

BD03. Interpretación de diagramas Entidad/Relación

MENÚ

1.	Análisis y diseño de bases de datos.....	2
2.	¿Qué es el Modelo E/R?.....	4
3.	Entidades.....	5
	3.1. Tipos de entidades.....	6
4.	Atributos	8
	4.1. Tipos de atributos.....	10
	4.2. Claves.....	12
	4.3. Atributos de una relación.....	14
5.	Relaciones.....	15
	5.1. Grado de una relación.....	16
	5.2. Cardinalidad de relaciones.....	17
	5.3. Cardinalidad de entidades.....	19
6.	Simbología del modelo E/R.....	21
7.	Modelo E/R Extendido.....	22
	7.1. Restricciones en las relaciones.....	23
	7.2. Generalización y especialización.....	25
	7.3. Agregación.....	27
8.	Elaboración de diagramas E/R.....	29
	8.1. Identificación de entidades y relaciones.....	30
	8.2. Identificación de atributos, claves y jerarquías.....	31
	8.3. Metodologías.....	32
	8.4. Redundancia en diagramas E/R.....	33
	8.5. Propiedades deseables de un diagrama E/R.....	35
9.	Paso del diagrama E/R al Modelo Relacional.....	36
	9.1. Simplificación previa de diagramas.....	38
10.	Paso del diagrama E/R.....	41
11.	Modelo E/R Extendido al Modelo Relacional.....	46
	11.1. Tipos de dependencias.....	48
	11.2. Formas normales.....	50

1. Análisis y diseño de bases de datos.

Como se ha descrito en unidades anteriores, un modelo de datos es una colección de herramientas conceptuales que permiten llevar a cabo la descripción de los datos, sus relaciones, su semántica o significado y las restricciones que se les pueden aplicar. Sabemos que los SGBD cuentan con una arquitectura que simplifica, a los diferentes usuarios de la base de datos, su labor. El objetivo fundamental de esta arquitectura es separar los programas de aplicación de la base de datos física, proponiendo tres niveles de abstracción: **nivel interno o físico, nivel lógico o conceptual y nivel externo o de visión del usuario.**

El **Nivel lógico o conceptual** describe la estructura completa de la base de datos a través de lo que llamamos **Esquema Conceptual**, que se encarga de representar la información de una manera totalmente independiente del Sistema Gestor de Base de Datos.

Cuando hemos de desarrollar una base de datos se distinguen claramente dos fases de trabajo: **Análisis y Diseño**.

Pasos de las fases de Análisis y de Diseño

Fase de Análisis	Fase de Diseño
Análisis de entidades: se trata de localizar y definir las entidades y sus atributos.	Diseño de tablas.
Análisis de relaciones: se definirán las relaciones existentes entre las entidades.	Normalización.
Obtención del Esquema Conceptual a través del modelo E/R.	Aplicación de retrodiseño si fuese necesario.
Fusión de vistas: se reúnen en un único esquema todos los esquemas existentes, en función de las diferentes vistas de cada perfil de usuario.	Diseño de transacciones: localización del conjunto de operaciones o transacciones que operarán sobre el esquema conceptual.
Aplicación el enfoque de datos relacional.	Diseño de sendas de acceso: se formalizan los métodos de acceso dentro de la estructura de datos.

BD03. Interpretación de diagramas Entidad/Relación

Llevando a cabo una correcta fase de análisis estaremos dando un paso determinante en el desarrollo de nuestras bases de datos. El hecho de saltarse el esquema conceptual conlleva un problema de pérdida de información respecto al problema real a solucionar. El esquema conceptual debe reflejar todos los aspectos relevantes del mundo real que se va a modelar.

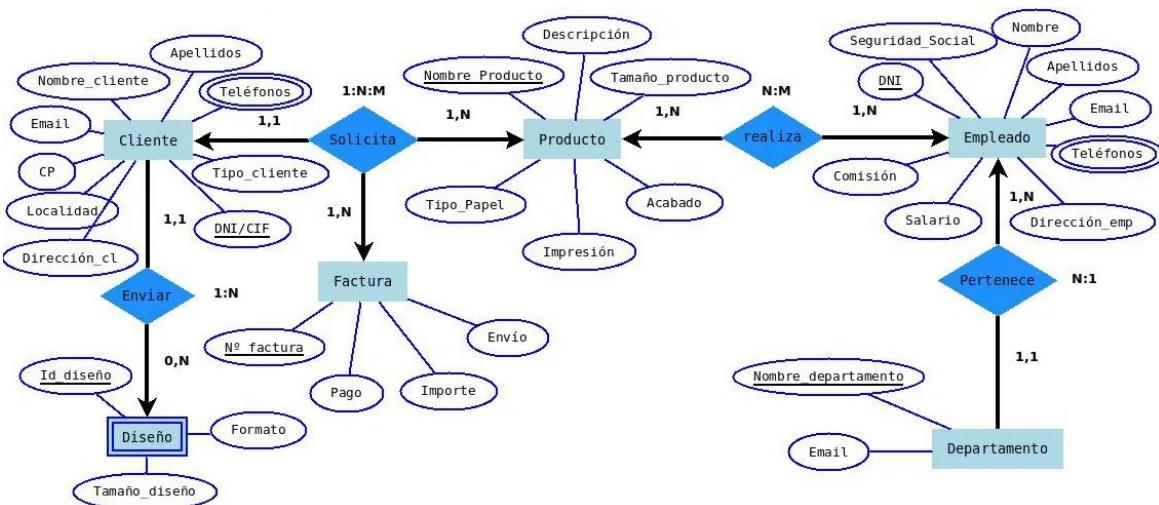
Para la realización de esquemas que ofrezcan una visión global de los datos, Peter Chen en 1976 y 1977 presenta dos artículos en los que se describe el **modelo Entidad/Relación** (entity/relationship). Con el paso del tiempo, este modelo ha sufrido modificaciones y mejoras. Actualmente, el modelo **entidad/relación extendido (ERE)** es el más aceptado, aunque existen variaciones que hacen que este modelo no sea totalmente un estándar. Ambos modelos serán estudiados a lo largo de esta unidad.

2. ¿Qué es el Modelo E/R?

El **modelo Entidad/Relación** es una herramienta de referencia para la representación conceptual de problemas del mundo real. Su objetivo principal, facilitar el diseño de bases de datos permitiendo la especificación de un esquema que representa la estructura lógica completa de una base de datos. Este esquema partirá de las descripciones textuales de la realidad, que establecen los requerimientos del sistema, buscando ser lo más fiel posible al comportamiento del mundo real para modelarlo.

El **modelo de datos E/R** representa el significado de los datos, es un modelo semántico. De ahí que no esté orientado a ningún sistema físico concreto y tampoco tiene un ámbito informático puro de aplicación, ya que podría utilizarse para describir procesos de producción, estructuras de empresa, etc. Además, las características actuales de este modelo favorecen la representación de cualquier tipo de sistema y a cualquier nivel de abstracción o refinamiento, lo cual da lugar a que se aplique tanto a la representación de problemas que vayan a ser tratados mediante un sistema informatizado, como manual.

Gracias al modelo Entidad-Relación, creado por Peter Chen en los años setenta, se puede representar el mundo real mediante una serie de símbolos y expresiones determinados. El modelo de datos Entidad/Relación (E/R ó E-R) está basado en una percepción consistente en objetos básicos llamados **entidades** y relaciones entre estos objetos. Estos y otros conceptos se desarrollan a continuación.



3. Entidades.

Si utilizamos las bases de datos para guardar información sobre cosas que nos interesan o que interesan a una organización, ¿No crees que hay que identificar esas cosas primero para poder guardar información sobre ellas? Para ello, vamos a describir un primer concepto, el de **Entidad**.

Una **entidad** puede ser un objeto físico, un concepto o cualquier elemento que queramos modelar, que tenga importancia para la organización y del que se deseé guardar información. Cada entidad debe poseer alguna característica, o conjunto de ellas, que lo haga único frente al resto de objetos. Por ejemplo, podemos establecer una entidad llamada **ALUMNO** que tendrá una serie de características. El alumnado podría ser distinguido mediante su número de identificación escolar (**NIE**), por ejemplo.

Entidad: objeto real o abstracto, con características diferenciadoras capaces de hacerse distinguir de otros objetos, acerca del cual se desea guardar información.

¿Ponemos otro ejemplo? Supongamos que tienes que desarrollar el esquema conceptual para una base de datos de mapas de montaña, los elementos: camping, pista forestal, valle, río, pico, refugio, etc., son ejemplos de posibles entidades. A la hora de identificar las entidades, hemos de pensar en nombres que tengan especial importancia dentro del lenguaje propio de la organización o sistema que vaya a utilizar dicha base de datos. Pero no siempre una entidad puede ser concreta, como un camping un río, en ocasiones puede ser abstracta, como un préstamo, una reserva en un hotel o un concepto.

Un **conjunto de entidades** serán un grupo de entidades que poseen las mismas características o propiedades. Por ejemplo, al conjunto de personas que realizan reservas para un hotel de montaña determinado, se les puede definir como el conjunto de entidades cliente. El conjunto de entidades río, representará todos los ríos existentes en una determinada zona. Por lo general, se suele utilizar el término entidad para identificar conjuntos de entidades. Cada elemento del conjunto de entidades será una ocurrencia de entidad.

Si establecemos un símil con la Programación Orientada a Objetos, podemos decir que el concepto de entidad es análogo al de instancia de objeto y que el concepto de conjunto de entidades lo es al de clase.

3.1. Tipos de entidades.

Las entidades pueden ser clasificadas en dos grupos:

- **Entidades Fuertes o Regulares.**

Son aquellas que tienen existencia por sí mismas, es decir, **su existencia NO depende de la existencia de otras entidades.**

En el modelo E/R las entidades fuertes se representan con el nombre de la entidad encerrado dentro de un **rectángulo**.



- **Entidades débiles.**

Son aquellas **cuya existencia depende de la existencia de otras instancias de entidad.**

En el modelo E/R una entidad débil se representa con el nombre de la entidad encerrado en un **rectángulo doble**.

Por ejemplo, consideremos las entidades **EDIFICIO** y **AULA**. Supongamos que puede haber aulas identificadas con la misma numeración pero en edificios diferentes.

Otro ejemplo de entidad débil es la sucursal bancaria que depende del banco al que pertenece y su código identificativo incluye al código del banco del que depende.



Entidad Débil: Es un tipo de entidad cuyas propiedades o atributos no la identifican completamente, sino que sólo la identifican de forma parcial. Esta entidad debe participar en una relación que ayude a identificarla.

BD03. Interpretación de diagramas Entidad/Relación

Tipos de dependencias

- **Dependencia en existencia.**

Si desaparece una instancia de entidad fuerte desaparecerán las instancias de entidad débiles que dependan de la primera.

La representación de este tipo de dependencia incluirá una E en el interior de la relación débil.

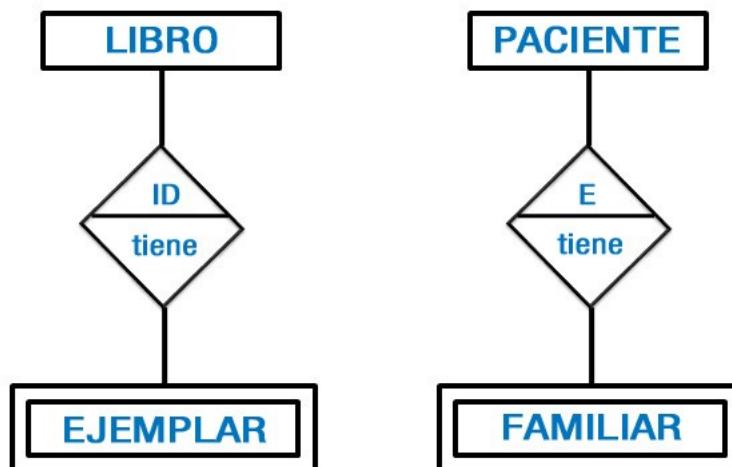
Ejemplo: Cada Paciente tiene un número indeterminado de Familiares, dichos Familiares lo son del Paciente, necesitan de su existencia pero SÍ pueden identificarse por sí mismos.

- **Dependencia en identificación.**

Debe darse una dependencia en existencia y además, una ocurrencia de la entidad débil NO puede identificarse por sí misma, debiendo hacerse mediante la clave de la entidad fuerte asociada.

La representación de este tipo de dependencia incluirá una ID en el interior de la relación débil.

Ejemplo: Cada Libro tiene un número indeterminado de Ejemplares, así el Ejemplar lo es de un Libro, del que toma su identificador y lo une con el propio: ejemplar 1 del Quijote, ejemplar 2 del Quijote, etc.



4. Atributos.

¿Cómo guardamos información de cada entidad? A través de sus atributos. Las entidades se representan con su conjunto de atributos. Éstos describen características o propiedades que posee cada miembro de un conjunto de entidades. El mismo atributo establecido para una entidad concreta, por ejemplo el atributo fecha de nacimiento para la entidad PACIENTE, almacenará información parecida para cada ocurrencia de entidad (es decir para cada paciente). Pero, cada ocurrencia de entidad tendrá su propio valor para cada atributo.

Atributo: Cada una de las propiedades o características de un tipo de entidad o de relación se denomina atributo; los atributos toman valores de uno o varios dominios.

Por tanto, un atributo se utilizará para guardar información sobre alguna característica o propiedad de una entidad o relación. Ejemplos de atributos pueden ser: **altura, color, peso, DNI, fecha**, etc. todo dependerá de la información que sea necesaria almacenar.



En el modelo Entidad/Relación los atributos de una entidad son representados mediante el nombre del atributo rodeado por una elipse. La elipse se conecta con la entidad mediante una línea recta. Cada atributo debe tener un nombre único que haga referencia al contenido de dicho atributo. Los nombres de los atributos se deben escribir en letra minúscula. En el gráfico se representan algunos de los atributos para la entidad **PACIENTE**.

BD03. Interpretación de diagramas Entidad/Relación

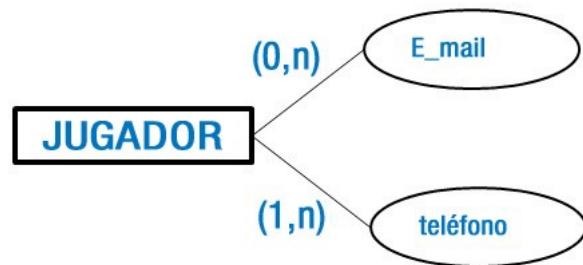
Al conjunto de valores permitidos para un atributo se le denomina **dominio**. Todos los posibles valores que puede tomar un atributo deberán estar dentro del dominio. Varios atributos pueden estar definidos dentro del mismo dominio. Por ejemplo, los atributos nombre, apellido primero y apellido segundo de la entidad **PACIENTE**, están definidos dentro del dominio de cadenas de caracteres de una determinada longitud.

Aunque los dominios suelen ser amplios (números enteros, reales, cadenas de caracteres, etc.), a la hora de llevar a cabo el desarrollo de una base de datos, es mejor establecer unos límites adecuados para que el sistema gestor de la base de datos lleve a cabo las verificaciones oportunas en los datos que se almacenen, garantizando así la integridad de éstos.

4.1. Tipos de atributos.

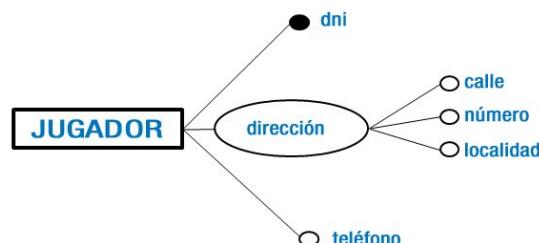
¿Todos los atributos son iguales? Claro que no. Existen varias características que hacen que los atributos asociados a una entidad o relación sean diferentes, los clasificaremos según varios criterios.

- **Atributos obligatorios y opcionales.**
 - **Atributo obligatorio:** es aquél que ha de estar siempre definido para una entidad o relación. Por ejemplo, para la entidad **JUGADOR** será necesario tener algún atributo que identifique cada ocurrencia de entidad, podría ser su DNI. Una clave o llave es un atributo obligatorio.
 - **Atributo opcional:** es aquél que podría ser definido o no para la entidad. Es decir, puede haber ocurrencias de entidad para las que ese atributo no esté definido o no tenga valor.
- **Atómicos o compuestos.**
 - **Atributo simple o atómico:** es un atributo que no puede dividirse en otras partes o atributos, presenta un único elemento. No es posible extraer de este atributo partes más pequeñas que puedan tener significado. Un ejemplo de este tipo de atributos podría ser el atributo **dni** de la entidad **JUGADOR** del gráfico.
 - **Atributo compuesto:** son atributos que pueden ser divididos en subpartes, éstas constituirán otros atributos con significado propio. Por ejemplo, la dirección del jugador podría considerarse como un atributo compuesto por la calle, el número y la localidad.
- **Atributos monovaluados o multivaluados.**
 - **Atributo monovaluado:** es aquél que tiene un único valor para cada ocurrencia de entidad. Un ejemplo de este tipo de atributos es el **dni**.
 - **Atributo multivaluado:** es aquél que puede tomar diferentes valores para cada ocurrencia de entidad. Por ejemplo, la dirección de e-mail de un empleado podría tomar varios valores para alguien que posea varias cuentas de correo. En este tipo de atributos hay que tener en cuenta los siguientes conceptos:
 - La **cardinalidad de un atributo** indica número mínimo y número máximo de valores que puede tomar para cada ejemplar de la entidad o relación a la que pertenece.
 - La **cardinalidad mínima** indica la cantidad de valores del atributo que debe existir para que la entidad sea válida. Este número casi siempre es 0 ó 1. Si es 0, el atributo podría no contener ningún valor y si es 1, el atributo debe tener un valor.
 - La **cardinalidad máxima** indica la cantidad máxima de valores del atributo que puede tener la entidad. Por lo general es 1 o n. Si es 1, el atributo no puede tener más que un valor, si es n, el atributo puede tener múltiples valores y no se especifica la cantidad absoluta.



El atributo **E_mail** de la figura, puede ser opcional y no contener ningún valor, o bien, almacenar varias cuentas de correo electrónico de un jugador. Como ves, la cardinalidad representada en la imagen es (0,n).

- **Atributos derivados, calculados o almacenados:** el valor de este tipo de atributos puede ser obtenido del valor o valores de otros atributos relacionados. Un ejemplo clásico de atributo derivado o calculado es la edad. Si se ha almacenado en algún atributo la fecha de nacimiento, la edad es un valor calculable a partir de dicha fecha. No debemos almacenar la edad ya que es un valor variable en el tiempo. En su lugar almacenaremos la fecha de nacimiento y la edad se obtendrá a partir de ella.



4.2. Claves.

En el apartado anterior hablábamos de un tipo de atributo especial obligatorio, **las claves o llaves**. Ahora es el momento de abordar con mayor detalle este concepto.

Identificación de ocurrencias de entidad o relación

Está claro que es necesario identificar correctamente cada ocurrencia de entidad o relación, de este modo el tratamiento de la información que se almacena podrá realizarse adecuadamente.

Esta distinción podría llevarse a cabo tomando todos los valores de todos los atributos de una entidad o relación. Pero, en algunas ocasiones, sabemos que puede no ser necesario utilizar todos, bastando con un subconjunto de ellos. Aunque puede ocurrir que ese subconjunto tenga idénticos valores para varias entidades, por lo que cualquier subconjunto no será válido.

Identificación de valores únicos

Por tanto, los valores de los atributos de una entidad deben ser tales que permitan **identificar únicamente** a la entidad. En otras palabras, no se permite que ningún par de entidades tengan exactamente los mismos valores de sus atributos.

Conceptos asociados

- **Superclave (Superllave)**: Es cualquier conjunto de atributos que permite identificar de forma única a una ocurrencia de entidad. Una superclave puede tener atributos no obligatorios, es decir, que no identificarían por si solos una ocurrencia de entidad.
- **Clave candidata**: Si de una superclave no es posible obtener ningún subconjunto que sea a su vez superclave, decimos que dicha superclave es clave candidata. Para elegir las claves candidatas nos basamos en su dominio y tendremos en cuenta lo siguiente:
 - Sus valores deben ser conocidos, es decir, distinto de nulos. La memoria que ocupen debe ser la menor posible.
 - La codificación sencilla.
 - El contenido de sus valores no deben variar.
- **Clave primaria (Primary Key)**: También llamada llave primaria o clave principal. De todas las claves candidatas, el diseñador de la base de datos ha de escoger una, que se denominará clave principal o clave primaria. La clave primaria es un atributo o conjunto de ellos, que toman valores únicos y distintos para cada ocurrencia de entidad, identificándola únicamente. No puede contener valores nulos.

Criterios de elección de clave primaria

- Elegir la de menor longitud
- Elegir las simples sobre las compuestas
- Numéricas sobre no numéricas
- Codificadas sobre no codificadas
- Las de ámbito local sobre las de ámbito más general

BD03. Interpretación de diagramas Entidad/Relación

¿Qué pasa con el resto de las claves candidatas?

Una vez elegida la clave primaria, las **restantes** claves candidatas son las denominadas **Claves alternativas o secundarias**.

Representación gráfica

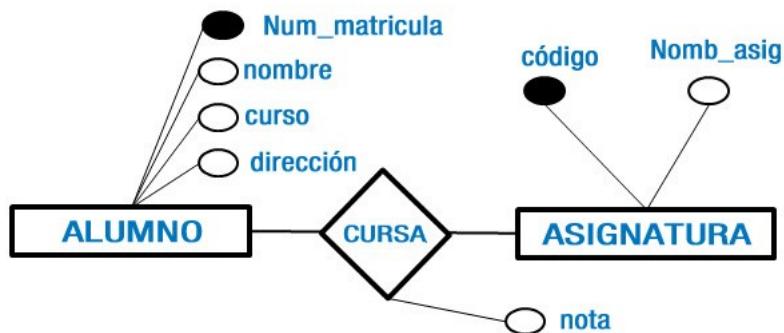
La representación en el modelo Entidad/Relación de las claves primarias puede realizarse de dos formas:

- Si se utilizan elipses para representar los atributos, se subrayarán aquellos que formen la clave primaria.
- Si se utilizan círculos para representar los atributos, se utilizará un círculo negro en aquellos que formen la clave primaria.



4.3. Atributos de una relación.

Una relación puede también tener atributos que la describan. Para ilustrar esta situación, observa el siguiente ejemplo.



Consideremos la relación **CURSA** entre las entidades **ALUMNO** y **ASIGNATURA**. Podríamos asociar a la relación **CURSA** un atributo **nota** para especificar la nota que ha obtenido un alumno/a en una determinada asignatura.

Otro ejemplo típico son las relaciones que representan históricos. Este tipo de relaciones suele constar de datos como fecha y hora. Cuando se emite una factura a un cliente o se le facilita un duplicado de la misma, es necesario registrar el momento en el que se ha realizado dicha acción. Para ello, habrá que crear un atributo asociado a la relación entre la entidad **CLIENTE** y **FACTURA** que se encargue de guardar la fecha de emisión.

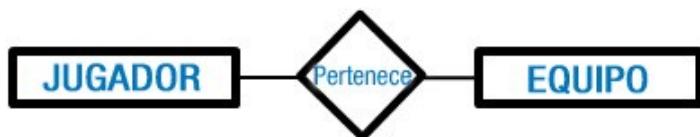
En el modelo Entidad/Relación la representación de atributos asociados a relaciones es exactamente igual a la que utilizábamos para entidades. Podremos utilizar una elipse con el nombre del atributo en su interior, conectada con una línea a la relación, o bien, un círculo blanco conectado con una línea a la relación y junto a él, el nombre del atributo. En el gráfico puedes ver esta segunda representación.

5. Relaciones.

¿Cómo interactúan entre sí las entidades? A través de las relaciones. La relación o interrelación es un elemento del modelo Entidad/Relación que permite relacionar datos entre sí. En una relación se asocia un elemento de una entidad con otro de otra entidad.

Relación: es una asociación entre diferentes entidades. En una relación no pueden aparecer dos veces relacionadas las mismas ocurrencias de entidad.

La representación gráfica en el modelo Entidad/Relación corresponde a un rombo en cuyo interior se encuentra inscrito el nombre de la relación expresado con un verbo. El rombo estará conectado con las entidades a las que relaciona, mediante líneas rectas, que podrán o no acabar en punta de flecha según el tipo de relación. Al interpretarlo sólo es necesario leerlo de izquierda a derecha o de arriba a abajo. Así, en el ejemplo, leeríamos: "jugador pertenece a equipo", y "a un equipo pertenecen jugadores" pero aún nos falta incluir algo fundamental: la cardinalidad.



En algunas ocasiones, es interesante que en las líneas que conectan las entidades con la relación, se indique el papel o rol que desempeña cada entidad en la relación. Como se verá más adelante, los papeles o roles son especialmente útiles en relaciones reflexivas.

Para describir y definir adecuadamente las relaciones existentes entre entidades, es imprescindible conocer los siguientes conceptos:

- Grado de la relación.
- Cardinalidad de la relación.
- Cardinalidades de las entidades.

A continuación desarrollamos cada uno de ellos.

BD03. Interpretación de diagramas Entidad/Relación

5.1. Grado de una relación.

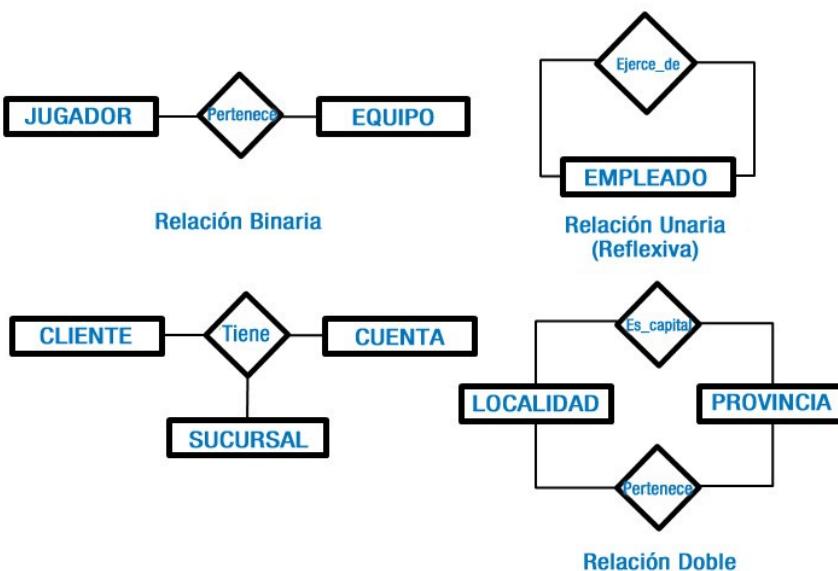
¿Pueden intervenir varias entidades en una misma relación? Claro que sí, en una relación puede intervenir una única entidad o varias.

Grado de una relación: número de entidades que participan en una relación.

En función del grado se pueden establecer diferentes tipos de relaciones:

- **Relación Unaria o de grado 1:** Es aquella relación en la que participa una única entidad. También llamadas reflexivas o recursivas.
- **Relación Binaria o de grado 2:** Es aquella relación en la que participan dos entidades. En general, tanto en una primera aproximación, como en los sucesivos refinamientos, el esquema conceptual de la base de datos buscará tener sólo este tipo de relaciones.
- **Relación Ternaria o de grado 3:** Es aquella relación en la que participan tres entidades al mismo tiempo.
- **Relación N-aria o de grado n:** Es aquella relación que involucra n entidades. Este tipo de relaciones no son usuales y deben ser simplificadas hacia relaciones de menor grado.
- **Relación doble:** ocurre cuando dos entidades están relacionadas a través de dos relaciones. Este tipo de relaciones son complejas de manejar.

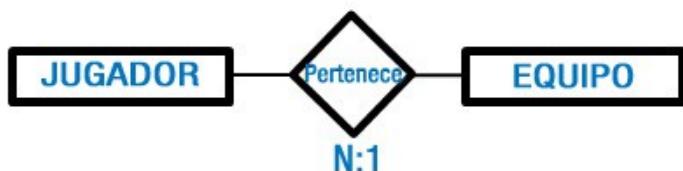
En este gráfico puedes observar cada uno de los tipos de relaciones en función de su grado y su representación gráfica en el modelo Entidad/Relación.



5.2. Cardinalidad de las relaciones.

¿Qué es eso de la cardinalidad? En matemáticas, el cardinal de un conjunto es el número de elementos que lo forman. Este concepto puede extrapolarse a las relaciones con las que estamos tratando.

Cardinalidad de una relación: Es el número máximo de ocurrencias de cada entidad que pueden intervenir en una ocurrencia de relación. La cardinalidad vendrá expresada siempre para relaciones entre dos entidades. Dependiendo del número de ocurrencias de cada una de las entidades pueden existir relaciones uno a uno, uno a muchos, muchos a uno y muchos a muchos.



Observa el siguiente ejemplo, la cardinalidad indicará el número de ocurrencias de la entidad **JUGADOR** que se relacionan con cada ocurrencia de la entidad **EQUIPO** y viceversa. Podríamos hacer la siguiente lectura: un jugador pertenece a un equipo y a un equipo pueden pertenecer varios jugadores.

Una posible representación de la cardinalidad de las relaciones es la que hemos visto en el ejemplo anterior. Podríamos representar el resto de cardinalidades mediante las etiquetas 1:1, 1:N, N:1, M:N que se leerían respectivamente: uno a uno, uno a muchos, muchos a uno y muchos a muchos.

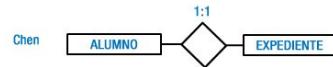
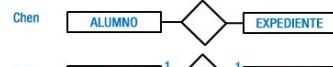
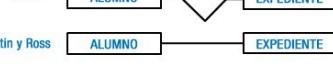
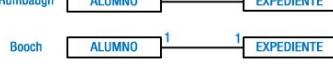
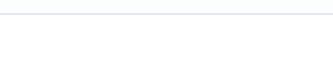
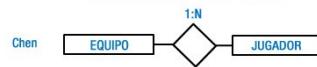
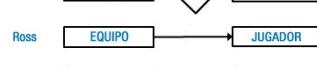
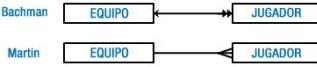
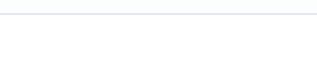
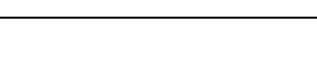
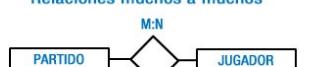
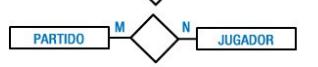
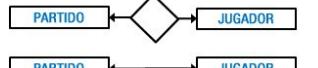
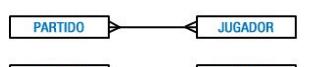
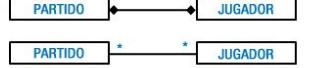
BD03. Interpretación de diagramas Entidad/Relación

Tipos de cardinalidades

- **Relaciones uno a uno (1:1).** Sean las entidades A y B, una instancia u ocurrencia de la entidad A se relaciona únicamente con otra instancia de la entidad B y viceversa. Por ejemplo, para cada ocurrencia de la entidad **ALUMNO** sólo habrá una ocurrencia relacionada de la entidad **EXPEDIENTE** y viceversa. O lo que es lo mismo, un alumno tiene un expediente asociado y un expediente sólo pertenece a un único alumno.
- **Relaciones uno a muchos (1:N).** Sean las entidades A y B, una ocurrencia de la entidad A se relaciona con muchas ocurrencias de la entidad B y una ocurrencia de la entidad B sólo estará relacionada con una única ocurrencia de la entidad A. Por ejemplo, para cada ocurrencia de la entidad **DOCENTE** puede haber varias ocurrencias de la entidad **ASIGNATURA** y para varias ocurrencias de la entidad **ASIGNATURA** sólo habrá una ocurrencia relacionada de la entidad **DOCENTE** (si se establece que una asignatura sólo puede ser impartida por un único docente). O lo que es lo mismo, un docente puede impartir varias asignaturas y una asignatura sólo puede ser impartida por un único docente.
- **Relaciones muchos a uno (N:1).** Sean las entidades A y B, una ocurrencia de la entidad A está asociada con una única ocurrencia de la entidad B y un ejemplar de la entidad B está relacionado con muchas ocurrencias de la entidad A. Por ejemplo, Un **JUGADOR** pertenece a un único **EQUIPO** y a un **EQUIPO** pueden pertenecer muchos jugadores.
- **Relaciones muchos a muchos (M:N).** Sean las entidades A y B, un ejemplar de la entidad A está relacionado con muchas ocurrencias de la entidad B y viceversa. Por ejemplo, un alumno puede estar matriculado en varias asignaturas y en una asignatura pueden estar matriculados varios alumnos.

Representación

La cardinalidad de las relaciones puede representarse de varias maneras en los esquemas del modelo Entidad/Relación. A continuación, te ofrecemos un resumen de las notaciones clasificadas por autores, más empleadas en la representación de cardinalidad de relaciones.

Relaciones uno a uno.	Relaciones uno a muchos.	Relaciones muchos a muchos.
<p>Relaciones uno a uno</p>       	<p>Relaciones uno a muchos</p>        	<p>Relaciones muchos a muchos</p>       

5.3. Cardinalidad de las entidades.

Si existe cardinalidad en las relaciones, supondrás que también existe para las entidades. Estás en lo cierto, la cardinalidad con la que una entidad participa en una relación especifica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ejemplar de dicha entidad. Indica el número de relaciones en las que una entidad puede aparecer.

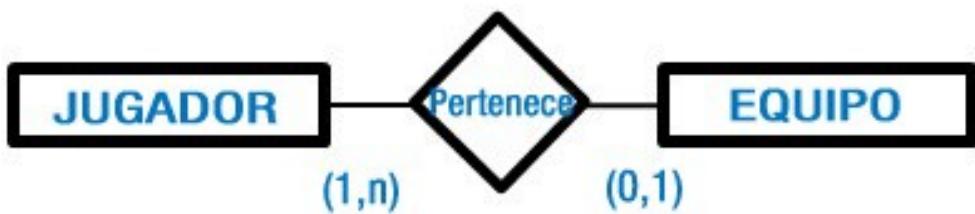
Sean las entidades A y B, la participación de la entidad A en una relación es obligatoria (total) si la existencia de cada una de sus ocurrencias necesita como mínimo de una ocurrencia de la entidad B (ver figura). En caso contrario, la participación es opcional (parcial).

La cardinalidad de una entidad se representa con el número mínimo y máximo de correspondencias en las que puede tomar parte cada ejemplar de dicha entidad, entre paréntesis. Su representación gráfica será, por tanto, una etiqueta del tipo (0,1), (1,1), (0,N) o (1,N). El significado del primer y segundo elemento del paréntesis corresponde a (cardinalidad mínima, cardinalidad máxima):

- **Cardinalidad mínima.** Indica el número mínimo de asociaciones en las que aparecerá cada ocurrencia de la entidad (el valor que se anota es de cero o uno, aunque tenga una cardinalidad mínima de más de uno, se indica sólo un uno). El valor 0 se pondrá cuando la participación de la entidad sea opcional.
- **Cardinalidad máxima.** Indica el número máximo de relaciones en las que puede aparecer cada ocurrencia de la entidad. Puede ser uno, otro valor concreto mayor que uno (tres por ejemplo) o muchos (se representa con n).

BD03. Interpretación de diagramas Entidad/Relación

Veámoslo más claro a través del siguiente ejemplo: un **JUGADOR** pertenece como mínimo a ningún **EQUIPO** y como máximo a uno (0,1) y, por otra parte, a un **EQUIPO** pertenece como mínimo un **JUGADOR** y como máximo varios (1,n). Como puedes ver, la cardinalidad (0,1) de **JUGADOR** se ha colocado junto a la entidad **EQUIPO** para representar que un jugador puede no pertenecer a ningún equipo o como máximo a uno. Para la cardinalidad de **EQUIPO** ocurre igual, se coloca su cardinalidad junto a la entidad **JUGADOR** para expresar que en un equipo hay mínimo un jugador y máximo varios.



Ten en cuenta que cuando se representa la cardinalidad de una entidad, el paréntesis y sus valores han de colocarse junto a la entidad con la que se relaciona. Es decir en el lado opuesto a la relación.

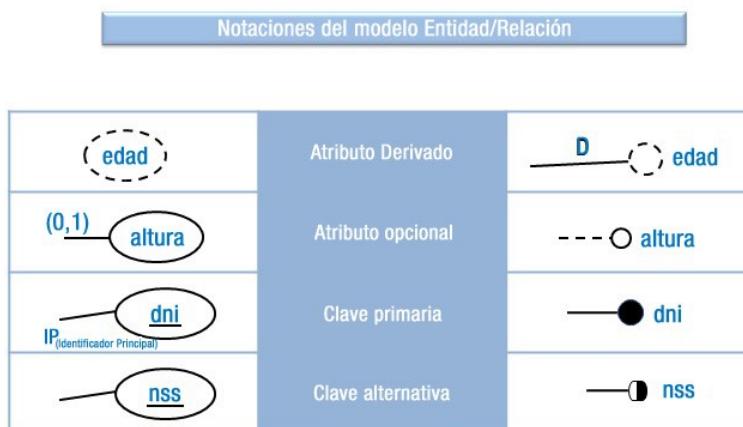
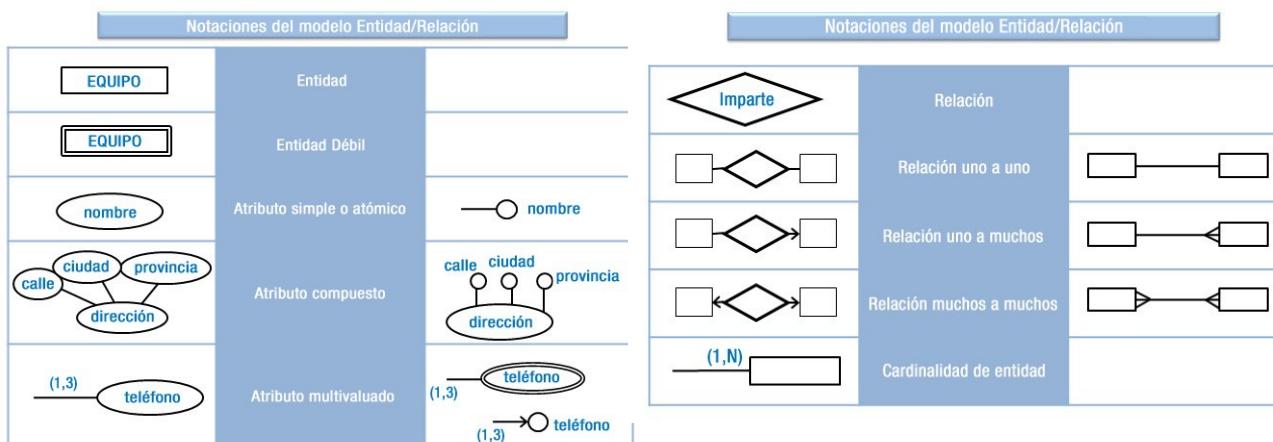
La cardinalidad de entidades también puede representarse en el modelo Entidad/Relación con la notación que se representa en la imagen de la derecha. Por tanto, el anterior ejemplo quedaría representado así:



6. Simbología del modelo E/R.

¿Recuerdas todos y cada uno de los símbolos que hemos utilizado a lo largo de esta unidad? Es probable que no. Para facilitar tu aprendizaje, te ofrecemos a continuación un resumen básico de los símbolos utilizados en el modelo Entidad/Relación. Verás que existen diferentes maneras de representar los mismos elementos, las que aquí se resumen te servirán para interpretar la gran mayoría de esquemas con los que te puedas encontrar.

Resumen básico de la simbología del modelo Entidad/Relación.



7. Modelos E/R extendido.

Hemos visto que a través del modelo Entidad/Relación se pueden modelar la gran mayoría de los requisitos que una base de datos debe cumplir. Pero existen algunos que ofrecen especial dificultad a la hora de representarlos a través de la simbología tradicional del modelo E/R. Para solucionar este problema, en el modelo Entidad/Relación Extendido se han incorporado nuevas extensiones que permiten mejorar la capacidad para representar circunstancias especiales. Estas extensiones intentan eliminar elementos de difícil o incompleta representación a través de la simbología existente, como por ejemplo relaciones con cardinalidad N:M, o la no identificación clara de entidades.

A continuación, se detallan estas nuevas características que convierten al modelo E/R tradicional en el modelo Entidad/Relación Extendido, como son: tipos de restricciones sobre las relaciones, especialización, generalización, conjuntos de entidades de nivel más alto y más bajo, herencia de atributos y agregación.

7.1. Restricciones en las relaciones.

La primera extensión que el modelo Entidad/Relación Extendido incluye, se centra en la representación de una serie de restricciones sobre las relaciones y sus ejemplares, vamos a describirlas:

a) Restricción de exclusividad.

Cuando existe una entidad que participa en dos o más relaciones y cada ocurrencia de dicha entidad sólo puede pertenecer a una de las relaciones únicamente, decimos que existe una restricción de exclusividad. Si la ocurrencia de entidad pertenece a una de las relaciones, no podrá formar parte de la otra. O se produce una relación o se produce otra pero nunca ambas a la vez.

Por ejemplo, supongamos que un músico puede dirigir una orquesta o tocar en ella, pero no puede hacer las dos cosas simultáneamente. Existirán por tanto, dos relaciones **dirige** y **toca**, entre las entidades **MUSICO** y **ORQUESTA**, estableciéndose una relación de exclusividad entre ellas.

La representación gráfica en el modelo Entidad/Relación Extendido de una restricción de exclusividad se realiza mediante un arco que engloba a todas aquellas relaciones que son exclusivas.

b) Restricción de exclusión.

Este tipo de restricción se produce cuando las ocurrencias de las entidades sólo pueden asociarse utilizando una única relación.

Pongamos un ejemplo, supongamos que un monitor puede impartir diferentes cursos de perfeccionamiento para monitores, y que éste puede a su vez recibirlos. Pero si un monitor imparte un determinado curso, no podrá estar recibiendo simultáneamente y viceversa.

Se establecerá, por tanto, una restricción de exclusión que se representa mediante una línea discontinua entre las dos relaciones, tal y como se muestra en el ejemplo al final.

BD03. Interpretación de diagramas Entidad/Relación

c) Restricción de inclusividad.

Este tipo de restricciones se aplican cuando es necesario modelar situaciones en las que para que dos ocurrencias de entidad se asocien a través de una relación, tengan que haberlo estado antes a través de otra relación.

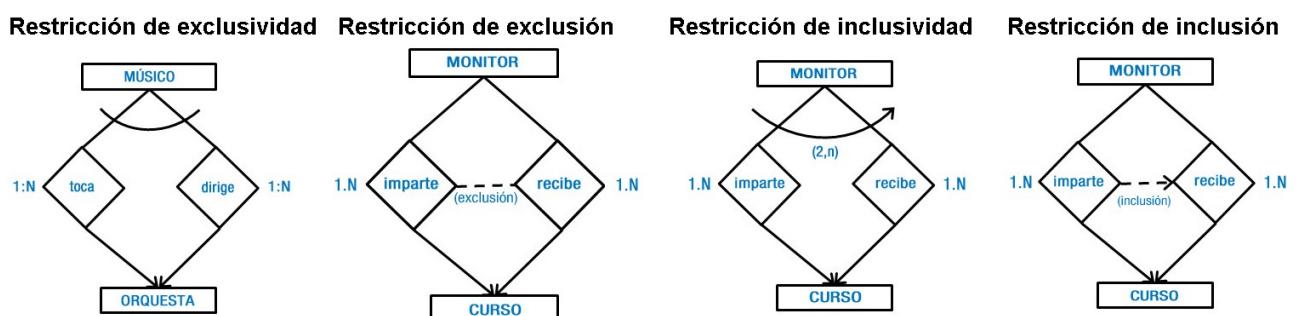
Siguiendo con el ejemplo anterior, supongamos que para que un monitor pueda impartir cursos de cocina sea necesario que reciba previamente dos cursos: nutrición y primeros auxilios. Como puedes ver, es posible que los cursos que el monitor deba recibir no tengan que ser los mismos que luego pueda impartir. Aplicando una restricción de inclusividad entre las relaciones **imparte** y **recibe**, estaremos indicando que cualquier ocurrencia de la entidad **MONITOR** que participa en una de las relaciones (**imparte**) tiene que participar obligatoriamente en la otra (**recibe**).

Se representará mediante un arco acabado en flecha, que partirá desde la relación que ha de cumplirse primero hacia la otra relación. Se indicará junto al arco la cardinalidad mínima y máxima de dicha restricción de inclusividad. En el ejemplo, (2,n) indica que un monitor ha de recibir 2 cursos antes de poder impartir varios.

d) Restricción de inclusión.

En algunas ocasiones aplicar una restricción de inclusividad no representa totalmente la realidad a modelar, entonces se hace necesario aplicar una restricción de inclusión que es aún más fuerte.

En nuestro ejemplo, si hemos de modelar que un monitor pueda impartir un curso, si previamente lo ha recibido, entonces tendremos que aplicar una restricción de inclusión. Con ella toda ocurrencia de la entidad **MONITOR** que esté asociada a una ocurrencia determinada de la entidad **CURSO**, a través de la relación **imparte**, ha de estar unida a la misma ocurrencia de la entidad **CURSO** a través de la relación **recibe**.



7.2. Generalización y especialización.

La segunda extensión incorporada en el modelo Entidad/Relación Extendido se centra en nuevos tipos de relaciones que van a permitir modelar la realidad de una manera más fiel. Estos nuevos tipos de relación reciben el nombre de jerarquías y se basan en los conceptos de generalización, especialización y herencia.



Supertipos y subtipos

Cuando estamos diseñando una base de datos puede que nos encontremos con conjuntos de entidades que posean características comunes, lo que permitiría crear un tipo de entidad de nivel más alto que englobe dichas características. Y a su vez, puede que necesitemos dividir un conjunto de entidades en diferentes subgrupos de entidades por tener éstas, características diferenciadoras. Este proceso de refinamiento ascendente/descendente, permite expresar mediante la generalización la existencia de tipos de entidades de nivel superior que engloban a conjuntos de entidades de nivel inferior. A los conjuntos de entidades de nivel superior también se les denomina **superclase o supertipo**. A los conjuntos de entidades de nivel inferior se les denomina **subclase o subtipo**.

Por tanto, existirá la posibilidad de realizar una especialización de una superclase en subclases, y análogamente, establecer una generalización de las subclases en superclases. La generalización es la reunión en una superclase o supertipo de entidad de una serie de subclases o subtipos de entidades, que poseen características comunes. Las subclases tendrán otras características que las diferenciarán entre ellas.

¿Cómo detectamos una generalización?

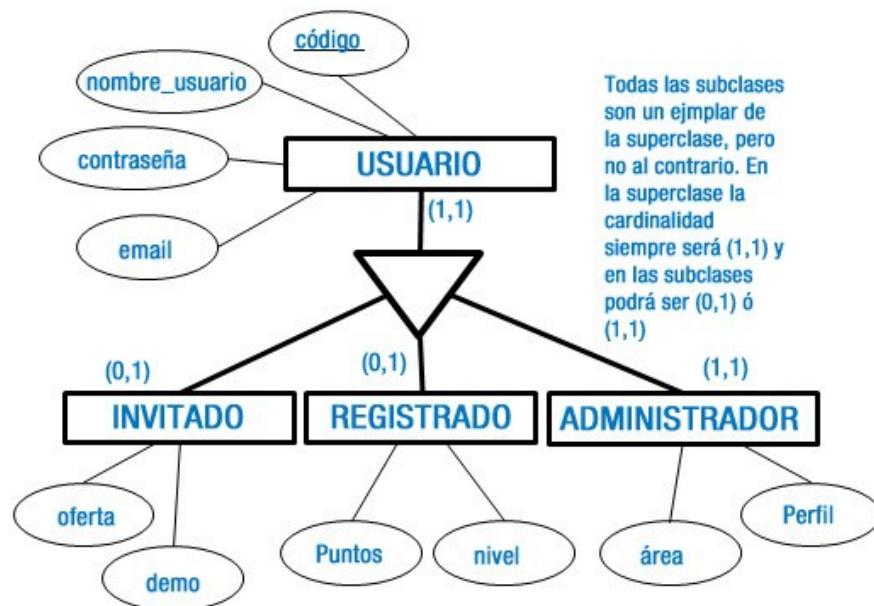
Podremos identificar una generalización cuando encontramos una serie de atributos comunes a un conjunto de entidades, y otros atributos que sean específicos. Los atributos comunes conforman la superclase o supertipo y los atributos específicos la subclase o subtipo.

Las jerarquías se caracterizan por un concepto que hemos de tener en cuenta, la herencia. A través de la herencia los atributos de una superclase de entidad son heredados por las subclases. Si una superclase interviene en una relación, las subclases también lo harán.

¿Cómo se representa una generalización o especialización?

Existen varias notaciones, pero hemos de convenir que la relación que se establece entre una superclase de entidad y todos sus subtipos se expresa a través de las palabras ES UN, o en notación inglesa IS A, que correspondería con ES UN TIPO DE. Partiendo de este punto, una jerarquía se representa mediante un triángulo invertido, sobre él quedará la entidad superclase y conectadas a él a través de líneas rectas, las subclases.

BD03. Interpretación de diagramas Entidad/Relación



En el ejemplo de la imagen, las subclases **INVITADO**, **REGISTRADO** y **ADMINISTRADOR** constituyen subclases de la superclase **USUARIO**. Cada una de ellas aporta sus propias características y heredan las pertenecientes a su superclase.

Restricciones semánticas

Una generalización/especialización podrá tener las siguientes restricciones semánticas:

- **Totalidad:** una generalización/especialización será total si todo ejemplar de la superclase pertenece a alguna de las subclases.
- **Parcialidad:** una generalización/especialización será parcial si no todos los ejemplares de la superclase pertenecen a alguna de las subclases.
- **Solapamiento:** una generalización/especialización presentará solapamiento si un mismo ejemplar de la superclase puede pertenecer a más de una subclase.
- **Exclusividad:** una generalización/especialización presentará exclusividad si un mismo ejemplar de la superclase pertenece sólo a una subclase.

7.3. Agregación.

Abordamos ahora la tercera de las extensiones del modelo Entidad/Relación Extendido, la **Agregación**.

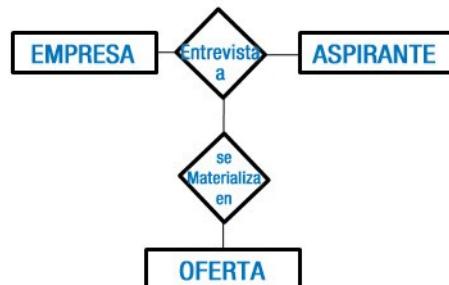
En el modelo Entidad/Relación no es posible representar relaciones entre relaciones. La agregación es una abstracción a través de la cual las relaciones se tratan como entidades de nivel más alto, siendo utilizada para expresar relaciones entre relaciones o entre entidades y relaciones.

Ejemplo

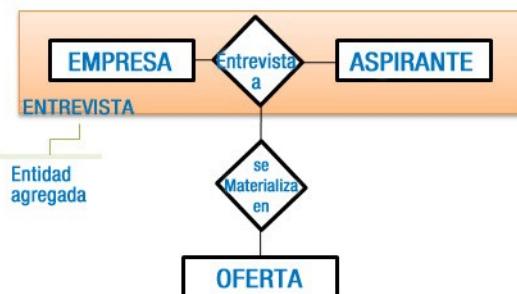
Supongamos un ejemplo en el que hemos de modelar la siguiente situación: una empresa de selección de personal realiza entrevistas a diferentes aspirantes. Puede ser que, de algunas de estas entrevistas a aspirantes, se derive una oferta de empleo, o no. En el siguiente gráfico se representan tres soluciones,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



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



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



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

Como has podido observar, la representación gráfica de una agregación se caracteriza por englobar con un rectángulo las entidades y relación a abstraer. De este modo, se crea una nueva entidad agregada que puede participar en otras relaciones con otras entidades. En este tipo de relación especial de agregación, la cardinalidad máxima y mínima de la entidad agregada siempre será (1,1) no indicándose por ello en el esquema.

BD03. Interpretación de diagramas Entidad/Relación

Tipos de agregaciones

- **Compuesto/componente.**

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

Teniendo esto en cuenta, esta abstracción permite representar que un todo o agregado se obtiene por la unión de diversas partes o componentes que pueden ser tipos de entidades distintas y que juegan diferentes roles en la agregación.

- **Miembro/Colección.**

Un todo se obtiene por la unión de diversas partes del mismo tipo y que desempeñan el mismo papel en la agregación.

Teniendo esto en cuenta, esta abstracción permite representar un todo o agregado como una colección de miembros, todos de un mismo tipo de entidad y todos jugando el mismo rol.

Esta agregación puede incluir una restricción de orden de los miembros dentro de la colección (indicando el atributo de ordenación). Es decir, permite establecer un orden entre las partes.



8. Elaboración de diagramas E/R.

Llegados a este punto, te surgirán varias dudas ¿Cómo creo un diagrama E/R? ¿Por dónde empiezo? ¿Y qué puedo hacer con todo lo visto? Son cuestiones totalmente normales cuando se comienza, no te preocupes, vamos a darte una serie de orientaciones para que puedas aplicar todos los conceptos aprendidos hasta ahora en la elaboración de diagramas Entidad/Relación.

Sabemos que en la fase de diseño conceptual de la base de datos, en la que nos encontramos, hemos de generar el diagrama E/R que representará de manera más sencilla el problema real a modelar, independientemente del Sistema Gestor de Base de Datos. Este esquema será como un plano que facilite la comprensión y solución del problema. Este diagrama estará compuesto por la representación gráfica, a través de la simbología vista, de los requisitos o condiciones que se derivan del problema a modelar.

Saltarnos este paso en el proceso de creación e implementación de una base de datos, supondría pérdida de información. Por lo que esta fase, requerirá de la creación de uno o varios esquemas previos más cercanos al mundo real, antes del paso a tablas del modelo relacional.

Te darás cuenta que, como en la programación, la práctica es fundamental. Los diagramas no siempre se crean del mismo modo y, en ocasiones, hay que retocarlos e incluso rehacerlos. A través de la resolución de diferentes problemas y la elaboración de múltiples diagramas, obtendrás la destreza necesaria para generar esquemas que garanticen una posterior y correcta conversión del modelo Entidad/Relación al modelo Relacional.

Lo primero que hemos de tener a nuestra disposición para poder generar un diagrama E/R adecuado es el conjunto de requerimientos, requisitos o condiciones que nuestra base de datos ha de cumplir. Es lo que se denomina el documento de especificación de requerimientos: el enunciado del problema a modelar.

8.1. Identificación de entidades y relaciones.

¿Cómo empezamos?

- a) **Identificación de entidades:** Es un proceso bastante intuitivo. Para localizar aquellos elementos que serán las entidades de nuestro esquema, analizaremos la especificación de requerimientos en busca de nombres o sustantivos. Si estos nombres se refieren a objetos importantes dentro del problema probablemente serán entidades. No es un proceso automático y es necesario aplicar la lógica y los criterios comentados para determinar si es, o no, una entidad. Tendremos en cuenta que nombres referidos a características, cualidades o propiedades no se convertirán en entidades.

Otra forma de identificar entidades es localizando objetos o elementos que existen por sí mismos, acerca de los cuales interesa guardar información. La información que guardemos de la entidad se convertirá en los atributos de la entidad. Por ejemplo: VEHICULO, PIEZA, etc. En otras ocasiones, la localización de varias características o propiedades puede dejar ver la existencia de una entidad.

¿Esto puede ser una entidad o no? Es una pregunta que se repite mucho cuando estamos en esta etapa. Algunos autores indican que para poder considerarse como entidad se deben cumplir tres reglas:

- Existencia propia.
- Cada ejemplar de un tipo de entidad debe poder ser diferenciado del resto de ejemplares.
- Todos los ejemplares de un tipo de entidad deben tener las mismas propiedades.

El número de entidades obtenidas debe ser manejable y según se vayan identificando se les otorgará nombres, preferiblemente en mayúsculas y en singular, representativos de su significado o función. De esta manera el diagrama será cada vez más legible.

- b) **Identificación de relaciones:** Localizadas las entidades, debemos establecer qué relación existe entre ellas. Para ello, analizaremos de nuevo el documento de especificación de requerimientos en busca de verbos o expresiones verbales que conecten unas entidades con otras. De nuevo hay que aplicar la lógica para no repetir relaciones o establecer relaciones que no lo son y determinar relaciones que tengan significado. En la gran mayoría de ocasiones encontraremos que las relaciones se establecen entre dos entidades (relaciones binarias), pero prestaremos especial atención a las relaciones entre más entidades y a las relaciones recursivas o relaciones unarias.

Cada una de las relaciones establecidas deberá tener asignado un nombre, preferiblemente un verbo en minúsculas, representativo del significado o acción de la relación.

Dependiendo de la notación elegida, el siguiente paso será la representación de la cardinalidad (mínima y máxima) de las entidades participantes en cada relación y del tipo de correspondencia de la relación (1 a 1, 1 a muchos o muchos a muchos).

8.2. Identificación de atributos, claves y jerarquías.

- a) **Identificación de atributos:** Volvemos sobre el documento de especificación de requerimientos para buscar nombres relativos a características, propiedades, identificadores o cualidades de entidades o relaciones. Resulta más sencillo si nos preguntamos ¿Qué información es necesario tener en cuenta de una u otra entidad o relación? Quizás no todos los atributos estén reflejados directamente en el documento de especificación de requerimientos, aplicando el sentido común el diseñador podrá establecerlos en algunos casos y en otros, será necesario consultar e indagar en el problema.

Tendremos en cuenta si los atributos localizados son simples o compuestos, derivados o calculados y si algún atributo o conjunto de ellos se repite en varias entidades. Si se da este último caso, deberemos detenernos y plantear la posibilidad de establecer una jerarquía de especialización, o bien, dejar las entidades tal y como han sido identificadas.

Cada atributo deberá tener asignado un nombre, preferiblemente en minúsculas, representativo de su contenido o función. Además, siempre es recomendable recopilar la siguiente información de cada atributo:

- Nombre y descripción.
- Atributos simples que lo componen, si es atributo compuesto.
- Método de cálculo, si es atributo derivado o calculado.

En el caso de encontrar atributos asociados a relaciones con cardinalidad uno a muchos, se valorará asignar ese atributo o atributos a la entidad con mayor cardinalidad participante en la relación.

- b) **Identificación de claves:** Del conjunto de atributos de una entidad se establecerán una o varias claves candidatas, escogiéndose una de ellas como clave o llave primaria de la entidad. Esta clave estará formada por uno o varios atributos que identificarán de manera única cada ocurrencia de entidad. El proceso de identificación de claves permitirá determinar la fortaleza (al menos una clave candidata) o debilidad (ninguna clave candidata) de las entidades encontradas. Se representará la existencia de esta clave primaria mediante la notación elegida para la elaboración del diagrama E/R. Del mismo modo, se deberán representar adecuadamente las entidades fuertes o débiles.
- c) **Determinación de jerarquías:** Como se ha comentado anteriormente, es probable que existan entidades con características comunes que puedan ser generalizadas en una entidad de nivel superior o superclase (jerarquía de generalización). Pero también, puede ser necesario expresar en el esquema las particularidades de diferentes ejemplares de un tipo de entidad, por lo que se crearán subclases o subtipos de una superclase o supertipo (jerarquía de especialización). Para ello, habrá que analizar con detenimiento el documento de especificación de requerimientos.

Si se identifica algún tipo de jerarquía, se deberá representar adecuadamente según el tipo de notación elegida, determinando si la jerarquía es total/parcial o exclusiva/con solapamiento.

8.3. Metodologías.

Hasta aquí, tenemos identificados los elementos necesarios para construir nuestro diagrama, pero ¿Existe alguna metodología para llevarlo a cabo? Sí, y además podremos utilizar varias. Partiremos de una versión preliminar del esquema conceptual o diagrama E/R que, tras sucesivos refinamientos, será modificado para obtener el diagrama E/R definitivo.

Las metodologías o estrategias para la elaboración del esquema conceptual

- **Metodología Descendente (Top-Down):** Se trata de partir de un esquema general e ir descomponiendo éste en niveles, cada uno de ellos con mayor número de detalles. Se parte de objetos muy abstractos, que se refinan paso a paso hasta llegar al esquema final.
- **Metodología Ascendente (Bottom-Up):** Inicialmente, se parte del nivel más bajo, los atributos. Se irán agrupando en entidades, para después ir creando las relaciones entre éstas y las posibles jerarquías hasta obtener un diagrama completo. Se parte de objetos atómicos que no pueden ser descompuestos y a continuación se obtienen abstracciones u objetos de mayor nivel de abstracción que forman el esquema.
- **Metodología Dentro-fuera (Inside-Out):** Inicialmente se comienza a desarrollar el esquema en una parte del papel y a medida que se analiza la especificación de requerimientos, se va completando con entidades y relaciones hasta ocupar todo el documento.
- **Metodología Mixta:** Es empleada en problemas complejos. Se dividen los requerimientos en subconjuntos que serán analizados independientemente. Se crea un esquema que servirá como estructura en la que irán interconectando los conceptos importantes con el resultado del análisis de los subconjuntos creados. Esta metodología utiliza las técnicas ascendente y descendente. Se aplicará la técnica descendente para dividir los requerimientos y en cada subconjunto de ellos, se aplicará la técnica ascendente.

¿Cuál de estas metodologías utilizar?

Cualquiera de ellas puede ser válida, todo dependerá de lo fácil y útil que te resulte aplicarlas. Probablemente y, casi sin ser consciente de ello, tú mismo crearás tu propia metodología combinando las existentes. Pero, como decíamos hace algunos epígrafes, la práctica es fundamental. Realizando gran cantidad de esquemas, analizándolos y llevando a cabo modificaciones en ellos es como irás refinando tu técnica de elaboración de diagramas E/R. Llegará un momento en que sólo con leer el documento de especificación de requerimientos serás capaz de ir construyendo en tu mente cómo será su representación sobre el papel, pero paciencia y ve paso a paso.

8.4. 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 ¿Y qué es la redundancia?

Redundancia: reproducción, repetición, reiteración, insistencia, reincidencia, reanudación. En bases de datos hace referencia al almacenamiento de los mismos datos varias veces en diferentes lugares.

Problemas asociados.

- **Aumento de la carga de trabajo:** al estar almacenado un dato en varios lugares, las operaciones de grabación o actualización de datos necesitan realizarse en varias ocasiones.
- **Gasto extra de espacio de almacenamiento:** al estar repetidos, los datos ocupan mayor cantidad de espacio en el medio de almacenamiento. Cuanto mayor sea la base de datos, más patente se hará esta problema.
- **Inconsistencia:** es el problema más importante ya que produce que la información no sea fiable. Se produce cuando los datos que están repetidos, no contienen los mismos valores. Es decir, se ha actualizado su valor en un lugar y en otro no, por lo que no se sabría qué dato es válido y cual erróneo.

Funcionamiento óptimo

Para que una base de datos funcione óptimamente, hay que empezar realizando un buen diseño de ella. Es imprescindible que nuestros diagramas E/R controlen la redundancia y, para ello, debemos analizar el esquema y valorar qué elementos pueden estar incorporando redundancia a nuestra solución.

¿Dónde buscamos indicios de redundancia en nuestros esquemas?

- Atributos redundantes cuyo contenido se calcula en función de otros. Un atributo derivado puede ser origen de redundancia.
- Varias entidades unidas circularmente o cíclica a través de varias relaciones, es lo que se conoce como un ciclo. En caso de existir un ciclo, deberemos tener en cuenta las siguientes condiciones, antes de poder eliminar dicha relación redundante:
 - Que el significado de las relaciones que componen el ciclo sea el mismo.
 - Que si eliminamos la relación redundante, el significado del resto de relaciones es el mismo.
 - Que si la relación eliminada tenía atributos asociados, éstos puedan ser asignados a alguna entidad participante en el esquema, sin que se pierda su significado.

Pero hay que tener en cuenta que no siempre que exista un ciclo estaremos ante una redundancia. Es necesario analizar detenidamente dicho ciclo para determinar si realmente existe o no redundancia.

BD03. Interpretación de diagramas Entidad/Relación

Apreciación final

Para finalizar, una apreciación. No toda redundancia es perjudicial.

Hay ciertas circunstancias y condiciones en las que es conveniente (sobre todo a efectos de rendimiento) introducir cierta redundancia controlada en una base de datos.

Por ejemplo, si el método de cálculo del valor de un determinado atributo derivado es complejo (varias operaciones matemáticas o de cadenas de caracteres, varios atributos implicados, etc.) y ralentiza el funcionamiento de la base de datos, quizás sea conveniente definir dicho atributo desde el principio y no considerarlo como un atributo redundante.

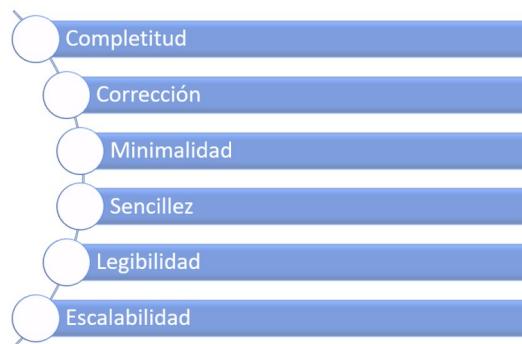
La incorporación o no de redundancia controlada dependerá de la elección que haga el diseñador.

8.5. Propiedades deseables de un diagrama E/R.

Cuando construimos un diagrama Entidad/Relación existen una serie de propiedades o características que éste debería cumplir. Quizá no se materialicen todas, pero hemos de intentar cubrir la gran mayoría de ellas. De este modo, conseguiremos que nuestros diagramas o esquemas conceptuales tengan mayor calidad.

Estas características o propiedades deseables se desglosan a continuación:

- **Completitud:** Un diagrama E/R será completo si es posible verificar que cada uno de los requerimientos está representado en dicho diagrama y viceversa, cada representación del diagrama tiene su equivalente en los requerimientos.
- **Corrección:** Un diagrama E/R será correcto si emplea de manera adecuada todos los elementos del modelo Entidad/Relación. La corrección de un diagrama puede analizarse desde dos vertientes:
 - **Corrección sintáctica:** Se producirá cuando no se produzcan representaciones erróneas en el diagrama.
 - **Corrección semántica:** Se producirá cuando las representaciones signifiquen exactamente lo que está estipulado en los requerimientos. Posibles errores semánticos serían: la utilización de un atributo en lugar de una entidad, el uso de una entidad en lugar de una relación, utilizar el mismo identificador para dos entidades o dos relaciones, indicar erróneamente alguna cardinalidad u omitirla, etc.
- **Minimalidad:** Un diagrama E/R será mínimo si se puede verificar que al eliminar algún concepto presente en el diagrama, se pierde información. Si un diagrama es redundante, no será mínimo.
- **Sencillez:** Un diagrama E/R será sencillo si representa los requerimientos de manera fácil de comprender, sin artificios complejos.
- **Legibilidad:** Un diagrama E/R será legible si puede interpretarse fácilmente. La legibilidad de un diagrama dependerá en gran medida del modo en que se disponen los diferentes elementos e interconexiones. Esta propiedad tiene mucho que ver con aspectos estéticos del diagrama.
- **Escalabilidad:** Un diagrama E/R será escalable si es capaz de incorporar posibles cambios derivados de nuevos requerimientos.



9. Paso del diagrama E/R al Modelo Relacional.

Si analizamos todo el proceso descrito hasta el momento, la fase de diseño conceptual desarrollada, y que se materializa en el diagrama E/R, permite una gran independencia de las cuestiones relativas a la implementación física de la base de datos. El tipo de SGBD, las herramientas software, las aplicaciones, lenguajes de programación o hardware disponible no afectarán, al menos hasta el momento, a los resultados de esta fase.

Nuestro esquema conceptual habrá sido revisado, modificado y probado para verificar que se cumplen adecuadamente todos y cada uno de los requerimientos del problema a modelar. Este esquema representará el punto de partida para la siguiente fase, el diseño lógico de la base de datos.

El diseño lógico consistirá en la construcción de un esquema de la información relativa al problema, basado en un modelo de base de datos concreto. El esquema conceptual se transformará en un esquema lógico que utilizará los elementos y características del modelo de datos en el que esté basado el SGBD, para implementar nuestra base de datos. Como pudimos ver anteriormente, estos modelos podrán ser: el modelo en red, el modelo jerárquico y, sobre todo, el modelo relacional y el modelo orientado a objetos.

Para esta transformación será necesario realizar una serie de pasos preparatorios sobre el esquema conceptual obtenido en la fase de diseño conceptual. Nos centraremos en la simplificación y transformación del esquema para que el paso hacia el modelo de datos elegido (en este caso el modelo relacional) sea mucho más sencilla y efectiva.

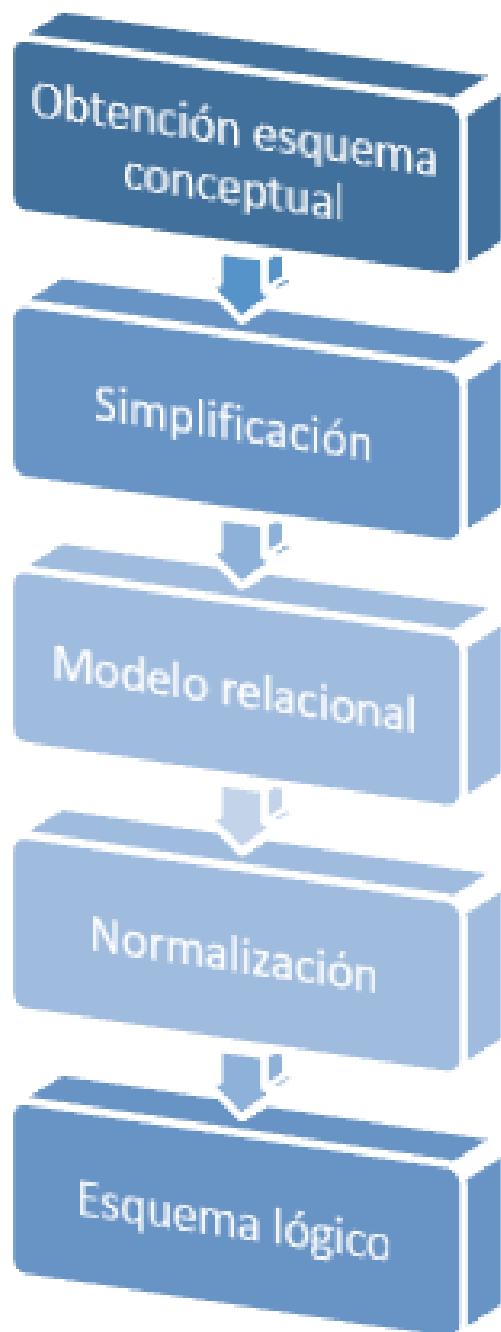
Seguidamente, tomando como referencia el esquema modificado/simplificado, se realizará el paso de éste al modelo de datos relacional. Esta transformación requerirá de la aplicación de determinadas reglas y condiciones que garanticen la equivalencia entre el esquema conceptual y el esquema lógico.

Como paso posterior, sobre la información del esquema lógico obtenido, será necesario llevar a cabo un proceso que permitirá diseñar de forma correcta la estructura lógica de los datos. Este proceso recibe el nombre de normalización, que se conforma como un conjunto de técnicas que permiten validar esquemas lógicos basados en el modelo relacional.

Entonces, ¿qué pasos son los siguientes a dar? Resumiendo un poco, simplificaremos nuestro diagrama E/R, lo transformaremos al modelo relacional, aplicaremos normalización y obtendremos lo que se conoce en el argot como el paso a tablas del

BD03. Interpretación de diagramas Entidad/Relación

esquema conceptual o, lo que es lo mismo, el esquema lógico. Desde ese momento, basándonos en este esquema, podremos llevarnos nuestra base de datos a cualquier SGBD basado en el modelo relacional e implementarla físicamente. Esta implementación física será totalmente dependiente de las características del SGBD elegido.



9.1. Simplificación previa de diagramas.

Hay un conjunto de procedimientos y normas que es necesario aplicar a nuestros diagramas E/R para que su transformación al modelo lógico basado en el modelo relacional, sea correcta y casi automática. Si aplicas correctamente estas pautas, conseguirás que el proceso de transformación sea fácil y fiable.

Las transformaciones de las que estamos hablando son las siguientes:

- Transformación de relaciones n-arias en binarias.
- Eliminación de relaciones cíclicas.
- Reducción a relaciones jerárquicas (uno a muchos).
- Conversión de entidades débiles en fuertes.

Transformación de atributos compuestos

Los atributos compuestos de una entidad han de ser descompuestos en los atributos simples por los que están formados.

El modelo relacional no admite atributos compuestos.

Transformación de atributos multivaluados

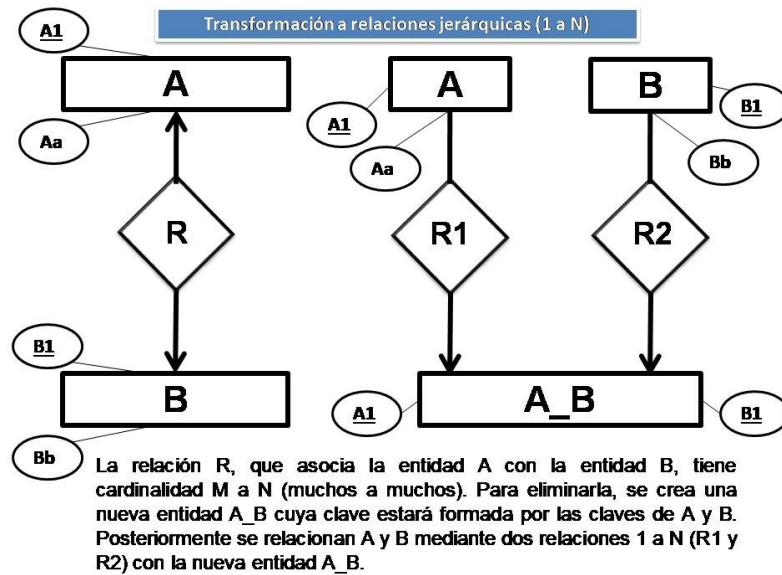
Si nuestro diagrama incluye la existencia de un atributo multivaluado, este se ha de convertir en una entidad relacionada con la entidad de la que procede.

El modelo relacional no admite atributos multivaluados. Para esta nueva entidad se elegirá un nombre adecuado y tendrá un único atributo (el correspondiente al antiguo atributo múltiple). Este atributo es posible que funcione correctamente como clave primaria de la entidad pero a veces es posible que no. En este caso, la entidad que hemos creado puede que sea débil. Deberemos ajustar en cualquier caso correctamente las claves primarias.

BD03. Interpretación de diagramas Entidad/Relación

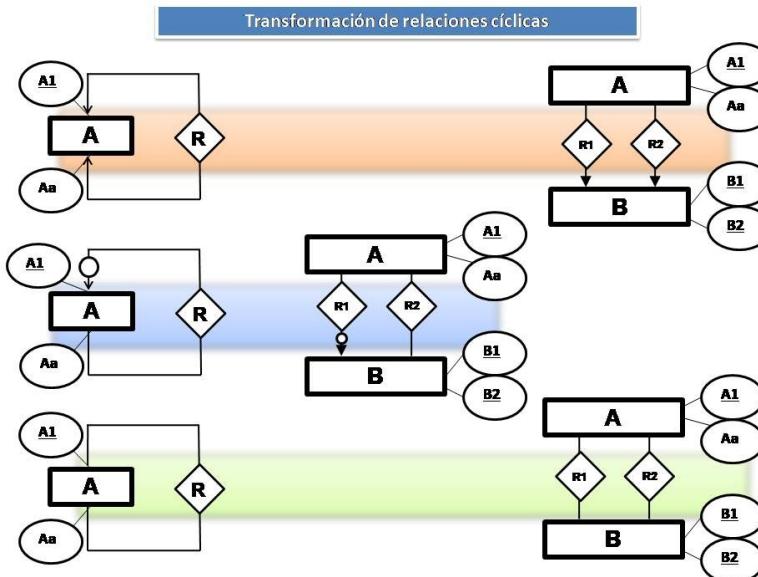
Transformación a relaciones jerárquicas

Se trata de transformar las relaciones con cardinalidad muchos a muchos (M a N) en relaciones con cardinalidad uno a muchos (1 a N). Observa la animación para comprender cómo se realiza la transformación. Si existiese algún atributo asociado a la relación n-aria, quedaría asociado a la nueva entidad que se crea.



Transformación de relaciones cíclicas

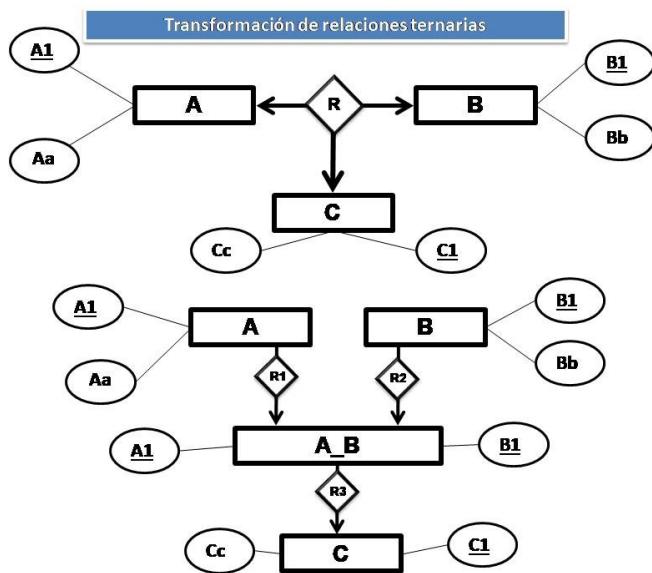
De forma general, si tenemos una entidad sobre la que existe una relación cíclica, para eliminar dicha relación, se crea una nueva entidad cuya clave estará formada por dos atributos, que contendrán las claves de las ocurrencias relacionadas. Entre ambas entidades se establecen dos relaciones, cuya cardinalidad dependerá de la cardinalidad que tuviera la relación cíclica en un principio.



BD03. Interpretación de diagramas Entidad/Relación

Transformación de relaciones ternarias

El tratamiento de las relaciones ternarias es similar al realizado para atributos asociados a relaciones, ya que una relación ternaria puede considerarse como una relación binaria a la que se le asocia una entidad. Por consiguiente, si en lugar de ser un conjunto de atributos los asociados a la relación es una entidad, se asociaría ésta mediante una nueva relación a la entidad resultante de eliminar la relación binaria.



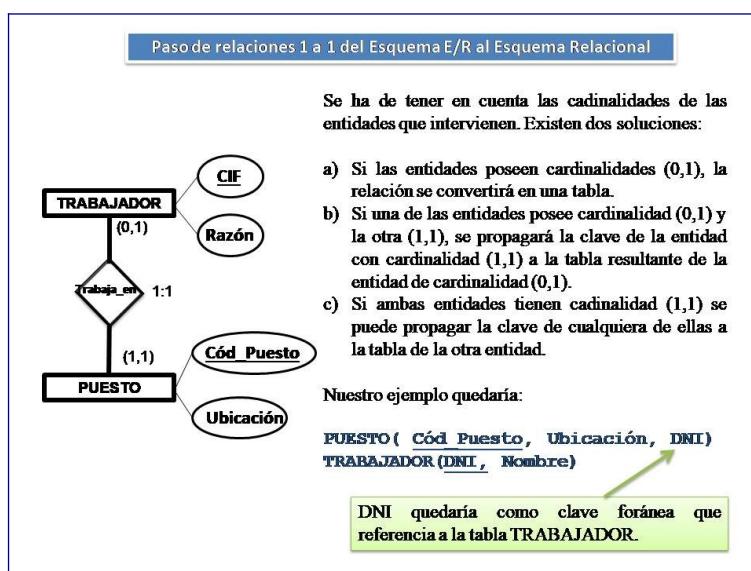
Transformación de entidades débiles en fuertes

Para esta transformación sólo es necesario añadir a la entidad débil los atributos clave de la entidad que hace posible la identificación de las ocurrencias. La clave de esta nueva entidad fuerte estará formada por los atributos clave de la que fuera entidad débil más los atributos adicionales.

10. Paso del diagrama E/R al Modelo Relacional.

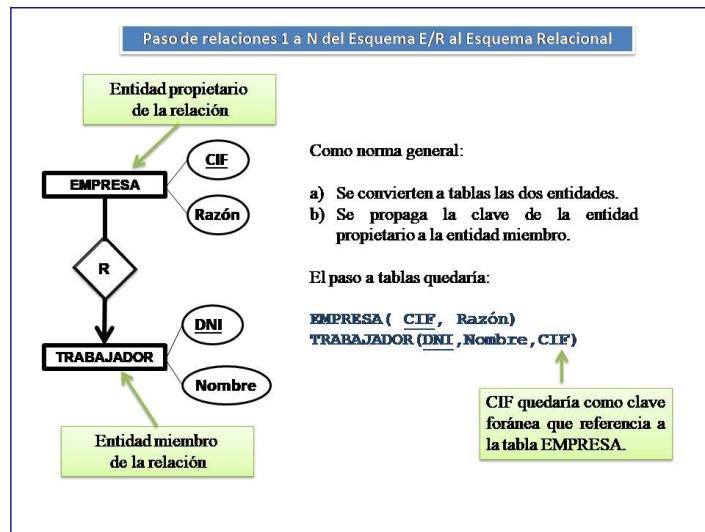
Si se ha llevado a cabo el proceso preparatorio de nuestro esquema conceptual o diagrama E/R, según se ha indicado en epígrafes anteriores, dispondremos de un Esquema Conceptual Modificado (ECM) en el que sólo existirán exclusivamente entidades fuertes con sus atributos y relaciones jerárquicas (1 a N). Pues bien, la aplicación del modelo de datos Relacional es automática, para ello se deben tener en cuenta las siguientes cuestiones:

- Toda **entidad** se transforma en una tabla.
- Todo **atributo** se transforma en columna dentro de una tabla.
- El **atributo clave** de la entidad se convierte en **clave primaria** de la tabla y se representará subrayado en la tabla.
- Cada **entidad débil** generará una tabla que incluirá todos sus atributos, añadiéndose **los atributos que son clave primaria de la entidad fuerte** con la que esté relacionada. Estos atributos añadidos se constituyen como clave foránea que referencia a la entidad fuerte. Seguidamente, se escogerá una clave primaria para la tabla creada.
- Las **relaciones Uno a Uno** podrán generar una nueva tabla o propagar la clave en función de la cardinalidad de las entidades.

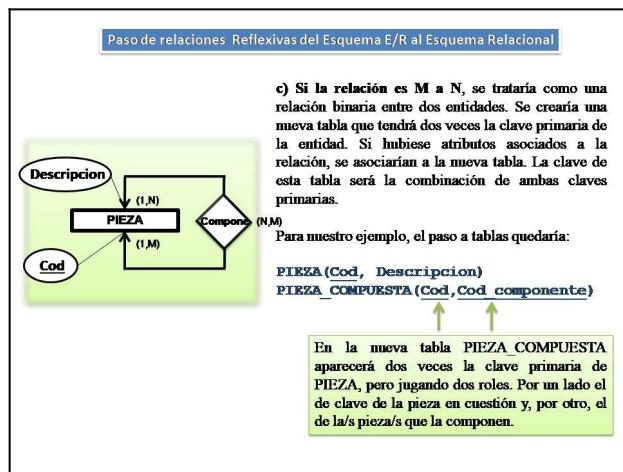
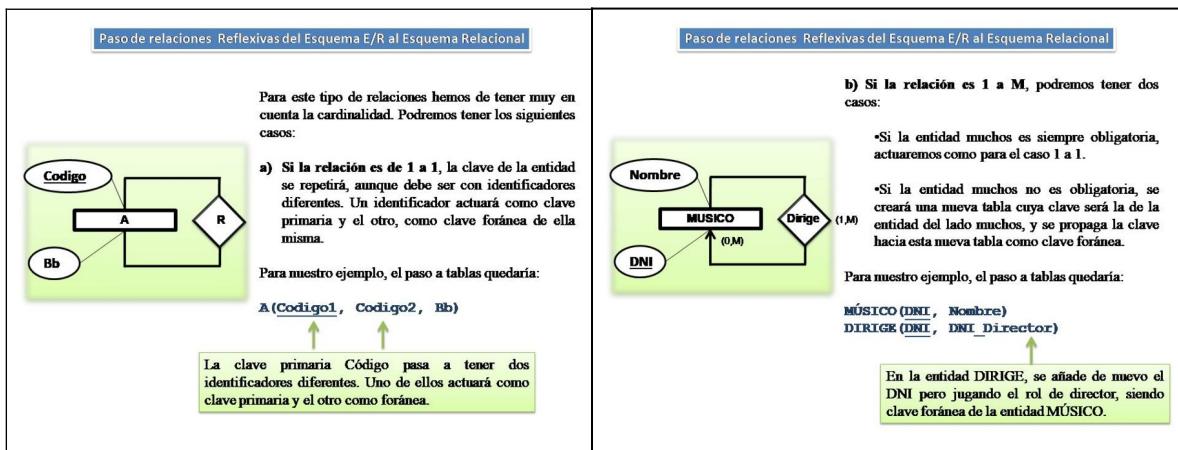


BD03. Interpretación de diagramas Entidad/Relación

- Las relaciones Uno a Muchos podrán generar una nueva tabla o propagar la clave.

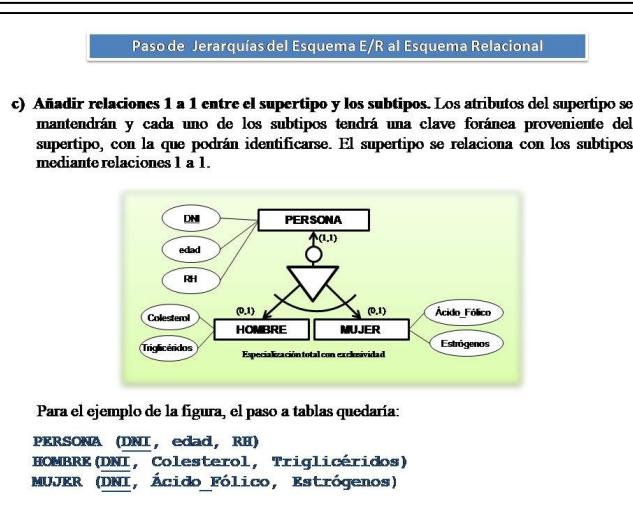
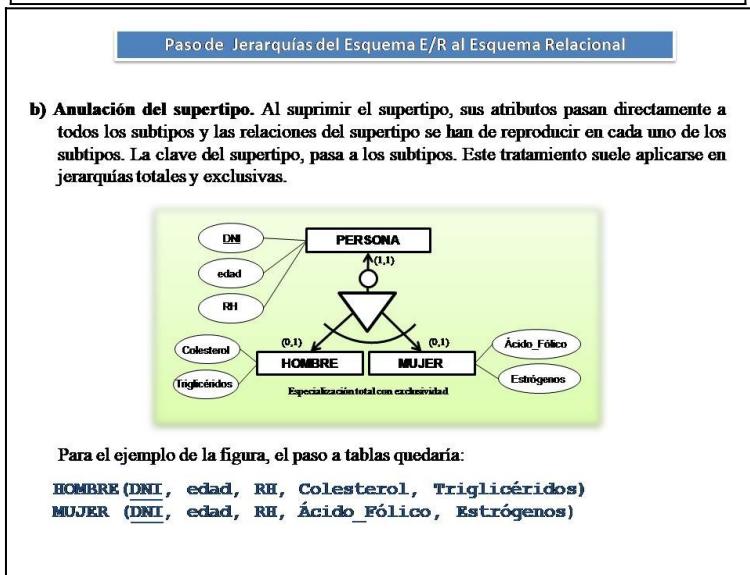
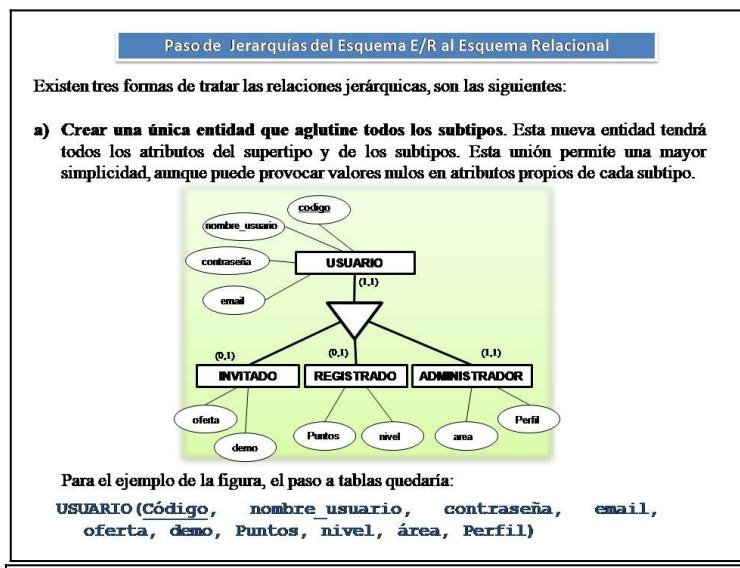


- Las relaciones reflexivas o cíclicas podrán generar una o varias tablas en función de la cardinalidad de la relación. Se muestran a continuación las tres formas posibles.



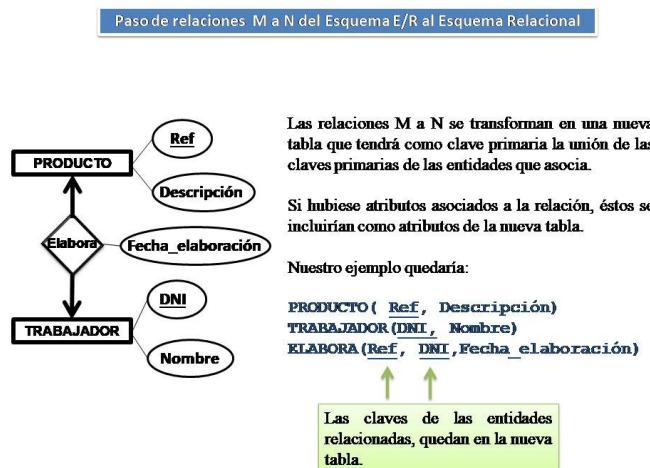
BD03. Interpretación de diagramas Entidad/Relación

- Las jerarquías generarán la reunión, eliminación o creación de relaciones 1 a 1. Se muestran a continuación las tres formas posibles.



BD03. Interpretación de diagramas Entidad/Relación

- Las **relaciones Muchos a Muchos** se transforman en una tabla que tendrá como clave primaria la unión de las claves primarias de las entidades que asocia.

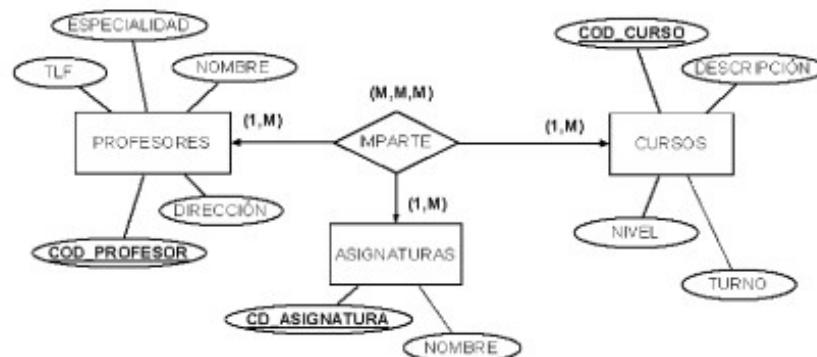


- Las **relaciones N-arias** que agrupan 3 o más entidades, cada entidad se convierte en tabla y también la relación que contendrá sus atributos propios más las claves de todas las entidades. La clave principal será la concatenación de las claves de las entidades.

Pueden darse dos casos dependiendo de las cardinalidades:

- Si la relación es N:M:N, es decir todas las entidades participan con cardinalidad máxima M, la clave de la tabla resultante es la unión de las claves de las entidades que relaciona.

Supongamos una relación ternaria entre las entidades PROFESORES-CURSOS-ASIGNATURAS, en la que un profesor imparte en varios cursos varias asignaturas y además las asignaturas son impartidas por varios profesores en varios cursos.



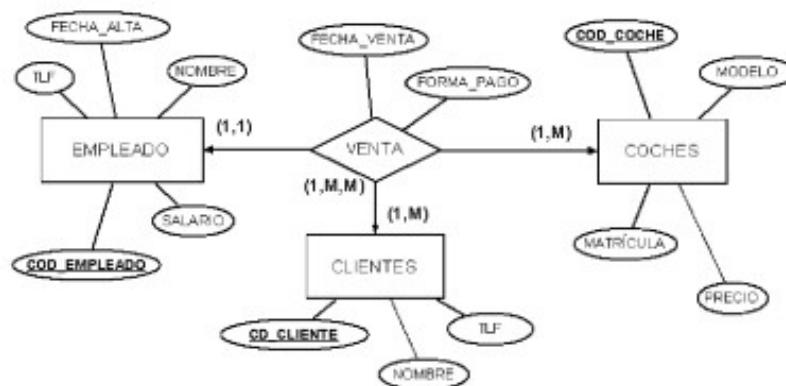
El resultado en el modelo relacional será:

PROFESORES (CodProfesor, Dirección, Nombre, Teléfono, Especialidad)
CURSOS (CodCurso, Descripción, Nivel, Turno)
ASIGNATURAS (CodAsignatura, Nombre)
IMPARTE (CodProfesor (FK), CodCurso (FK), CodAsignatura (FK))

BD03. Interpretación de diagramas Entidad/Relación

- Si la relación es 1:N:M, es decir una de las entidades participa con cardinalidad máxima 1, la clave de esta entidad no pasa a formar parte de la clave de la tabla resultante, pero forma parte de la relación como un atributo más.

Supongamos el caso de una tienda de venta de coches en la que un empleado vende muchos coches a muchos clientes y los coches son vendidos por un solo empleado. En la venta hay que tener en cuenta la forma de pago y la fecha de venta.



El resultado en el modelo relacional es:

CLIENTES (CodCliente, Nombre, Tfno)

EMPLEADO (CodEmpleado, Nombre, Tfno, Salario, FechaAlta)

COCHES (CodCoche, Matrícula, Modelo, Precio)

VENTA (CodCoche (FK), CodCliente (FK), CodEmpleado (FK), FormaPago, FechaVenta)

No obstante, si en el proceso de generación del diagrama E/R o esquema conceptual hemos aplicado correctamente las reglas de simplificación de diagramas, nuestro Esquema Conceptual Modificado nos permitirá el paso a tablas teniendo en cuenta sólo las transformaciones asociadas a entidades, relaciones 1 a N, 1 a 1 y Jerarquías.

11. Normalización de modelos relationales.

¿Crees que tu base de datos ya podría construirse directamente sobre el SGBD relacional que hayas elegido? La respuesta podría ser afirmativa, pero si queremos que nuestra base de datos funcione con plena fiabilidad, es necesario antes llevar a cabo un proceso de normalización de las tablas que la componen.

¿Y qué es eso de la normalización?

Normalización: Proceso que consiste en imponer a las tablas del modelo Relacional una serie de restricciones a través de un conjunto de transformaciones consecutivas. Este proceso garantizará que las tablas contienen los atributos necesarios y suficientes para describir la realidad de la entidad que representan, permitiendo separar aquellos atributos que por su contenido podrían generar la creación de otra tabla.

A principios de la década de los setenta, concretamente en 1972, Codd establece una técnica para llevar a cabo el diseño de la estructura lógica de los datos representados a través del modelo relacional, a la que denominó normalización. Pero esta técnica no ha de utilizarse para el diseño de la base de datos, sino como un proceso de refinamiento que debe aplicarse después de lo que conocemos como "paso a tablas", o lo que formalmente se denomina traducción del esquema conceptual al esquema lógico.

Objetivos

- Suprimir dependencias erróneas entre atributos.
- Optimizar los procesos de inserción, modificación y borrado en la base de datos.

Bases de la normalización

- El proceso de normalización se basa en el análisis de las dependencias entre atributos.
- Para ello tendrá en cuenta los conceptos de:
 - Dependencia funcional.
 - Dependencia funcional completa.
 - Dependencia transitiva.

BD03. Interpretación de diagramas Entidad/Relación

¿Y cómo se aplica la normalización?

Es un proceso que se realiza en varias etapas secuenciales. Cada etapa está asociada a una forma normal, que establece unos requisitos a cumplir por la tabla sobre la que se aplica.

Formas Normales

Hay varias formas normales:

- Primera.
- Segunda.
- Tercera.
- Boyce-Codd.
- Cuarta.
- Quinta.
- Dominio-Clave.

Como hemos indicado, el paso de una forma normal a otra es consecutivo, si no se satisface una determinada forma normal no puede pasarse al análisis de la siguiente. Según vamos avanzando en la normalización, los requisitos a cumplir serán cada vez más restrictivos, lo que hará que nuestro esquema relacional sea cada vez más robusto.

En este [enlace](#) (pdf - 336 KB) tienes un PDF con un ejemplo de tablas desnormalizadas (no normalizadas), los inconvenientes en el sistema de información y las ventajas de aplicar las formas normales al diseño.

Como norma general, para garantizar que no existan problemas en la actualización de datos, es recomendable aplicar el proceso de normalización hasta Tercera Forma Normal o incluso hasta Forma Normal de Boyce-Codd.

En los siguientes epígrafes se describen las características y requisitos de cada una de las formas normales.

11.1. Tipos de dependencias.

Para aplicar las formas normales y analizar las relaciones entre los atributos es necesario conocer el concepto de dependencia funcional y sus variantes.

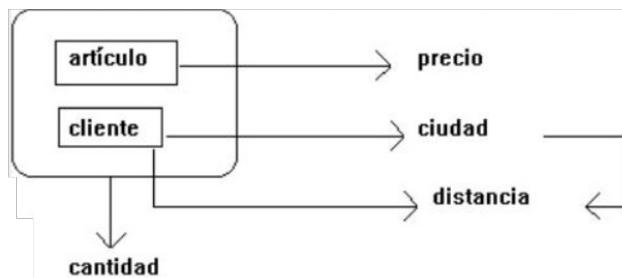
Vamos a desarrollar aquí esos conceptos:

- **Dependencia Funcional.**

Dados los atributos A y B, se dice que B depende funcionalmente de A, sí, y solo sí, para cada valor de A sólo puede existir un valor de B. La dependencia funcional siempre se establece entre atributos de una misma tabla.

El atributo A se denomina determinante, ya que A determina el valor de B. Para representar esta dependencia funcional utilizamos la siguiente notación: $A \rightarrow B$. Hay que indicar que A y B podrían ser un solo atributo o un conjunto de ellos.

Por ejemplo: entre los atributos artículo, cliente, precio, ciudad, distancia y cantidad hay varias dependencias funcionales:



Es preciso estudiar las DF (Dependencia Funcionales) para encontrar las claves candidatas, y a partir de ellas obtener el mínimo conjunto posible de atributos tales que una vez conocidos sus valores en las tuplas los demás queden definidos. Será la clave principal.

- **Dependencia Funcional Parcial o Incompleta.**

Dados los atributos A₁, A₂, ...A_k y B, se dice que B depende funcionalmente de forma completa de A₁, A₂, ...A_k, si y solo si B depende funcionalmente del conjunto de atributos A₁, A₂, ...A_k. Asociada a la 1FN.

$\text{artículo, cliente} \rightarrow \text{precio, ciudad, distancia, cantidad}$

- **Dependencia Funcional Completa**

Dados los atributos A₁, A₂, ...A_k y B, se dice que B depende funcionalmente de forma completa de A₁, A₂, ...A_k, si y solo si B depende funcionalmente del conjunto de atributos A₁, A₂, ...A_k, pero no de ninguno de sus posibles subconjuntos. Asociada a la 2FN.

Por ejemplo: precio depende únicamente de artículo, ciudad y distancia de cliente, y, cantidad de artículo y cliente.

$\text{Artículo} \rightarrow \text{precio}$
 $\text{Cliente} \rightarrow \text{ciudad, distancia}$
 $\text{Artículo, Cliente} \rightarrow \text{cantidad}$

BD03. Interpretación de diagramas Entidad/Relación

- **Dependencia Transitiva.**

Dados tres atributos A , B y C , se dice que existe una dependencia transitiva entre A y C , si B depende funcionalmente de A y C depende funcionalmente de B . A , B y C podrían ser un solo atributo o un conjunto de ellos.

Por ejemplo: distancia depende de ciudad y también de cliente, y ciudad depende de cliente. El atributo distancia es transitivamente dependiente del atributo Cliente.

Artículo → precio
Cliente → ciudad
Ciudad → distancia
Artículo, Cliente → cantidad

11.2. Formas normales.

Una vez conocidos los conceptos sobre los que se basa el proceso de normalización, se han de llevar a cabo una serie de procesos consecutivos en los que se aplicarán las propiedades de cada una de las formas normales definidas por Codd.

Requisitos a cumplir por las tablas de la base de datos según la forma normal aplicada:

1ª Forma Normal

Una tabla está en Primera Forma Normal (1FN o FN1) sí, y sólo sí, todos los atributos de la misma contienen valores atómicos, es decir, no hay grupos repetitivos. Dicho de otra forma, estará en 1FN si los atributos no clave, dependen funcionalmente de la clave. ¿Cómo se normaliza a Primera Forma Normal?

- Se crea, a partir de la tabla inicial, una nueva tabla cuyos atributos son los que presentan dependencia funcional de la clave primaria. La clave de esta tabla será la misma clave primaria de la tabla inicial. Esta tabla ya estará en 1FN.
- Con los atributos restantes se crea otra tabla y se elige entre ellos uno que será la clave primaria de dicha tabla. Comprobaremos si esta segunda tabla está en 1FN. Si es así, la tabla inicial ya estará normalizada a 1FN y el proceso termina. Si no está en 1FN, tomaremos la segunda tabla como tabla inicial y repetiremos el proceso.

2ª Forma Normal

Una tabla está en Segunda Forma Normal (2FN o FN2) sí, y sólo sí, está en 1FN y, además, todos los atributos que no pertenecen a la clave dependen funcionalmente de forma completa de ella. Es obvio que una tabla que esté en 1FN y cuya clave esté compuesta por un único atributo, estará en 2FN. ¿Cómo se normaliza a Segunda Forma Normal?

- Se crea, a partir de la tabla inicial, una nueva tabla con los atributos que dependen funcionalmente de forma completa de la clave. La clave de esta tabla será la misma clave primaria de la tabla inicial. Esta tabla ya estará en 2FN.
- Con los atributos restantes, se crea otra tabla que tendrá por clave el subconjunto de atributos de la clave inicial de los que dependen de forma completa. Se comprueba si esta tabla está en 2FN. Si es así, la tabla inicial ya está normalizada y el proceso termina. Si no está en 2FN, tomamos esta segunda tabla como tabla inicial y repetiremos el proceso.

3ª Forma Normal

Una tabla está en Tercera Forma Normal (3FN o FN3) sí, y sólo sí, está en 2FN y, además, cada atributo que no está en la clave primaria no depende transitivamente de la clave primaria. ¿Cómo se normaliza a Tercera Forma Normal?

- Se crea, a partir de la tabla inicial, una nueva tabla con los atributos que no poseen dependencias transitivas de la clave primaria. Esta tabla ya estará en 3FN.
- Con los atributos restantes, se crea otra tabla con los dos atributos no clave que intervienen en la dependencia transitiva, y se elige uno de ellos como clave primaria, si cumple los requisitos para ello. Se comprueba si esta tabla está en 3FN. Si es así, la tabla inicial ya está normalizada y el proceso termina. Si no está en 3FN, tomamos esta segunda tabla como tabla inicial y repetiremos el proceso.

BD03. Interpretación de diagramas Entidad/Relación

Forma Normal de Boyce Codd

Una tabla está en Forma Normal de Boyce-Codd (FNBC o BCFN) sí, y sólo sí, está en 3FN y todo determinante es una clave candidata.

Un **determinante** será todo atributo simple o compuesto del que depende funcionalmente de forma completa algún otro atributo de la tabla.

Aquellas tablas en la que todos sus atributos forman parte de la clave primaria, estarán en FNBC. Por tanto, si encontramos un determinante que no es clave candidata, la tabla no estará en FNBC. Esta redundancia suele ocurrir por una mala elección de la clave.

Para normalizar a FNBC tendremos que descomponer la tabla inicial en dos, siendo cuidadosos para evitar la pérdida de información en dicha descomposición.

Otras formas normales

Existen también la Cuarta Forma Normal (4FN o FN4), Quinta Forma Normal (5FN o FN5) y Forma Normal de Dominio-Clave (DKFN), aunque se ha recomendado normalizar hasta 3FN o FNBC. La 4FN se basa en el concepto de Dependencias Multivaluadas, la 5FN en las Dependencias de Join o de reunión y la DKFN en las restricciones impuestas sobre los dominios y las claves.

En este enlace [Despues de la 3FN](#) (pdf - 395 KB) tienes un PDF sobre la FNBC, 4FN y 5FN.

Aquí tienes un resumen de cada FN [Resumen FN](#) (pdf - 235 KB) y cómo se aplica. Puede ser útil que lo tengas a mano.