

2.1. INTRODUCCIÓN ¶

En este tema veremos como hacer el diseño conceptual y lógico de una base de datos. Empezaremos elaborando el modelo conceptual usando diagramas Entidad-Relación y Entidad-Relación extendidos. Este diseño es de más alto nivel, más próximo al usuario y más alejado del diseño físico de la BD. A continuación, a partir del modelo Entidad-Relación, procederemos a generar el modelo relacional, el cual ya se halla muy próximo al modelo físico de BD. Veremos las reglas de transformación que hemos de seguir para ello. Por último deberemos normalizar las tablas obtenidas para evitar redundancias. Resumiendo, los 2 modelos lógicos, de mayor a menor nivel de abstracción, que veremos en este tema son:

Modelo Entidad-Relación (extendido)
Modelo Relacional

En el siguiente tema, realizaremos el diseño físico de la BD a partir del modelo relacional.

2.2. DISEÑO DE BD ¶

El diseño de una base de datos consiste en extraer todos los datos relevantes de un problema, por ejemplo, saber que datos están implicados en el proceso de facturación de una empresa que vende artículos de informática, o, que datos son necesarios para llevar el control de pruebas diagnósticas en un centro de radiológico. Para extraer estos datos, se debe realizar un análisis en profundidad del problema, para averiguar qué datos son esenciales para la base de datos y descartar los que no sean necesarios. Una vez extraídos los datos esenciales comenzamos a construir los modelos adecuados. Es decir, construimos, mediante una herramienta de diseño de base de datos, un esquema que exprese con total exactitud todos los datos que el problema requiere almacenar. Ya dijimos en el tema anterior, que es algo equivalente al dibujo de un plano previo a la construcción de un edificio. También introdujimos en el tema 1, las distintas fases por las que atraviesa el proceso de diseño de una Base de Datos. Además, previo al diseño es necesario realizar una primera fase denominada de análisis.

2.2.1. Fase de Análisis: Especificación de requisitos Software (E.R.S.) ¶

Antes de pasar a diseñar una BD hay que tener claro que es lo que queremos hacer. Para ello, típicamente los informáticos se reúnen con los futuros usuarios del sistema para recopilar la información que necesitan para saber que desean dichos usuarios. Normalmente se hace una reunión inicial a y partir de ella se elabora una batería de preguntas para entrevistar a los usuarios finales en una segunda reunión y obtener de ella una información detallada de lo que se espera de nuestra BD. De estas entrevistas, se extrae el documento más importante del análisis, el documento de Especificación de Requisitos Software o E.R.S. A partir de dicha E.R.S. Se extrae toda la información necesaria para la modelización de datos.

2.2.2. Fase 1 del diseño. Diseño Conceptual: Modelo Entidad/Relación (E/R) ¶

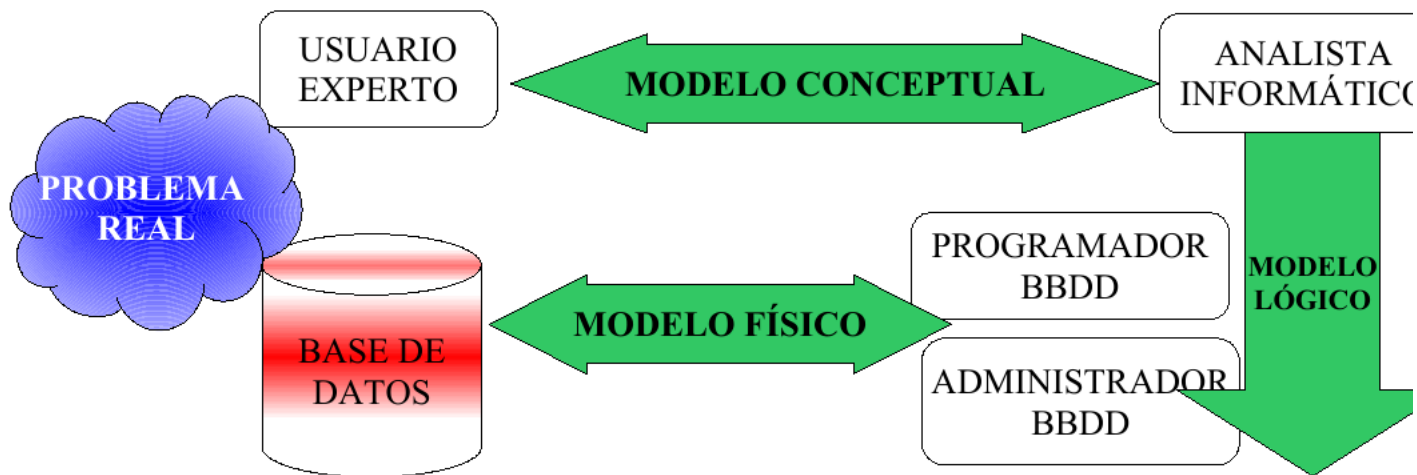
Habitualmente quien realiza la modelización es un analista informático que no tiene porqué ser un experto en el problema que pretende resolver (Contabilidad, Gestión de Reservas hoteleras, medicina, economía, etc.). Es por esto que es imprescindible contar con la experiencia de un futuro usuario de la BD que conozca a fondo todos los entresijos del negocio, y que, a su vez, no tienen porqué tener ningún conocimiento de informática. El objetivo de esta fase del diseño consiste en representar la información obtenida del usuario final y concretada en el E.R.S. mediante estándares para que el resto de la comunidad informática pueda entender y comprender el modelo realizado. El modelo que se utiliza en esta primera fase del diseño tiene un gran poder expresivo para poder comunicarse con el usuario que no es experto en informática y se denomina Modelo Conceptual. El modelo conceptual que utilizaremos es el Modelo Entidad/Relación e iremos profundizando en él a lo largo de esta unidad.

2.2.3. Fase 2 del diseño. Diseño Lógico: Modelo Relacional ¶

Este modelo es más técnico que el anterior porque está orientado al personal informático y generalmente tiene traducción directa al modelo físico que entiende el SGBD. Se obtienen a partir del modelo conceptual y dependerá de la implementación de la BD. Así, no es lo mismo implementar una base de datos jerárquica u orientada a objetos que una BD relacional. El modelo que se usará en este módulo es el Modelo Relacional.

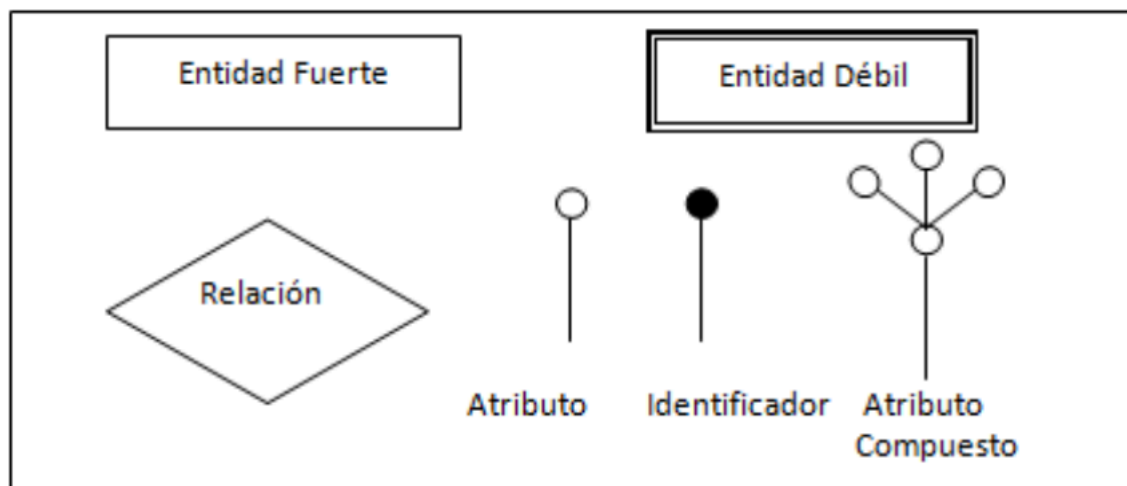
2.2.4. Fase 3 del diseño. Diseño Físico: Modelo Físico ¶

Es el resultado de aplicar el modelo lógico a un SGBD concreto. Generalmente está expresado en un lenguaje de programación de BBDD tipo SQL. En este módulo, transformaremos el Modelo Relacional en el modelo físico mediante el sublenguaje DDL de SQL. Esto se estudiará en el próximo tema.



2.3. MODELO ENTIDAD/RELACIÓN ¶

El modelo Entidad-Relación es el modelo más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976 y se basa en la existencia de objetos a los que se les da el nombre de entidades, y asociaciones entre ellos, llamadas relaciones. Sus símbolos principales se representan en el cuadro siguiente.



A continuación se detallan los elementos fundamentales de este modelo.

2.3.1. Entidades ¶

Una entidad es cualquier objeto o elemento acerca del cual se pueda almacenar información en la BD. Las entidades pueden ser concretas como una persona o abstractas como una fecha. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.

Tipos de entidades

Hay dos tipos de entidades: fuertes y débiles. Una entidad débil es una entidad cuya existencia depende de la existencia de otra entidad. Una entidad fuerte es una entidad que no es débil.



2.3.2. Atributos ¶

Una entidad se caracteriza y distingue de otra por los atributos, en ocasiones llamadas propiedades o campos, que representan las características de una entidad. Los atributos de una entidad pueden tomar un conjunto de valores permitidos al que se le conoce como dominio del atributo. Dando valores a estos atributos, se obtienen las diferentes ocurrencias de una entidad.

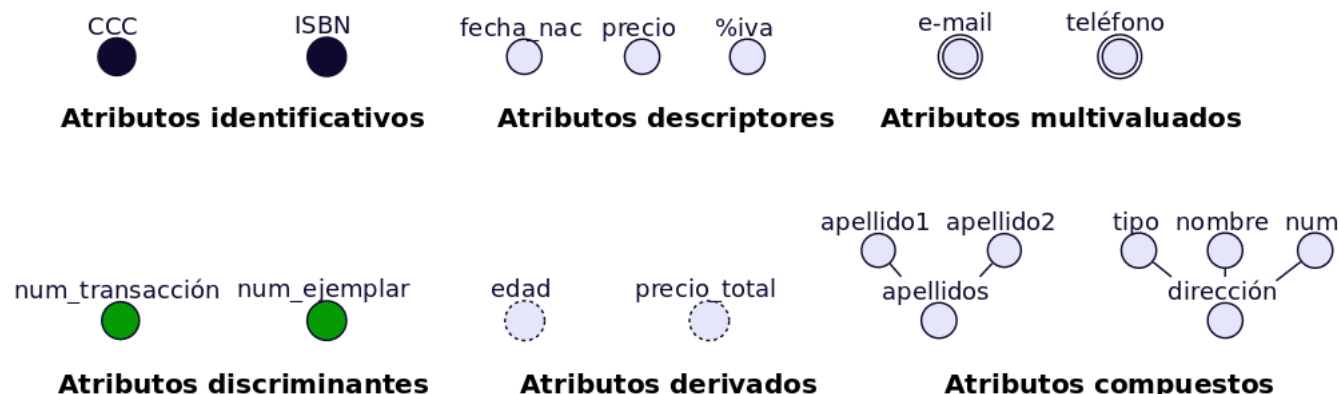
En esencia, existen dos tipos de atributos: - Identificadores de entidad (también llamados clave primaria o clave principal): son atributos que identifican de manera unívoca cada ocurrencia de una entidad. - Descriptores de entidad: son atributos que muestran unas características de la entidad.

Siempre debe existir, al menos, un atributo identificativo.

Ejemplos de atributos:



2.3.2.1. Tipos de atributos ¶



Atributos identificadores o identificativos: Son atributos cuyos valores no se repiten dentro de una misma entidad o relación. Sirven para identificar de forma unívoca cada ocurrencia. Actúan como clave principal o primaria. Por ejemplo CCC (Código Cuenta Corriente) que identifica cada cuenta bancaria. O ISBN (International Standard Book Number) que identifica cada libro que se publica. Un atributo identificativo puede ser un atributo compuesto. Por ejemplo CCC podría descomponerse en 3 atributos: num_banco, num_sucursal y num_cuenta.

Atributos discriminadores o discriminantes: Son atributos que discriminan distintas ocurrencias de una entidad débil en identificación dentro de la entidad fuerte de la que dependen. Lo representaremos con un círculo relleno de un color distinto a los atributos identificadores y descriptivos. Por ejemplo num_transacción dentro de una CCC o num_ejemplar dentro de un ISBN.

Atributos descriptores o descriptivos: Son los atributos que describen diversas propiedades de una entidad o relación (¡la relaciones también pueden tener atributos!). Son los más frecuentes.

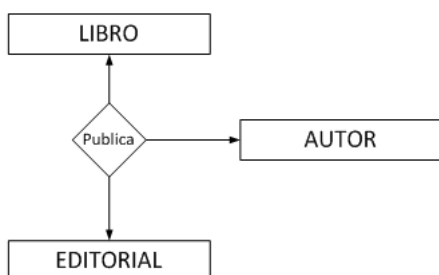
Atributos derivados: Son atributos cuyos valores se calculan a partir de los valores de otros atributos. Por ejemplo podemos disponer de un atributo fecha_nac que sería un atributo descriptivo normal y calcular el valor del atributo edad a partir de él. El precio_total también podría calcularse a partir del precio + %iva.

Atributos multivaluados: Son atributos descriptores que poseen varios valores de un mismo dominio. Por ejemplo, si necesitamos almacenar varios e-mail de una misma persona entonces deberemos utilizar un atributo multivaluado. Igual sucede con el teléfono. Si sólo necesitamos almacenar un sólo valor utilizaremos un atributo descriptivo normal.

Atributos compuestos: Muchas veces se confunden con los anteriores, aunque no tienen nada que ver con ellos. Un atributo compuesto es un atributo que puede ser descompuesto en otros atributos pertenecientes a distintos dominios.

2.3.3. Relaciones ¶

Una relación es la asociación que existe entre dos a más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior. Normalmente le pondremos de nombre la primera o primeras letras de las entidades que relaciona. Las entidades que están involucradas en una determinada relación se denominan entidades participantes. El número de participantes en una relación es lo que se denomina grado de la relación. Por ejemplo la relación CLIENTE-COCHE es de grado 2 o binaria, ya que intervienen dos entidades.



Observa que el nombre que ponemos a la relación usa las primeras letras de cada entidad. En este caso como ambas empiezan por "C" se añade algunas letras más para hacer referencia a CLIENTES. También podríamos haber puesto como nombre de la relación uno más descriptivo de la misma, por ejemplo "Compra" (CLIENTE compra COCHE), pero esta nomenclatura puede conducir a confusión a la hora de determinar la cardinalidad de la relación cuando estamos aprendiendo. La relación PUBLICAR, es de grado 3, ya que involucra las entidades LIBRO, EDITORIAL y AUTOR.



Cuando una entidad está relacionada consigo misma, hablamos de relación reflexiva.



Aunque el modelo E-R permite relaciones de cualquier grado, la mayoría de las aplicaciones del modelo sólo consideran relaciones del grado 2.

2.3.3.1. El Papel o Rol de una entidad en una relación ¶

Es la función que tiene en una relación. Se especifican los papeles o roles cuando se quiera aclarar el significado de una entidad en una relación. A continuación mostramos los mismos ejemplos del punto anterior pero incluyendo el papel o rol de cada entidad en las relaciones:

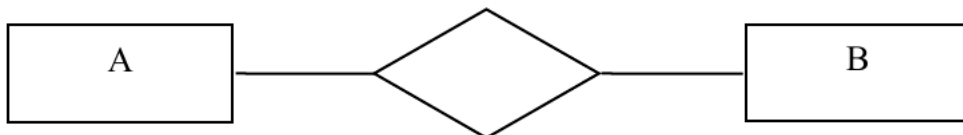


2.3.3.2. La Cardinalidad de una relación ¶

Cuando la relación es binaria, cosa que ocurre en la mayoría de los casos, la cardinalidad es el número de ocurrencias de una entidad asociadas a una ocurrencia de la otra entidad. Existen principalmente tres tipos de cardinalidades binarias:

Relación uno a uno 1:1

A cada elemento de la primera entidad le corresponde no más de un elemento de la segunda entidad, y a la inversa. Es representado gráficamente de la siguiente manera:

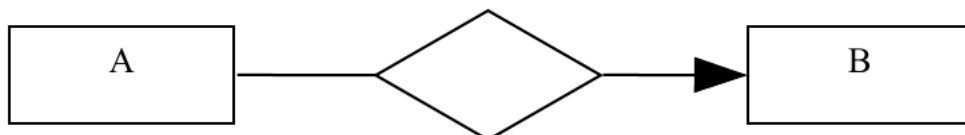


Ejemplo cardinalidad 1:1



Relación uno a muchos 1:N

Significa que cada elemento de una entidad del tipo A puede relacionarse con cualquier cantidad de elementos de una entidad del tipo B, y un elemento de una entidad del tipo B solo puede estar relacionado con un elemento de una entidad del tipo A. Su representación gráfica es la siguiente: Nótese en este caso que el extremo punteado de la flecha de la relación de A y B, indica un elemento de A conectado a muchos de B.

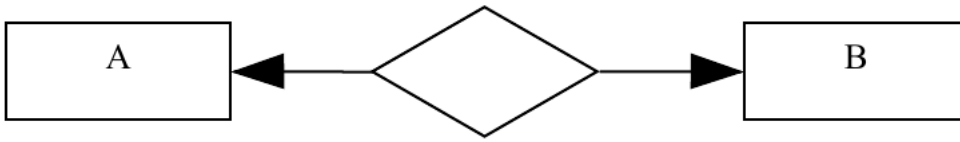


Ejemplo cardinalidad 1:N

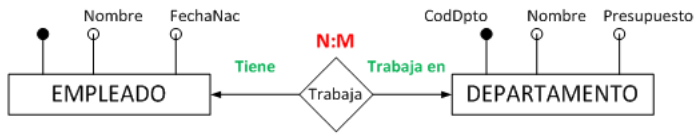


Muchos a muchos N:M

Establece que cualquier cantidad de elementos de una entidad del tipo A pueden estar relacionados con cualquier cantidad de elementos de una entidad del tipo B. El extremo de la flecha que se encuentra punteada indica el "varios" de la relación.



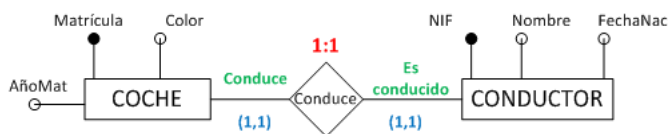
Ejemplo cardinalidad N:M



2.3.3.3. La Participación de una entidad ¶

La participación de una entidad también se conoce como cardinalidad de la entidad dentro de una relación. Una misma entidad puede tener distinta cardinalidad dentro de distintas relaciones. Para obtener la participación, se debe fijar una ocurrencia concreta de una entidad y averiguar cuántas ocurrencias de la otra entidad le corresponden como mínimo y como máximo. Después realizar lo mismo en el otro sentido. Estas ocurrencias mínimas y máximas (llamadas también participación de una entidad) se representarán entre paréntesis y con letras minúsculas en el lado de la relación opuesto a la entidad cuyas ocurrencias se fijan. Para determinar la cardinalidad nos quedamos con las participaciones máximas de ambas y se representan con letras mayúsculas separadas por dos puntos junto al símbolo de la relación. Veamos algunos ejemplos:

Ejemplo 1



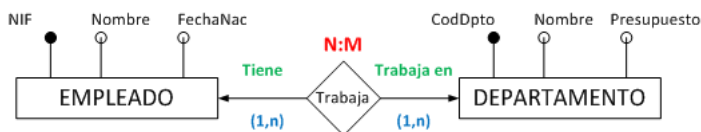
Un conductor “conduce” como mínimo 1 coche y como máximo 1 coche → Participación (1,1) y se pone en el lado opuesto a CONDUCTOR, es decir, junto a COCHE. Un coche “es conducido” como mínimo por 1 conductor y como máximo por 1 conductor → Participación (1,1) y se pone en el lado opuesto a COCHE, es decir, junto a CONDUCTOR. Para determinar la cardinalidad nos quedamos con las dos participaciones máximas. Es decir → 1:1.

Ejemplo 2



Un cliente “compra” como mínimo 1 coche y como máximo puede comprar más de un coche, es decir, varios coches. Ese varios se representa con la letra “n” → Participación (1,n) y se pone en el lado opuesto a CLIENTE, es decir, junto a COCHE. Un coche “es comprado” como mínimo por 1 cliente y como máximo por 1 cliente → Participación (1,1) y se pone en el lado opuesto a COCHE, es decir, junto a CLIENTE. Para determinar la cardinalidad nos quedamos con las dos participaciones máximas y la “n” se pone en mayúsculas “N”. Es decir → 1:N.

Ejemplo 3

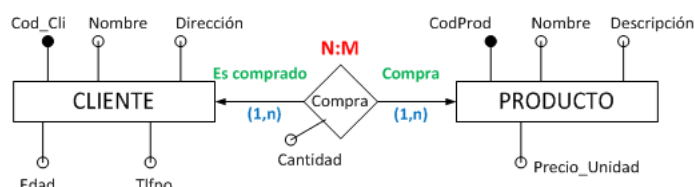


Un empleado “trabaja” como mínimo 1 departamento y como máximo puede trabajar en varios. Ese varios se representa con la letra “n” → Participación(1,n) y se pone en el lado opuesto a EMPLEADO, es decir, junto a DEPARTAMENTO. Un departamento “tiene” como mínimo por 1 empleado y como máximo puede tener varios → Participación (1,n) y se pone en el lado opuesto a DEPARTAMENTO, es decir, junto a EMPLEADO. Para determinar la cardinalidad nos quedamos con las dos participaciones máximas y la “n” se pone en mayúsculas “N” y para diferenciar el otro “varios” en lugar de “N” ponemos “M” (Igual que cuando en matemáticas había dos variables no se ponía x e x sino x e y). Es decir → N:M.

2.3.3.4. Atributos propios de una relación ¶

Las relaciones también pueden tener atributos, se les denominan atributos propios. Son aquellos atributos cuyo valor sólo se puede obtener en la relación, puesto que dependen de todas las entidades que participan en la relación. Veamos un ejemplo.

Ejemplo:



Tenemos la relación “Compra” entre cliente y producto. Así un cliente puede comprar uno o varios productos, y un producto puede ser comprado por uno o varios clientes. Encontramos una serie de atributos propios de cada una de las entidades [CLIENTE (Cod_Cliente, Nombre, Dirección, edad, teléfono) y PRODUCTO (Cod_Producto, Nombre, Descripción, Precio_Unidad)], pero también podemos observar como el atributo “Cantidad” es un atributo de la relación. ¿Por qué? Pues porque un mismo cliente puede comprar distintas cantidades de distintos productos y un mismo producto puede ser comprado en distintas cantidades por distintos clientes. Es decir el atributo cantidad depende del cliente y del producto de que se traten.

2.3.3.5. Relaciones de dependencia: Entidades Fuertes y Entidades Débiles ¶

Al definir las entidades hablamos de dos tipos de ellas: fuertes y débiles. Una entidad débil está unida a una entidad fuerte a través de una relación de dependencia. Hay dos tipos de relaciones de dependencia:

Dependencia en existencia Se produce cuando una entidad débil necesita de la presencia de una fuerte para existir. Si desaparece la existencia de la entidad fuerte, la de la débil carece de sentido. Se representa con una barra atravesando el rombo y la letra E en su interior. Son relaciones poco frecuentes.

Ejemplo:



En la figura se muestra el caso de que un empleado puede tener ninguno, uno o varios hijos, por lo que los datos de los hijos deben sacarse en una entidad aparte, aunque siguen siendo datos propios de un empleado. Si se eliminase un registro de un empleado, no tendría sentido seguir manteniendo en la base datos la información sobre sus hijos.

Dependencia en identificación Se produce cuando una entidad débil necesita de la fuerte para identificarse. Por sí sola la débil no es capaz de identificar de manera unívoca sus ocurrencias. La clave de la entidad débil se forma al unir la clave de la entidad fuerte con los atributos identificadores de la entidad débil.

Ejemplo:

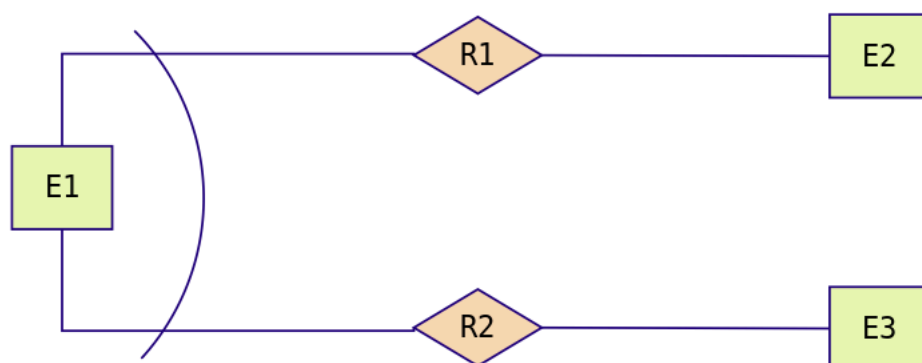


En la figura se observa que la provincia tiene uno o varios municipio y que un municipio pertenece a una sola provincia. Ahora bien si lo que identifica a los municipios es el código que aparece en el código postal, se tiene que las dos primeras cifras corresponden al código de la provincia y las tres últimas al del municipio. Por ejemplo, el C.P de Écija es 41400, dónde 41 es el código de la provincia y 400 el del municipio. De esta forma, habrá distintos municipios con código 400 en distintas provincias. Uno de estos municipios se distinguirá del resto al anteponerle las dos primeras cifras correspondientes al código de la provincial.

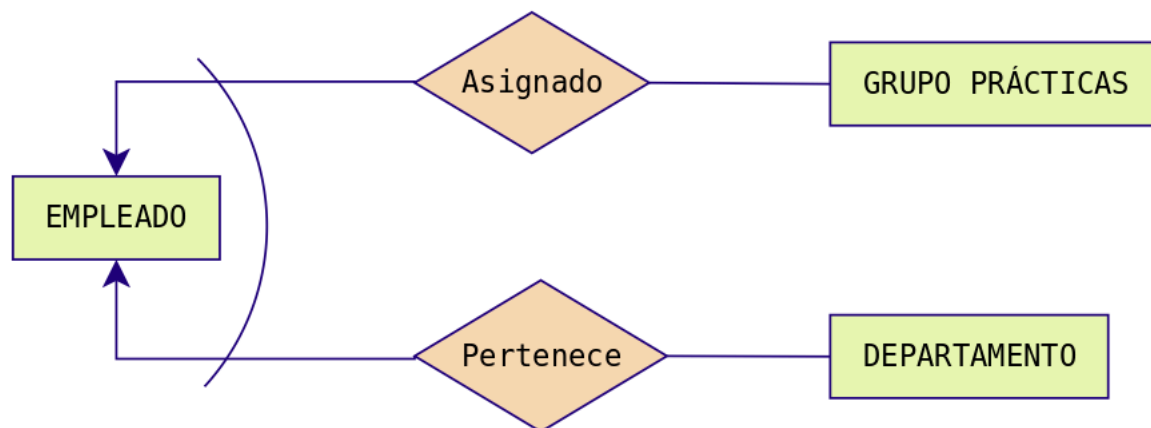
2.3.3.6. Símbolos de exclusividad o inclusividad entre relaciones ¶

Otros símbolos usados en el modelo E/R son los siguientes:

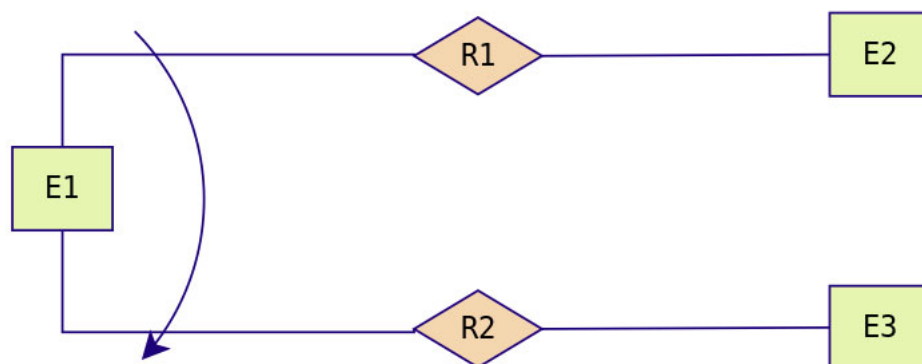
Restricción de exclusividad entre dos tipos de relaciones R1 y R2 respecto a la entidad E1. Significa que E1 está relacionada, o bien con E2 o bien con E3, pero no pueden darse ambas relaciones simultáneamente.



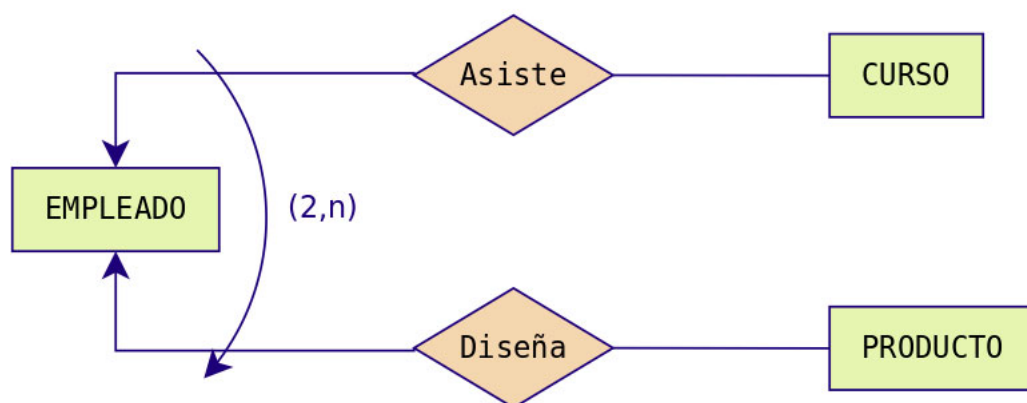
Ejemplo: Un empleado puede estar en una empresa, o bien realizando prácticas, en cuyo caso está asignado a un grupo de prácticas y no pertenece a ningún departamento en concreto. O bien puede ser empleado en plantilla y en este caso pertenece a un departamento.



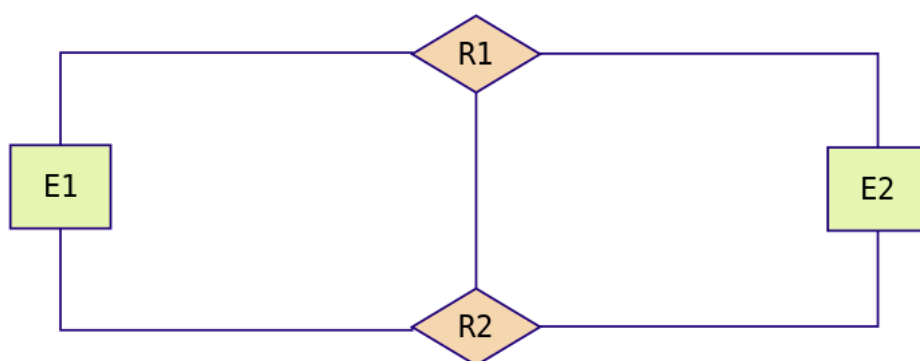
Restricción de inclusividad entre dos tipos de relaciones R1 y R2 respecto a la entidad E1. Para que la entidad E1 participe en la relación R2 debe participar previamente en la relación R1.



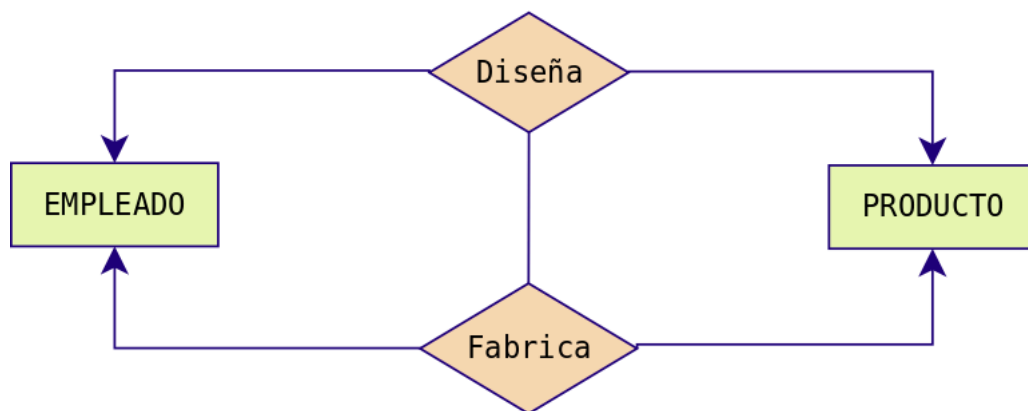
Ejemplo: Para que un empleado pueda trabajar como diseñador de productos debe haber asistido, al menos, a dos cursos.



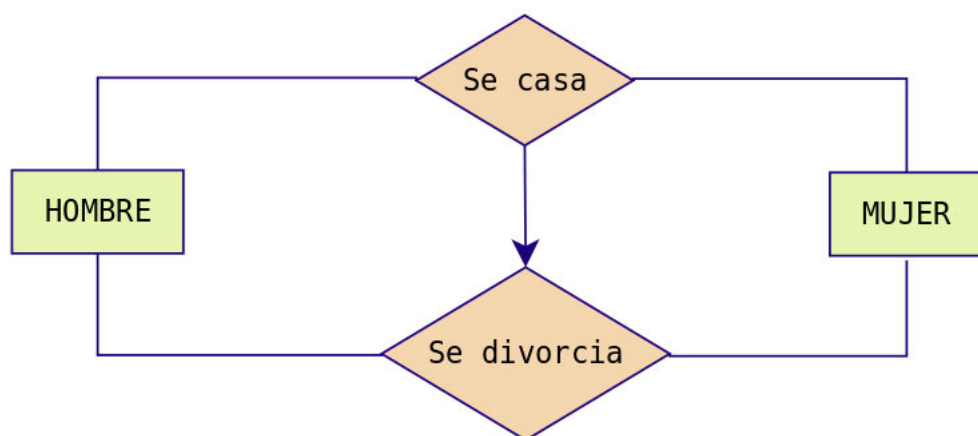
Restricción de exclusión entre dos tipos de relaciones R1 y R2. Significa que E1 está relacionada con E2 bien mediante R1, o bien mediante R2 pero que no pueden darse ambas relaciones simultáneamente.



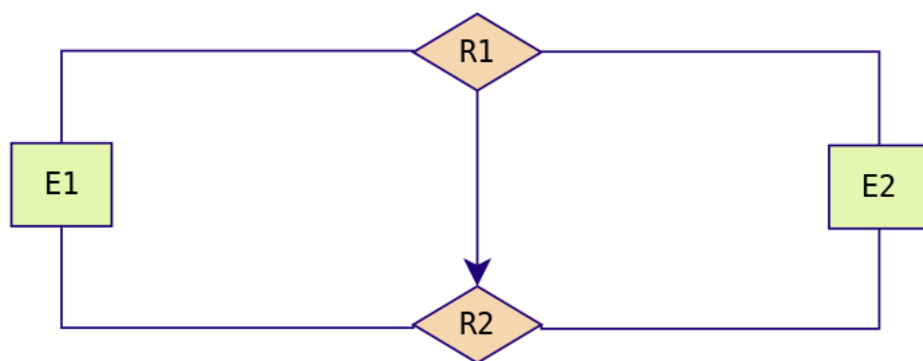
Ejemplo: Los empleados, en función de sus capacidades, o son diseñadores de productos o son operarios y los fabrican, no es posible que ningún empleado sea diseñador y fabricante a la misma vez.



Restricción de inclusión entre dos tipos de relaciones R1 y R2. Para que la entidad E1 participe en la relación R2 con E2 debe participar previamente en la relación R1.



Ejemplo: Para que un hombre se divorcie de una mujer, previamente ha de haberse casado con ella.

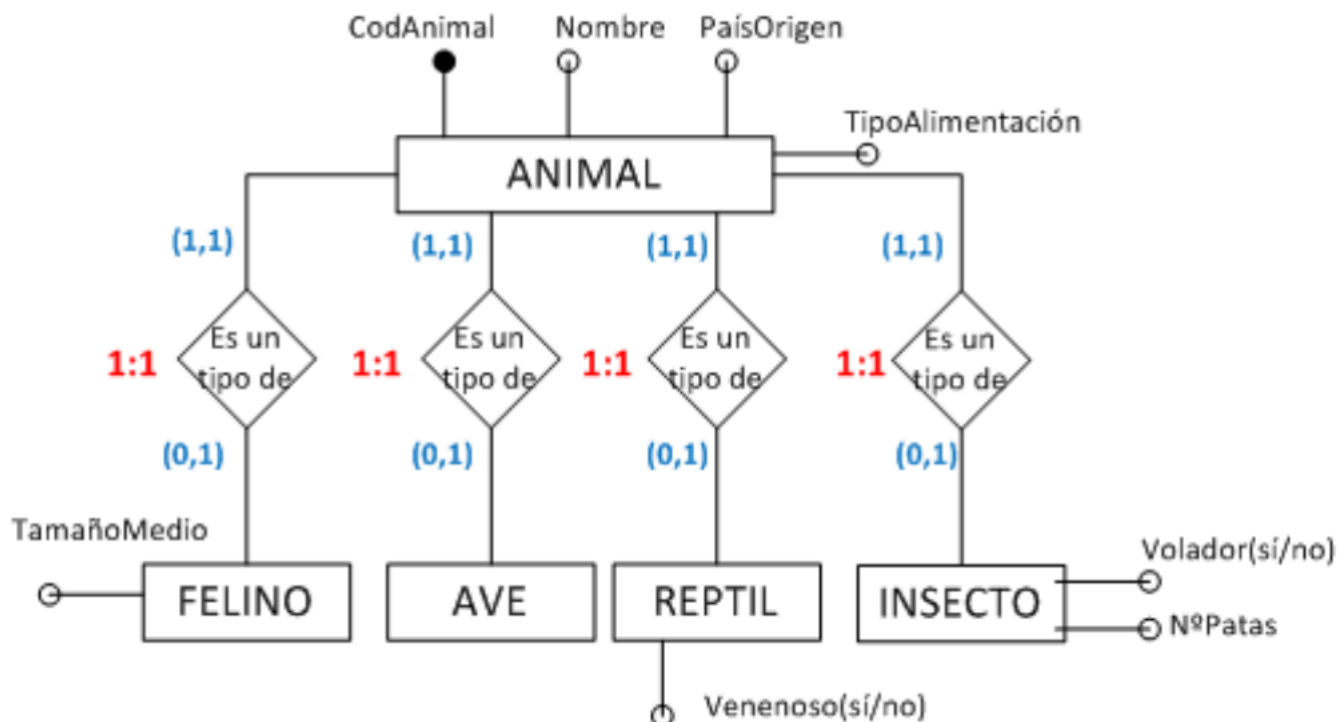


2.4. MODELO E/R EXTENDIDO ¶

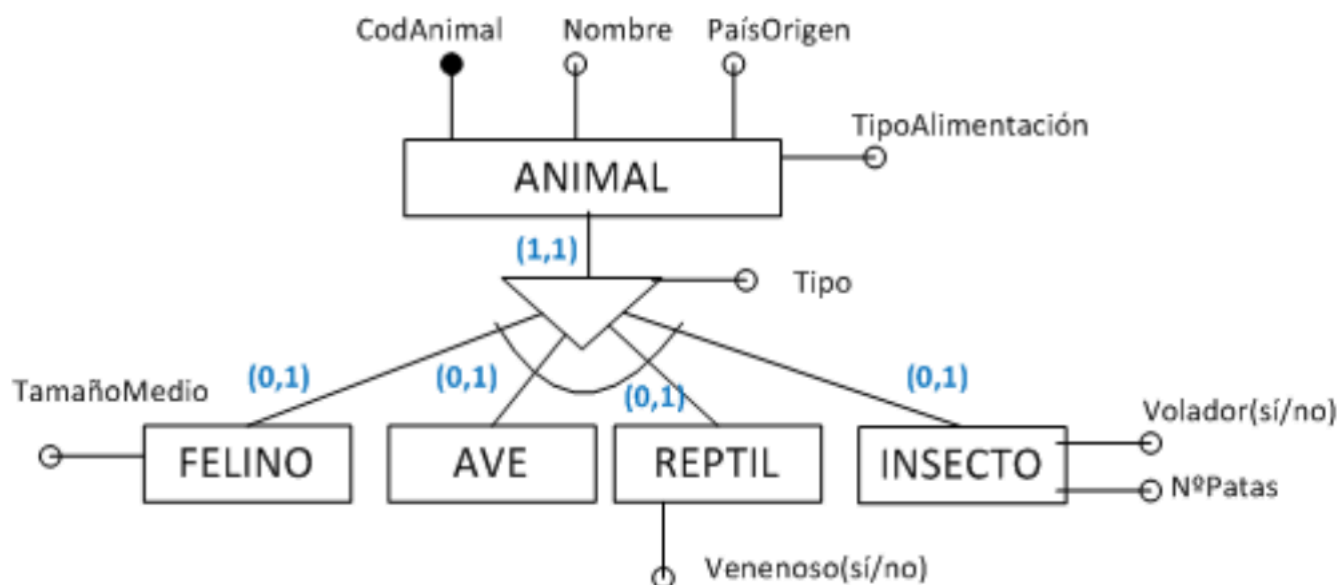
El modelo Entidad/Relación extendido incluye todo lo visto en el modelo Entidad/Relación pero además las **Relaciones de Jerarquía**. Una relación de jerarquía se produce cuando una entidad se puede relacionar con otras a través de una relación cuyo rol sería "Es un tipo de".

Por ejemplo, imaginemos la siguiente situación.

Queremos hacer una BD sobre los animales de un Zoo. Tenemos las entidades ANIMAL, FELINO, AVE, REPTIL, INSECTO. FELINO, AVE, REPTIL e INSECTO tendrían el mismo tipo de relación con ANIMAL: "son un tipo de". Ahora bien, su representación mediante el E/R clásico sería bastante engorrosa:



Para evitar tener que repetir tantas veces el rombo de la misma relación, se utilizan unos símbolos especiales para estos casos y se sustituyen todos los rombos de relación "es un tipo de" por un triángulo invertido, donde las entidades de abajo son siempre un tipo de la entidad de arriba y se llaman subtipo e entidades hijas. La de arriba se denominará supertipo o entidad padre. Las relaciones jerárquicas siempre se hacen en función de un atributo que se coloca al lado de la relación "es_un". En la figura siguiente sería "tipo". El ejemplo anterior quedaría del modo siguiente utilizando símbolos del E/R extendido.



2.4.1. Relaciones de Jerarquía ¶

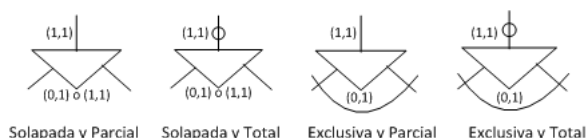
Vamos a ver los distintos tipos de relaciones de jerarquía existentes:

Total: Subdividimos la entidad Empleado en: Ingeniero, Secretario y Técnico y en nuestra BD no hay ningún otro empleado que no pertenezca a uno de estos tres tipos.

Parcial: Subdividimos la entidad Empleado en: Ingeniero, Secretario y Técnico pero en nuestra BD puede haber empleados que no pertenezcan a ninguno de estos tres tipos.

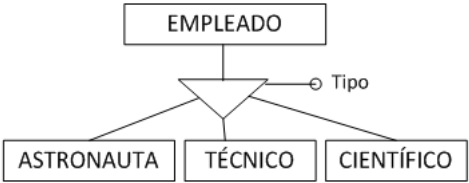
Solapada: Subdividimos la entidad Empleado, en: Ingeniero, Secretario y Técnico y en nuestra BD puede haber empleados que sean a la vez Ingenieros y secretarios, o secretarios y técnicos, etc.

Exclusiva: Subdividimos la entidad Empleado en: Ingeniero, Secretario y Técnico. En nuestra BD ningún empleado pertenece a más de una subentidad.



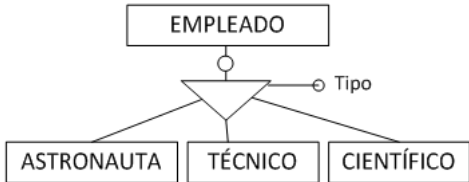
Ejemplos:

2.4.1.1. Jerarquía solapada y parcial ¶



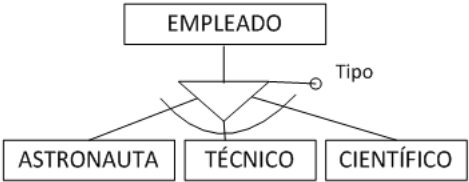
En esta BD un empleado podría ser simultáneamente técnico, científico y astronauta o técnico y astronauta, etc. (solapada). Además puede ser técnico, astronauta, científico o desempeñar otro empleo diferente (parcial).

2.4.1.2. Jerarquía solapada y total ¶



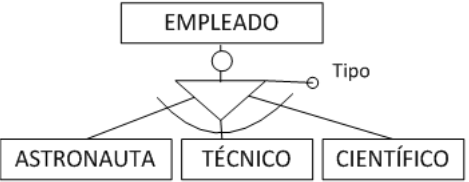
En esta BD un empleado podría ser simultáneamente técnico, científico y astronauta o técnico y astronauta, etc. (solapada). Además puede ser solamente técnico, astronauta o científico (total).

2.4.1.3. Jerarquía exclusiva y parcial ¶



En esta BD un empleado sólo puede desempeñar una de las tres ocupaciones (exclusiva) . Además puede ser técnico, o ser astronauta, o ser científico o también desempeñar otro empleo diferente, por ejemplo, podría ser FÍSICO (parcial).

2.4.1.4. Jerarquía exclusiva y total ¶



En esta BD un empleado sólo puede desempeñar una de las tres ocupaciones (exclusiva) y únicamente será astronauta, técnico o científico (total).