:

COMPRA2 (cod\_compra, cod\_prod, fecha, cantidad, precio, fecha\_rec, cod\_prov)

PROVEEDOR (cod\_prov, nomb\_prov, tfno)

**Comprobamos FNBC:**

PRODUCTO está en FNBC, ya que está en 3FN y todo determinante es clave candidata. COMPRA2 está en FNBC, ya que está en 3FN y todo determinante es clave candidata. PROVEEDOR está en FNBC, ya que está en 3FN y todo determinante es clave candidata.

La tabla inicial COMPRAS queda normalizada hasta FNBC del siguiente modo:

PRODUCTO (cod\_prod, nomb\_prod)

COMPRA2 (cod\_compra, cod\_prod, fecha, cantidad, precio, fecha\_rec, cod\_prov)

PROVEEDOR (cod\_prov, nomb\_prov, tfno)

---

De haber hecho el modelo de entidad-relación correctamente a partir del documento de requerimientos (universo del discurso) no habríamos llegado a esos casos iniciales...