

Unidad 2: Interpretación de diagramas

Entidad/Relación.

Índice de contenidos

1	Análisis y diseño de BD.....	2
2	¿Qué es el modelo Entidad/Relación?	2
3	Entidades.....	3
3.1	Tipos: fuertes y débiles.	3
4	Atributos.....	4
4.1	Tipos de atributos.	5
4.2	Claves.	6
4.3	Atributos de una relación.....	7
5	Relaciones.....	7
5.1	Grado de una relación.....	8
5.2	Cardinalidad de relaciones.	8
5.3	Cardinalidad de entidades.	10
6	Simbología del modelo E/R.....	11
7	El modelo E/R Extendido.	11
7.1	Restricciones en las relaciones.....	12
7.2	Generalización y especialización.....	13
7.3	Agregación.	14
8	Elaboración de diagramas E/R.....	15
8.1	Identificación de entidades y relaciones.....	16
8.2	Identificación de atributos, claves y jerarquías.....	16
8.3	Metodologías.	17
8.4	Redundancia en diagramas E/R.....	18
8.5	Propiedades deseables de un diagrama E/R.....	19
9	Modelo Relacional.	22
9.1	Introducción.	22
9.2	El modelo relacional.....	22
9.3	Estructura y características.	24
9.4	Restricciones del modelo relacional.....	24
9.4.1	Restricciones y operaciones relacionales.	26
10	Paso del diagrama E/R al modelo relacional.....	27
10.1	Relaciones M:N.....	27
10.2	Relaciones 1:N.....	28
10.3	Relaciones 1:1.....	29
10.4	Relaciones Reflexivas.....	30
10.5	Otras.	31
10.6	Pérdida de la semántica en la transformación al modelo relacional.	33
10.7	Grafos relacionales.	33
11	Normalización de modelos relacionales.	34
11.1	Tipos de dependencias.	35
11.2	Formas Normales.	36
11.2.1	1 ^a Forma Normal.	36
11.2.2	2 ^a Forma Normal.	37
11.2.3	3 ^a Forma Normal.	38
11.2.4	Forma Normal de Boyce-Codd.....	38
11.2.5	Otras formas normales.	39
11.3	Desnormalización.	40

Como se ha descrito en la unidad anterior, 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 BD, su labor. El **objetivo fundamental** de esta arquitectura es separar los programas de aplicación de la BD física, proponiendo tres niveles de abstracción: nivel interno o físico, nivel lógico o conceptual y nivel externo o de visión del usuario.

1 Análisis y diseño de BD.

El **nivel lógico o conceptual** describe la estructura completa de la BD a través de lo que llamamos **Esquema Conceptual**, que se encarga de representar la información de una manera totalmente independiente del SGBD.

Cuando hemos de desarrollar una BD se distinguen claramente **dos fases** de trabajo: **análisis y diseño**. En la siguiente tabla se muestran los pasos que forman parte de cada fase:

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 entidades.	Normalización.
Obtención del esquema conceptual a través del modelo E/R.	Aplicación de retrodiseño, si fuera 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 del enfoque de datos relacional.	Diseño de sendas de acceso: se formalizan los métodos de acceso dentro de la estructura de datos.

Llevando a cabo una correcta fase de análisis estaremos dando un paso determinante en el desarrollo de nuestras BD. 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 Entidad/Relación?

Es una **herramienta de referencia para la representación conceptual de problemas del mundo real**. Su objetivo principal es facilitar el diseño de BD permitiendo la especificación de un esquema que representa la estructura lógica completa de una BD. 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 o 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 BD 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 desee 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.*

¿Ponemos otro ejemplo? Supongamos que tienes que desarrollar el esquema conceptual para una BD 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 BD. Pero no siempre una entidad puede ser concreta, como un camping o 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.

En el modelo Entidad/Relación, la representación gráfica de las entidades se realiza mediante el nombre de la entidad encerrado dentro de un rectángulo. A continuación, se muestra la representación de la entidad *CLIENTE*.



CLIENTE

3.1 Tipos: fuertes y débiles.

Las entidades pueden ser clasificadas en dos grupos:

a) Entidades fuertes o regulares.

Son aquellas que tienen existencia por sí mismas, es decir, su existencia no depende de la existencia de otras entidades. Por ejemplo, en una BD hospitalaria, la existencia de instancias concretas de la entidad *DOCTOR* no depende de la existencia de instancias u objetos de la entidad *PACIENTE*. En el modelo E/R las entidades fuertes se representan como hemos indicado anteriormente, con el nombre de la entidad encerrado dentro de un rectángulo.

b) Entidades débiles.

Son aquellas cuya existencia depende de la existencia de otras instancias de entidad. Por ejemplo,

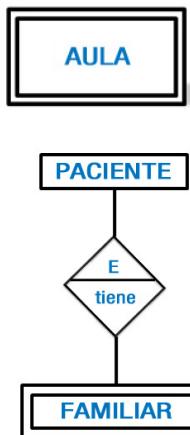
consideremos las entidades *EDIFICIO* y *AULA*. Supongamos que puede haber aulas identificadas con la misma numeración, pero en edificios diferentes. La numeración de cada aula no identificará completamente cada una de ellas. Para poder identificar completamente un aula es necesario saber también en qué edificio está localizada. Por tanto, **la existencia de una instancia de una entidad débil depende de la existencia de una instancia de la entidad fuerte con la que se relaciona.**

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.

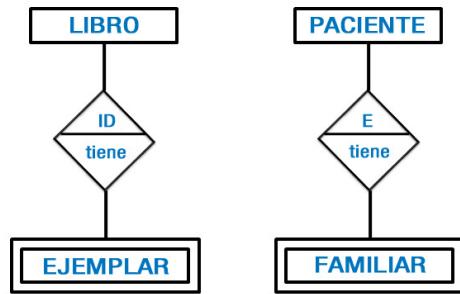
En el modelo E/R una entidad débil **se representa con el nombre de la entidad encerrado en un rectángulo doble**. En el gráfico se muestra la representación de la entidad *AULA*.

Las entidades débiles **presentan dos tipos de dependencia:**

- ✓ **Dependencia en existencia:** entre entidades, **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.**
- ✓ **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.**



Tanto las entidades fuertes como las débiles se nombran habitualmente con sustantivos en singular. Puede ser que haya algunos conceptos que aún no hemos desarrollado (relación, atributo y clave) y que se están utilizando para describir los tipos de dependencias, no te preocupes, en los siguientes puntos los describimos claramente.



Ejercicio resuelto

Identifica cuál de las siguientes entidades no podría ser considerada como entidad débil:

- ✓ **PROVEEDOR** (perteneciente a una BD de gestión de stocks).
- ✓ **PAGO** (perteneciente a una BD bancaria).
- ✓ **FAMILIAR** (perteneciente a una BD hospitalaria).

Solución: *PROVEEDOR* → Esta entidad puede existir por sí misma sin depender de otras ocurrencias de entidad. Además, posee propiedades o atributos propios que la identifican frente a otras ocurrencias de la misma entidad.

PAGO no podría ser ya que no podría ser sin la existencia de otra entidad como podría ser *PRÉSTAMO*. Si desaparece el *PRÉSTAMO* desaparecen también los pagos.

FAMILIAR no podría ser ya que en una BD hospitalaria, esta entidad podría depender de otra que sería *PACIENTE*. Si se estuviesen gestionando las visitas, cada familiar estaría asociado a un determinado paciente.

4 Atributos.

¿Cómo guardamos información de cada entidad? A través de sus **atributos**. Éstos **describen características o propiedades que posee cada miembro de un conjunto de entidades**. El mismo atributo establecido para un conjunto de entidades o, lo que es lo mismo, para un tipo de entidad, almacenará información parecida para cada ocurrencia de entidad. Pero, cada ocurrencia de entidad tendrá su propio valor para cada atributo.

Atributo: cada una de las propiedades o características que tiene un tipo de entidad o un tipo 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.



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 BD, es mejor establecer unos límites adecuados para que el SGBD lleve a cabo las verificaciones oportunas en los datos que se almacenen, garantizando así la integridad de éstos.

4.1 Tipos de atributos.

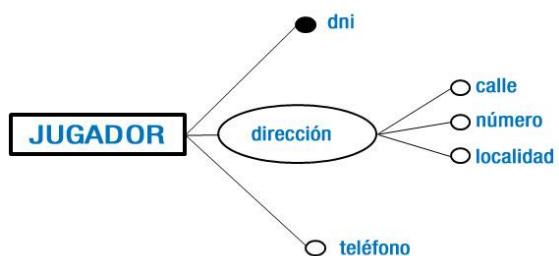
¿Todos los atributos son iguales? Claro que no. Existen varias características que hacen que los atributos asociados a una entidad o relación sean diferentes, los clasificaremos según varios criterios:

a) Atributos obligatorios yopcionales.

- **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.

b) Atributos 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.

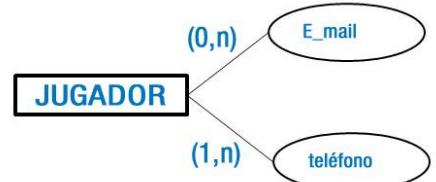


c) Atributos monovaluados o multivaluados.

- **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 el **número mínimo y el 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 o 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_email* 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).



- d) **Atributos derivados 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 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.

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.

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.

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 todos sus atributos**. Teniendo en cuenta esto, presta atención a los siguientes conceptos:

Identificador o superclave: cualquier conjunto de atributos que permite identificar de forma única a cada entidad.

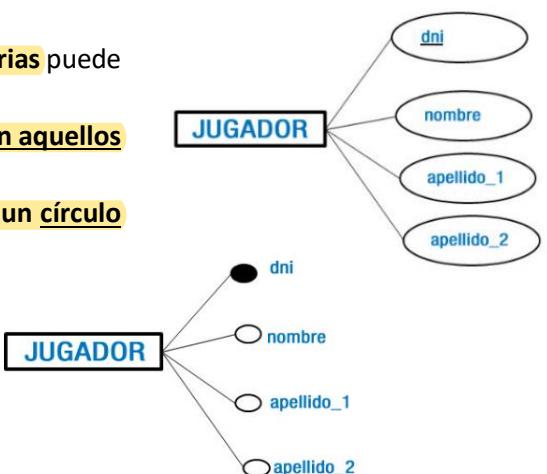
Clave candidata: cada una de las superclaves formadas por el mínimo número de atributos posibles.

Clave primaria o principal (Primary Key): es la clave candidata seleccionada por el diseñador de la BD.

Claves alternativas: son el resto de claves candidatas que no han sido escogidas como clave primaria.

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**.



Ejercicio resuelto

Sea la entidad **TRABAJADOR**, con los atributos **nombre**, **apellido_1**, **apellido_2**, **dni**, **numero_afiliacion_ss**, **fecha_nacimiento** y **código_empresa**. ¿Los atributos **nombre**, **apellido_1** y **apellido_2** podrían formar una clave candidata?

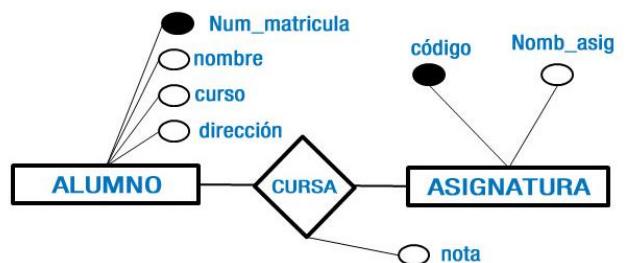
Solución: No, si tenemos en cuenta que puede haber varios trabajadores con el mismo nombre y apellidos. Los atributos **dni** y **numero_afiliacion_ss**, serían dos claves candidatas adecuadas. Si escogemos **dni** como clave primaria, **numero_afiliacion_ss** quedaría como clave alternativa.

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 alumn@ 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 la imagen anterior puedes ver esta segunda representación.

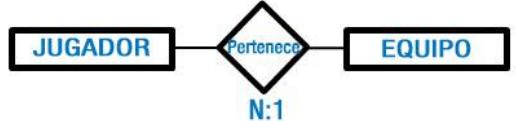


5 Relaciones.

¿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**. 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.



Cuando se deba **dar un nombre a una relación** hay que procurar que éste **haga referencia al objetivo o motivo de la asociación de entidades**. Se suelen utilizar verbos en singular. Algunos ejemplos podrían ser: *forman, poseen, atiende, contrata, hospeda, supervisa, imparte, etc.*

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. Normalmente es implícito y no se suele especificar (son útiles cuando el significado de una relación necesita ser calificado).

Normalmente las relaciones no tienen atributos, cuando surge una relación con atributos significa que debajo hay una entidad que aún no se ha definido. A esa entidad se le llama **entidad asociada** (esta entidad dará origen a una tabla que contendrá esos atributos).

Un **conjunto de relaciones** es un conjunto de relaciones del mismo tipo, por ejemplo, entre **JUGADOR** y **EQUIPO** todas las asociaciones existentes entre los jugadores y los equipos. **La mayoría de los conjuntos de relaciones en un sistema de BD son binarias** (dos entidades) **aunque puede haber conjunto de relaciones que impliquen más de dos conjuntos de entidades.**

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.

5.1 Grado de una relación.

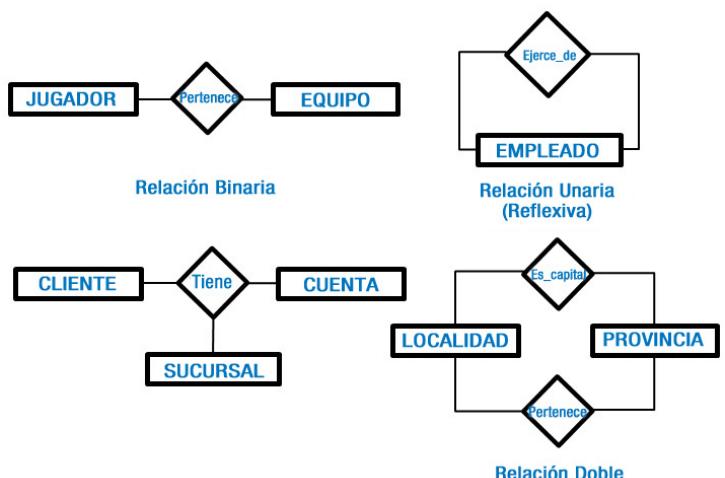
¿Pueden intervenir varias entidades en una misma relación? Claro que sí, en una relación puede intervenir una única entidad o varias.

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 BD 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 la imagen anterior se puede observar cada uno de los tipos de relaciones en función de su grado y su representación gráfica en el modelo Entidad/Relación.

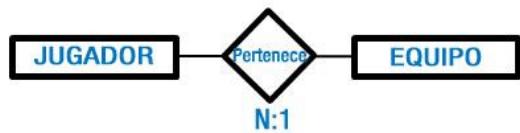


5.2 Cardinalidad de relaciones.

¿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.



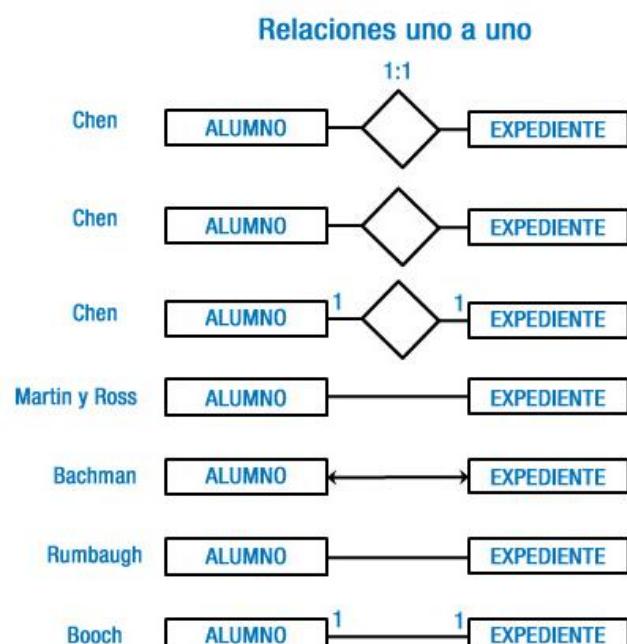
Una posible representación de la **cardinalidad de las relaciones** es la que hemos visto en el ejemplo anterior. Podríamos representar el resto de cardinalidades mediante las etiquetas **1:1, 1:N, N:1, M:N** que se leerían respectivamente: uno a uno, uno a muchos, muchos a uno y muchos a muchos.

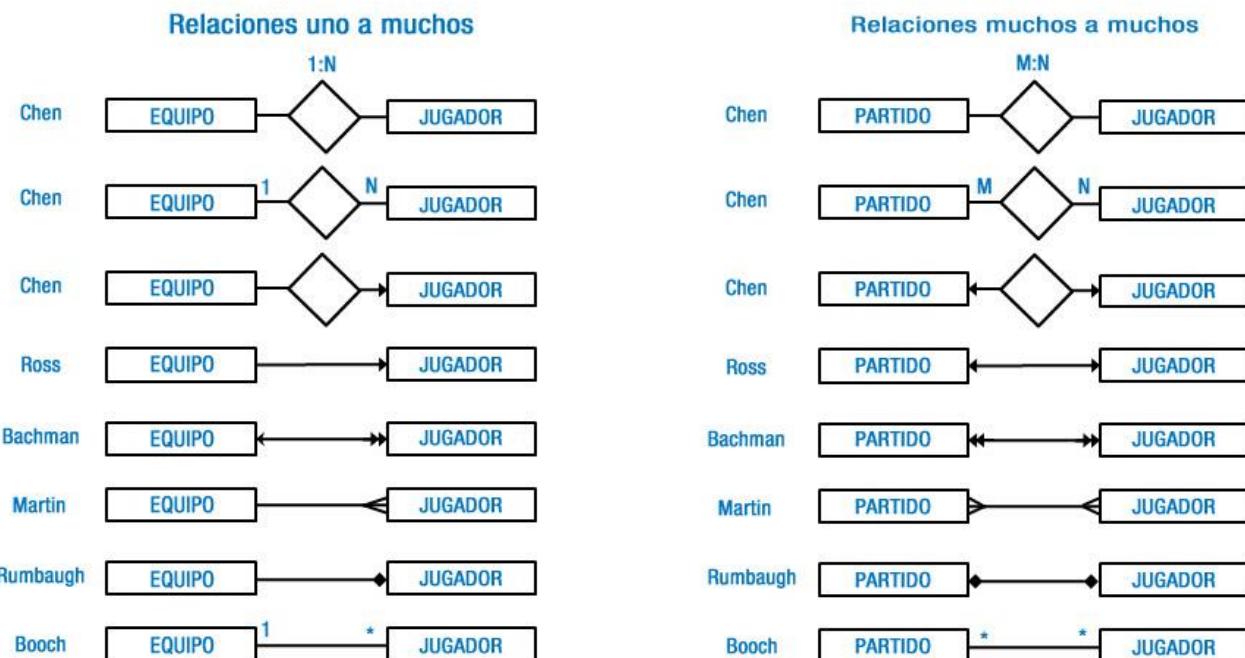
Veamos en detalle el significado de cada una de estas **cardinalidades**:

- ✓ **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.

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.**

Notaciones para representación de cardinalidad de relaciones:

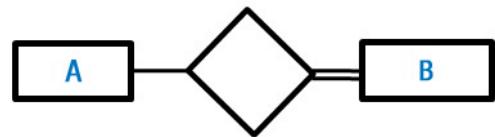




5.3 Cardinalidad de entidades.

Si existe cardinalidad en las relaciones, supondrás que también existe para las entidades. 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.

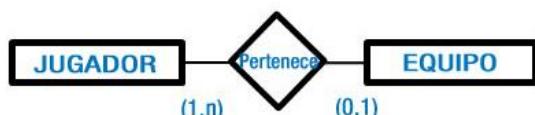
Sean las entidades A y B, la participación de la entidad A en una relación es **obligatoria (total)** si la existencia de cada una de sus ocurrencias necesita como mínimo de una ocurrencia de la entidad B (ver imagen). En caso contrario, la participación es **opcional (parcial)**.



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).

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:



Notación alternativa para representar cardinalidad de entidades

	Muchos
	Uno
	De cero a muchos
	De uno a muchos
	De cero a uno

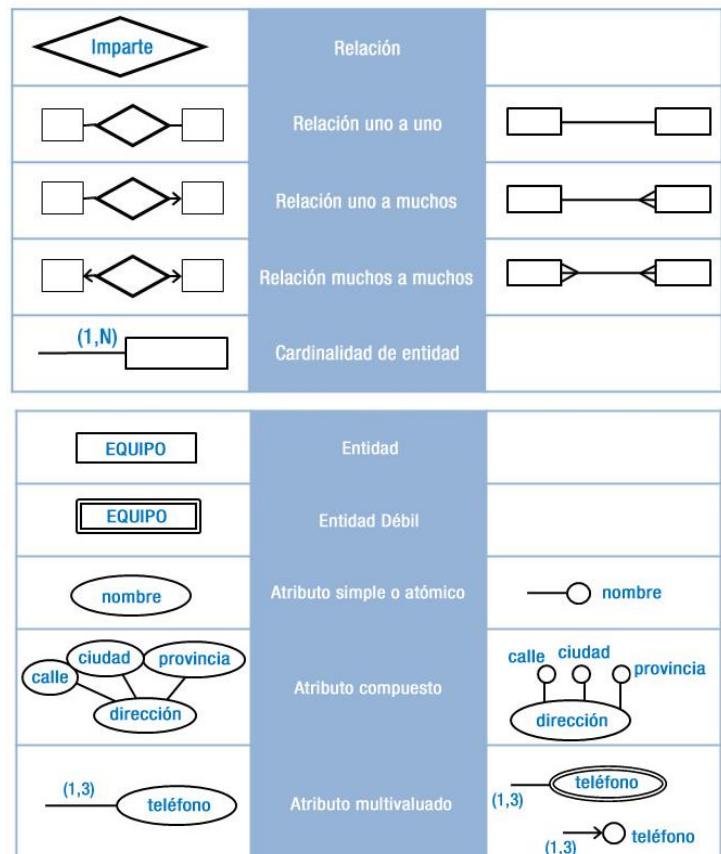
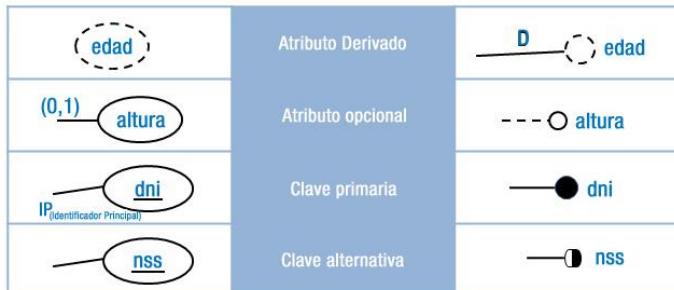
Ejercicio resuelto

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

Solución: (1,1) y (0,n) → Con estas cardinalidades estamos indicando que un usuario puede crear varias partidas, o ninguna. Por otra parte, una partida deberá estar creada exclusivamente por un único usuario.

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.



7 El modelo E/R Extendido.

Hemos visto que a través del modelo Entidad/Relación se pueden modelar la gran mayoría de los requisitos que una BD 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 y agregación.**

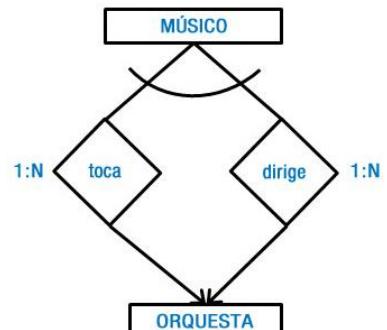
7.1 Restricciones en las relaciones.

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, ni a la vez ni por separado.

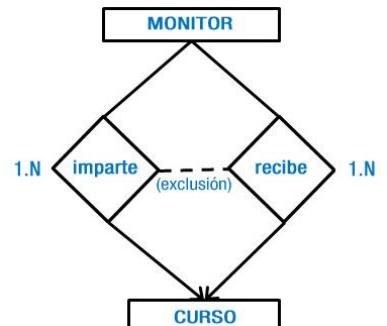
La representación gráfica en el modelo Entidad/Relación Extendido de una restricción de exclusividad se realiza mediante un arco que engloba a todas aquellas relaciones que son exclusivas.



b) Restricción de exclusión.

Este tipo de restricción se produce cuando las ocurrencias de las entidades sólo pueden asociarse utilizando una única relación. O se produce una relación o se produce otra, pero nunca ambas de forma simultánea. La entidad si podrá actuar en todas las relaciones, pero siempre por separado.

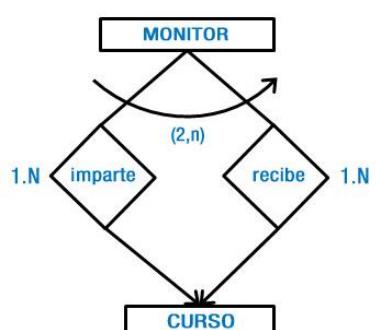
Pongamos un ejemplo, supongamos que un monitor puede impartir diferentes cursos de perfeccionamiento para monitores, y que éste puede a su vez recibirlas. Pero si un monitor imparte un determinado curso, no podrá estar recibiendo simultáneamente y viceversa. Se establecerá, por tanto, una restricción de exclusión que se representa mediante una línea discontinua entre las dos relaciones, tal y como se muestra en el ejemplo.



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).

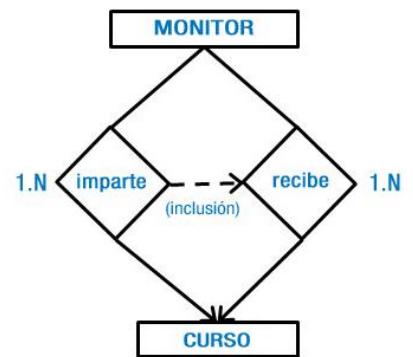


Se representará mediante un arco acabado en flecha, que partirá desde la relación que ha de cumplirse la última hacia la otra relación. Se indicará junto al arco la cardinalidad mínima y máxima de dicha restricción de inclusividad. En el ejemplo, (2,n) indica que un monitor ha de recibir 2 cursos antes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



Ejercicio resuelto

Supongamos que hemos de modelar mediante el modelo Entidad/Relación Extendido el siguiente requerimiento de una BD: Para que un hombre se divorcie de una mujer, primero ha de haber estado casado con ella.

Las entidades participantes son *MUJER* y *HOMBRE*, que estarán asociadas a través de dos relaciones: se casa, se divorcia. No tendremos en cuenta la cardinalidad de ambas relaciones.

¿Qué tipo de restricción sobre las relaciones hemos de establecer en nuestro esquema para representar correctamente este requisito?

Solución: Restricción de inclusión → Este tipo de restricción establece la obligatoriedad de haber un casamiento para que pueda haber un divorcio y, además, las entidades que se relacionan a través de la relación se casa, deben ser las mismas que las participantes en se divorcia.

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**.



Cuando estamos diseñando una BD **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 encontremos 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.

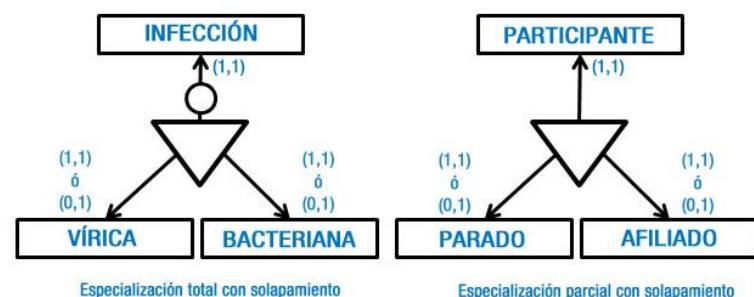
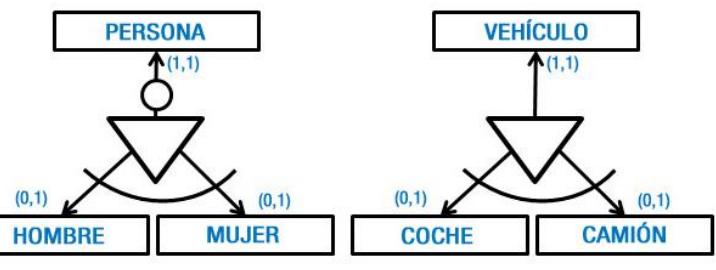
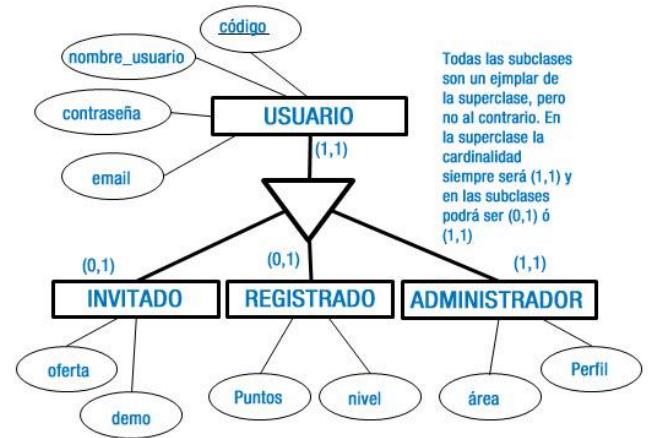
¿Cómo se representa una generalización o especialización? Existen varias notaciones, pero hemos de convenir que la relación que se establece entre una superclase de entidad y todos sus subtipos se expresa a través de las palabras ES UN, o en notación inglesa IS A, que correspondería con ES UN TIPO DE. 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.**

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.

Una generalización/especialización podrá tener las siguientes restricciones semánticas:

- ✓ **Totalidad:** si todo ejemplar de la superclase pertenece a alguna de las subclases.
- ✓ **Parcialidad:** si no todos los ejemplares de la superclase pertenecen a alguna de las subclases.
- ✓ **Solapamiento:** si un mismo ejemplar de la superclase puede pertenecer a más de una subclase.
- ✓ **Exclusividad:** si un mismo ejemplar de la superclase pertenece sólo a una subclase.

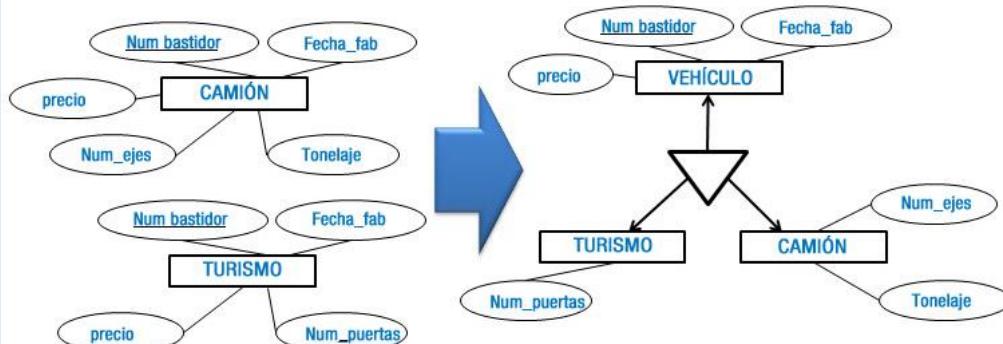
Las diferentes restricciones semánticas descritas tienen su representación gráfica, a través del gráfico que a continuación se muestra podrás entender mejor su funcionamiento.



Ejercicio resuelto

Supongamos la existencia de dos entidades TURISMO y CAMIÓN. Los atributos de la entidad TURISMO son: Num_bastidor, Fecha_fab, precio y Num_puertas. Los atributos de la entidad CAMIÓN son: Num_bastidor, Fecha_fab, precio, Num_ejes y Tonelaje.

Si analizamos ambas entidades existen algunos atributos comunes y otros que no. Por tanto, podremos establecer una jerarquía. Para ello, reuniremos los atributos comunes y los asociaremos a una nueva entidad superclase denominada VEHÍCULO. Las subclases TURISMO y CAMIÓN, con sus atributos específicos, quedarán asociadas a la superclase VEHÍCULO mediante una jerarquía parcial con solapamiento. En el siguiente gráfico puedes apreciar la transformación.



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.

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.

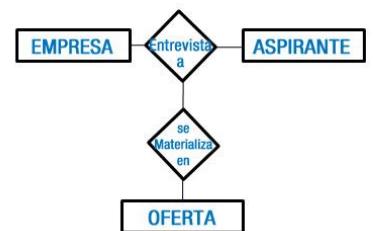
Como has podido observar, la representación gráfica de una agregación se caracteriza por englobar con un rectángulo las entidades y relación a abstraer. De este modo, se crea una nueva entidad agregada que puede participar en otras relaciones con otras entidades. En este tipo de relación especial de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

Existen dos clases de agregaciones:

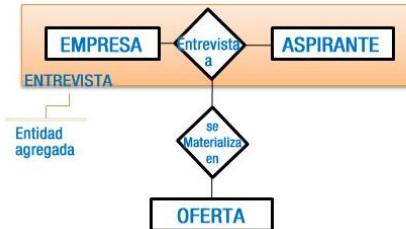
- ✓ **Compuesto/componente:** un todo se obtiene por la unión de diversas partes, que pueden ser objetos distintos y que desempeñan papeles distintos en la agregación. Teniendo esto en cuenta, esta abstracción permite representar que un todo o agregado se obtiene por la unión de diversas partes o componentes 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.



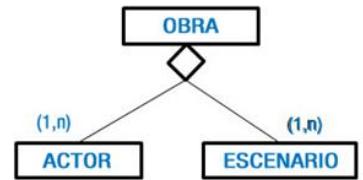
Solución 1: Errónea, ya que estaríamos representando que, por cada entrevista realizada por una empresa a un aspirante, se genera una oferta de empleo



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"



8 Elaboración de diagramas E/R.

Llegados a este punto, 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 ver una serie de orientaciones para poder aplicar todos los conceptos aprendidos hasta ahora en la elaboración de diagramas Entidad/Relación.

Sabemos que, en la fase de diseño conceptual de la BD, 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 SGBD. 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 BD, 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, se obtendrá la destreza necesaria para generar esquemas que garanticen una posterior y correcta conversión del modelo Entidad/Relación al modelo Relacional.

8.1 Identificación de entidades y relaciones.

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 BD ha de cumplir**. Es lo que se denomina el **documento de especificación de requerimientos**. En otras palabras, el enunciado del problema a modelar. Cuanto más completa y detallada sea la información de la que dispongamos, mucho mejor.

Suponiendo que conocemos la simbología del modelo Entidad/Relación y que entendemos su significado ¿Cómo empezamos? Las **etapas para la creación del diagrama E/R** se detallan a continuación:

- 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. 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**. 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, representativos** de su significado o función. De esta manera el diagrama será cada vez más legible.

- 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**. 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 en minúsculas, representativo** de su significado o acción.

En ocasiones, **el identificador de una relación está compuesto por varias palabras**, como, por ejemplo: *es supervisado*, *trabaja para*, etc. Es recomendable utilizar guiones bajos para unir las palabras que forman el identificador.

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). Si hemos encontrado alguna **relación recursiva, reflexiva o unaria**, **hemos de representar en nuestro esquema los roles desempeñados** por la entidad en dicha relación.

8.2 Identificación de atributos, claves y jerarquías.

Sólo con la localización de entidades y relaciones no está todo hecho. Hemos de completar el proceso realizando

las siguientes tareas:

- 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 el 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 disponibles para la elaboración del esquema conceptual** son las siguientes:

- ✓ **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 se decía hace algunos puntos, la práctica es fundamental. Realizando gran cantidad de esquemas, analizándolos y llevando a cabo modificaciones en ellos es como se irá refinando la técnica de elaboración de diagramas E/R. Llegará un momento en que sólo con leer el documento de especificación de requerimientos se sea capaz de ir construyendo en la mente cómo será su representación sobre el papel, pero paciencia y vamos paso a paso.

8.4 Redundancia en diagramas E/R.

Una de las principales razones por las que las BD aparecieron fue la eliminación de la redundancia en los datos ¿Y qué es la redundancia?

Redundancia: en BD hace referencia al almacenamiento de los mismos datos varias veces en diferentes lugares.

La **redundancia de datos** puede provocar problemas como:

- ✓ **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 este problema.
- ✓ **Inconsistencia:** 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.

Para que una BD 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? Existen lugares y elementos que podrían presentar redundancia, por ejemplo:

- ✓ **Atributos redundantes cuyo contenido se calcula en función de otros.** Un atributo derivado puede ser origen de redundancia.
- ✓ **Varias entidades unidas circularmente 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 existe un ciclo estaremos ante una redundancia**. Es necesario analizar detenidamente dicho ciclo para determinar si realmente existe o no redundancia.

Para finalizar, una apreciación. **No toda redundancia es perjudicial.** Existen ciertas circunstancias y condiciones en las que es conveniente (sobre todo a efectos de rendimiento) introducir cierta **redundancia controlada** en una BD. 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 BD, quizá 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:** 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:** si emplea de manera adecuada todos los elementos del modelo Entidad/Relación. La corrección de un diagrama puede analizarse desde dos vertientes:
 - **Corrección sintáctica:** cuando no se produzcan representaciones erróneas en el diagrama.
 - **Corrección semántica:** 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:** 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:** si representa los requerimientos de manera fácil de comprender, sin artificios complejos.
- ✓ **Legibilidad:** 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:** si es capaz de incorporar posibles cambios derivados de nuevos requerimientos.

Ejercicio resuelto

Si en un diagrama E/R asociamos un atributo a una entidad, pero este atributo debe asociarse realmente a una relación en la que interviene dicha entidad, estaríamos incumpliendo la propiedad de:

Solución: Corrección semántica → La Corrección semántica se centra en analizar si cada representación del diagrama significa exactamente lo mismo que lo estipulado por el documento de especificación de requerimientos.

Ejercicio resuelto

Vamos a realizar el diagrama de estructura de datos en el modelo E/R. Supongamos que en un centro educativo se imparten muchos cursos. Cada curso está formado por un grupo de alumnos, de los cuales uno de ellos es el delegado del grupo. Los alumnos cursan asignaturas, y una asignatura puede o no ser cursada por los alumnos.

Lo primero que hay que hacer es identificar las entidades, luego las relaciones y las cardinalidades y, por último, los atributos de las entidades y de las relaciones (si los hubiera).

1. Identificación de entidades.

Una entidad es un objeto del mundo real, algo que tiene interés para la empresa. Se hace un análisis del enunciado, de donde sacamos los candidatos a entidades: *CENTRO, CURSO, ALUMNO, ASIGNATURA* y *DELEGADO*. Si analizamos esta última vemos que los delegados son alumnos, por lo tanto, los tenemos recogidos en *ALUMNO*. Esta posible entidad la eliminaremos. También eliminaremos la posible entidad *CENTRO* pues se trata de un único centro, si se tratara de una gestión de centros tendría sentido incluirla.

2. Identificar las relaciones.

Construimos una matriz de entidades en la que las filas y las columnas son los nombres de entidades y cada celda

puede contener o no la relación, las relaciones aparecen explícitamente en el enunciado. En este ejemplo, las relaciones no tienen atributos. Del enunciado sacamos lo siguiente:

- ✓ Un curso está formado por muchos alumnos. La relación entre estas dos entidades la llamaremos *PERTENECE*, pues a un curso pertenecen muchos alumnos, relación 1:N. Consideraremos que es obligatorio que existan alumnos en un curso. Para calcular los máximos y mínimos hacemos la pregunta: a un CURSO, ¿cuántos ALUMNOS pertenecen, como mínimo y como máximo? Y se ponen los valores en la entidad ALUMNOS, en este caso (1,N). Para el sentido contrario, hacemos lo mismo: un ALUMNO, ¿a cuántos CURSOS va a pertenecer? Como mínimo a 1, y como máximo a 1, en este caso pondremos (1,1) en la entidad CURSOS.
- ✓ De los alumnos que pertenecen a un grupo, uno de ellos es *DELEGADO*. Hay una relación de grado 1 entre la entidad ALUMNO que la podemos llamar *ES DELEGADO*. La relación es 1:N, un alumno es delegado de muchos alumnos. Para calcular los valores máximos y mínimos preguntamos: ¿un ALUMNO de cuántos alumnos ES DELEGADO? Como mínimo es 0, pues puede que no sea delegado, y como máximo es N, pues si es delegado lo será de muchos; pondremos en el extremo (0,N). Y en el otro extremo pondremos (1,1), pues obligatoriamente el delegado es un alumno.
- ✓ Entre ALUMNO y ASIGNATURA surge una relación N:M, pues un alumno cursa muchas asignaturas y una asignatura es cursada por muchos alumnos. La relación se llamará *CURSA*. Consideramos que puede haber asignaturas sin alumnos. Las cardinalidades serán (1,N) entre ALUMNO-ASIGNATURA, pues un alumno, como mínimo, cursa una asignatura, y, como máximo muchas. La cardinalidad entre ASIGNATURA-ALUMNO será (0,N), pues una asignatura puede ser cursada por 0 alumnos o por muchos.

En la tabla siguiente se muestra la matriz de entidades y relaciones entre ellas:

	CURSO	ALUMNO	ASIGNATURA
CURSO	-----	PERTENECE (1:N)	-----
ALUMNO	X	ES DELEGADO (1:N)	CURSA (N:M)
ASIGNATURA	-----	X	-----

Las celdas que aparecen con una x indican que las relaciones están ya identificadas. Las que aparecen con guiones indican que no existe relación.

3. Identificar los atributos.

Como en el enunciado no explicita ningún tipo de característica de las entidades nos imaginamos los atributos, que pueden ser los siguientes:

- ✓ CURSO: idCurso (clave primaria), descripción, nivel, turno y etapa.
- ✓ ALUMNO: idMatrícula (clave primaria), nombre, dirección, población, teléfono y numHermanos.
- ✓ ASIGNATURA: idAsignatura (clave primaria), denominación y tipo.

Diagrama de estructura de datos resultante en el modelo E/R (notación de Chen), creado con DIA:

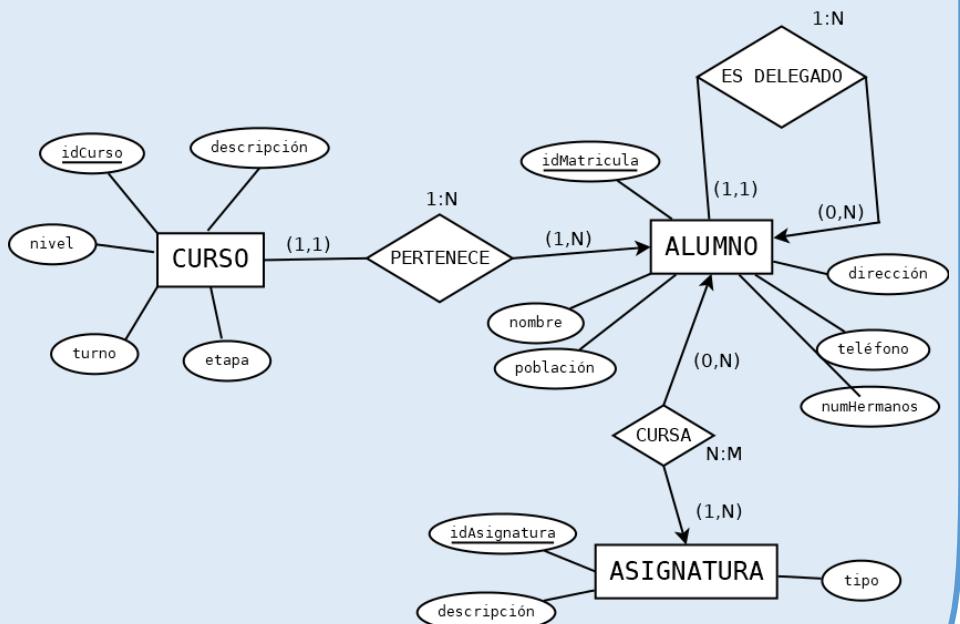
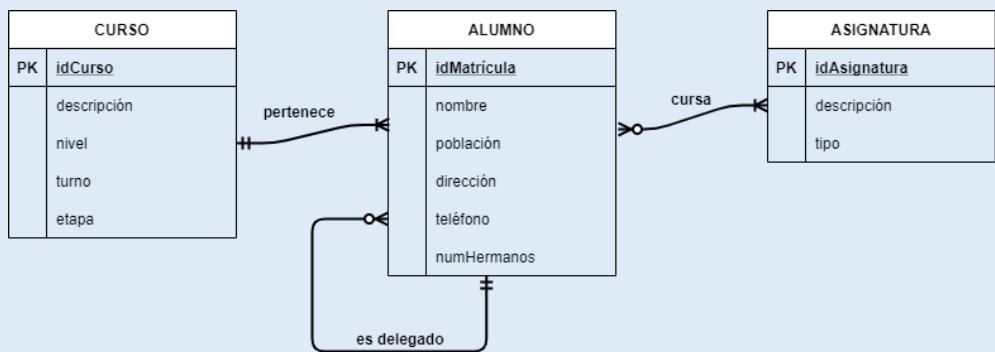


Diagrama de estructura de datos resultante en el modelo E/R (notación de Martin), creado con el editor online <http://draw.io> :



TAREA - MODELO ENTIDAD/RELACIÓN

Se desea realizar el diagrama de estructuras de datos en el modelo E/R, correspondiente de los siguientes enunciados:

- Supongamos el bibliobús que proporciona un servicio de préstamo de libros a los socios de un pueblo. Los libros están clasificados por temas. Un tema puede contener varios libros. Un libro es prestado a muchos socios, y un socio puede coger varios libros. En el préstamo de libros es importante saber la fecha de préstamo y la fecha de devolución. De los libros nos interesa saber el título, el autor y el número de ejemplares.

- En una estación de autobuses contamos con unos autobuses que recorren una serie de lugares y que son conducidos por varios conductores. Se quiere representar los lugares que son recorridos por cada autobús, conducidos por cada conductor y la fecha en la que se visita el lugar.

Define las entidades, los posibles atributos, identifica los atributos clave y los datos importantes para la relación entre las entidades. Indica también la cardinalidad.

- Una compañía de distribución de productos para el hogar dispone de proveedores que le suministran artículos. Un artículo sólo puede proveerlo un proveedor.

La empresa tiene tres tipos de empleados: oficinistas, transportistas y vendedores. Estos últimos venden los artículos. Un artículo es vendido por varios vendedores, y un vendedor puede vender varios artículos en distintas zonas de venta. De las ventas nos interesa saber la fecha de venta y las unidades vendidas.

- La Consejería de Educación gestiona varios tipos de centros: públicos, privados y concertados. Los privados tienen un atributo específico que es la cuota y los concertados la agrupación y la comisión. También asigna plazas a los profesores de la comunidad para impartir clase en esos centros. Un profesor puede impartir clase en varios centros.

Define las entidades, los posibles atributos, identifica los atributos clave y los datos importantes para la relación entre las entidades. Indica también la cardinalidad.

- A un taller de automóviles llegan clientes a comprar coches. De los coches nos interesa saber la marca, el modelo, el color y el número de bastidor.

Los coches pueden ser nuevos y de segunda mano. De los nuevos nos interesa saber las unidades que hay en el taller. De los viejos el año de fabricación, el número de averías y la matrícula.

Los mecánicos se encargan de poner a punto los coches usados del taller. Un mecánico pone a punto a varios coches usados.

Un cliente puede comprar varios coches; un coche puede ser comprado por varios clientes. De la compra nos interesa la fecha y el precio.

Define las entidades, los atributos, las relaciones, sus atributos si los hubiera y las cardinalidades.

- Una agencia de viajes está formada por varias oficinas que se ocupan de atender a los posibles viajeros. Cada oficina oferta un gran número de viajes. Los viajes trabajan con una serie de destinos y una serie de procedencias. Cada viaje tiene un único destino y una única procedencia. Sin embargo, un destino puede ser objetivo de varios viajes y una procedencia ser punto de partida de varios viajes. Cada viaje tiene muchos viajeros.

- Una entidad bancaria está formada por varias sucursales y cada sucursal tiene un gran número de cuentas que son propiedad de los clientes. Los datos saldo, debe y haber deben aparecer en cada una de las cuentas. Las

cuentas son de dos tipos: cuenta de ahorro con el atributo específico tipo de interés y cuenta corriente, con el atributo específico cantidad de descubierto. Las cuentas o son corrientes o son de ahorro.

Un cliente puede tener varias cuentas. Una cuenta es sólo propiedad de un cliente.

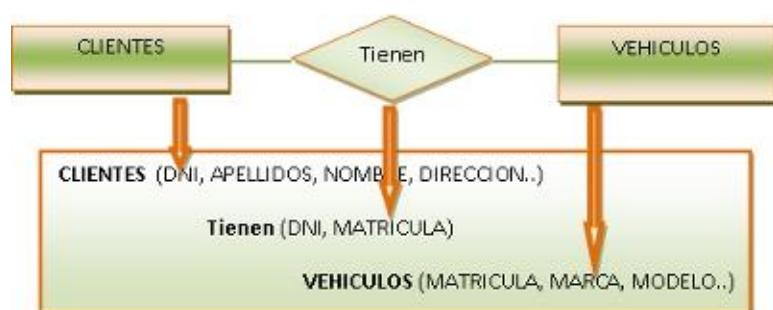
Las cuentas realizan una serie de movimientos, en los que además de otros datos deben aparecer la cantidad implicada y la fecha. Existe una serie de tipos de movimientos reconocidos por el banco. Un movimiento pertenece a un tipo. Sin embargo, de un tipo de movimiento puede haber varios movimientos.

9 Modelo Relacional.

9.1 Introducción.

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 BD. 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 BD**.



El diseño lógico **consistirá en la construcción de un esquema de la información relativa al problema, basado en un modelo de BD 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 BD. 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.

En este caso, **tomando como referencia el esquema E/R, 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, 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 esquema conceptual o, lo que es lo mismo, el esquema lógico. Desde ese momento, basándonos en este esquema, podremos llevarnos nuestra BD a cualquier SGBD basado en el modelo relacional e implementarla físicamente. Esta implementación física será totalmente dependiente de las características del SGBD elegido.

9.2 El modelo relacional.

Edgar Frank Codd es el padre del modelo relacional y lo definió en un artículo publicado en 1970. Este **modelo está basado en dos teorías matemáticas: la teoría de conjuntos y la lógica de predicados de primer orden**. Pero no es necesario aprender estas teorías para poder utilizar el modelo relacional.

Este modelo persigue, al igual que la mayoría de los modelos de datos, los siguientes **objetivos**:

- ✓ **Independencia física de los datos:** el modo de almacenamiento de los datos no debe influir en su

manipulación lógica.

- ✓ **Independencia lógica de los datos:** los cambios que se realicen en los objetos de la BD no deben repercutir en los programas y usuarios que accedan a ella.
- ✓ **Flexibilidad:** presentar los datos a los usuarios de la forma más adecuada a la aplicación que utilicen.
- ✓ **Uniformidad:** presentación de las estructuras lógicas en forma de tablas, fáciles de comprender y manipular por los usuarios.
- ✓ **Sencillez:** las características anteriores, junto con los lenguajes de usuario hacen que este modelo sea fácil de entender y utilizar por el usuario.

Para conseguir estos objetivos, Codd introduce el concepto de **relación (tabla)** como **estructura básica del modelo**. Todos los datos de una BD se representan en forma de relaciones cuyo contenido varía con el tiempo. Los **conceptos básicos de este modelo son:**

- ✓ **Interrelación** es la asociación entre dos tablas.
- ✓ **El conjunto de columnas representa las propiedades de la tabla y se denominan atributos.**
- ✓ **El conjunto de las filas se denomina tuplas y contienen los valores** que toman cada uno de los atributos para cada elemento de la relación.

Para Codd lo importante es el diseño lógico, un modelo abstracto, por lo que no indica nada acerca de la implementación de este modelo en SGBD comerciales. Sin embargo, los trabajos de Codd y otros investigadores dieron lugar a diversas bases de datos comerciales y es el modelo lógico en el que se basan la mayoría de los SGBD existentes actualmente en el mercado.

Suele haber confusión entre la terminología que utilizan las BD y los sistemas de archivos. En ocasiones encontraremos que se hace referencia a las filas como registros, a las columnas como campos y a las tablas como archivos. La diferencia está en que la tabla en una BD es un concepto lógico, mientras que los conceptos archivo, registro y campo son conceptos físicos.

En 1985, Codd publica sus famosas **doce reglas** que debe cumplir cualquier BD para ser considerada relacional:

- ✓ **Regla de información:** toda información de una BD relacional está representada mediante valores en tablas.
- ✓ **Regla de acceso garantizado:** se garantiza que todos los datos de una BD relacional son lógicamente accesibles a través de una combinación de nombre de tabla, valor de clave primaria y nombre de columna.
- ✓ **Tratamiento sistemático de valores nulos:** los valores nulos se soportan en los SGBD para representar la falta de información de un modo sistemático e independiente de los tipos de datos.
- ✓ **Catálogo en línea dinámico basado en el modelo relacional:** la descripción de la BD se representa en el ámbito lógico de la misma forma que los datos ordinarios.
- ✓ **Regla de sublenguaje completo de datos:** un sistema relacional puede soportar varios lenguajes y varios modos de uso terminal.
- ✓ **Regla de actualización de vista:** todas las vistas, que sean teóricamente actualizables, son también actualizables por el sistema.
- ✓ **Inserción, actualización y supresión de alto nivel.**
- ✓ **Independencia física de datos.**
- ✓ **Independencia lógica de los datos.**
- ✓ **Independencia de la integridad:** las restricciones de integridad específicas de una BD relacional deben ser definidas mediante el sublenguaje de datos relacional y almacenarse en el catálogo de la BD.
- ✓ **Independencia de la distribución:** un SGBD es independiente de la distribución.
- ✓ **Regla de no subversión:** si un SGBD tiene un lenguaje de bajo nivel no se puede utilizar para destruir o evitar las reglas de integridad o las restricciones expresadas en el lenguaje relacional de alto nivel.

El modelo relacional propone una representación de la información que:

- ✓ Origine esquemas que representen fielmente la información, los objetos y las relaciones.
- ✓ Sea fácilmente entendida por los usuarios.
- ✓ Sea posible ampliar el esquema de la BD sin modificar la estructura lógica existente y los programas de aplicación.
- ✓ Permita la máxima flexibilidad en la formulación de los interrogantes sobre la información mantenida en la BD.

9.3 Estructura y características.

El modelo relacional se representa gráficamente como una tabla bidimensional en la que las filas corresponden a registros individuales y las columnas corresponden a campos o atributos de esos registros.

Se han introducido algunos elementos que intervienen en el modelo relacional. De forma más detallada son:

- ✓ Una relación es una tabla con columnas y filas. El usuario percibe la BD como un conjunto de tablas. Una tabla sirve para almacenar los datos de una entidad.
- ✓ Un atributo es el nombre de una columna de una relación. Ejemplo: atributo nombre definido sobre el dominio: conjunto de 15 caracteres.
- ✓ Una tupla es cada fila de una relación (o tabla). El orden de las tuplas no es relevante.
- ✓ El grado de una relación es el número de atributos que contiene, es decir el número de columnas de la tabla.
- ✓ La cardinalidad de una relación es el número de tuplas que contiene.
- ✓ El valor es la intersección entre una fila y una columna.

Una tabla tiene una serie de características:

- ✓ Una tabla se percibe como una estructura bidimensional compuesta de filas y columnas.
- ✓ Cada tabla o relación tiene un nombre que la diferencia de las demás.
- ✓ Cada fila de la tabla se denomina tupla y representa la ocurrencia de una entidad.
- ✓ No admiten filas duplicadas.
- ✓ El orden de las filas no es significativo.
- ✓ Cada columna representa un atributo, y cada columna tiene un nombre distinto.
- ✓ El orden de las columnas es irrelevante. No están ordenadas.
- ✓ La tabla es plana, es decir, en la intersección fila/columna solo puede haber un valor, no se admiten atributos multivaluados.
- ✓ Cada tabla debe tener un atributo o conjunto de atributos que identifique a cada fila de forma única.
- ✓ Cada columna tiene un mismo tipo de datos y un intervalo de valores específico: dominio de un atributo.

Veamos el siguiente ejemplo: Relación *EMPLEADOS*.

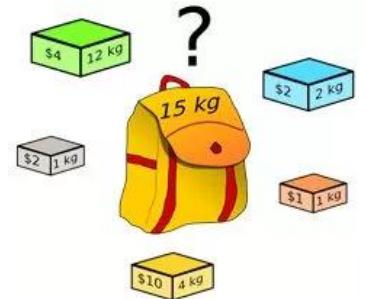


La visualización desde un punto de vista lógico de una BD relacional se corresponde con un conjunto de tablas relacionadas. Para el usuario una tabla es un conjunto de entidades.

9.4 Restricciones del modelo relacional.

En todos los modelos de datos existen restricciones que deben tenerse en cuenta a la hora de diseñar una BD. Los datos almacenados en la BD deben cumplir con una serie de reglas para garantizar que sean correctos. El modelo relacional, como todo modelo de datos, también presenta ocurrencias no permitidas. Los tipos de restricciones pueden ser:

1. Restricciones inherentes al modelo: son restricciones propias del modelo e indican las características propias de una relación, por tanto, han de cumplirse obligatoriamente.

- ✓ Ausencia de tuplas repetidas.
 - ✓ Irrelevancia del orden de las tuplas.
 - ✓ Irrelevancia del orden de los atributos.
 - ✓ Cada atributo solo puede tomar un único valor del dominio al que pertenece.
2. **Restricciones semánticas o de usuario:** son impuestas para cada problema concreto, desde la definición de los dominios de un campo a condiciones impuestas a un campo de acuerdo con el valor de otros. Para establecer estas condiciones disponemos de las siguientes restricciones:
- ✓ **La restricción de clave primaria (PRIMARY KEY):** permite declarar uno o varios atributos como clave primaria de una relación.
- En nuestro ejemplo la clave principal de la relación *CLIENTES* sería el *CodCliente*.
- ✓ **Restricciones de verificación (CHECK):** comprueban en una actualización si se cumplen las condiciones exigidas en la restricción o no, antes de ejecutarla.
- Por ejemplo: que la edad de un trabajador sea mayor o igual a 16
- ✓ **Restricción de unicidad (UNIQUE):** permite definir claves alternativas. Los valores de los atributos no pueden repetirse.
- Por ejemplo: en la tabla *CLIENTES* la clave principal seleccionada es el *CodCliente*, pero el atributo *DNI*, que es clave alternativa en dicha tabla, tampoco admite valores duplicados. Para ello establecemos una restricción de tipo *UNIQUE* para este campo.
- ✓ **Restricción de obligatoriedad (NOT NULL):** permite declarar si uno o varios atributos pueden tomar valores nulos. Por definición una clave principal no puede contener valores nulos en ninguno de los atributos que la componen (integridad de clave) por tanto se definirán como *NOT NULL*.
- Por ejemplo: la restricción *NOT NULL* puede ser necesaria también con otro tipo de atributos como el *DNI* citado en el ejemplo anterior, el *nombre*, etc.
- ✓ **Los disparadores (TRIGGER):** son acciones, métodos según la terminología orientada a objetos, que ejecuta un objeto, como respuesta a un evento o acción que se realiza sobre él, cuando se cumple una determinada condición.
- Por ejemplo, si tenemos dos tablas: una con los datos de los vehículos de nuestros clientes (matrícula y resto de datos que identifican al vehículo) y otra que recoge cada una de las entradas en el taller de esos vehículos (matrícula, fecha y hora), podemos crear un trigger que cada vez que se inserte un vehículo en la tabla de vehículos añada una fila en la tabla entradas en taller con los datos anteriores.
- 
3. **Otras restricciones** inherentes son:
- ✓ **Integridad de clave:** es una restricción que exige que todos los atributos de la clave primaria (*PRIMARY KEY*), han de contener valores no nulos (*NOT NULL*).
- En el ejemplo de la relación *FACTURAS-LÍNEAS*, la clave principal de la tabla *LÍNEAS* es una clave compuesta de los campos *NumFactura+NumLinea*. Esta restricción implica que ambos atributos deben contener valores que no sean nulos.
- ✓ **Integridad referencial (FOREIGN KEY):** consiste en que no puede haber un valor en una clave ajena de una tabla, si antes no existe en la tabla de la que ese campo o conjunto de campos formen la clave primaria.
- Para explicar este concepto denominamos tabla hija a la tabla que contiene la clave ajena y tabla padre a la tabla que contiene la clave principal.
- Se permiten valores nulos en las claves ajenas; es decir que una clave ajena debe coincidir con un valor de clave primaria de la relación a la que apunta o tener valor nulo.

Por ejemplo: para relacionar un vehículo con su propietario establecemos una clave ajena (*CodCliente*) en la tabla *VEHÍCULOS* que coincide con el *CodCliente* correspondiente en la tabla *CLIENTES*. La integridad referencial establece que no podemos introducir ningún código de cliente (clave ajena) en la tabla (*VEHICULOS*) si no existe previamente ese código en la tabla *CLIENTES* (clave principal).

9.4.1 Restricciones y operaciones relacionales.

Además de definir las claves ajenas hay que tener en cuenta las operaciones de borrado y actualización que se realizan sobre las filas de la tabla relacionada.

A la hora de hacer operaciones de inserción, actualización o borrado aparecen problemas, ya que hay que tener en cuenta los tres tipos de restricciones establecidas, sobre todo las restricciones de clave primaria e integridad de clave.

Respecto a la **actualización de campos** hay que tener en cuenta:

- ✓ No pueden modificarse los atributos que forman parte de la clave primaria para dar origen a registros con valores de clave primaria redundantes.
- ✓ La modificación de atributos que forman la clave principal no puede dar lugar a que aparezcan en esos atributos valores nulos.
- ✓ Para aceptar los cambios en un atributo, se han de cumplir las restricciones de clave, referenciales y de usuario que hayan sido definidas para ese atributo en la tabla a la que pertenezca.
- ✓ Si se modifica el valor de un atributo que forme parte de la clave primaria, habrá que sustituir los valores de los atributos que forman claves ajenas, en los que aparece el valor antes de modificar el contenido del atributo, por su nuevo valor. Es lo que se denomina actualización en cascada.

Respecto de la de **inserción de campos**:

- ✓ No pueden añadirse registros si alguno de los atributos que forman parte de la clave primaria posee un valor nulo.
- ✓ No pueden añadirse registros si el conjunto de valores de los atributos que forman parte de la clave primaria origina una clave primaria redundante.
- ✓ Como consecuencia de la actualización en cascada, para añadir un valor a un atributo que forma parte de la clave ajena, ese valor ha de existir en la clave primaria a la que hace referencia la clave ajena.
- ✓ Para aceptar los cambios en un atributo se han de cumplir las restricciones de clave, referenciales y de usuario que hayan sido definidas para ese atributo en la tabla a la que pertenezca.

Respecto del **borrado de campos**:

- ✓ Al borrar una la que tenga asociados, mediante la clave primaria, otras filas en otras tablas, se pierde la consistencia de la tabla. Para solucionarlo existen dos posibilidades:
 - **Borrado en cascada:** consiste en borrar todas las filas de las tablas en los que aparezca como clave ajena el conjunto de atributos que forman la clave primaria del registro a borrar.
 - **Para borrar un registro cuya clave primaria es clave ajena en otras tablas, primero hay que borrar los registros con la clave primaria asociada, para después borrar el registro deseado.**
- ✓ Teniendo en cuenta las consideraciones anteriores las posibilidades son las siguientes:
 - **Borrado y/o modificación en cascada (CASCADE).** El borrado o modificación de una fila en la relación padre (relación con la clave primaria) ocasiona un borrado o modificación de las filas relacionadas en la relación hija (relación que contiene la clave ajena).
 - **Borrado y/o modificación restringido (RESTRICT).** En este caso no es posible realizar el borrado o la modificación de las filas de la relación padre si existen las relacionadas en la relación hija.
 - **Borrado y/o modificación con puesta a nulos (SET NULL).** Esta restricción permite poner la clave ajena en la tabla referenciada a NULL, si se produce el borrado o modificación en la tabla primaria o padre.
 - **Borrado y/o modificación con puesta a valor por defecto (SET DEFAULT).** El valor que se pone en las claves ajenas de la tabla referenciada es un valor por defecto que se habrá especificado en la creación de la tabla.

10 Paso del diagrama E/R al modelo relacional.

Una vez obtenido el esquema conceptual mediante el modelo E/R, hay que definir el modelo lógico de datos. Las **reglas básicas** para transformar un esquema conceptual E/R a un esquema relacional son las siguientes:

- ✓ Toda **entidad** se transforma en **una tabla**.
- ✓ Todo **atributo** se transforma en **columna dentro de la tabla**.
- ✓ El **identificador único de la entidad** se convierte en **clave primaria**.
- ✓ Como las **relaciones** del modelo E/R no tienen equivalente en el modelo relacional, ya que sólo existen tablas y operaciones entre ellas, es necesario aplicar lo siguiente:
 - En las **relaciones M:N** se crea una **nueva tabla que tendrá como clave primaria la concatenación de los atributos clave de las entidades que asocia y con los atributos propios de la relación si los hay**. Esta tabla posee dos claves ajenas, una por cada entidad con la que está relacionada.



- En las **relaciones 1:N** la **entidad del lado N de la relación añade el conjunto de campos necesarios para incorporar a sus atributos la totalidad de la clave primaria de la entidad del lado 1, creando una clave ajena**, de modo que se puedan relacionar ambas tablas mediante operadores relacionales. El nombre de la relación desaparece.



- Las **relaciones 1:1** se transforman en función de las cardinalidades:
 - Propagando cualquiera de los atributos identificadores y sus atributos asociados creando una **única tabla con el conjunto de los atributos de ambas entidades**. La clave primaria sería cualquiera de las dos. Ambas entidades participan con cardinalidades (1,1).
 - Propagando la clave de cualquier tabla a la otra, según cuál sea la que tenga accesos más frecuentes. Ambas cardinalidades (1,1).
 - Propagar la clave de la entidad con cardinalidad (1,1) a la entidad que tenga (0,1).



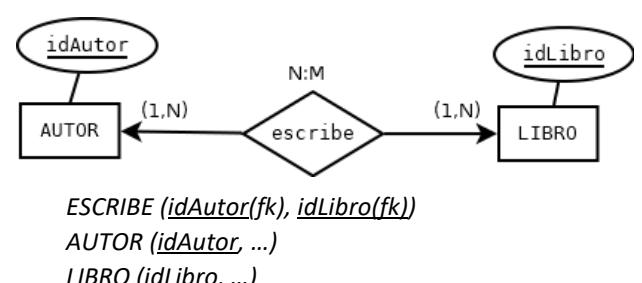
- **Crear una nueva tabla a partir de la relación con las dos claves de ambas**. Cuando ambas tablas tienen cardinalidades (0,1).

Además de las reglas de transformación que acabamos de ver y que se aplican con carácter general, existen otros aspectos a tener en cuenta a la hora de obtener un esquema relacional a partir de un modelo entidad/relación.

10.1 Relaciones M:N.

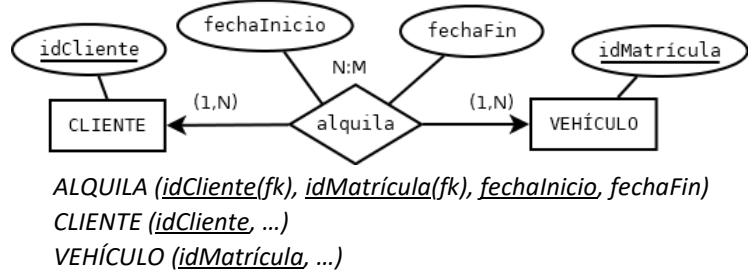
TRANSFORMACIÓN DE INTERRELACIÓN M:N.

Regla general: **se transforman en una nueva tabla cuya clave se forma, al menos, con la concatenación de las claves de las entidades que participan en la relación**, que son además claves ajenas que referencian a las tablas en las que son claves primarias. El **nombre asignado a la tabla es el que tenía la relación**.



TRANSFORMACION DE LA DIMENSION TEMPORAL.

En algunos **casos en que la relación tenga atributos de tipo fecha, será necesario incluir al menos una fecha como parte del atributo identificador principal para recoger la dimensión temporal del modelo.** En otros casos la fecha puede ser una entidad más o solo un atributo.



10.2 Relaciones 1:N.

TRANSFORMACION DE INTERRELACION 1:N.

Como norma general **se propaga la clave de la entidad que tiene cardinalidad máxima 1 a la que tiene cardinalidad máxima N.**

EXCEPCIONES:

En los siguientes casos interesa más crear una nueva tabla a partir de la relación como en el caso de correspondencias M:N :

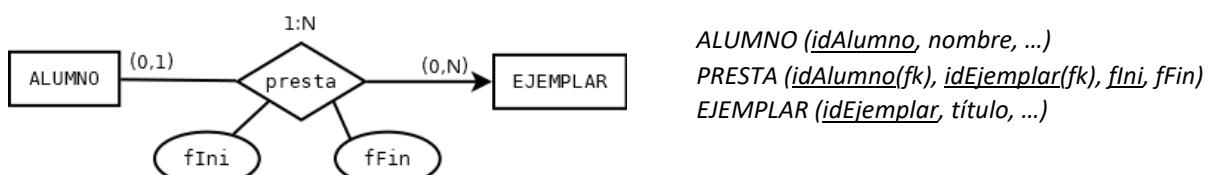
- 1) **Cuando el número de ocurrencias de la entidad que propaga la clave es muy pequeño y cabe la posibilidad de que al propagar la clave quedan muchos valores repetidos o nulos** (Ej.: cuando el número de empleados que estén afiliados es muy pequeño).



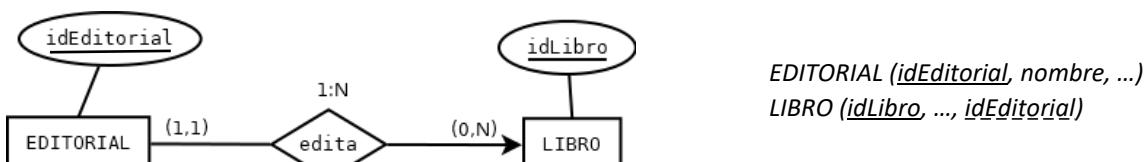
- 2) **Cuando se prevea que en el futuro se puede convertir en una relación N:M** (Ej.: si en un futuro una misma pieza puede ser suministrada por distintos proveedores).



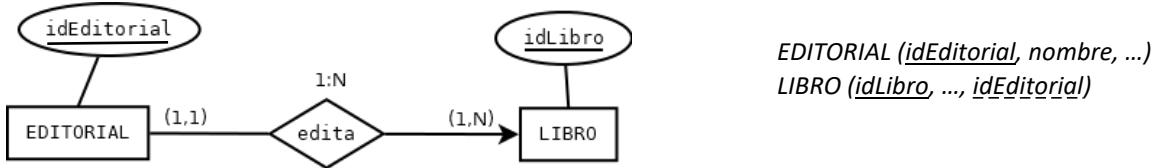
- 3) **Cuando la relación tenga atributos propios.** En algunos casos se pueden migrar estos atributos junto con la clave, pero en general, se crea una nueva tabla (Ej.: un alumno puede retirar varios ejemplares. No se recoge la dimensión temporal).

CARDINALIDADES:

- 1) **Cuando la entidad que tiene cardinalidad máxima 1, tiene también 1 de cardinalidad mínima, tendremos que tener en cuenta al propagar la clave que en la tabla que recibe la clave, como clave extranjera, no pueda tener valores nulos.**

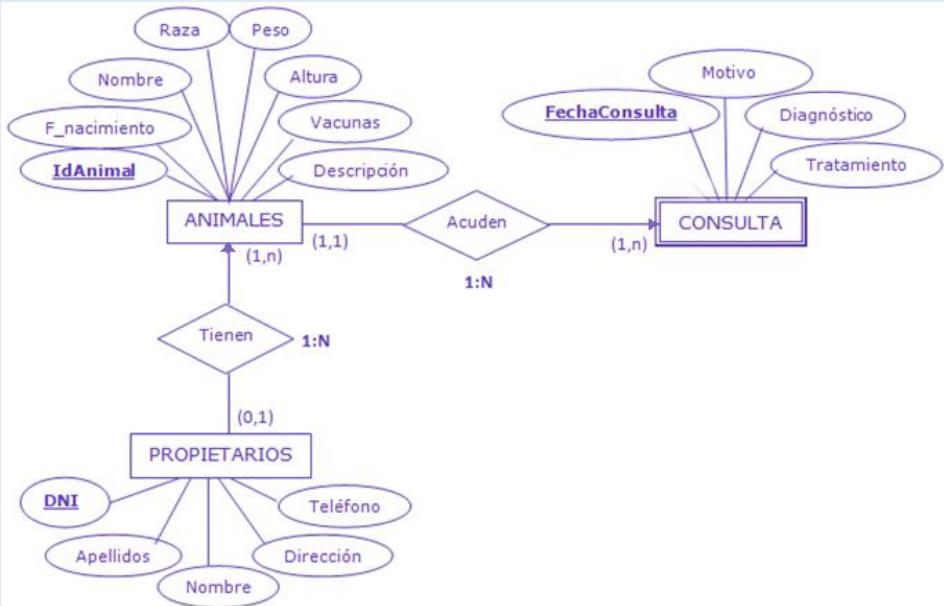


- 2) **Cuando la entidad que tiene cardinalidad máxima n, tiene de cardinalidad mínima 1, tendremos que controlar por software que, al dar de alta una, la de la otra tabla se introduzca al menos una fila en esta.**



Ejercicio resuelto

Partiendo del diagrama E/R entre las entidades PROPIETARIOS, ANIMALES Y CONSULTAS. Se pide obtener el modelo relacional correspondiente.



Solución:

ANIMALES (*IdAnimal*, *F_nacimiento*, *Nombre*, *Raza*, *Peso*, *Altura*, *Vacunas*, *Descripción*, *DNI*)

CONSULTA (*IdAnimal*(fk), *FechaConsulta*, *Motivo*, *Diagnóstico*, *Tratamiento*)

PROPIETARIOS (*DNI*, *Apellidos*, *Nombre*, *Dirección*, *Teléfono*)

10.3 Relaciones 1:1.

No hay una regla fija, puede optarse por la solución basada en:

- ✓ Recoger la mayor cantidad de semántica posible.
- ✓ Tener en cuenta las cardinalidades mínimas.
- ✓ Evitar los valores nulos.
- ✓ Motivos e eficiencia.

1) Cuando las **cardinalidades de ambas entidades son (1,1)** se pueden adoptar distintas soluciones:

- a) No se necesitan 2 tablas, se puede **crear una única tabla en la que se incluyan los atributos de las dos entidades, cuya clave principal será cualquiera de los atributos identificadores principales.**

ESTAR_CASADO (idEsposo, ..., idEsposa, ...)



- b) **Propagar la clave de cualquiera de ellas a la otra tabla, teniendo en cuenta a cuál de ellas se le efectúan los accesos más frecuentes.**

ESPOSO (idEsposo, ..., idEsposa)

ESPOSA (idEsposa, ...)

- c) **Propagar las dos claves, introduce redundancias que se controlarán mediante restricciones.**

ESPOSO (idEsposo, ..., idEsposa)

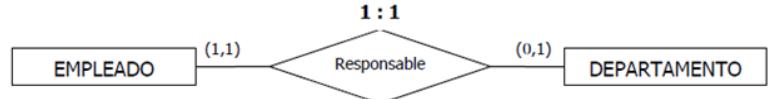
ESPOSA (idEsposa, ..., idEsposo)

- 2) Cuando las **cardinalidades de ambas entidades son (0,1)**, la relación se transforma en una tabla para evitar los valores nulos.



*MATRIMONIO (idHombre(fk), idMujer(fk))
HOMBRE (idHombre, ...)
MUJER (idMujer, ...)*

- 3) Si **una tiene cardinalidad (1,1) y la otra (0,1)**, se propaga la clave de la entidad que tiene cardinalidad (1,1) a la que tiene (0,1).



*EMPLEADO (idEmpleado, ...)
DEPARTAMENTO (idDepartamento, ..., idEmpleado)*

Habrá que controlar además que la clave propagada como clave ajena no pueda tomar valores nulos.

10.4 Relaciones Reflexivas.

TRANSFORMACION DE UNA INTERRELACION REFLEXIVA.

Se trata de relaciones en las que solo participa una entidad. **Como regla general toda relación reflexiva se convierte en dos tablas: una para la entidad y otra para la relación.**

Se pueden presentar los siguientes casos:

Relación 1:1.

No se crea una tabla para la relación. La clave de la entidad se repite, con lo que la tabla resultante tendrá ese atributo dos veces, una como clave primaria y otra como clave ajena de ella misma.

PERSONA (DNI, nombre, ..., DNI_Espos@)

Relación 1:N.

En este caso hay que tener en cuenta la cardinalidad del lado muchos:

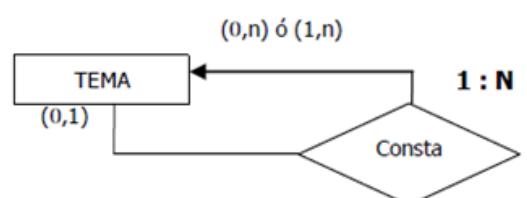
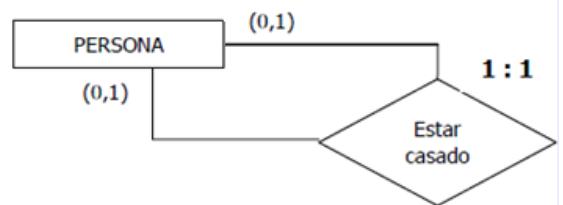
- a) **Si no es obligatoria se crea una nueva tabla cuya clave será la de la entidad del lado muchos (idTemaS) y se propaga la clave a la nueva tabla como clave ajena (cardinalidad mínima 0 o 1).**

CONSTA (idTemaS, idTemaP)

TEMA (idTemaS, ...)

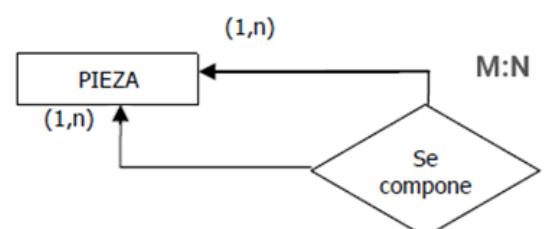
- b) **Si es siempre obligatoria no se crea una nueva tabla (cardinalidad mínima 1).**

TEMA (idTemaS, ..., idTemaP)



Relación M:N.

La tabla que resulta de la relación contendrá dos veces la clave primaria de la entidad del lado muchos, más los atributos de la relación, si los hay. La clave de esta nueva tabla será la combinación de las dos.
Ej.: cada pieza se compone de muchas piezas que, a su vez están compuestas por piezas.



PIEZA (idPieza, descripción, ...)

SE_COMPONE (idPiezaC(fk), idPieza(fk)) → idPiezaC es la pieza que se compone de otras

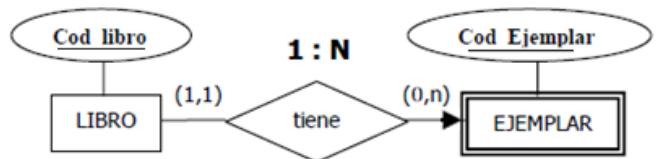
10.5 Otras.

DEPENDENCIAS.

- ✓ En existencia: se propaga la clave, creando una clave ajena con las características de
 - No permitir nulos.
 - Modificación en cascada.
 - Borrado en cascada.
- ✓ En existencia e identificación: (relaciones fuertes: IDENTIFICATIVAS) la propagación de la clave se hace de forma que la clave primaria de la entidad débil estará formada por la concatenación de las dos claves.

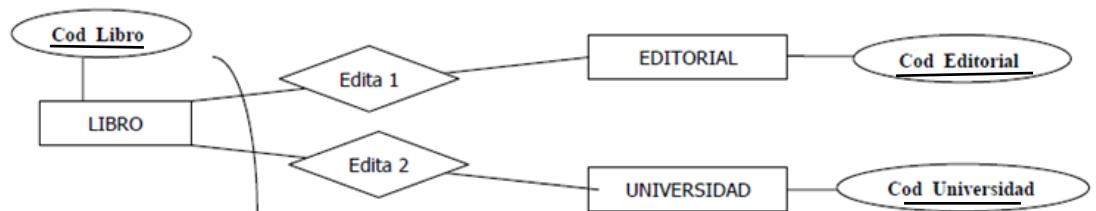
LIBRO (Cod libro, ...)

EJEMPLAR (Cod libro/fk, Cod Ejemplar, ...)



RELACIONES EXCLUSIVAS.

Se propaga la clave de las entidades que se excluyen, a la otra entidad, como claves ajenas.



Para que se cumpla la exclusividad es necesario definir restricciones en cada caso. Por ejemplo, chequeando que, si una de las claves ajenas tiene datos, la otra sea nula y viceversa.

LIBRO (Cod libro, ..., Cod Universidad, Cod Editorial)

UNIVERSIDAD (Cod Universidad, ...)

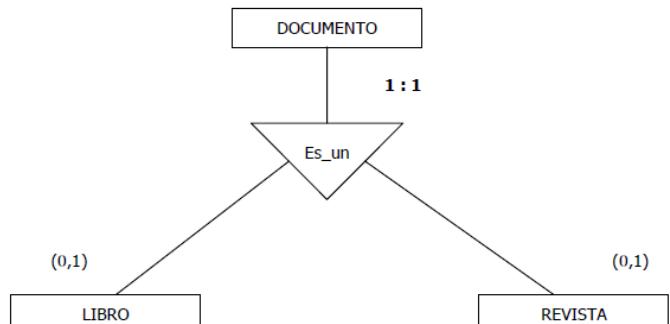
EDITORIAL (Cod Editorial, ...)

TRANSFORMACIÓN DE JERARQUÍAS.

Pueden darse 3 opciones:

- Crear una sola tabla con todos los atributos de la entidad y de los subtipos, añadiendo como un atributo más el atributo discriminante (integración de los subtipos en el supertipo). Esto se aplica cuando:

- ✓ Los subtipos se diferencian en muy pocos atributos.
- ✓ Las relaciones que los asocian al resto de las entidades sean las mismas para los subtipos.



Si la jerarquía es:

- ✓ Total: el atributo discriminante no admitirá nulos.
- ✓ Parcial: el atributo discriminante si admitirá nulos.

Si entre los subtipos puede haber:

- ✓ Solapamiento: se forman grupos repetitivos, por tanto, será necesario crear una nueva tabla que asocie el atributo discriminante con el supertipo.
- ✓ Exclusividad: no es necesaria una tabla nueva.

DOCUMENTO (Código, título, idioma, ..., tipo)

- Crear una tabla para cada tipo y subtipos que haya. Esto se aplica cuando:

- ✓ Existen muchos atributos distintos entre los subtipos.
- ✓ Se quieren mantener los atributos comunes en una tabla.

DOCUMENTO (Código, título, idioma, ...)

LIBRO (Código, ...)

REVISTA (Código, ...)

- c) Crear **una tabla por cada subtipo, incluyendo los atributos comunes en cada una** (eliminación del supertipo).

Esto se aplica cuando:

- ✓ Existen **muchos atributos distintos entre los subtipos**.
- ✓ **Los accesos a los datos de los subtipos siempre afectan a los atributos comunes.**

LIBRO (Código, título, idioma, ...)

REVISTA (Código, título, idioma, ...)

Ventajas e inconvenientes:

Opción a) → Es la más rápida por tener que acceder a una sola entidad.

Opción b) → La menos eficiente. La mejor desde un punto de vista semántico.

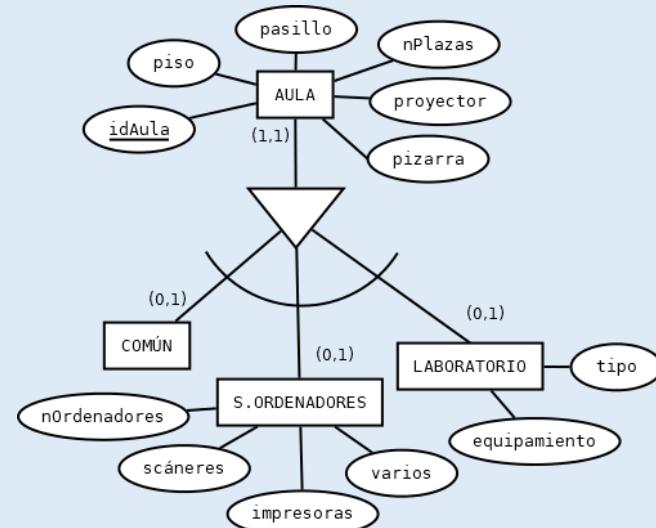
Opción c) → Más eficiente en consultas que afecten a todos los atributos de un subtipo. Menos eficiente en consultas que afecten a los atributos comunes. Introduce redundancias. Es la que pierde más semántica.

Ejercicio resuelto

Partiendo del ejercicio resuelto en el que se establece el modelo entidad-relación correspondiente a la clasificación de la entidad AULA en los subtipos: COMÚN, S. ORDENADORES y LABORATORIO. Obtener el modelo relacional correspondiente.

Solución: opción b)

- ✓ AULA (idAula, piso, pasillo, nPlazas, proyector, pizarra)
- ✓ COMÚN (idAula)
- ✓ S. ORDENADORES (idAula, nOrdenadores, scáneres, impresoras, varios)
- ✓ LABORATORIO (idAula, tipo, equipamiento)

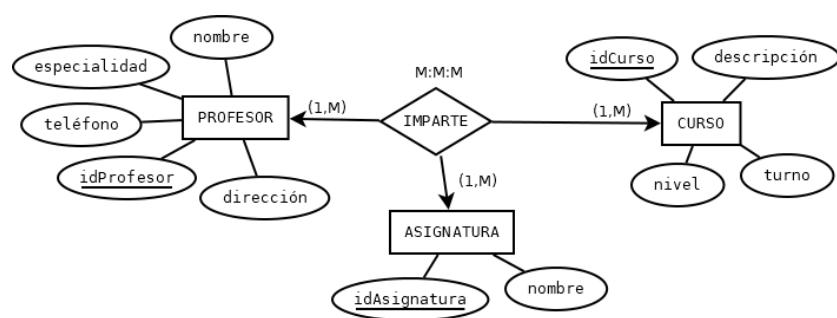


RELACIONES N-ARIAS.

En este tipo de relaciones **se agrupan 3 o más entidades, y para pasar al modelo de datos relacional cada entidad se convierte en una tabla, así como la relación, que va a contener los atributos propios de ella más las claves de todas las entidades. La clave de la tabla resultante será la concatenación de las claves de las entidades**. Hay que tener en cuenta:

- ✓ Si la **relación es M:M:M**, es decir, si 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**.

Esa tabla incluirá los atributos de la relación si los hubiera.



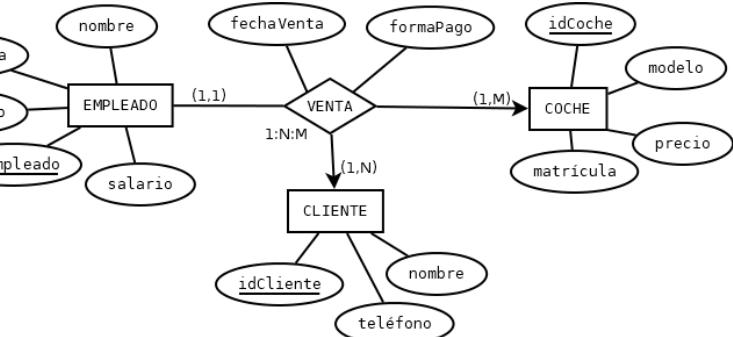
PROFESOR (idProfesor, nombre, dirección, teléfono, especialidad)

CURSO (idCurso, descripción, nivel, turno)

ASIGNATURA (idAsignatura, nombre)

IMPARTE (idProfesor(fk), idCurso(fk), idAsignatura(fk))

- ✓ 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.



CLIENTE (idCliente, nombre, teléfono)

COCHE (idCoche, matrícula, modelo, precio)

EMPLEADO (idEmpleado, nombre, teléfono, salario, fechaAlta)

VENTA (idCoche(fk), idCliente(fk), idEmpleado, formaPago, fechaVenta)

10.6 Pérdida de la semántica en la transformación al modelo relacional.

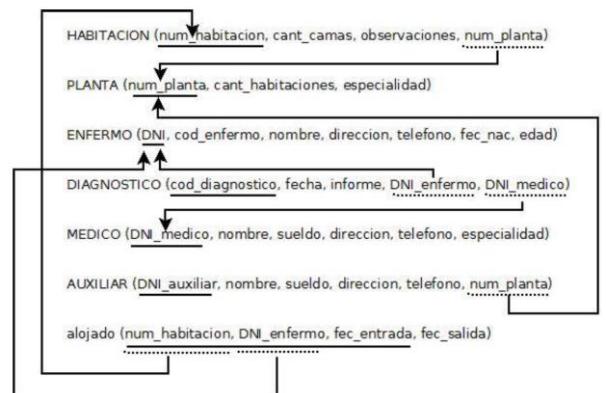
Algunas restricciones son necesarias controlarlas con mecanismos externos al modelo relacional dado que muchos SGBD no implementan el modelo relacional completo (hay que recurrir a otros medios como disparadores, procedimientos almacenados o aplicaciones externas).

Algunas de las **restricciones** de los esquemas E/R que deben contemplarse en la transformación al modelo relacional mediante checks, aserciones o disparadores son:

- ✓ **Cardinalidades mínimas de 1 en las relaciones N:M y 1:N** (excluyendo aquellas que se controlan con las restricciones NOT NULL cuando se realiza una propagación de clave).
- ✓ **Cardinalidades máximas conocidas** en las relaciones binarias N:M y 1:N y relaciones ternarias.
- ✓ **Exclusividad en las generalizaciones.**
- ✓ **Inserciones y borrado en las generalizaciones.**
- ✓ **Restricciones que no figuren en el enunciado inicial pero que se consideran adecuadas** o convenientes.

10.7 Grafos relacionales.

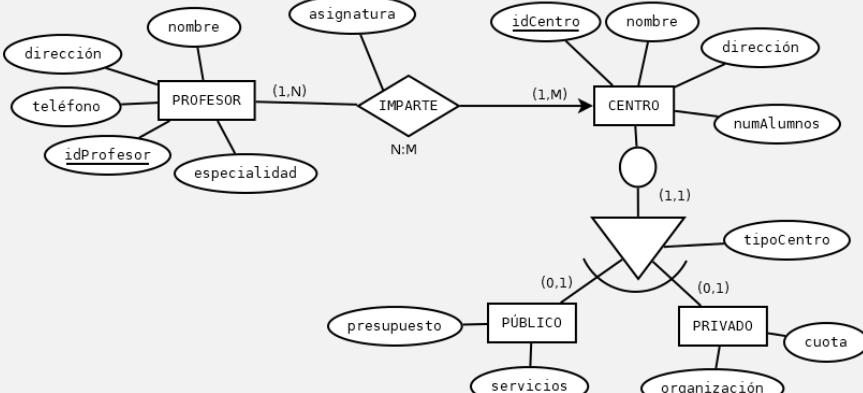
Es un esquema relacional en el que hay líneas que enlazan las claves principales con las claves secundarias para representar mejor las relaciones.



TAREA

8. Dado el esquema E/R siguiente.

Según cada una de las reglas indicadas, ¿Qué esquema E/R obtendremos? ¿con qué tablas contará el modelo relacional resultante?



9. Realiza el diagrama E/R que cumpla

las especificaciones y pásalo al modelo de datos relacional.

Se desea mecanizar la biblioteca de un centro educativo. En la biblioteca existen fichas de autores y libros. Un autor puede escribir varios libros, y un libro puede ser escrito por varios autores. Un libro está formado por

ejemplares que son los que se prestan a los usuarios.

Así un libro tiene muchos ejemplares y un ejemplar pertenece sólo a un libro. De los ejemplares nos interesa saber la localización dentro de la biblioteca. Los ejemplares son prestados a los usuarios, un usuario puede tomar prestados varios ejemplares y un ejemplar puede ser prestado a varios usuarios. Del préstamo nos interesa saber la fecha de préstamo y la de devolución.

10. Realiza el diagrama E/R que cumpla las especificaciones y pásalo al modelo de datos relacional.

Se desea informatizar la gestión de los proyectos del departamento de química de una universidad siguiendo las siguientes especificaciones:

- *Al departamento llegan una serie de clientes que quieren realizar proyectos. Generalmente los clientes son empresas que realizan contratos con el grupo de investigadores del departamento. Un cliente puede realizar varios proyectos.*
- *Un proyecto es de un cliente. Cada proyecto tiene asignada una cuantía de dinero que se utilizará para pagar los gastos del proyecto. De esta cuantía se saca el dinero para realizar los pagos a los colaboradores. También nos interesa saber de los proyectos el nombre, la fecha de comienzo, la de fin, entre otros.*
- *De cada proyecto se realizan muchos pagos para pagar a los colaboradores.*
- *De los pagos nos interesa saber el concepto, la cantidad, el IVA aplicado y la fecha del pago.*
- *Existen varios tipos de pagos (por ejemplo: nómina, representación, material, etc.). Un pago es de un tipo de pago, y a un tipo de pago pueden pertenecer muchos pagos.*
- *Existen una serie de colaboradores que son personas o entidades que van a recibir el dinero de los pagos en concepto de una tarea realizada o la compra de material. Un pago sólo puede ser para un colaborador. Este a su vez puede recibir muchos pagos.*
- *De los colaboradores nos interesa saber: nombre, NIF, domicilio, teléfono, retención, banco y nº de cuenta.*

11. Obtén el esquema relacional a partir del modelo E/R del ejercicio 6.

12. Obtén el esquema relacional a partir del modelo E/R del ejercicio 7.

11 Normalización de modelos relationales.

¿Crees que tu BD ya podría construirse directamente sobre el SGBD relacional que hayas elegido? La respuesta podría ser afirmativa, pero si queremos que nuestra BD 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 BD, 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. Este proceso de refinamiento conseguirá los siguientes objetivos:

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

El proceso de normalización.

El esquema relacional debe modelizar la realidad y para obtenerlo hemos seguido el camino que consiste en, primero, efectuar el diagrama Entidad-Relación para luego efectuar la traducción al modelo relacional. Si el diagrama Entidad-Relación era correcto, habremos obtenido un esquema relacional del todo correcto. Este sería el camino aconsejable. Pero no siempre es así. **Es posible que un buen diseño produzca tablas defectuosas con redundancia de**

datos e inconsistencia. Estos problemas, cuando existan, pueden ser eliminados mediante el proceso de 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 y dependencia funcional transitiva**.

¿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. Existen **varias formas normales: Primera, Segunda, Tercera, Boyce-Codd, Cuarta, Quinta y Dominio-Clave**. Como se ha 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.

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 (3FN)**. En los siguientes epígrafes se describen las características y requisitos de cada una de las formas normales.

11.1 Tipos de dependencias.

Vamos a desarrollar aquí los conceptos sobre los que se basa el análisis de dependencias entre atributos, que se lleva a cabo en el proceso de normalización antes indicado, son los siguientes:

- ✓ **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.
- ✓ **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. Notación: A₁, A₂, ... → B.
- ✓ **Dependencia Funcional 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. Notación A → B → C.

Ejercicio resuelto

Dadas las siguientes tablas:

EMPLEADO (DNI, Nombre, Dirección, Localidad, Cod_Localidad, Nombre_hijo, Edad_hijo)

LIBRO (Título_libro, Num_ejemplar, Autor, Editorial, Precio)

Resuelve las siguientes cuestiones:

- Indica qué atributos presentan una dependencia funcional de la clave primaria de la tabla EMPLEADO.
- Indica qué atributos presentan una dependencia funcional completa en la tabla LIBRO.
- Indica qué atributos presentan una dependencia transitiva en la tabla EMPLEADO.

Apartado a)

Los atributos Nombre, y Dirección dependen funcionalmente de DNI, ya que para un DNI específico sólo podrá haber un nombre y una dirección. Pero los atributos Nombre_hijo y Edad_hijo no presentan esa dependencia funcional de DNI, ya que para un DNI específico podríamos tener varios valores diferentes en esos atributos (varios hijos). Expresemos estas dependencias funcionales mediante su notación:

$DNI \rightarrow Nombre, Dirección$

Apartado b)

Los atributos Editorial y Precio dependen funcionalmente del conjunto de atributos que forman la clave primaria de la tabla, pero no dependen de Título_libro o de Num_ejemplar por separado, por lo que presentan una dependencia

funcional completa de la clave. El atributo Autor depende funcionalmente sólo y exclusivamente de Titulo_libro, por lo que no presenta una dependencia funcional completa de los atributos que forman la clave.

Apartado c)

Los atributos Cod_Localidad y Localidad dependen funcionalmente de DNI, pero entre Cod_Localidad y Localidad existe otra dependencia funcional. Por tanto, se establece que Localidad depende funcionalmente de Cod_Localidad, y a su vez, Cod_Localidad depende funcionalmente de DNI. Con lo que podemos armar que existe una dependencia transitiva entre Localidad y DNI. Si lo representamos con la notación asociada a las dependencias funcionales, quedaría: DNI → Cod_Localidad → Localidad.

11.2 Formas Normales.

11.2.1 1^a Forma Normal.

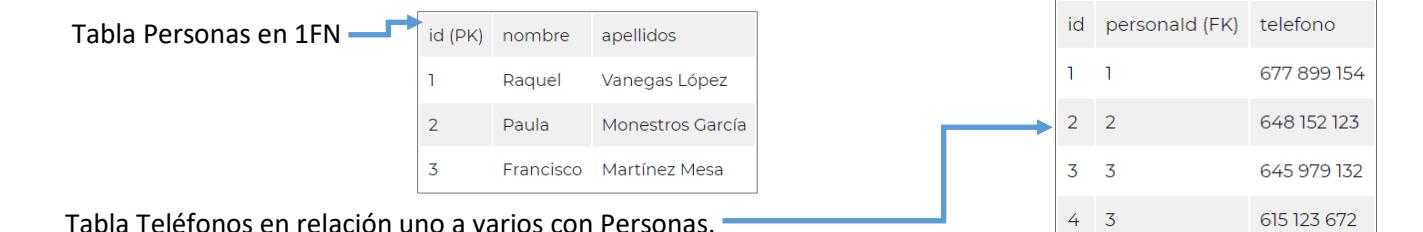
Una tabla está en Primera Forma Normal (1FN o FN1) si, y sólo si, todos los atributos de la misma contienen valores atómicos, es decir, no hay grupos repetitivos (elimina grupos repetitivos).

La 1FN especifica que **cada atributo en nuestras tablas debe ser atómico**. Significa que la información que contienen no puede ser descompuesta en elementos más pequeños (todos los atributos, valores almacenados en las columnas, deben ser indivisibles). Si fuera así, **tendríamos que separarlos, o en otras columnas, o ponerlos en una tabla separada**.

Por ejemplo, miremos a una tabla *Personas* siguiente:

Esta tabla no está en 1FN. El atributo teléfonos no es atómico. Si sólo permitimos 2 números de teléfono por persona, lo que podemos hacer es crear dos columnas, teléfono1 y teléfono2. Obviamente, lo mismo no vale para los apellidos, ya que no nos interesa acceder a ellos por separado.

Pero, ¿qué hacemos si la cantidad de números de teléfono no es limitada? Tendremos que crear otra tabla que contenga los números de teléfono con una relación uno a varios con la tabla Personas.



Otro ejemplo: supongamos que tienes en una tabla una columna Dirección para almacenar la dirección completa, dato que se compondría del nombre de la calle, el número, el piso, la puerta, el código postal, la población y la provincia.

Una tabla con esta estructura plantea problemas a la hora de recuperar información. Imagina que necesitas conocer todas las entradas correspondientes a una determinada población, o que quieres buscar a partir del código postal. Al ser la dirección completa una secuencia de caracteres de estructura libre no resultaría nada fácil.

Existirán más columnas, pero cada una de ellas contendrá un valor simple e indivisible que facilitará la realización de las operaciones antes mencionadas.

Además, se debe evitar la repetición de los datos de la población y provincia en cada una de las filas. Siempre que al muestrear la información de una tabla aparezcan datos repetidos, existe la posibilidad de crear una tabla independiente con ellos.

La mayoría de las veces no será necesario aplicar este proceso ya que nuestras tablas se encontrarán en 1FN si hemos hecho un buen diseño conceptual y lógico. La aplicación de la 1FN será la correcta elección de la clave principal.

11.2.2 2^a 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 (elimina las dependencias parciales de la clave primaria). Es obvio que **una tabla que esté en 1FN y cuya clave esté compuesta por un único atributo, estará en 2FN**.

La **segunda forma normal (2FN)** sólo tiene que ver con las tablas con claves primarias compuestas.

Para estar en 2FN, tenemos que asegurarnos primero de respetar la 1FN y respetar una regla más: ningún atributo de la tabla puede depender sólo de una parte de la clave primaria.

Miremos por ejemplo la tabla HabilidadEmpleados siguiente, donde la clave primaria es la combinación de los atributos empleado y habilidad.

El atributo dirección es dependiente de solo una parte de la clave compuesta, por lo cual tenemos redundancia en la columna dirección. Estamos repitiendo innecesariamente el valor de este atributo. Peor aún, si en algún momento actualizamos la dirección en una tupla de esta tabla, tendremos un problema de consistencia del valor. Esta tabla no respeta la 2FN.

Empleado (PK)	Habilidad (PK)	dirección
Jones	mecánico	131 calle bondad
Jones	carpintero	131 calle bondad
Marvin	mecánico	2 calle sol
Beckerd	limpieza	25 avenida de la borsa
Beckerd	construcción	25 avenida de la borsa

La solución consiste otra vez en sacar el atributo dirección de la tabla y colocarlo en otra tabla con relación uno a varios.

La nueva tabla Empleados

Empleado (PK, FK)	direccion
Jones	131 calle bondad
Marvin	2 calle sol
Beckerd	25 avenida de la bolsa

La tabla HabilidadEmpleados

empleado (PK)	habilidad (PK)
Jones	mecánico
Jones	carpintero
Marvin	mecánico
Beckerd	limpieza
Beckerd	construcción

Este diseño sí está en 2FN. Si queremos actualizar la dirección de un empleado, lo haremos a través de la tabla Empleados, sin riesgo de incoherencia en los datos.

¿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.

Cuando la clave primaria de una tabla se compone de un solo atributo automáticamente está en 2FN

Ejercicio resuelto

La siguiente relación no está en 2FN. ¿Qué relaciones resultan de pasarla a 2FN?

CONSULTA (codPaciente, codMédico, fechaConsulta, nombrePaciente, diagnóstico, tratamiento, especialidad)

Solución: Relaciones resultantes al pasar a 2FN

CONSULTA (codPaciente, codMédico, fechaConsulta, diagnóstico, tratamiento)

MEDICO (codMédico, especialidad)

PACIENTE (codPaciente, nombrePaciente)

11.2.3 3^a 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 (elimina las dependencias transitivas de la clave primaria).

La tercera forma normal 3FN también depende de las formas normales anteriores. Se parece a la 2FN, pero trata de los atributos que no forman parte de la clave.

Tomemos como ejemplo una BD dónde guardamos los tricks realizados por los concursantes en un concurso de skateboard.

Tomemos la siguiente tabla TricksParticipantes

id	nombre	trick	puntos
1	Maria	Olle	4
2	Rafael	Heelflip	5
3	Amalia	50-50 rail	5
4	Bert	Heelflip	5
5	Bert	BS 180	3
6	José	FS 360	6
7	José	Ollie	4

Para cumplir con la 3FN, debemos sacar la columna puntos y meterla en otra tabla, ya que cada trick vale un número determinado de puntos.

id (PK)	nombre	trick
1	Maria	Ollie
2	Rafael	Heelflip
3	Amalia	50-50 rail
4	Bert	Heelflip
5	Bert	BS 180
6	José	FS 360
7	José	Ollie

Tendremos pues la tabla TricksParticipantes en 3FN

id	trick (FK)	puntos
1	Ollie	4
2	Heelflip	2
3	50-50 rail	5
4	BS 180	3
5	FS 360	6

Y la tabla ValoresTricks con una clave ajena haciendo referencia a los tricks de la tabla TricksParticipantes.

Así nos aseguramos que cada trick tiene la misma puntuación como tiene que ser.

¿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 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.

11.2.4 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 (elimina dependencias sobre claves candidatas). **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.

Es **muy difícil que una tabla que está en 3FN no esté en FNBC**, pero podemos «lograrlo» eligiendo mal las claves de nuestras tablas. Por ejemplo, si tenemos una tabla con idTrabajador, idDepartamento, idResponsable, donde el idResponsable es la persona responsable del trabajador. La clave sería, si cada trabajador puede trabajar en varios departamentos y tener distintos responsables (idTrabajador, idDepartamento, idResponsable). Pero si resulta que cada responsable lo es de un único departamento, entonces idResponsable dependería de idDepartamento, lo que convierte a idResponsable en «determinante» (atributo que depende de otro atributo), pero no es clave candidata.

11.2.5 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. 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.

Ejercicio resuelto

Sea la siguiente tabla. Se pide normalizarla hasta FNBC.

COMPRAS (cod_compra, cod_prod, nomb_prod, fecha, cantidad, precio, fecha_rec, cod_prov, nomb_prov, tfno)

- **Comprobamos 1FN:**

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

- **Comprobamos 2FN:**

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

Veamos las dependencias:

$\text{cod_prod} \rightarrow \text{nomb_prod}$, y cod_prod es parte de la clave primaria.

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

COMPRA1 (cod_compra, cod_prod, fecha, cantidad, precio, fecha_rec, cod_prov, nomb_prov, tfno).

PRODUCTO (cod_prod, nomb_prod).

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

- **Comprobamos 3FN:**

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

Veamos las dependencias:

$\text{cod_prov} \rightarrow \text{nomb_prov}$

$\text{cod_prov} \rightarrow \text{tfno}$

(siendo cod_prov el código del proveedor y nomb_prov el nombre del proveedor)

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

COMPRA2 (cod_compra, cod_prod, fecha, cantidad, precio, fecha_rec, cod_prov)

PROVEEDOR (cod_prov, nomb_prov, tfno)

- **Comprobamos FNBC:**

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

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

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

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

PRODUCTO (cod_prod, nomb_prod)

COMPRA2 (cod_compra, cod_prod, fecha, cantidad, precio, fecha_rec, cod_prov)

PROVEEDOR (cod_prov, nomb_prov, tfno)

11.3 Desnormalización.

Cada vez que avanzamos en el nivel de normalización se necesitan más uniones entre las entidades y eso ralentiza la respuesta del sistema. Como la respuesta rápida a las demandas del usuario debe tenerse en cuenta a la hora del diseño de una BD, a veces es necesario desnormalizar alguna parte de la misma.

El proceso de denormalización consiste en crear redundancia a propósito en casos concretos, donde perdemos eficiencia en almacenamiento para ganar rapidez en las consultas.

Desnormalizar es transformar una BD en un nivel de normalización inferior. No obstante, es necesario tener en cuenta que a cambio de un desempeño más rápido deberemos soportar una mayor redundancia de datos.

Los diseñadores de una BD deben conciliar tres requerimientos a menudo incompatibles entre sí:

- ✓ Un diseño apropiado.
- ✓ Almacenamiento de la información, controlando las redundancias.
- ✓ Velocidad de procesamiento.

En ocasiones es necesario aceptar una cierta redundancia para que la BD cumpla mejor con el objetivo de mostrar la información adecuadamente.



TAREA

13. Dada la siguiente tabla, transformarla a 3FN.

idEmpleado	Nombre	idDepartamento	nombreDepartamento	añosDepartamento
1	Juan	6	Contabilidad	6
2	Pedro	3	Sistemas	3
2	Pedro	6	Contabilidad	5
3	Sonia	2	I+D	1
4	Verónica	3	Sistemas	10
4	Verónica	6	Contabilidad	2

14. Dada la siguiente tabla, normalizar hasta 3FN.

DNI	Nombre	Apellidos	Dirección	codPostal	Población	Provincia
413245-B	Juan	Ramos	Las Cañas, 59	19005	Guadalajara	Guadalajara
413245-B	Juan	Ramos	Pitón, 12	45589	Caleruela	Toledo
23456-J	Pedro	Pérez	Vitoria, 3	28804	Alcalá de Henares	Madrid
23456-J	Pedro	Pérez	El Altozano	10392	Berrocalejo	Cáceres
34561-B	María	Rodríguez	Sanz Vázquez, 2	19004	Guadalajara	Guadalajara
222346-J	Juan	Cabello	El ensanche, 3	28802	Alcalá de Henares	Madrid
222346-J	Juan	Cabello	Los abedules, 10	10300	Navalmoral de la Mata	Cáceres

15. Supongamos que tenemos la siguiente relación **EMPLEADOS** en la que representamos los datos de los empleados de una fábrica, normalizar a FNBC.

DNI	numSegSocial	Nombre	Apellidos	Departamento	Puesto	Salario
413245-B	28-1234566	Juan	Ramos	Compras	Gerente	2.300
23456-J	28-2345686	Pedro	Pérez	Nóminas	Auxiliar	1.200
123123-C	19-458766	María	Gil	Almacén	Conserje	1.530
123455-B	45-223344	Antonio	Sanz	Compras	Gestión	2.200

16. Normalizar la siguiente relación, utilizada para almacenar información sobre los artículos que un dependiente vende, además de información del propio dependiente.

DNI	Calle	Ciudad	Comunidad	Código	cantidad
413245-B	Bravo Murillo	Madrid	Madrid	1	10
413245-B	Bravo Murillo	Madrid	Madrid	2	3
123123-C	Bravo Murillo	Barcelona	Cataluña	1	4
123455-B	Goya	Sevilla	Andalucía	3	7