

DISEÑO LÓGICO RELACIONAL

Tabla de Contenidos

Bloque I

Modelo Conceptual

1.- Introducción.....	1
2.- Modelo E/R.....	1
2.1.- Entidad.....	1
2.2.- Atributos.....	2
2.2.1.- Tipos de atributos.....	2
2.2.2.- Claves.....	3
2.2.3.- Representación gráfica.....	3
2.2.4.- Dominio de un atributo.....	4
2.3.- Relaciones.....	4
2.3.1.- Grado.....	5
2.3.2.- Cardinalidad.....	5
2.3.3.- Modalidad.....	6
2.3.4.- Atributo de relación.....	8
2.3.5.- Representaciones.....	8
2.3.6.- Relaciones débiles.....	8
3.- Modelo E/R Extendido.....	9
3.1.- Restricciones en las relaciones.....	9
3.1.1.- Exclusividad.....	9
3.1.2.- Exclusión.....	10
3.1.3.- Inclusividad.....	10
3.1.4.- Inclusión.....	11
3.2.- Jerarquías de Generalización / Especialización.....	11
3.3.- Entidad asociativa y Agregación.....	14
3.3.1.- Entidad asociativa.....	14
3.3.2.- Agregación.....	15
4.- Guía para la creación del modelo E/R.....	16
4.1.- Redundancia en diagramas E/R.....	16
4.2.- Propiedades deseables de un diagrama E/R.....	17

Bloque II

Modelo Relacional

1.- Modelo Relacional.....	20
1.1.- Terminología relacional.....	20
1.2.- Restricciones en una base de datos relacional.....	22
1.2.1.- Restricciones implícitas o inherentes basadas en el modelo.....	22
1.2.2.- Restricciones semánticas o de usuario.....	23
2.- Representación del modelo relacional.....	25
2.1.- Representación textual del modelo relacional.....	25
2.2.- Representación gráfica del modelo relacional.....	26

Bloque III

Transformación del Modelo Conceptual E/R al Modelo Lógico Relacional

1.- Conversión del E/R al Relacional.....	28
1.1.- Conversiones previas.....	28
1.1.1.- Eliminación de atributos compuestos.....	28
1.1.2.- Eliminación de atributos multivaluados.....	28
1.2.- Transformación de dominios.....	29
1.3.- Transformación de entidades.....	29
1.4.- Transformación de relaciones binarias.....	30
1.4.1.- Relaciones N:M.....	30
1.4.2.- Relaciones 1:N.....	31
1.4.3.- Relaciones 1:1.....	32
1.4.4.- Relaciones débiles.....	34
1.5.- Transformación de relaciones ternarias.....	35
1.5.1.- Relaciones N:M:P.....	35
1.5.2.- Relaciones N:M:1.....	36
1.5.3.- Relaciones N:1:1.....	36
1.5.4.- Relaciones 1:1:1.....	37
1.6.- Relaciones reflexivas.....	37
1.7.- Transformación de jerarquías.....	38
1.8.- Transformación de la dimensión temporal.....	40
1.9.- Transformación de atributos derivados.....	41
2.- El modelo relacional y los SGBD.....	42

Bloque IV

Normalización del Modelo Lógico Relacional

1.- Normalización.....	44
1.1.- Dependencia funcional simple.....	45
1.2.- Dependencia funcional plena.....	45
1.3.- Primera Forma Normal.....	46
1.4.- Segunda Forma Normal.....	46
1.5.- Tercera Forma Normal.....	47

BLOQUE I

MODELO CONCEPTUAL

1.- Introducción

Cualquier desarrollo de un sistema de información parte de un proceso de análisis estructurado cuyo objetivo es identificar de forma precisa, exhaustiva y verificable las necesidades del sistema. Una vez que la abstracción conceptual del problema tome forma en papel, se procederá a implementarla como solución informatizada.

El modelo obtenido deberá contener la información más relevante de la misma para el campo de aplicación que se está estudiando así como las relaciones que se producen entre los datos. Por tanto, el objetivo es obtener una *foto* realista que sea una descripción fiel de la realidad que se quiere modelar y al mismo tiempo, lo bastante simple para poder implementarla en un sistema real de bases de datos.

Para crear el modelo conceptual emplearemos el modelo Entidad / Relación (E/R), una técnica propuesta por Peter Chen en 1976 junto con una notación bien definida para cada elemento del modelo. Aunque hoy en día existen muchas notaciones gráficas, para comenzar nuestro estudio del modelo E/R emplearemos la notación de Chen.

El primer paso será ver los diferentes elementos que tiene este modelo para representar la información.

2.- Modelo E/R

El modelo E/R es una técnica cuyo objetivo es la representación y definición de todos los datos que se introducen, almacenan, transforman y producen dentro de un sistema de información, sin tener en cuenta las necesidades de la tecnología existente ni otras restricciones.

Con esto se deduce el carácter de independencia del modelo E/R con respecto a la implementación final. Es un modelo semántico, ya que representa el significado de los datos y dará solución al problema planteado sin importar cuál sea el SGBD que se vaya a utilizar.

El modelo entidad-relación utiliza los conceptos abstractos de entidad, atributo y relación.

2.1.- Entidad

Es el elemento fundamental que hay que caracterizar. En la descripción del universo del discurso se representa mediante el uso de sustantivos.

Una entidad es cualquier objeto concreto o abstracto del cual se desea almacenar información en el sistema.

Ejemplo de entidad concreta puede ser un coche, un libro, un cliente,... y de entidad abstracta puede ser profesión, asignatura, departamento,...

Cada elemento concreto de una entidad es una **ocurrencia** o **instancia**. Así, para la entidad EMPLEADO, cada uno de los empleados es una ocurrencia de dicha entidad.

El modelo E/R distingue dos tipos de entidades:

- **Fuertes**, también llamadas propias o regulares, tienen existencia por sí mismas y sus ocurrencias no dependen de otra entidad. Corresponde a la inmensa mayoría de las entidades y se representan mediante un rectángulo con el nombre en medio.
- **Débiles**, la existencia de sus ocurrencias depende de la existencia en las ocurrencias en otra entidad. Se representan mediante un doble rectángulo.

En general, el sustantivo que representa la entidad se indica en singular y se escribe en mayúscula.

En el ejemplo tenemos como entidad fuerte un PEDIDO y como entidad débil cada una de las LÍNEAS DEL PEDIDO, pues una línea de pedido no puede existir por sí sola, depende de la existencia previa de un pedido al que pertenezca.



2.2.- Atributos

Los atributos son las propiedades o características que deseamos guardar de *una entidad* o de *una relación*. Se definen en minúscula con un nombre único que hace referencia a su contenido.

Por ejemplo, algunos de los atributos que puede tener la entidad EMPLEADO pueden ser DNI, Nombre, Domicilio, Email y Telefono.

2.2.1.- Tipos de atributos

Para matizar el valor que debe almacenar un atributo se establecen las siguientes clasificaciones:

- **Compuesto**, si contiene un valor que puede ser dividido en partes más pequeñas con significado propio. Por ejemplo, el atributo dirección se puede subdividir en calle, número, localidad, provincia y código postal.
- **Multivaluado**, cuando para la misma instancia el atributo puede tomar varios valores. Por ejemplo, si se desea almacenar más de un teléfono para cualquier instancia de la entidad EMPLEADO.
- **Opcional**, si se permite que pueda haber instancias de la entidad en las que el atributo no esté definido o no tenga valor.
- **Derivado**, cuando el valor del atributo se puede obtener a partir del valor o valores de otros atributos relacionados. Por ejemplo, la edad de una persona se puede calcular a partir de la fecha de nacimiento y la fecha del sistema.
- **Clave**, si el valor es único para cada ocurrencia de la entidad y por tanto permite identificar de manera única los ejemplares de la entidad. Un atributo clave no puede ser nulo.

2.2.2.- Claves

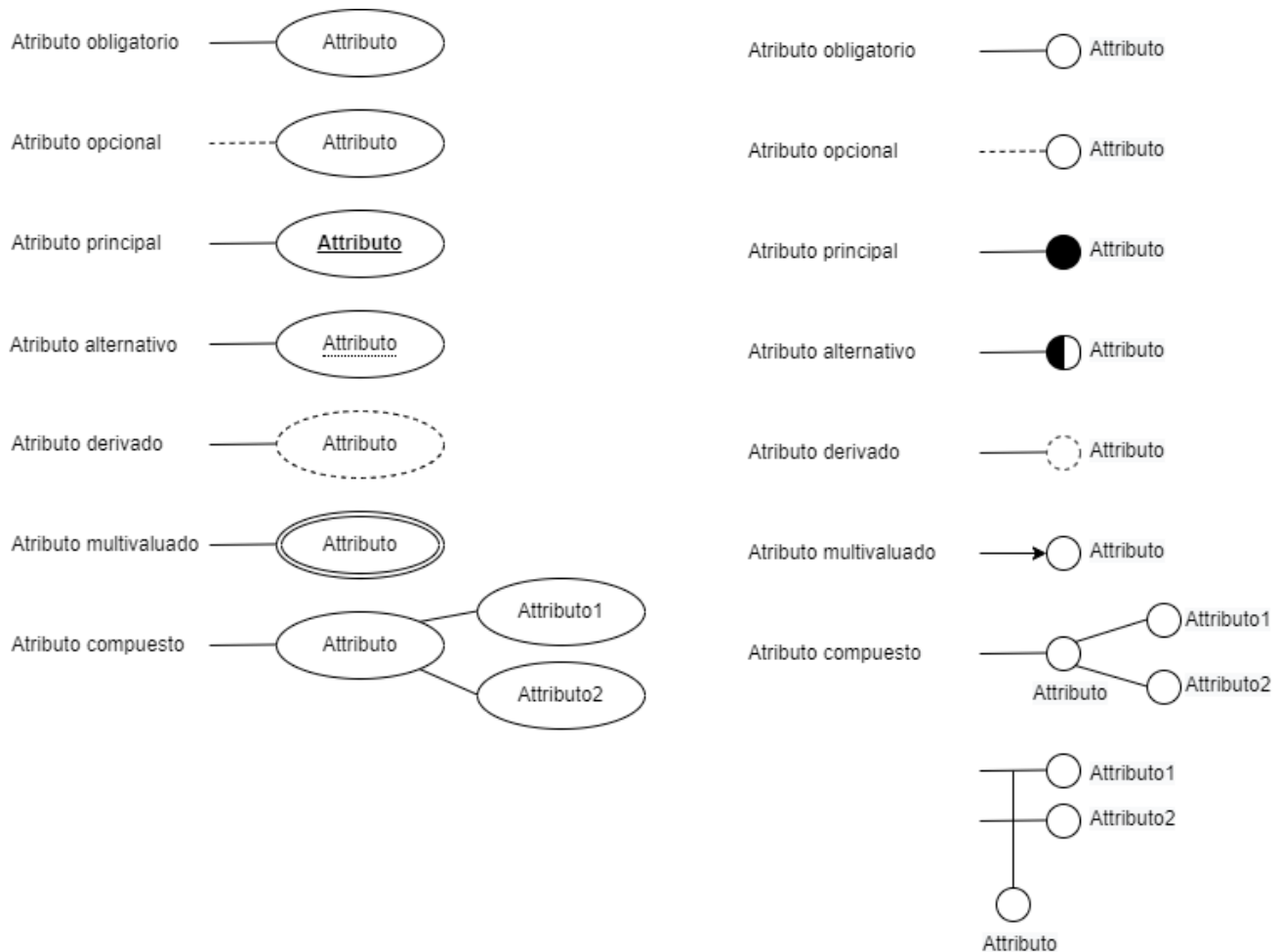
En toda entidad se debe identificar unívocamente cualquier instancia de la misma.

Si la entidad es fuerte, se determina qué atributo lo consigue por sí mismo o si es necesario alguna combinación de éstos. A todos los resultados obtenidos se les denomina **claves candidatas** y de entre éstas se debe seleccionar una, generalmente atendiendo al contexto en el que se esté trabajando, como **clave principal**. Las claves candidatas que no han sido seleccionadas se conocen como **claves alternativas**.

Cuando la entidad es débil y no dispone de atributos para identificar unívocamente sus instancias, su clave se forma con la clave principal de la entidad fuerte de la que depende más la **clave parcial (discriminante)** de la entidad débil.

2.2.3.- Representación gráfica

La representación varía según la notación elegida.



Las restricciones de los atributos pueden combinarse entre sí.

Es fácil intuir que realizar un modelo que requiera cierta complejidad utilizando la representación gráfica anterior para describir los atributos de las relaciones será complicado de dibujar. Afortunadamente, existen otras notaciones que arrojan mayor claridad y que se asoman al diseño de bases de datos orientadas a objeto.

Por lo general, estas representaciones incorporan en un mismo rectángulo a la entidad (cuyo nombre se indica en la parte superior) y sus atributos. Las restricciones de los atributos se representan mediante algún símbolo que lo acompaña o alguna abreviatura.



2.2.4.- Dominio de un atributo

El dominio de un atributo es el conjunto de valores permitidos para un atributo en particular y hace referencia al tipo de datos con el que se almacenará (cadena de caracteres, número entero, fecha,...) o a restricciones en los valores que el atributo puede tomar. En el primer caso el dominio queda definido de forma implícita y en el segundo de forma explícita. Por ejemplo, el atributo *dni* se define implícitamente indicando que puede tomar 9 dígitos numéricos mientras que el atributo *localidad* se definiría en el conjunto explícito Málaga, Granada, Jaén, Huelva, Cádiz, Sevilla, Córdoba, Almería.

Cualquier atributo de una entidad quedará definido por el par (atributo, valor). Sin embargo, los dominios no se suelen representar en el modelo por problemas de espacio y se añaden como descripción textual adicional.

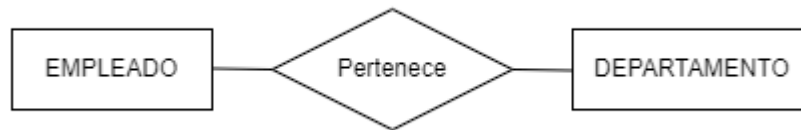
2.3.- Relaciones

También llamadas *interrelaciones*, una relación representa una asociación o correspondencia entre entidades exigiendo que compartan ciertos atributos de forma indispensable.

Sin las interrelaciones las bases de datos carecen de sentido. Es el mecanismo empleado para cambiar información y obtener resultados de dicha información. Establecen el modo en que las ocurrencias de una entidad se relacionan con las ocurrencias de otra u otras entidades.

En la notación de Chen, las relaciones se representan mediante un rombo. En su interior se escribe el nombre de la relación que debe ser un verbo o una acción verbal.

Veamos un ejemplo que aparece en la gran mayoría de las empresas en las que existen varios departamentos y cada uno de los empleados pertenece a un determinado departamento. Tendríamos dos entidades, por un lado EMPLEADO y por otro lado DEPARTAMENTO. La asociación que utilizamos en este caso es Pertenece.



2.3.1.- Grado

Se denomina **Grado** de una relación al número de entidades que participan en esa relación.

Si una relación asocia instancias de una entidad con ella misma, la relación se dice que es **reflexiva** o en **anillo**. Para comprenderlas correctamente hay que tener en cuenta que las ocurrencias que se relacionan, aún perteneciendo a la misma entidad, juegan papeles distintos.

Si en una relación intervienen dos entidades la relación se denomina **binaria**. Entre dos entidades pueden existir tantas relaciones binarias como se consideren oportunas. Es el tipo de relación más común.

Si participan tres entidades, la relación se denomina **ternaria**. En estos casos es necesario estudiar cómo se relacionan cada par de entidades con la tercera entidad. Este tipo de interrelaciones añaden complejidad al diseño conceptual, por eso es necesario abordar si se puede transformar en dos relaciones binarias sin perder la semántica.

Es raro y poco deseable encontrar en el diseño relaciones de grado mayor a las presentadas. En cualquier caso a este tipo de relación se le conoce como **n-aria**.

2.3.2.- Cardinalidad

La cardinalidad de una relación es el **número máximo** de veces que una ocurrencia de una entidad puede asociarse con ocurrencias de la otra entidad dentro de una relación determinada.

Hay que analizar ambos sentidos de la relación. En función del número de ocurrencias asociadas se distinguen tres tipos de relaciones:

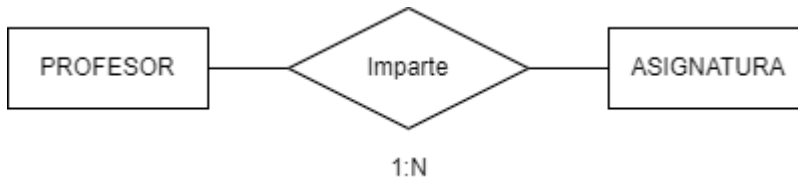
- **Relación 1:1**

Sean A y B dos entidades relacionadas con cardinalidad 1:1. Esto indica que una ocurrencia de la entidad A solo puede relacionarse con una ocurrencia de la entidad B. Y en el sentido contrario, esa misma ocurrencia de B solo se corresponde con dicha ocurrencia de A.



- **Relación 1:N**

Sean A y B dos entidades relacionadas con cardinalidad 1:N. Esto indica que una ocurrencia de la entidad A puede relacionarse con varias ocurrencias de la entidad B. Pero en el sentido contrario, cada ocurrencia de B solo se corresponde con dicha ocurrencia de A.



- **Relación M:N**

Sean A y B dos entidades relacionadas con cardinalidad M:N. Esto indica que una ocurrencia de la entidad A puede relacionarse con varias ocurrencias de la entidad B. Y en el sentido contrario, cada una de esas ocurrencias de B también se corresponden con varias ocurrencias de A.



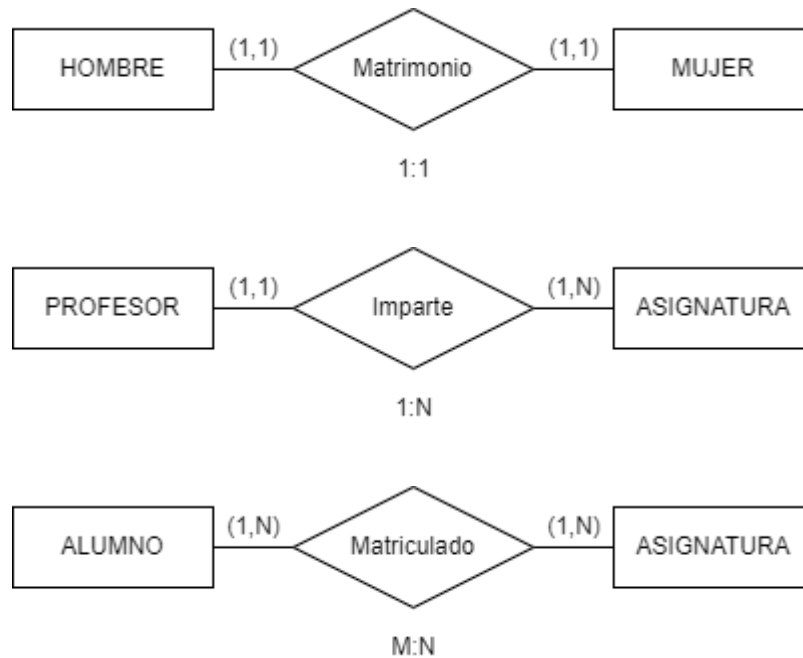
2.3.3.- Modalidad

Define el número mínimo y máximo de ocurrencias de una entidad que pueden estar relacionadas con una ocurrencia de otra u otras entidades, indicando relaciones optativas. En la representación gráfica, la modalidad se indica a ambos lados de la relación, y su valor máximo coincide con el valor de la cardinalidad correspondiente al lado de la relación en la que se encuentra.

Se dan los siguientes tipos de modalidades:

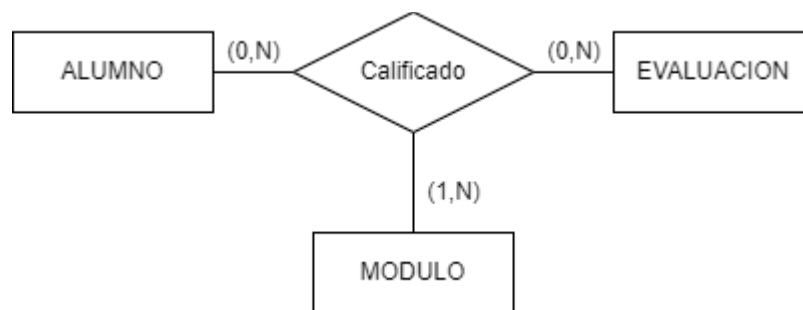
- **(0,1)** Cada ocurrencia de la primera entidad puede relacionarse con una o ninguna de la segunda entidad.
- **(1,1)** Cada ocurrencia de la primera entidad debe relacionarse obligatoriamente con una y solo una de la segunda entidad.
- **(1,N)** Cada ocurrencia de la primera entidad debe relacionarse con al menos una ocurrencia de la segunda entidad, pero también puede relacionarse con varias.
- **(0,N)** Cada ocurrencia de la primera entidad no tiene limitada su relación con ocurrencias de la segunda entidad. Puede relacionarse con una, varias o ninguna.

Para las relaciones vistas en el apartado anterior, las modalidades podrían ser:



Nótese que, una vez que se utiliza el concepto de modalidad, no es necesario indicar la cardinalidad, ya que esta se puede deducir a partir de aquella tomando los valores máximos a cada lado de la relación.

En las relaciones ternarias y n-arias la modalidad se lee de forma distinta porque el valor que se indica junto a la entidad representa *su* modalidad con respecto al resto de entidades que participan en la relación.

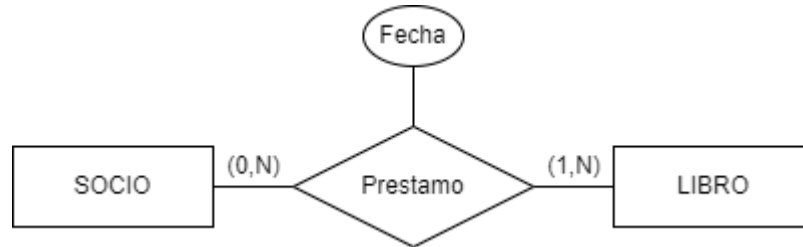


En el ejemplo, las modalidades responden a las siguientes preguntas:

- ✓ Para un alumno y módulo determinados, ¿en cuántas evaluaciones es calificado?
- ✓ Para una alumno y evaluación determinados, ¿en cuántos módulos es calificado?
- ✓ Para una evaluación y módulo determinados, ¿cuántos alumnos son calificados?

2.3.4.- Atributo de relación

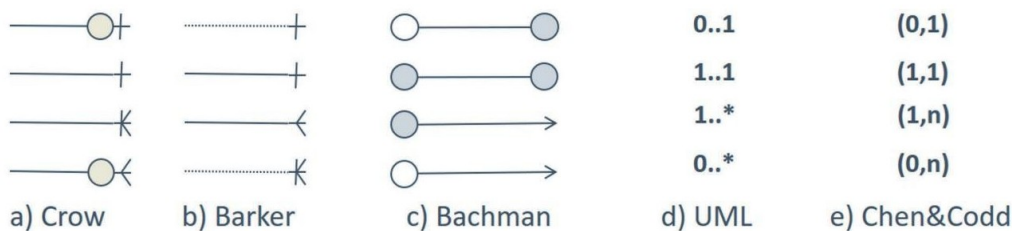
Es muy común que una relación tenga atributos en ella. Estos atributos se originan solo cuando las instancias se relacionan y no son propias de ninguno de los tipos de entidad que participan en la relación.



2.3.5.- Representaciones

Para definir las restricciones de las interrelaciones se pueden utilizar diferentes nomenclaturas.

Existen varias notaciones para realizar un diagrama E/R: Chen, Barker, IDE1X, Crow, UML, notación de flechas,... Una de las más utilizadas es la notación Crow, también conocida como *pata de gallos*, que usa símbolos en los extremos de las relaciones para representar la modalidad.



El siguiente enlace puede servirte como punto de partida para profundizar sobre ellos y decantar tu elección: <https://www.vertabelo.com/blog/comparison-of-erd-notations/>

2.3.6.- Relaciones débiles

Tienen lugar cuando una entidad débil se asocia con una entidad fuerte o regular de la que depende. Siempre son de grado dos y la modalidad de la entidad débil hacia la fuerte es siempre (1,1).

Se distinguen dos tipos de dependencia:

- **Existencia**

La entidad débil existe porque existe la entidad fuerte. Si esta desaparece, la débil también.

- **Identificación**

Además de la dependencia en existencia, la entidad débil necesita de la clave primaria de la entidad fuerte para identificar sus propias instancias.

Para indicar que la relación es por existencia, en el borde del rombo de la interrelación se escribe EX y si lo es por identificación la etiqueta es ID.

3.- Modelo E/R Extendido

El modelo original planteado por Peter Chen ha sido enriquecido con nuevas representaciones que ofrecen un mayor grado de abstracción para poder representar una semántica más amplia del universo del discurso. Este modelo se conoce como E/R Extendido.

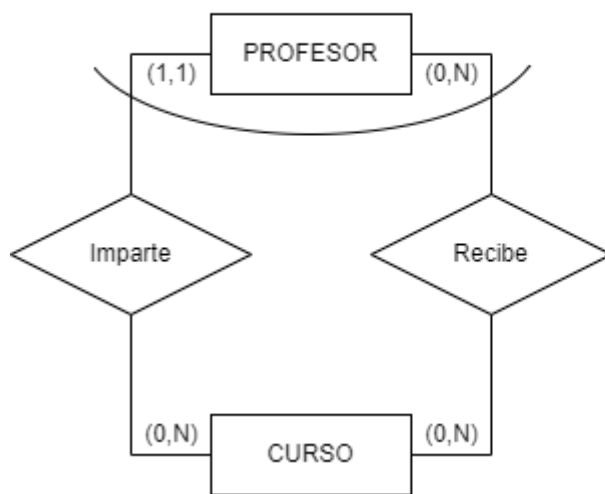
3.1.- Restricciones en las relaciones

3.1.1.- Exclusividad

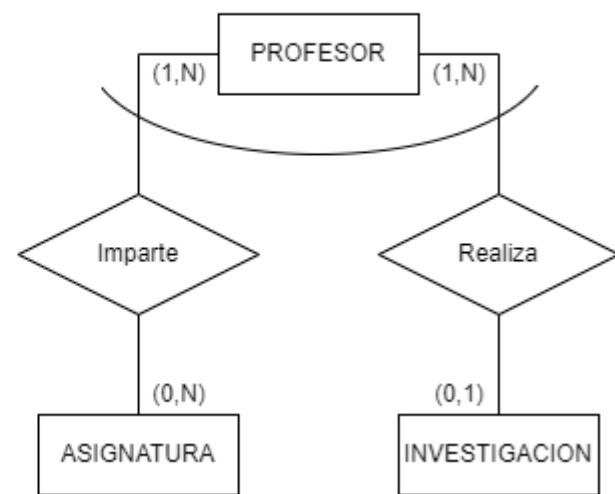
Dos o más tipos de relaciones tienen restricción de exclusividad con respecto a un tipo de entidad que participa en ellas cuando cada ocurrencia de dicha entidad sólo puede pertenecer a uno de los tipos de relaciones, pero en el momento en que pertenezca a uno ya no podrá formar parte del otro.

No es obligatorio que las relaciones exclusivas lo sean respecto al mismo tipo de entidad, sino que pueden serlo respecto a distintos tipos.

Se representa con un arco que engloba a todas aquellas relaciones que son exclusivas.



Un profesor o imparte o recibe un curso

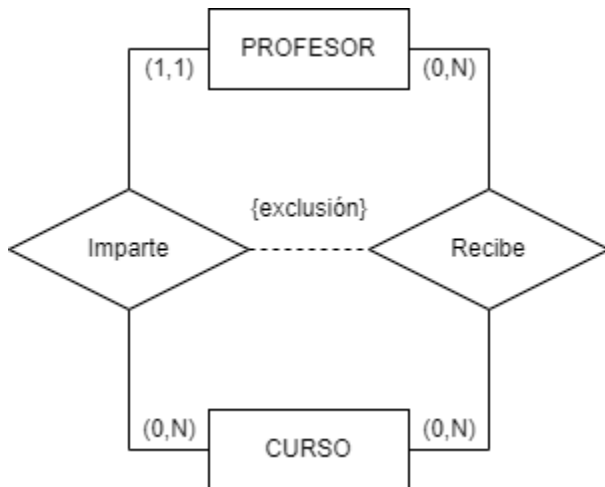


Un profesor o imparte una asignatura o realiza un trabajo de investigación

3.1.2.- Exclusión

Se produce cuando las ocurrencias de las entidades que se relacionan no pueden asociarse simultáneamente sino que deben asociarse utilizando una sola de las relaciones que las une.

Se representa mediante una línea discontinua.

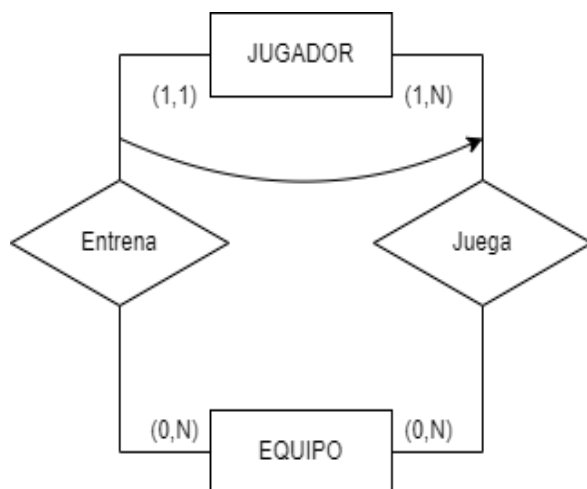


Un profesor que imparte un determinado curso no puede al mismo tiempo recibirlo, pero puede participar en otro que él no imparta.

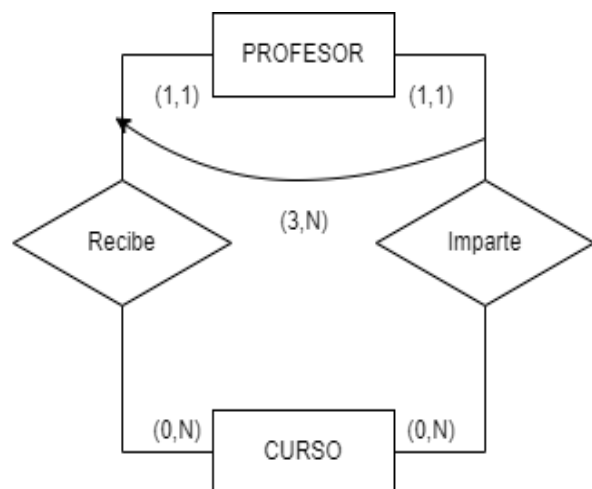
3.1.3.- Inclusividad

Sea una entidad que se relaciona con otra a través de dos relaciones. Para toda ocurrencia de la entidad que participe en una de las relaciones tiene obligatoriamente que participar también en la otra.

Se representa mediante una flecha arqueada que parte desde la relación que ha de cumplirse primero hacia la otra relación.



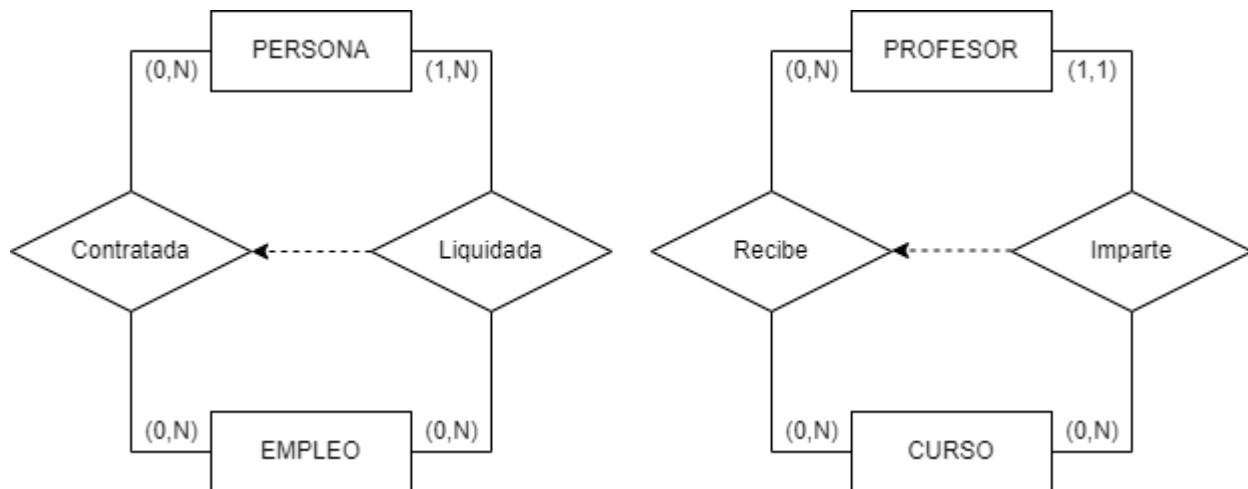
Un jugador, para entrenar un equipo primero debe haber jugado en algún equipo.



Un profesor tiene que haber recibido al menos tres cursos para poder impartir él alguno.

3.1.4.- Inclusión

En este caso, cuando las ocurrencias que se asocian mediante una de las relaciones tienen también que asociarse mediante la otra. Se representa mediante una flecha recta de línea discontinua.



Para liquidar a una persona de un empleo previamente ha debido estar contratada en dicho empleo

Un profesor puede impartir un curso si previamente ha recibido dicho curso.

3.2.- Jerarquías de Generalización / Especialización

Son un caso especial de relación entre un tipo de entidad más general llamado **supertipo** y un conjunto de varios tipos de entidades que se denominan **subtipos**. Por su propia naturaleza este tipo de relación presenta una característica muy importante que es la **herencia**, ya que toda propiedad del supertipo (atributos, identificadores, participaciones) pasa a serlo también de los subtipos. La herencia nunca se da de los subtipos al supertipo.

La aparición de estas jerarquías en el modelo puede surgir de dos formas distintas:

- **Generalización**

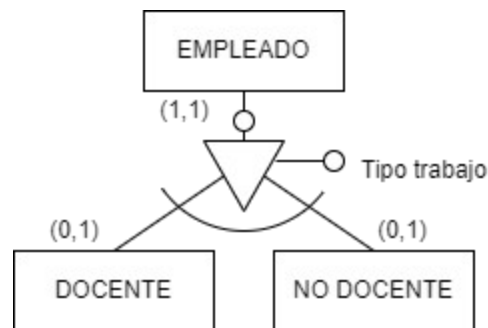
Se observa que dos o más tipos de entidad comparten propiedades (atributos y/o relaciones) de donde se deduce la existencia de un tipo de entidad de nivel superior (supertipo) que contiene los atributos y los tipos de relaciones comunes a todos los subtipos.

- **Especialización**

Se observa que un tipo de entidad tiene ciertas propiedades (atributos y/o relaciones) que tienen sentido para unos ejemplares pero no para otros, por lo que es conveniente definir uno o varios subtipos que contengan estas propiedades específicas, dejando en el supertipo los que son comunes.

La división en subtipos puede estar condicionada por los valores que tome un atributo (**atributo discriminante**) definido previamente.

Se representa mediante un triángulo invertido, sobre él quedará la entidad superclase y conectadas a él a través de líneas rectas, las subclases. Si existe atributo discriminante se conectará con el triángulo que representa la jerarquía. No es necesario incluir el nombre de la relación ya que por defecto responde a la noción de “es un tipo de” (*is a*).



Como se puede observar, si nos movemos de los subtipos hacia el supertipo se trata de una generalización, mientras que si primero identificamos el supertipo y a partir de él llegamos a los subtipos se trata de una especialización. Puede ocurrir que se formen jerarquías de más de un nivel, donde un subtipo es a su vez supertipo de otras entidades.

La generalización tiene una serie de restricciones semánticas atendiendo a si los subtipos se solapan o son disjuntos y a si la unión de los subtipos recubre o no al supertipo:

- **Solapamiento / Exclusividad**

Si un mismo ejemplar del supertipo puede pertenecer o no a más de un subtipo habrá solapamiento, y si sólo puede pertenecer a uno de los subtipos existirá exclusividad.

La exclusividad se representa mediante un arco debajo del triángulo.

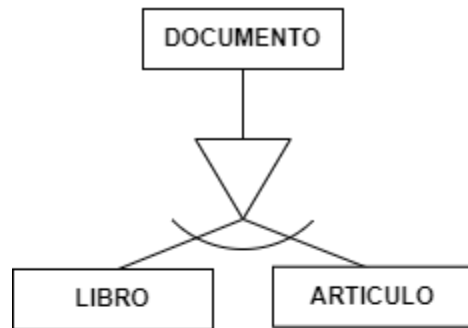
- **Totalidad / Parcialidad**

Si todo ejemplar del supertipo tiene que pertenecer a algún subtipo tendremos totalidad y, si por el contrario, no tiene obligatoriamente que pertenecer a algún subtipo habrá parcialidad.

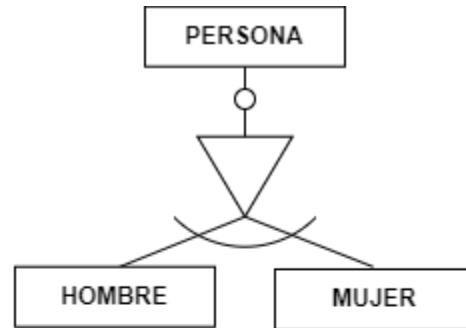
La totalidad se representa mediante un pequeño círculo antes del triángulo.

De estas características se deduce que la modalidad del supertipo será siempre (1,1) y la de cada uno de los subtipos (0,1).

La combinación de estas posibilidades da lugar a cuatro tipos de jerarquías:



Exclusiva / Parcial



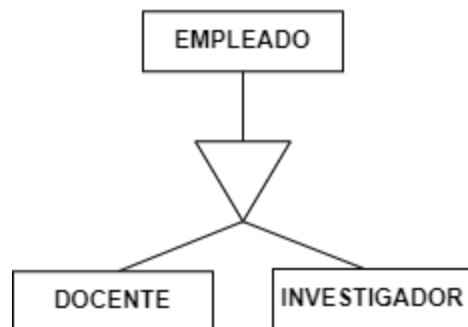
Exclusiva / Total

- **Exclusiva / Parcial**

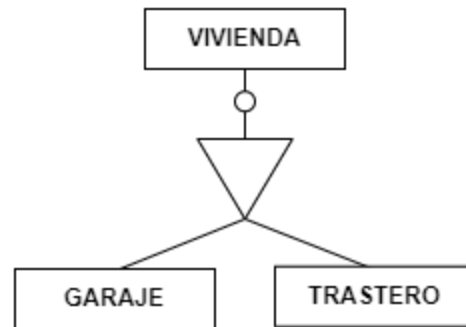
Un documento no es obligatoriamente libro o artículo (podría ser, por ejemplo, una revista), pero en caso de ser alguno de ellos lo es de solo uno.

- **Exclusiva / Total**

Una persona debe ser obligatoriamente hombre o mujer, pero no ambos a la vez.



Solapada / Parcial



Solapada / Total

- **Solapada / Parcial**

Un empleado no es obligatoriamente docente o investigador (podría ser, por ejemplo, becario), y en caso de ser alguno puede ser ambos a la vez.

- **Solapada / Total**

Una vivienda o tiene garaje, o tiene trastero o tiene garaje y trastero.

3.3.- Entidad asociativa y Agregación

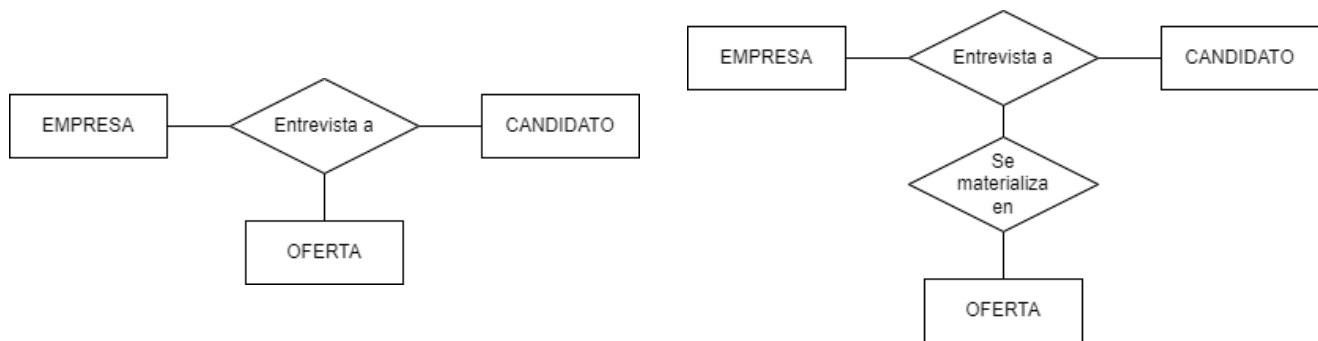
Una de las restricciones inherentes del modelo E/R es que no puede haber relaciones entre relaciones o entre una entidad y otra relación. Para solucionar este problema aparecen los conceptos de agregación y entidad asociativa.

3.3.1.- Entidad asociativa

Es una abstracción que permite crear una entidad que provenga de una relación entre varias entidades para que de este modo esta entidad creada pueda asociarse con otras entidades existentes.

Veamos este concepto mediante un ejemplo.

Supongamos una empresa de selección de personal que realiza entrevistas a diferentes candidatos. Puede ser que para algunas de algunas de estas entrevistas se derive una oferta de empleo.

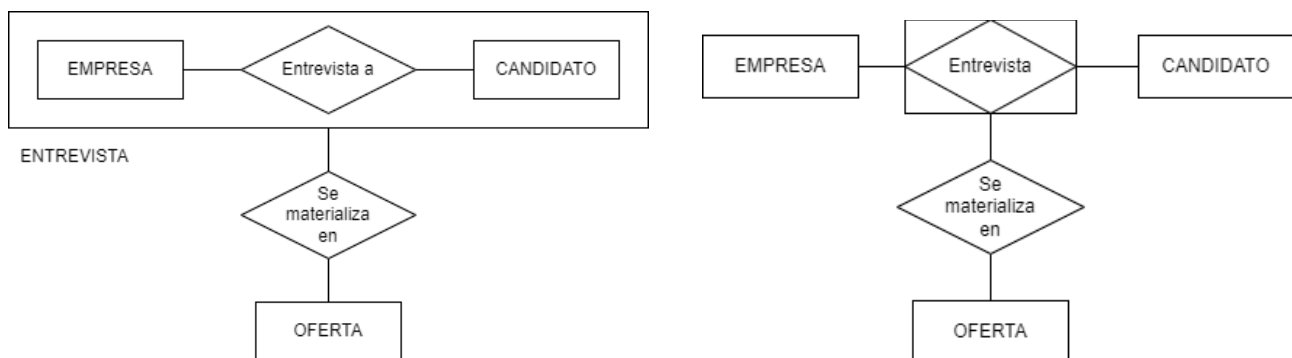


Esta solución no es correcta ya que representa que por cada entrevista se genera una oferta de empleo.

Esta solución tampoco es correcta porque en el modelo E/R no se pueden relacionar relaciones entre sí.

La solución pasa por crear la entidad asociativa ENTREVISTA compuesta por la relación entre EMPRESA y CANDIDATO que sí puede relacionarse con la entidad OFERTA.

Se puede encontrar representado de estas dos formas, aunque es más correcta la de la derecha:



3.3.2.- Agregación

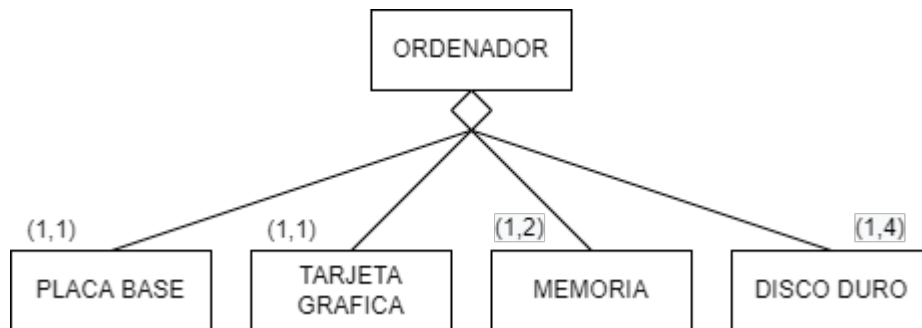
En este caso con la abstracción se busca distinguir un todo de una parte.

Al tipo compuesto nos referimos como el *todo* mientras que los componentes son las *partes*.

Existen dos tipos de agregaciones:

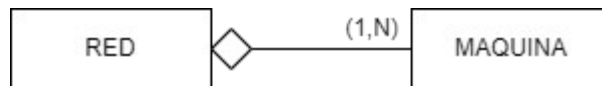
- **Composición**

Un todo se obtiene por la unión de diversas partes, que pueden ser entidades distintas y que desempeñan papeles distintos en la agregación.



- **Colección**

Un todo se obtiene por la unión de diversas partes, donde todas las partes son de un mismo tipo de entidad y desempeñan el mismo papel en la agregación.



En cuanto a las cardinalidades mínima y máxima de las entidades agregadas siempre son (1,1) y no se suelen indicar en la representación gráfica.

La agregación se indica mediante un pequeño rombo junto a la entidad que representa *el todo*.

4.- Guía para la creación del modelo E/R

A partir de los requisitos de usuario, realizaremos varias lecturas para identificar los distintos elementos del modelo de menor a mayor dificultad:

1. Entidades, teniendo que especificar los atributos que las caracterizan y los que las identifican. También seleccionaremos los atributos multivaluados si los hubiera.
2. Relaciones que vinculan a varias entidades o a una entidad más de una vez. Estableceremos las cardinalidades (min,max) de cada entidad respecto de las relaciones en las que participa.
3. Posibles atributos de relaciones.
4. Posibles jerarquías, distinguiendo las características comunes del supertipo de las específicas de los subtipos.
5. Posibles entidades débiles y si son en existencia o en identificación.
6. Posibles restricciones entre distintas relaciones (inclusión, exclusión, inclusividad o exclusividad).
7. Posible relaciones de grado superior a dos. Comprobaremos si reflejan adecuadamente la semántica de los requisitos de usuario y pueden transformarse en relaciones binarias o en entidades asociativas.
8. Posible relaciones redundantes, eliminándolas según el caso. Si no se eliminan habrá que crear mecanismos o procedimientos para controlar dicha redundancia.

4.1.- Redundancia en diagramas E/R

Una de las principales razones por las que las bases de datos aparecieron fue la eliminación de la redundancia en los datos.

La redundancia de datos puede provocar problemas como:

- Aumento de la carga de trabajo, ya que al estar un mismo dato almacenado en varios lugares, las operaciones de grabación o actualización deberá realizarse en varias ocasiones, tantas como veces esté duplicado el dato.
- Gasto extra de espacio de almacenamiento, que será más patente a medida que la base de datos aumente.
- Inconsistencia, si los datos repetidos no contienen los mismos valores, no pudiéndose saber cuál de ellos es el correcto.

Para que una base de datos funcione de forma óptima es fundamental realizar un buen diseño de ella, por tanto, es imprescindible que en el modelo E/R se controle la redundancia.

Existen dos formas principales de redundancia:

- En los **atributos** (derivados)

Aunque son redundantes, no dan lugar a inconsistencias siempre que en el esquema se indique su condición de derivados y la fórmula mediante la que han de ser calculados.

En determinadas ocasiones es conveniente introducir redundancia controlada en los atributos para permitir un buen rendimiento de la base datos.

- En las **relaciones**

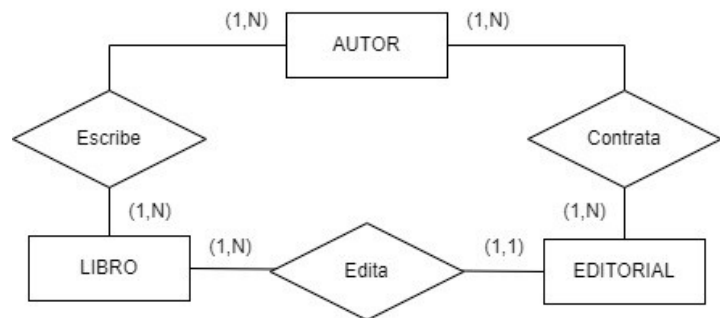
Una relación es redundante si su eliminación no implica pérdida de semántica porque existe la posibilidad de obtener la misma información por medio de otras relaciones.

Para que una relación sea redundante es condición necesaria pero no suficiente que forme parte de un ciclo.

El siguiente ejemplo muestra la redundancia presente en un ciclo:

Si se conocen los libros escritos por un autor se puede saber qué editoriales ha contratado a ese autor.

Si se conocen qué libros ha editado una determinada editorial se puede saber qué autores han escrito para esa editorial (y por tanto han sido contratados)



En este caso, la relación Contrata es redundante.

4.2.- Propiedades deseables de un diagrama E/R

Es deseable que el diagrama E/R cumpla con los siguientes requisitos:

- **Completitud**

Será completo si es posible verificar que cada uno de los requerimientos está representado en el diagrama y viceversa, cada representación del diagrama tiene su equivalente en los requerimientos.

- **Corrección**

Será correcto si emplea de manera adecuada todos los elementos del modelo E/R. Esto se puede analizar desde dos puntos de vista:

- Corrección **sintáctica**, si no se producen representaciones erróneas en el diagrama.
- Corrección **semántica**, si las representaciones significan exactamente lo que está estipulado en los requerimientos. Posibles errores semánticos son, por ejemplo, la

utilización de un atributo en lugar de una entidad, utilizar una entidad en lugar de una relación, usar el mismo nombre para dos entidades o dos relaciones, indicar erróneamente alguna modalidad / cardinalidad u omitirla, etc.

- **Minimalidad**

Será mínimo si se puede verificar que al eliminar algún concepto presente en el diagrama se pierde información. Por tanto, si tiene redundancia, no es mínimo.

- **Sencillez**

Si representa los requerimiento de manera fácil de comprender, sin artificios complejos.

- **Legibilidad**

Si se puede interpretar fácilmente, y esto dependerá en gran medida del modo en que se disponen los diferentes elementos e interconexiones.

- **Escalabilidad**

Si es capaz de incorporar posibles cambios derivados de nuevos requerimientos.

BLOQUE II

MODELO RELACIONAL

1.- Modelo Relacional

Con el modelo Entidad / Relación se ha obtenido una representación conceptual de la información que se desea manejar (universo del discurso) independientemente de cuál sea el sistema gestor en el que se implemente. En este bloque se dará el siguiente paso, que es convertir el modelo conceptual en un modelo lógico ya adaptado a un tipo concreto de SGBD. En este caso el tipo de SGBD que se va a utilizar es el relacional, presentado por Codd en 1970 y que sigue vigente en la actualidad.

Hasta ese momento los SGBD imperantes del mercado estaban basados en los modelos de red y jerárquico, que no habían superado el grave inconveniente que suponía la dependencia de las aplicaciones desarrolladas en ellos respecto a las estructuras de datos.

El modelo relacional está basado en lo que se conoce como **lógica de predicado** y la **teoría de conjunto**. El único constructor del modelo relacional es la **relación**, elemento al que hay que transformar cualquier objeto (entidad o interrelación) obtenido en el esquema conceptual. A la relación también se le denomina **tabla**, aunque ésta se la representación gráfica de aquella.

Una vez obtenido el modelo lógico, se podrá obtener de él el modelo físico de un SGBD concreto que ya se podrá utilizar para su explotación.

1.1.- Terminología relacional

Una base de datos relacional está compuesta por un conjunto de relaciones, de ahí su nombre.

El modelo relacional se ocupa de la estructura, de la integridad y de la manipulación de los datos. Cada una de estas partes tiene sus propios términos especiales. A continuación se exponen los términos relacionados con la estructura de datos.

- Una **relación** es un conjunto de tuplas con la misma estructura. Sería algo similar a una tabla con información.
- Una **tupla** es un conjunto de pares atributo-valor. Se podría asemejar a una fila de una tabla (relación). El número de tuplas de una relación se denomina **cardinalidad** de esa relación y al número de atributos se le llama **grado**.
- Un **atributo** es un dato simple con un nombre, que debe ser único dentro de una misma relación, aunque pueden existir atributos con el mismo nombre en relaciones distintas. El valor de un atributo es el dato que hay actualmente almacenado en dicho atributo. Se podría considerar como una celda o casilla de una fila de la tabla.
- Un **dominio** es un conjunto de valores. Todo atributo tiene asociado un dominio que indica cuales son los valores válidos que puede contener dicho atributo. En algunos casos a los dominios se les puede añadir el valor especial NULO. Un valor NULO en un atributo significa que a ese atributo no se le ha asignado ningún valor todavía. Si a un dominio **no** se le añade el valor NULO, esto significa que no pueden dejarse valores vacíos en los atributos que pertenezcan a dicho dominio.

De manera formal, se define una relación R sobre un conjunto de dominios D_1, D_2, \dots, D_n como una estructura con dos partes:

- **Cabecera**

La cabecera de una relación es un conjunto de pares atributo-dominio $\{(A_1, D_1), (A_2, D_2), \dots, (A_n, D_n)\}$.

La cabecera indica qué atributos tiene la relación (cómo se llama cada uno) y el dominio del que cada atributo puede obtener sus valores. La cabecera de una relación es estática, esto es, no varía con el tiempo. También se denomina *esquema de la relación*.

- **Cuerpo**

El cuerpo de una relación es un conjunto de tuplas con pares atributo-valor de la forma: $\{(A_1, V_1), (A_2, V_2), \dots, (A_n, V_n)\}$.

El cuerpo contiene la información almacenada en la relación (la información almacenada en la tabla). El número de tuplas es variable a lo largo del tiempo, según se añada o elimine información de la relación.

EJEMPLO

PROVEEDOR

CODIGO	NOMBRE	CIUDAD
1	Salazar	Londres
2	Jaime	París
3	Bernal	París
4	Corona	Londres
5	Aldana	Atenas

Esta tabla es la relación PROVEEDOR.

Las tuplas son cada una de estas filas

- 1, Salazar, Londres
- 2, Jaime, París
- 3, Bernal, París
- 4, Corona, Londres
- 5, Aldana, Atenas

La cardinalidad es 5 y el grado es 3.

La cabecera sería: $\{(CODIGO, NATURALES), (NOMBRE, TEXTO50), (CIUDAD, CIUDADES)\}$

El dominio NATURALES es el conjunto de los números naturales, el dominio CIUDADES es el conjunto de nombres de ciudades de los proveedores y el dominio TEXTO50 es el conjunto de todas las posibles combinaciones de letras de hasta 50 caracteres de longitud.

Y el cuerpo:

$\{(CODIGO, 1), (NOMBRE, Salazar), (CIUDAD, Londres)\}$
 $\{(CODIGO, 2), (NOMBRE, Jaime), (CIUDAD, París)\}$
 $\{(CODIGO, 3), (NOMBRE, Bernal), (CIUDAD, París)\}$
 $\{(CODIGO, 4), (NOMBRE, Corona), (CIUDAD, Londres)\}$
 $\{(CODIGO, 5), (NOMBRE, Aldana), (CIUDAD, Atenas)\}$

Por tanto:

- Las relaciones representan en el diseño lógico lo que las entidades en el diseño conceptual.
- Una relación tiene un nombre, una serie de atributos y un conjunto de tuplas. El nombre debe ser único, es decir, no puede haber dos relaciones con el mismo nombre en la misma base de datos.
- Los atributos representan las propiedades inherentes de las relaciones.
- Las tuplas representan los valores que toman los diferentes atributos para cada elemento de la relación.

Por otra parte:

- Las relaciones se representan mediante tablas bidimensionales.
- Las columnas de una tabla corresponden con los atributos de la relación.
- Las filas de una tabla se corresponden con las tuplas de la relación.
- Las tuplas coinciden con las ocurrencias de una entidad o de una interrelación del modelo E/R.

Se podrían establecer de forma informal las siguientes equivalencias:

- Relación Tabla
- Tupla Fila o registro
- Cardinalidad Número de filas
- Atributo Columna o campo
- Grado Número de columnas
- Dominio Conjunto de valores permitidos

1.2.- Restricciones en una base de datos relacional

1.2.1.- Restricciones implícitas o inherentes basadas en el modelo

Son las restricciones derivadas de la propia naturaleza del modelo, que no tienen que ser definidas por el usuario e imponen limitaciones a la hora de modelar nuestro universo del discurso.

La base matemática de la teoría de conjuntos en la que se apoya el modelo junto al único constructor que define, la relación, implica que:

- 1) No se permite la existencia de tuplas repetidas (un conjunto no puede tener elementos iguales) lo que obliga a la existencia de una clave primaria o identificador (conjunto mínimo de atributos que identifican de forma unívoca las tuplas de una relación).
- 2) Ningún atributo que forme parte de la clave primaria puede tomar valores nulos, caso de no ser así no se podría identificar unívocamente las tuplas de la relación. A esto se conoce como **integridad de entidad**.

- 3) Las relaciones son tablas de dos dimensiones, por lo que no se permiten grupos repetitivos en los atributos, es decir, que cada atributo sólo puede tomar un valor del dominio.
- 4) El orden de las tuplas y de los atributos es irrelevante dentro de la relación.
- 5) Dentro de una relación no pueden existir atributos con el mismo nombre.
- 6) Cada relación contiene un solo tipo de registro con un número fijo de campos.

1.2.2.- Restricciones semánticas o de usuario

Son restricciones que tiene que definir el diseñador a fin de que el esquema sea un reflejo lo más fiel posible del universo del discurso.

A) Restricción de unicidad (UNIQUE)

Los valores de uno o varios atributos no pueden repetirse en las distintas tuplas de la relación.

B) Restricción de obligatoriedad (NOT NULL)

Indica que el atributo debe tomar siempre un valor y no admite valores nulos (desconocidos o inexistentes).

C) Restricción de clave primaria (PRIMARY KEY)

Se denomina clave candidata al conjunto de atributos mínimo capaz de identificar unívocamente cualquier tupla de la relación. Si existe más de una clave candidata, a la elegida se le denomina clave primaria (PRIMARY KEY) y al resto que no lo ha sido se les denomina claves alternativas. En cualquier caso, a todas se les aplican las restricciones de unicidad y obligatoriedad.

En la práctica se elige como clave primaria la que se considera de más importancia. Como regla general se debe preferir las claves candidatas que tengan un menor número de atributos, idealmente sólo uno. De entre estos se debe elegir el que se considere que es más significativo como identificador de cada tupla.

D) Restricción de clave ajena (FOREIGN KEY)

Se utiliza para indicar que un conjunto de atributos de la relación es clave primaria en otra o la misma relación. De esta forma podemos enlazar relaciones entre sí. Los atributos que son clave ajena en una relación no necesitan tener los mismos nombres que los atributos clave de la relación primaria con la que se corresponden pero sí deben estar definidos necesariamente sobre el mismo dominio.

Las claves ajenas son el mecanismo que se utiliza en el modelo relacional para crear *asociaciones* entre datos contenidos en distintas tablas. Al incluir una clave ajena en una relación se está creando un vínculo entre cada tupla de la relación y una tupla en la relación objetivo. El tipo de asociación concreto no queda recogido en el modelo relacional, por lo que lo tiene que conocer el diseñador.

La clave ajena no necesita ser un componente de la clave primaria de la relación que la contiene.

E) Restricción de integridad referencial

Está vinculada directamente a la restricción de clave ajena. Indica que los valores de la clave ajena (relación hija o que referencia) se han de corresponder con los valores de la clave primaria (relación padre o referenciada) o bien ser nulos. Las relaciones padre e hija no han de ser necesariamente distintas.

El modelo relacional además de permitir enlazar relaciones entre sí dando lugar a la estructura de la base de datos, también permite definir las acciones a realizar cuando se produce un borrado o modificación de una tupla en la relación padre referenciada por una relación hija.

Las acciones pueden ser:

- **CASCADE**

Para la actualización consiste en permitir actualizar el valor de la clave primaria de una tupla de la relación padre y actualizar todos los valores de la clave ajena de la relación hija que la referencian.

Para el borrado consiste en permitir borrar una tupla de la relación padre y borrar todas las tuplas de la relación hija que la referencian.

- **NO ACTION / RESTRICT**

Si la relación padre tiene tuplas asociadas en la relación hija la operación no se permitirá.

- **SET NULL**

Para la actualización consiste en permitir actualizar el valor de la clave primaria de una tupla de la relación padre y actualizar todos los valores de la clave ajena de la relación que la referencian al valor NULL.

Para el borrado consiste en permitir borrar una tupla de la relación padre y actualizar todos los valores de la clave ajena de la relación hija que la referencian al valor NULL.

La clave ajena debe permitir valores nulos para que esta opción tenga sentido.

- **SET DEFAULT**

Su funcionamiento es similar al caso anterior con la excepción de que el valor al que se ponen las claves ajenas de la relación hija es un valor por defecto que se habrá especificado en la relación para dicha clave ajena.

F) Restricción de verificación (CHECK)

Establece la condición que debe cumplir un atributo o varios de una relación para realizar la operación (inserción, borrado o modificación), en caso contrario la operación se rechaza. No necesita tener nombre.

G) Restricción de aserción (ASSERTION)

Su funcionamiento es idéntico al de las restricciones de verificación con la diferencia de que el ámbito de aplicación (la verificación) se extiende a más de una relación. Deben tener nombre.

H) Disparadores (TRIGGER)

Al contrario que el resto de restricciones vistas en las que se rechaza la operación si no se cumple la condición, en el caso de los disparadores la acción especificada se ejecuta siempre y cuando sí se cumpla la condición.

Los disparadores son procedimientos especiales que ayudan al diseñador de la base de datos a recoger aquellos supuestos semánticos del universo del discurso que no se han podido mantener en el modelo conceptual.

2.- Representación del modelo relacional

El modelo relacional se representa habitualmente de dos formas: textual o gráfica.

2.1.- Representación textual del modelo relacional

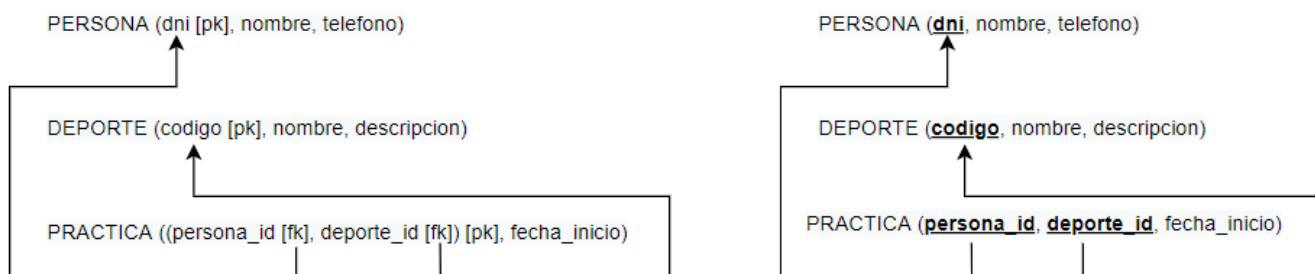
En la representación textual se da una lista con las relaciones y sus atributos. Mediante flechas se indica a qué relación corresponde una clave foránea.

No hay unanimidad para este tipo de representación más allá del uso de flechas. Algunas de las representaciones que se pueden encontrar son:

- Resaltar en negrita y subrayado la clave principal. Resaltar con línea discontinua la clave alternativa. Utilizar flechas entre claves ajenas y primarias.
- Junto al atributo principal se escribirá [pk], si es clave ajena [fk] y si es clave alternativa [ak]. Utilizar flechas entre claves ajenas y primarias.

En cualquier caso, por claridad no especificaremos en esta representación la obligatoriedad u opcionalidad de los atributos ni tampoco las acciones asociadas a las claves ajenas en las operaciones de modificación y borrado.

EJEMPLO



2.2.- Representación gráfica del modelo relacional

Lo más habitual es utilizar la notación gráfica con la notación Crow's Foot, que utiliza:

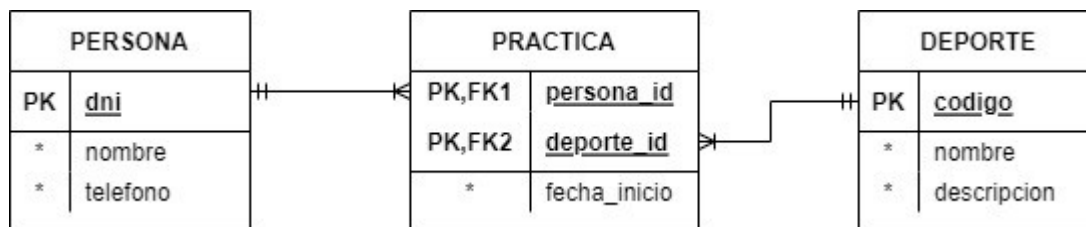
- Nodos (cajas) para representar las relaciones.
- Líneas que unen nodos para representar las modalidades de las entidades y las relaciones.

Correspondencia entre notaciones

(min, max)	Crow's foot
(0,N) - (1,1)	
(1,N) - (0,1)	
(0,1) - (1,1)	
(1,1) - (1,1)	
(0,1) - (0,1)	

- Símbolos para indicar determinadas restricciones:
 - ◆ Clave primaria _____ PK
 - ◆ Clave alternativa _____ AK
 - ◆ Clave foránea _____ FK
 - ◆ Clave primaria ajena _____ PF
 - ◆ Clave alternativa foránea _____ AF
 - ◆ Obligatorio (NOT NULL) _____ *

EJEMPLO



BLOQUE III

TRANSFORMACIÓN DEL MODELO CONCEPTUAL E/R AL MODELO LÓGICO RELACIONAL

1.- Conversión del E/R al Relacional

Con el modelo conceptual obtenido mediante el modelo E/R y sabiendo que el único constructor del modelo relacional es la **relación**, queda ahora aprender cómo convertir el modelo E/R en el modelo relacional equivalente.

Para ello se realizan en una serie de pasos bastante mecánicos a través de los cuales se van convirtiendo los distintos elementos de un modelo a sus equivalentes en el otro. Las reglas de transformación que se aplican en cada momento dependen del tipo de objeto del esquema conceptual que se deba transformar y en el caso de las relaciones, de las cardinalidades y modalidades de las entidades participantes.

Hay que hacer notar, sin embargo, que hay elementos que no se pueden transformar directamente y que hay que documentar en otra forma, por ejemplo con un documento anexo.

1.1.- Conversiones previas

Una restricción inherente al modelo relacional es que no pueden existir grupos repetitivos. Por tanto, para que los objetos obtenidos en el modelo conceptual satisfagan esta restricción hay que eliminar las siguientes anomalías:

1.1.1.- Eliminación de atributos compuestos

Todo atributo compuesto que provenga de una entidad o interrelación debe ser descompuesto en varios atributos simples, respetando los dominios que tenían en el atributo compuesto (que queda eliminado).

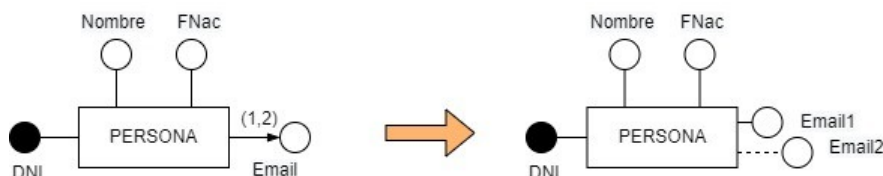


1.1.2.- Eliminación de atributos multivaluados

La eliminación del atributo multivaluado dará lugar a varias transformaciones dependiendo de los supuestos semánticos. Se distinguen los siguientes casos.

- **Primer caso**

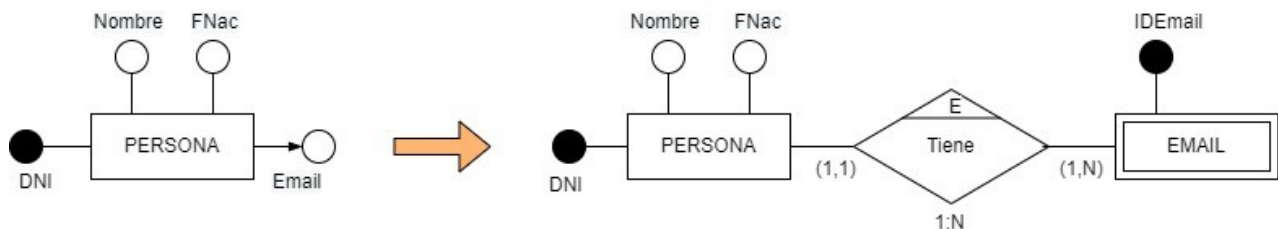
Si el atributo multivaluado puede tomar pocos valores y son conocidos para la mayoría de los ejemplares, es posible descomponerlo en varios atributos simples.



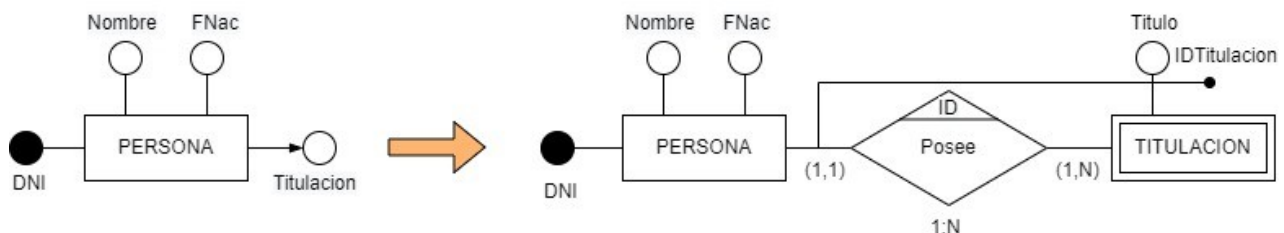
- **Segundo caso**

Si el atributo multivaluado puede tomar muchos valores y son desconocidos para la mayoría de los ejemplares. En este caso se crea una interrelación y una entidad. Hay dos alternativas que dependen del contexto del problema:

- 1) Se crea una nueva entidad débil en existencia con una relación de uno a muchos. La entidad débil tendrá un solo atributo univaluado que podrá identificar de manera única cualquier ejemplar de la entidad creada.



- 2) Esta opción es parecida a la anterior, pero un valor del atributo multivaluado puede pertenecer a más de un ejemplar de la entidad a la que describe, por lo que la dependencia es en identificación.



1.2.- Transformación de dominios

Un dominio es un conjunto de valores atómicos del mismo tipo y finito. Todo dominio tiene asociado un nombre y un formato que puede definirse por extensión (especificando la lista de valores permitida) o por intensión (especificando un tipo de datos).

Un dominio en el esquema conceptual será transformado en el mismo dominio en el modelo relacional. En el esquema conceptual no representaremos los dominios que más tarde deben ser reflejados en el esquema relacional para mantener la legibilidad en el diseño. Un dominio puede estar asignado a uno o varios atributos de distintas relaciones.

1.3.- Transformación de entidades

Toda entidad en el esquema conceptual se transformará en una relación (tabla) en el esquema relacional, cuyo nombre será el nombre de la entidad. Los atributos de la entidad serán los atributos (campos) de la tabla con las siguientes características o restricciones, dependiendo del tipo de atributo:

a) **Identificador principal**

El atributo o atributos que sea identificador principal se transforma en clave primaria de la tabla (PRIMARY KEY).

b) **Identificador alternativo**

A los atributos que son identificadores alternativos se les añadirán las restricciones de obligatoriedad (NOT NULL) y unicidad (UNIQUE).

c) **Atributo no identificador**

En este tipo de atributo puede ser que sea obligatorio conocer su valor (NOT NULL) o, por el contrario, que sea opcional (NULL).

1.4.- Transformación de relaciones binarias

Cualquier relación binaria en el esquema conceptual se transformará en una de estas dos formas:

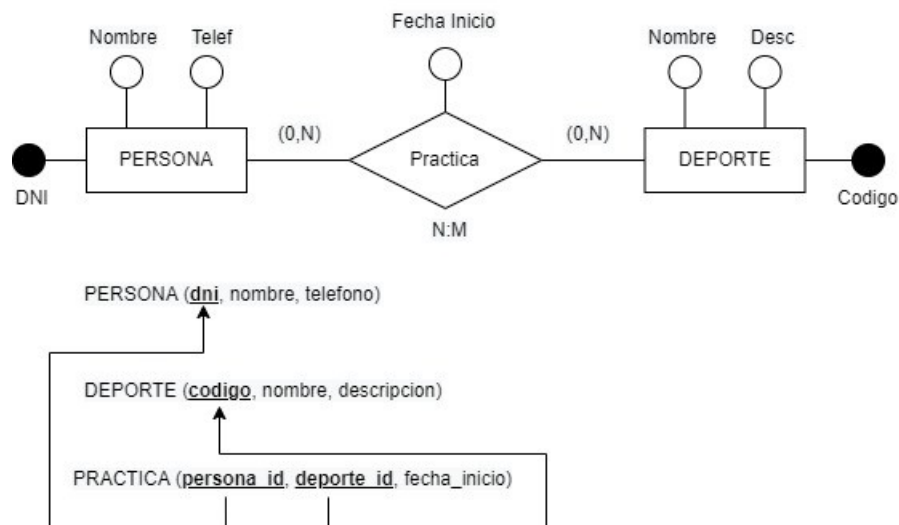
1. Creación de relación (tabla)
2. Propagación de clave

La elección de una u otra dependerá del tipo de la cardinalidad de la relación y de las modalidades de las entidades participantes.

1.4.1.- Relaciones N:M

Se transforma en una relación con los siguientes atributos: los identificadores de las entidades asociadas y los atributos propios de la relación si los tuviera.

La suma de todos estos identificadores será la clave primaria de la relación creada. Cada uno de estos identificadores es una clave ajena que referencia a la tabla de la que proviene.

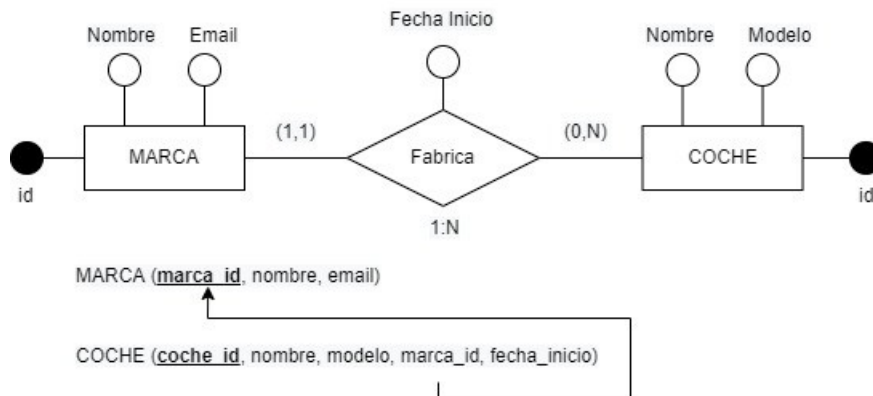


1.4.2.- Relaciones 1:N

Se puede modelizar de varias formas según como sea la cardinalidad mínima de la entidad con cardinalidad máxima 1.

- **Primer caso**

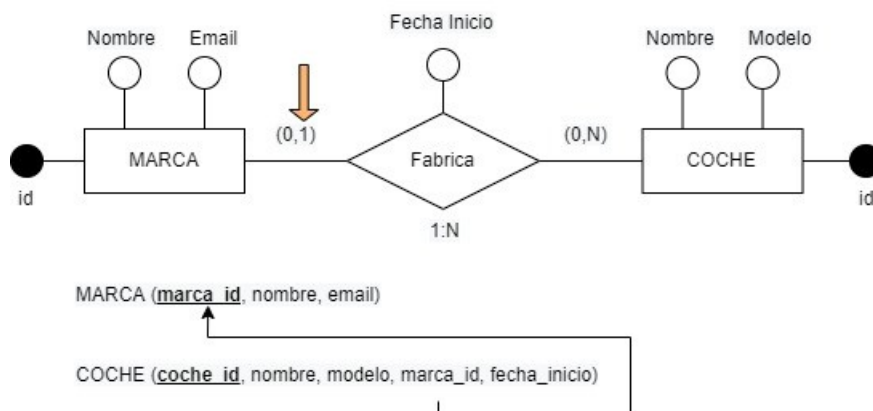
La entidad que tiene cardinalidad máxima 1 propagará su clave a la entidad de cardinalidad máxima N y no admitirá valores nulos. Si la relación tuviera atributos, estos también serán propagados en el mismo sentido.



- **Segundo caso**

Admite dos transformaciones:

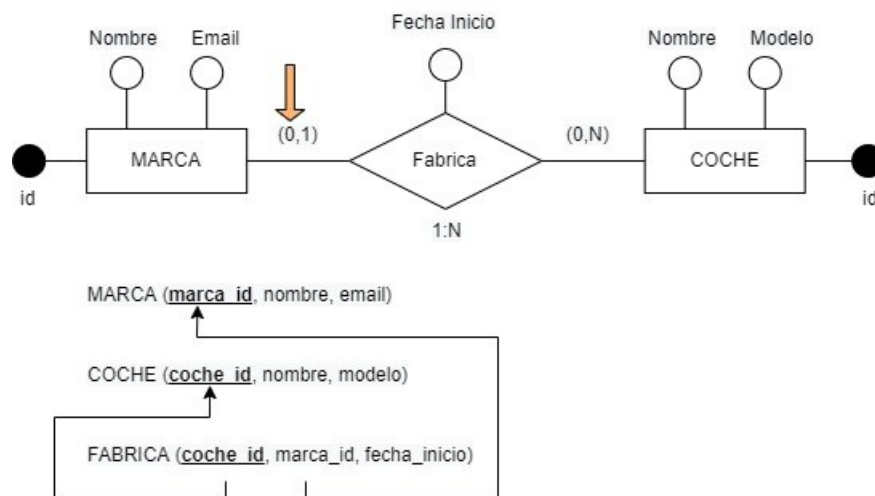
1. Si la mayoría de los ejemplares de la entidad que participa con cardinalidad máxima N conoce su ejemplar en la entidad que participa con cardinalidad máxima 1, entonces se modelizará como propagación de clave igual que en el caso anterior y la clave ajena admitirá valores nulos. Si la interrelación tuviera atributos estos también serán propagados en el mismo sentido.



2. Si la mayoría de los ejemplares de la entidad que participa con cardinalidad máxima N desconoce su ejemplar vinculado de la entidad que participa con cardinalidad máxima 1, entonces la interrelación se modelizará como creación de relación tomando como atributos los identificadores principales de las entidades asociadas y los propios de la interrelación si los tuviera.

Cada uno de estos identificadores es una clave ajena en la tabla creada y no admitirá valores nulos. La clave primaria de la nueva relación será la clave procedente de la entidad del lado N.

Esta transformación evita los valores nulos que aparecerían en caso de propagar la clave de la entidad del lado 1 a la entidad del lado N.



1.4.3.- Relaciones 1:1

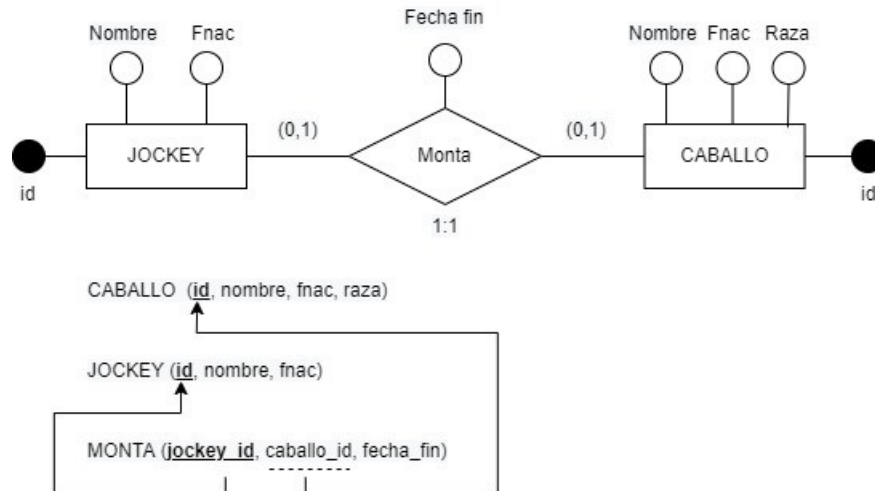
Se puede transformar de varias formas, dependiendo de si al menos una de las entidades participa con modalidad (1,1) o si por el contrario, ambas participan con modalidad (0,1).

En toda transformación de una relación con cardinalidad 1:1 se creará un índice sin repetidos sobre la clave ajena propagada.

- **Primer caso**

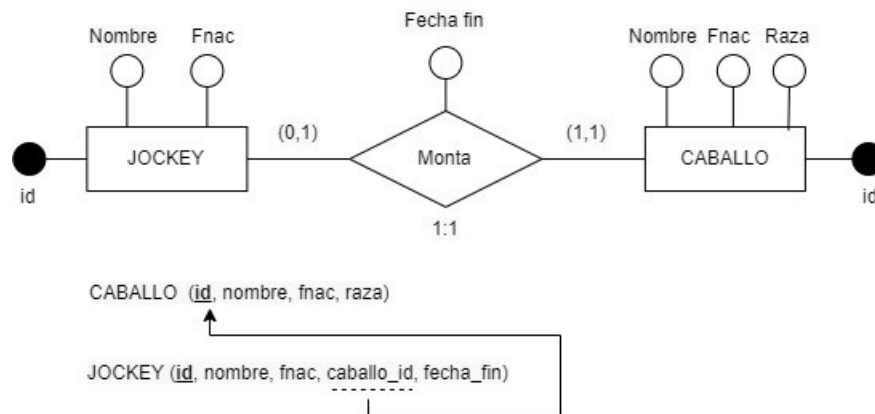
Si las dos entidades participan con cardinalidad (0,1) se creará una nueva relación que tendrá por atributos los identificadores de las entidades asociadas y los atributos propios de la interrelación si los tuviera.

La clave primaria de la tabla creada será cualquiera de los dos identificadores participantes. El que no sea elegido será clave alternativa y se le creará un índice sin repetidos.



- Segundo caso**

Si una de las entidades participa con cardinalidad (1,1) y la otra con cardinalidad (0,1), la entidad con cardinalidad (1,1) propagará su clave a la otra entidad como clave ajena y no permitirá valores nulos. Además esta clave será también clave alternativa y se creará un índice sin repetidos sobre ella. Si la interrelación tuviera atributos deberán ser propagados en el mismo sentido.



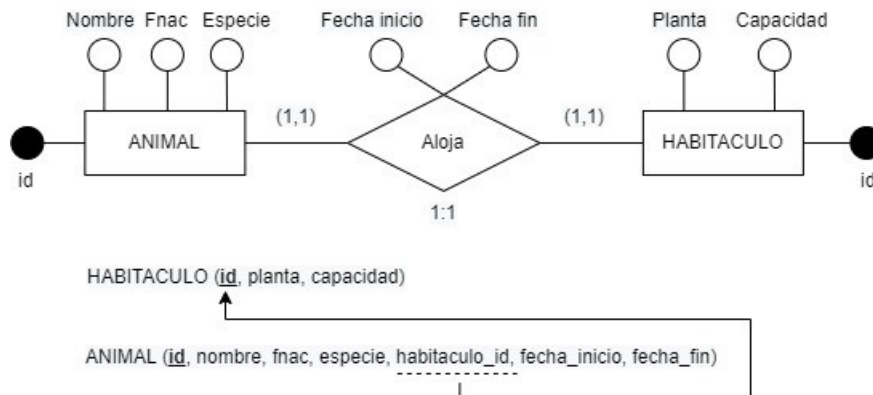
- Tercer caso**

Si las dos entidades participan con cardinalidad (1,1) existen varias opciones.

Una es propagar el identificador de una entidad al de la otra, como clave ajena y no admitirá valores nulos, teniendo en cuenta la elección en función de los accesos más frecuentes a los datos de las relaciones. Esta clave ajena será también clave alternativa por lo que crearemos un índice sin repetidos sobre la misma. Si la interrelación tuviera atributos también serían propagados en el mismo sentido.

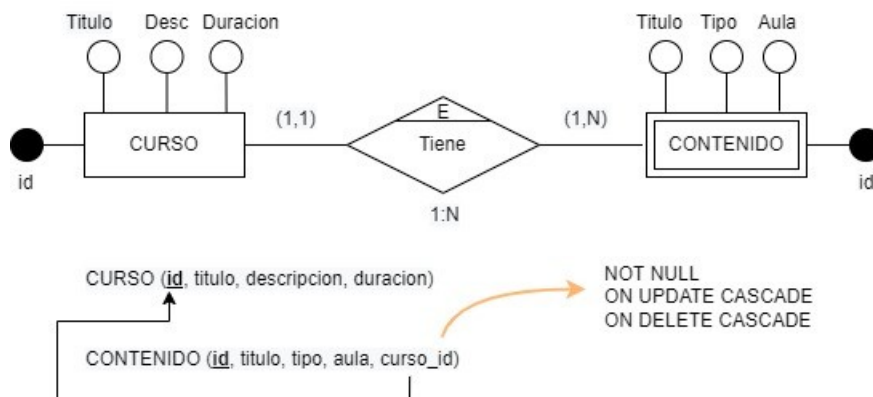
Otra opción es la propagación de los dos identificadores (en un sentido y en el otro), lo que puede provocar redundancias que han de ser controladas. No se aconseja realizar este tipo de transformación si la interrelación tiene atributos propios puesto que la redundancia sería aún mayor.

La última opción consiste en unir las relaciones provenientes de las dos entidades en una sola, combinando todos los atributos de las mismas y eligiendo como clave primaria uno de los dos identificadores y estableciendo el otro como clave alternativa.

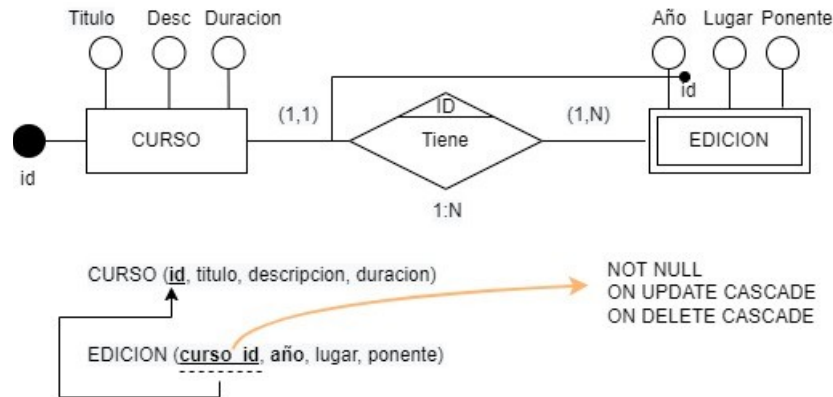


1.4.4.- Relaciones débiles

La interrelación que provenga de una entidad débil con dependencia en existencia o identificación se modeliza como propagación de clave y la clave ajena no admitirá valores nulos. Además, también tendrá asociada la acción CASCADE tanto para la modificación como para el borrado.



En el caso de que la dependencia sea en identificación, la clave primaria de la relación que proviene de la entidad débil estará formada al menos por el identificador de la entidad fuerte y en muchos casos, por algún o algunos atributos propios de la propia relación débil.

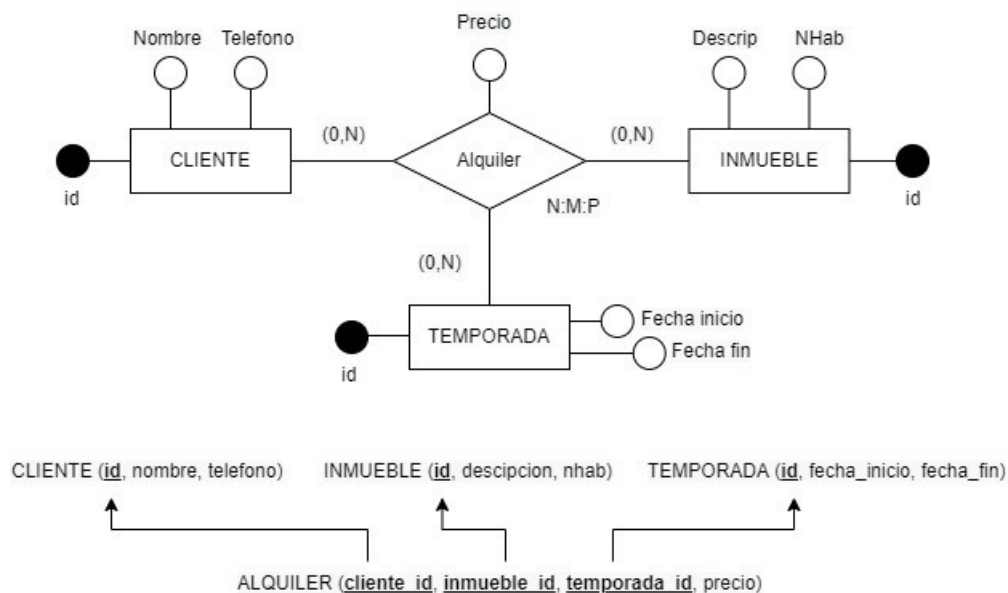


1.5.- Transformación de relaciones ternarias

Cualquier relación ternaria en el esquema conceptual se transformará como creación de relación. Los atributos que formarán la nueva relación serán la suma de todos los identificadores de las entidades asociadas más los atributos propios de la interrelación si ésta los tuviera.

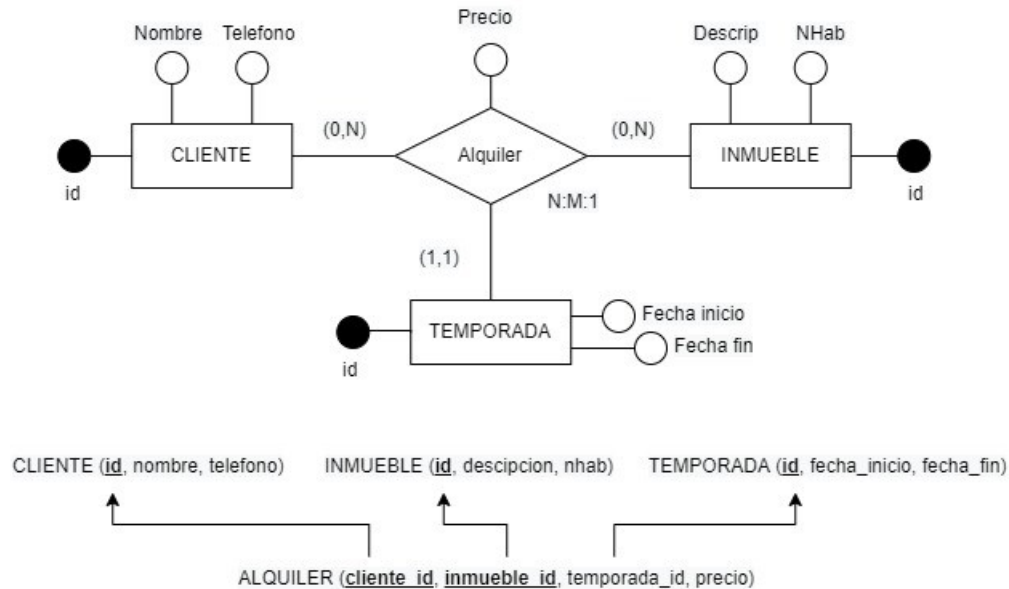
1.5.1.- Relaciones N:M:P

La clave primaria de la relación será la suma de todos los identificadores de las entidades participantes en la interrelación.



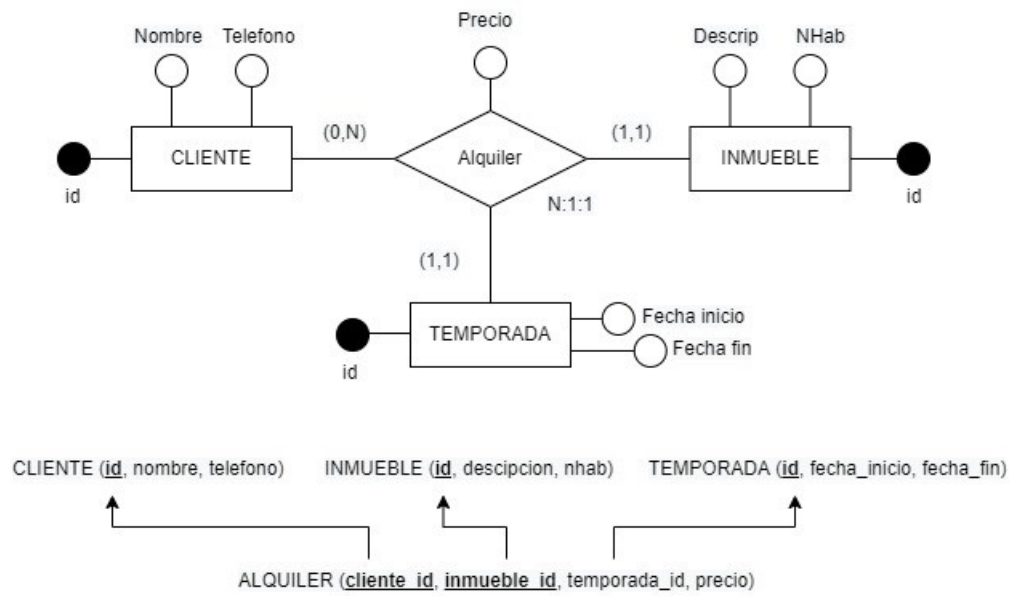
1.5.2.- Relaciones N:M:1

La clave primaria de la relación será la suma de los identificadores principales de las dos entidades que participan en la interrelación con cardinalidad máxima N.



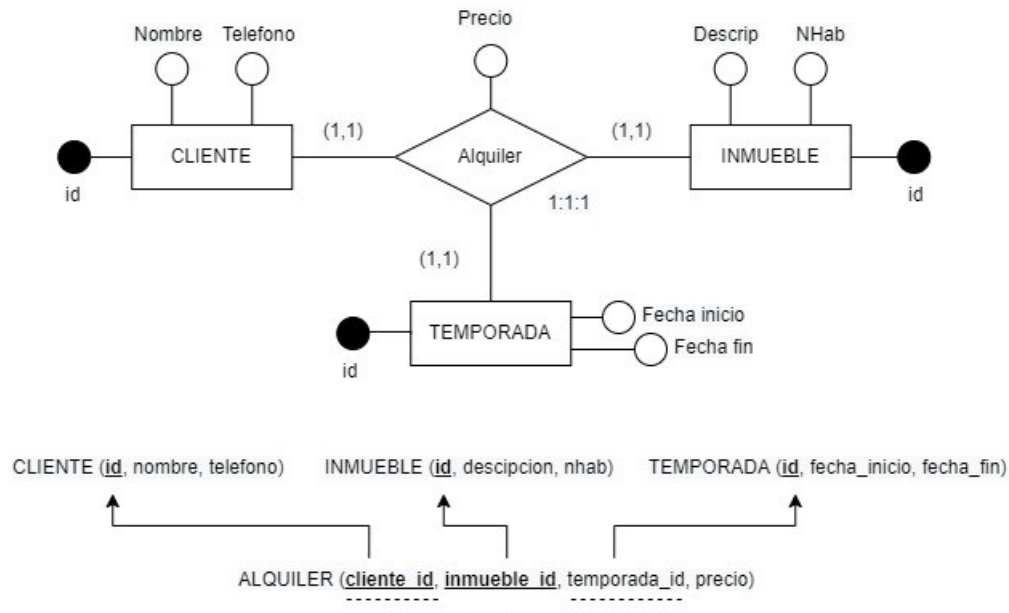
1.5.3.- Relaciones N:1:1

La clave primaria de la relación será la suma del identificador principal de la entidad que participa con cardinalidad máxima N y el identificador de una de las otras dos entidades participantes en la interrelación. Además se añadirá como clave alternativa la formada por el identificador de la entidad que participa con cardinalidad máxima N junto con el identificador de la entidad que no forma parte de la clave primaria de la relación creada.



1.5.4.- Relaciones 1:1:1

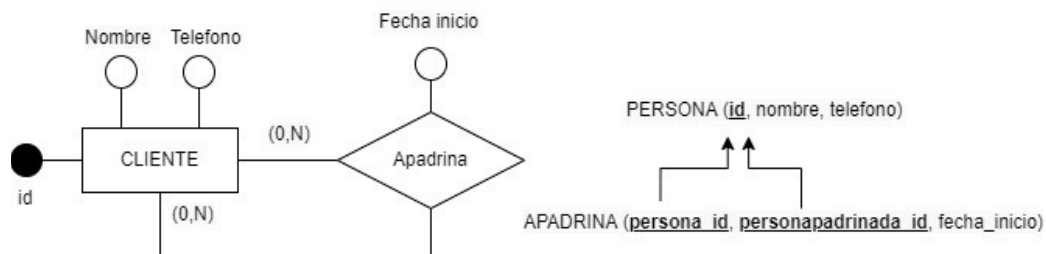
Se establecerán tres claves candidatas formadas por las tres combinaciones posibles de parejas de identificadores de las entidades participantes. Una de estas parejas será establecida como clave primaria de la relación creada y las otras dos parejas serán claves alternativas.



1.6.- Relaciones reflexivas

Se modelizan igual que cualquier relación binaria pero teniendo en cuenta que una entidad puede participar con más de un rol en la interrelación.

Tanto si el proceso se lleva a cabo por propagación de clave o por creación de relación, el identificador propagado no puede tener el mismo nombre que los atributos de la relación que los recibe. Esto es una restricción inherente al modelo relacional, todos los atributos de una misma relación han de tener un nombre distinto.



1.7.- Transformación de jerarquías

Las jerarquías en el modelo conceptual representan un tipo de relación especial entre dos tipos de entidades distintas llamadas supertipo y subtipo. Al transformarse al modelo relacional, como este tiene un solo constructor, se pierde parte de la semántica que reflejaba ese tipo de relación específica entre dichas entidades.

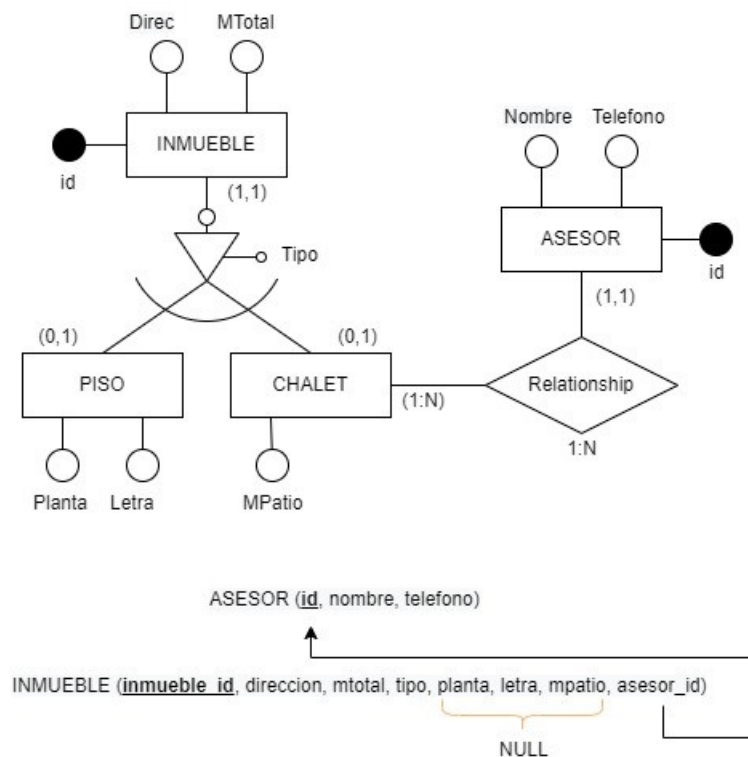
No hay un mecanismo directo de transformación, sino que dependiendo del tipo de jerarquía (parcial / total, inclusiva / exclusiva) y de otros factores se modelizará de una de las tres formas que veremos a continuación.

La semántica recogida en los distintos tipos de jerarquía puede recogerse en el modelo relacional mediante restricciones (CHECK) o procedimientos disparadores o triggers asociados a distintos eventos de actualización sobre distintas relaciones.

- **Primer caso**

Eliminar los subtipos y dejar sólo la relación del supertipo. Toda la información específica de los subtipos (atributos e interrelaciones) y el atributo discriminante de la jerarquía, si existe, se traslada al supertipo. En el caso de que alguno de los subtipos tuviera alguna interrelación con otra entidad, la interrelación vincularía esa entidad con el supertipo.

El esquema relacional es más reducido pero los valores de los atributos transferidos desde los subtipos al supertipo dejarán de ser obligatorios en el caso que los fueran. Esta transformación es aconsejable cuando no haya muchas diferencias en cuanto al número de atributos e interrelaciones entre el supertipo y los subtipos.

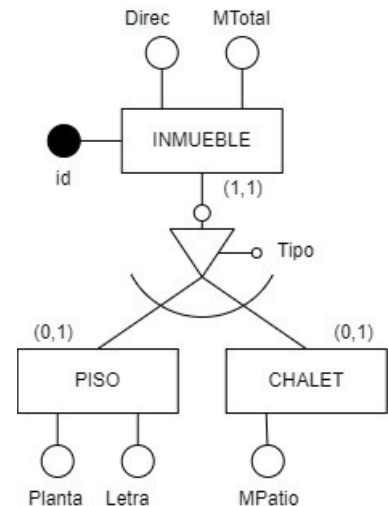
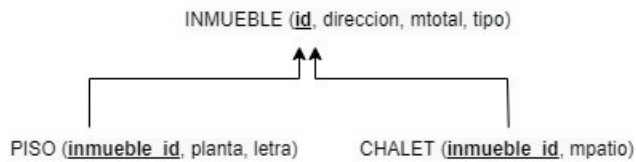


- **Segundo caso**

Se mantienen el supertipo y los subtipos y se modeliza creando una interrelación 1:1 entre cada pareja supertipo-subtipo.

Esta solución es la que mantiene la mayor semántica posible, aunque es la menos eficiente al ser el esquema relacional más extenso y requerir cierto tipo de consultas más combinaciones de tablas. Apropiaada cuando hay muchas diferencias en cuanto al número de atributos e interrelaciones entre el supertipo y los subtipos existentes.

Cada pareja se modelizará propagando el identificador de la entidad supertipo a la entidad subtipo. Si la entidad subtipo no tiene identificador propio, la clave ajena será también clave primaria de dicho subtipo. Si el subtipo tiene algún identificador, entonces la clave ajena será también clave alternativa.

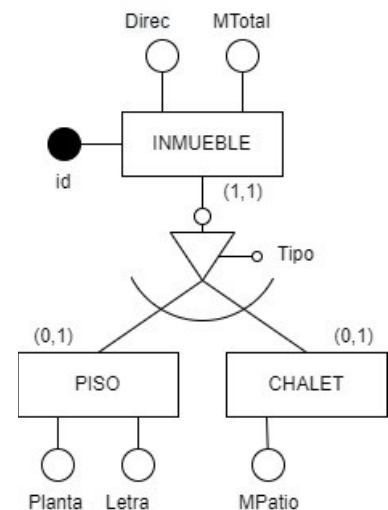
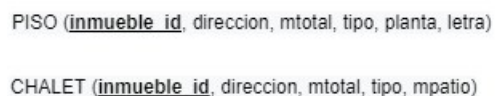


- **Tercer caso**

Se elimina el supertipo, quedando tantas relaciones como subtipos existan. Toda la información común del supertipo (atributos e interrelaciones) es llevada a cada uno de los subtipos, que contendrán además sus atributos específicos.

Se adoptará esta transformación cuando se cumplan dos condiciones:

- Existen muchas diferencias en cuanto al número de atributos e interrelaciones entre el supertipo y los subtipos existentes.
- La mayoría de las consultas que se realizan sobre algún subtipo afectan tanto a los atributos propios del mismo como a los atributos comunes del supertipo eliminado.



1.8.- Transformación de la dimensión temporal

Cuando en el modelo lógico, en una interrelación existe un atributo multivaluado que representa tiempo su transformación al modelo relacional se puede hacer de dos formas:

- **Primer caso**

Convertir el atributo multivaluado en atributo univaluado y analizar cuál es la clave primaria de la relación resultante dependiendo de los distintos supuesto semánticos.

- **Segundo caso**

Convertir el atributo multivaluado en entidad en el modelo E/R y transformar la interrelación inicial en otra de mayor grado.

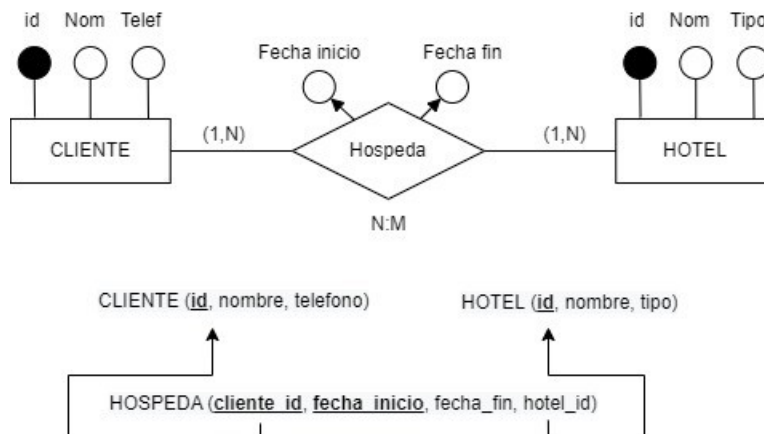
Vemos esto con un ejemplo.

Supongamos que una agencia de viajes desea crear una base de datos teniendo en cuenta los siguientes supuestos semánticos:

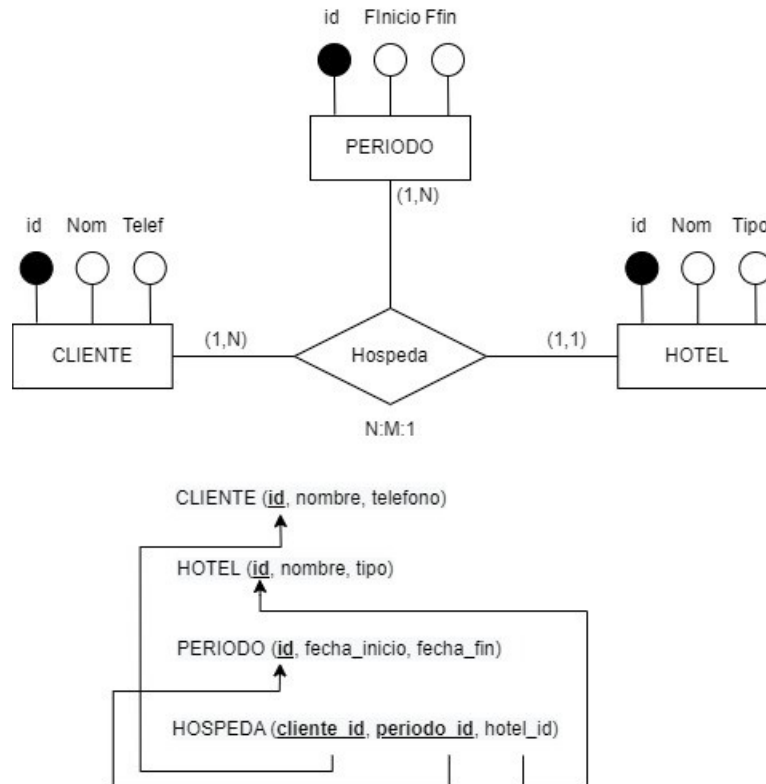
- En un hotel se pueden alojar varios clientes y estos se pueden hospedar en distintos hoteles a lo largo del tiempo.
- Se quiere saber en qué fechas se ha hospedado un cliente en un hotel.
- Además, se quiere imponer la restricción de que un cliente en una fecha solo puede hospedarse en un hotel.

Para el primer caso la clave de la relación Hospeda no será la suma de los identificadores de las entidades participantes y los atributos multivaluados como hemos visto en la transformación de la interrelación N:M, ya que dicha clave no sería mínima.

La clave primaria estará compuesta por el identificador de la entidad CLIENTE y el atributo multivaluado Fecha_inicio, respetando así el último supuesto semántico.



Para el segundo caso el aspecto temporal se convierte en la entidad PERIODO y se aumenta el grado de la interrelación Hospeda, pasando de binaria a ternaria y se ajustan las cardinalidades de las entidades en función a los supuestos semánticos.



1.9.- Transformación de atributos derivados

Hay dos posibilidades:

- Almacenar el atributo derivado como atributo propio de la relación a la que pertenece. En tal caso se deben crear procedimientos para calcular su valor cada vez que se realicen operaciones de actualización sobre los elementos que intervienen en el cálculo de este.
- No almacenar el atributo derivado y se calcula su valor cuando se demande dicha información.

2.- El modelo relacional y los SGBD

Las siguientes restricciones del esquema E/R se deben tener en cuenta en el modelo relacional mediante checks, aserciones o disparadores:

- Cardinalidades mínimas de 1 en interrelaciones N:M y 1:N (excluyendo aquellas que se controlan con la restricción NOT NULL cuando se realiza propagación de clave).
- Cardinalidades máximas conocidas en interrelaciones binarias N:M y 1:N e interrelaciones ternarias.
- Exclusividad en las generalizaciones.
- Inserción y borrado en generalizaciones.
- Atributos derivados.
- Exclusividad entre interrelaciones.
- Atributos multivaluados obligatorios.

BLOQUE IV

NORMALIZACIÓN

DEL

MODELO LÓGICO RELACIONAL

1.- Normalización

El modelo relacional describe los datos a través de relaciones que están enlazadas entre si por valores idénticos de campos clave: claves primarias y claves ajenas.

Pero... ¿Cómo sabemos si el esquema relacional está bien diseñado? ¿Estamos repitiendo características de un objeto en distintas relaciones? ¿Existen características definidas en un objeto que realmente pertenecen a otro?. Para responder a estas preguntas existe un proceso sistemático de eliminación de inconsistencias en bases de datos relacionales denominado *normalización*.

Un esquema relacional que no esté normalizado provoca redundancias que originan un mayor consumo de espacio en disco y anomalías en las operaciones de actualización (inserción, edición y eliminación).

Supongamos que se tiene una base de datos relacional sencilla que consta de una única relación, que se muestra a continuación:

R				
<u>CodProveedor</u>	<u>Ciudad</u>	<u>País</u>	<u>CodPieza</u>	<u>Cantidad</u>
1	Londres	Inglaterra	1	200
1	Londres	Inglaterra	2	1300
1	Londres	Inglaterra	4	345
2	Madrid	España	1	24
2	Madrid	España	6	988
3	Paris	Francia	2	200
4	Londres	Inglaterra	3	34

Se puede comprobar fácilmente que esta relación no está demasiado bien diseñada y que puede dar problemas de inconsistencia de datos por la abundancia de redundancias que aparecen. Por ejemplo, se repite muchas veces la información de que el proveedor 1 es de Londres.

Esta redundancia puede provocar graves problemas en la manipulación de los datos. Específicamente:

- **Problemas de actualización**

Si por ejemplo, el proveedor 1 cambia de ciudad habría que actualizar TODAS las tuplas en las que aparece dicho proveedor. Si el operador se equivoca y no modifica alguna de ellas, se podrían producir dos respuestas a la pregunta ¿En qué ciudad reside el proveedor?

- **Problemas de inserción**

Como los datos de un pedido y de un proveedor van entrelazados, no se podría añadir un proveedor hasta que no haga un pedido.

- **Problemas de eliminación**

Si eliminamos un proveedor también se eliminan sus pedidos.

Para evitar este tipo de problemas se realiza un proceso denominado *normalización*, establecido por Codd en 1972, para llevar la base de datos hasta lo que se llama una *forma normal*. El proceso consiste en descomponer la relación que no está normalizada en varias relaciones más pequeñas, es decir que tienen menos atributos, que sean equivalentes semánticamente y más normalizadas que la originaria.

Una forma normal es una serie de condiciones que deben cumplir todas las relaciones de la base de datos para asegurar que se reducen las redundancias al mínimo posible.

Existen diferentes niveles de normalización: 1FN, 2FN, 3FN, FNBC (FN de Boyce y Codd), 4FN, 5FN y Dominio-Clave. Una relación que está en una forma normal específica está también en la forma normal inmediatamente inferior.

Se considera que un diseño es aceptable si está en 3FN y se garantiza que no hay pérdida de información en el proceso de normalización. Sin embargo, un esquema muy normalizado puede establecer un número elevado de relaciones y algunas operaciones de consulta pueden requerir la combinación excesiva de tablas, disminuyendo la eficiencia en dichas operaciones.

Antes de definir las formas normales de Codd se deben definir dos conceptos fundamentales que se utilizan: *dependencia funcional simple* y *dependencia funcional plena*.

1.1.- Dependencia funcional simple

La dependencia funcional simple, o dependencia funcional a secas, entre dos atributos se define de la siguiente forma:

Dados dos atributos X e Y de una relación R, se dice que Y depende funcionalmente de X si, y sólo si, en cualquier instante cada valor de X tiene asociado, en R, un único valor de Y. Se suele denotar como $X \rightarrow Y$

Dicho de una forma informal se dice que un atributo Y depende de otro X si el valor de Y está determinado por el valor de X pero no al revés.

1.2.- Dependencia funcional plena

La dependencia funcional plena se define de la siguiente forma:

Dado un conjunto de atributos X (X_1, X_2, \dots, X_n) y otro atributo Y, se dice que Y tiene dependencia funcional plena de X si Y depende de X y no se puede encontrar ningún subconjunto de X del que dependa Y.

Dicho informalmente, si el valor de un atributo depende de una clave compuesta, debe depender de **toda** la clave y no de parte de ella.

1.3.- Primera Forma Normal

Una relación está en Primera Forma Normal (1FN) si se cumple la condición de que todos los dominios no contienen valores que sean conjuntos sino que son valores escalares.

Dicho en otras palabras, una relación está en 1FN si no hay ningún atributo cuyo valor sea un conjunto de valores.

Es una forma normal que, por definición, toda tabla que cumpla las condiciones del modelo relacional debe cumplir, por lo que usualmente todas las bases de datos deberían estar en esta forma normal.

En el ejemplo anterior, la relación ya está en 1FN porque no hay atributos cuyos valores sean conjuntos.

1.4.- Segunda Forma Normal

Una relación está en Segunda Forma Normal (2FN) si, además de estar en 1FN, cualquier atributo que no forme parte de la clave primaria tiene dependencia funcional plena con la clave primaria.

Si la clave primaria de la relación no es compuesta, la relación está automáticamente en 2FN.

Dicho informalmente, esta forma normal intenta hacer que todos los atributos de una relación dependan de **toda** la clave principal. Si sólo dependen de parte, esto quiere decir que hay un problema.

Para solucionarlo se toman los atributos que sólo dependen de parte de la clave y se mueven a otra nueva relación que sólo contendrá los atributos parcialmente dependientes y la parte de la clave principal de la que dependen. Esta parte será la clave principal de la nueva relación. Los campos de la clave principal permanecen en la relación original también.

En el ejemplo,

La clave es (CodProveedor, CodPieza).

Si se examinan los otros se puede comprobar lo siguiente:

- Ciudad depende sólo de CodProveedor, no de CodPieza. Si se examinan los datos se verá que hay valores distintos de Ciudad para el mismo valor de CodPieza. Por ejemplo, para CodPieza con valor 1 hay dos valores distintos de Ciudad: Londres y Madrid.
- Pais está en el mismo caso que Ciudad.
- Cantidad *si* depende de la clave completa.

Por lo tanto, dividimos la relación R en dos: R1 y R2

R1(CodProveedor,CodPieza,Cantidad)

R2(CodProveedor,Ciudad,Pais)

Si se vuelven a analizar ambas relaciones se comprobará que las dos están en 2FN, por lo que la base de datos está en 2FN.

1.5.- Tercera Forma Normal

Una relación está en Tercera Forma Normal (3FN) si está en 2FN y además ninguno de los atributos que no forman parte de la clave primaria tiene dependencias transitivas respecto de la clave de la relación.

Informalmente quiere decir que hay atributos que dependen de atributos que no forman parte de la clave primaria.

Para solucionar el problema se mueve el atributo dependiente y el intermedio a otra nueva relación en la que el intermedio será clave primaria. El atributo intermedio permanece en la relación original.

En el caso del ejemplo,

Está claro que la relación R1 está en 3FN.

¿Y la relación R2?

- ¿Ciudad depende de Proveedor? Sí.
- ¿País depende de Proveedor? No directamente. País depende de Ciudad, que a su vez depende de Proveedor.

Por lo tanto la relación R2 no cumple la 3FN.

Habría que dividir la relación R2 en dos relaciones: R21 y R22. La base de datos quedaría:

R1(CodProveedor, CodPieza, Cantidad)

R21(CodProveedor, Ciudad)

R22(Ciudad, País)

Y los datos

R1			R21		R22	
<u>CodProveedor</u>	<u>CodPieza</u>	Cantidad	<u>CodProveedor</u>	Ciudad	<u>Ciudad</u>	País
1	1	200	1	Londres	Londres	Inglaterra
1	2	1300	2	Madrid	Madrid	España
1	4	345	3	Paris	Paris	Francia
2	1	24	4	Londres		
2	6	988				
3	2	200				
4	3	34				

Como se puede comprobar se han eliminado completamente las redundancias.