

TEMA 2: BASE DE DATOS RELACIONALES.

BASE DE DATOS RELACIONALES	1
1. ANÁLISIS Y DISEÑO DE BASES DE DATOS RELACIONALES.	1
2. ¿QUÉ ES EL MODELO E/R?	2
3. ENTIDADES.	2
3.1 TIPOS: FUERTES Y DÉBILES.	3
4. ATRIBUTOS.	4
4.1 TIPOS DE ATRIBUTOS.....	4
4.2 CLAVES.....	6
4.3 ATRIBUTOS DE UNA RELACIÓN.	7
5. RELACIONES.	8
5.1 GRADO DE UNA RELACIÓN.....	9
5.2 CARDINALIDAD DE RELACIONES.....	10
5.3 CARDINALIDAD DE ENTIDADES.	12
6. SIMBOLOGÍA DEL MODELO E/R.	13
7. EL MODELO E/R EXTENDIDO.	14
7.1 RESTRICCIONES EN LAS RELACIONES.	14
7.2 GENERALIZACIÓN Y ESPECIALIZACIÓN.	16
7.3 AGREGACIÓN.....	18
8. ELABORACIÓN DE DIAGRAMAS E/R.	20
8.1 IDENTIFICACIÓN DE ENTIDADES Y RELACIONES.....	20
8.2 IDENTIFICACIÓN DE ATRIBUTOS, CLAVES Y JERARQUÍAS.	22
8.3 METODOLOGÍAS.....	23
8.4 REDUNDANCIA EN DIAGRAMAS E/R.	24
8.5 PROPIEDADES DESEABLES DE UN DIAGRAMA E/R.....	25
9. TRANSFORMACIÓN DE DIAGRAMAS E/R AL MODELO RELACIONAL.	26
9.1 EL MODELO RELACIONAL.	27
9.2 TRANSFORMACIÓN DE ENTIDADES.....	28
9.3 TRANSFORMACIÓN DE RELACIONES.	30
9.4 TRANSFORMACIÓN DE ATRIBUTOS.....	34
10. EJERCICIOS DE DISEÑO CONCEPTUAL Y LÓGICO DE BASES DE DATOS.	35
11. NORMALIZACIÓN DE MODELOS RELACIONALES.....	42
11.1 TIPOS DE DEPENDENCIAS.....	43
11.2 FORMAS NORMALES.	44
12. ENLACES DE REFUERZO Y AMPLIACIÓN.	47

BASE DE DATOS RELACIONALES

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 base de datos, su labor. El objetivo fundamental de esta arquitectura es separar los programas de aplicación de la base de datos física, proponiendo tres niveles de abstracción: **nivel interno o físico, nivel lógico o conceptual y nivel externo o de visión del usuario**.

1. ANÁLISIS Y DISEÑO DE BASES DE DATOS RELACIONALES.

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

Cuando hemos de desarrollar una base de datos relacional se distinguen claramente dos fases de trabajo: **Análisis y Diseño**. En la siguiente tabla te describimos las etapas 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 🏗️ <u>retrodiseño</u> , 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 bases de datos. El hecho de saltarse el esquema conceptual conlleva un problema de pérdida de información respecto al problema real a solucionar. El esquema conceptual debe reflejar todos los aspectos relevantes del mundo real que se va a modelar.

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

2. ¿QUÉ ES EL MODELO E/R?

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

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

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

3. ENTIDADES.

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

Una **entidad** puede ser un objeto físico, un concepto o cualquier elemento que queramos modelar, que tenga importancia para la organización y del que se 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 base de datos de mapas de montaña, los elementos: camping, pista forestal, valle, río, pico, refugio, etc., son ejemplos de posibles entidades. A la hora de identificar las entidades, hemos de pensar en nombres que tengan especial importancia dentro del lenguaje propio de la organización o sistema que vaya a utilizar dicha base de datos. Pero no siempre una entidad puede ser concreta, como un camping 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 le 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**.

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 base de datos 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.

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.

Recomendación

- ✓ 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 epígrafes te los describimos claramente.

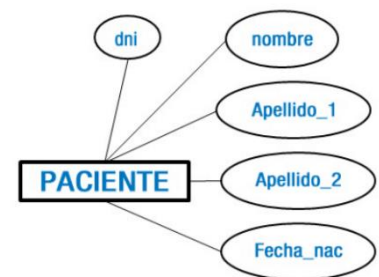
4. ATRIBUTOS.

¿Cómo guardamos información de cada entidad? A través de sus **atributos**. Las entidades se representan mediante un conjunto de **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.

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

4.1 TIPOS DE ATRIBUTOS.

Existen varias características que hacen que los atributos asociados a una entidad o relación sean diferentes, los clasificaremos según varios criterios.

- a. **Atributos obligatorios y opcionales.** **Atributo obligatorio:** es aquél que ha de estar siempre definido para una entidad o relación. Por ejemplo, para la entidad **JUGADOR** será necesario tener algún atributo que identifique cada ocurrencia de entidad, podría ser su DNI. Una clave o llave es un atributo obligatorio. **Atributo opcional:** es aquél que podría ser definido o no para la entidad. Es decir, puede haber ocurrencias de entidad para las que ese atributo no esté definido o no tenga valor.

TEMA 2: BASE DE DATOS RELACIONALES.

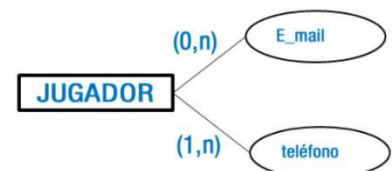
b. Atómicos o compuestos.

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



- 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 unívocamente** a la entidad. En otras palabras, no se permite que ningún par de entidades tengan exactamente los mismos valores de sus atributos. Teniendo en cuenta esto, presta atención a los siguientes conceptos:

Superclave (Superllave): Es cualquier conjunto de atributos que permite identificar de forma única a una ocurrencia de entidad. Una superclave puede tener atributos no obligatorios, es decir, que no identificarían por si solos una ocurrencia de entidad.

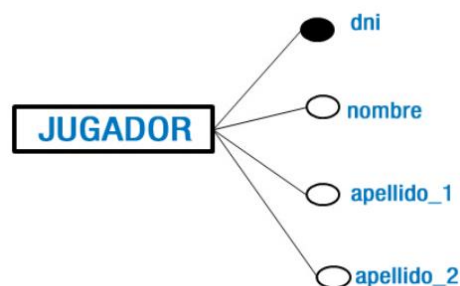
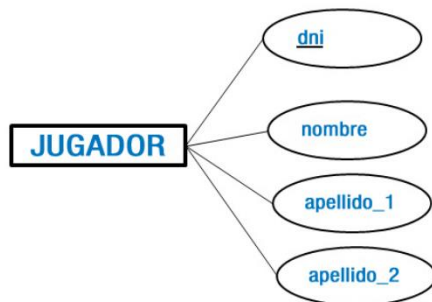
Clave candidata: Si de una superclave no es posible obtener ningún subconjunto que sea a su vez superclave, decimos que dicha superclave es clave candidata.

Clave primaria (Primary Key): También llamada llave primaria o clave principal. De todas las claves candidatas, el diseñador de la base de datos ha de escoger una, que se denominará clave principal o clave primaria. La clave primaria es un atributo o conjunto de ellos, que toman valores únicos y distintos para cada ocurrencia de entidad, identificándola unívocamente. No puede contener valores nulos.

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.



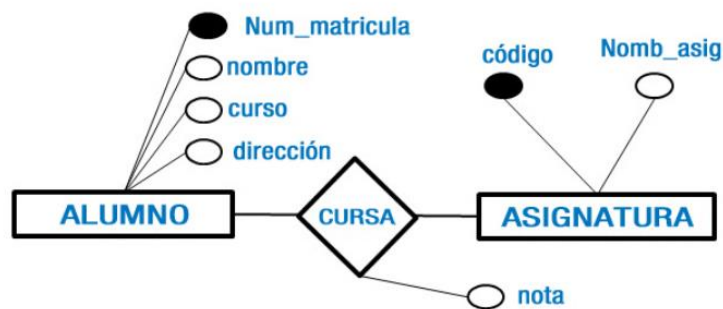
4.3 ATRIBUTOS DE UNA RELACIÓN.

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

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

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

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

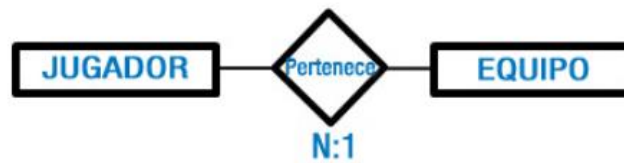


5. RELACIONES.

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

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

La representación gráfica en el modelo Entidad/Relación corresponde a un rombo en cuyo interior se encuentra inscrito el nombre de la relación. 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.



Recomendación

Cuando debas dar un nombre a una relación procura 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.

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.

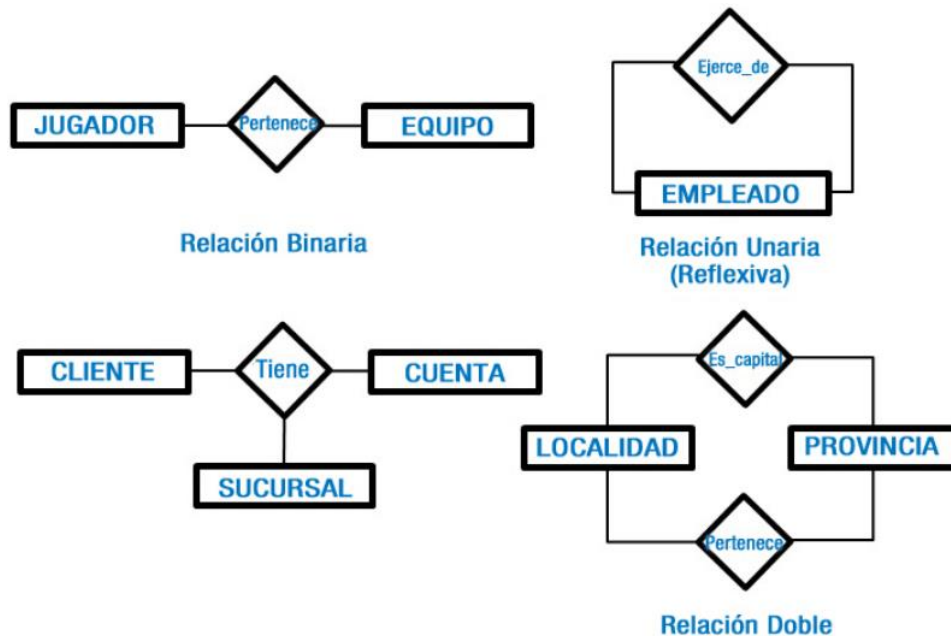
¿Pueden intervenir varias entidades en una misma relación? Claro que sí, en una relación puede intervenir una única entidad o varias.

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 base de datos buscará tener sólo este tipo de relaciones.
- ✓ **Relación Ternaria o de grado 3:** Es aquella relación en la que participan tres entidades al mismo tiempo.
- ✓ **Relación N-aria o de grado n:** Es aquella relación que involucra n entidades. Este tipo de relaciones no son usuales y deben ser simplificadas hacia relaciones de menor grado.
- ✓ **Relación doble:** ocurre cuando dos entidades están relacionadas a través de dos relaciones. Este tipo de relaciones son complejas de manejar.

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

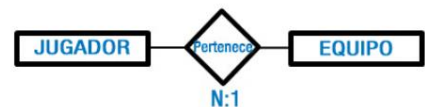


5.2 CARDINALIDAD DE RELACIONES.

¿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.

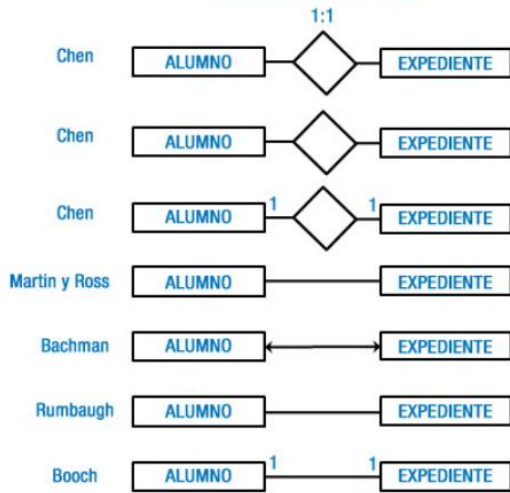
Veamos en detalle el significado de cada una de estas cardinalidades:

- ✓ **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.

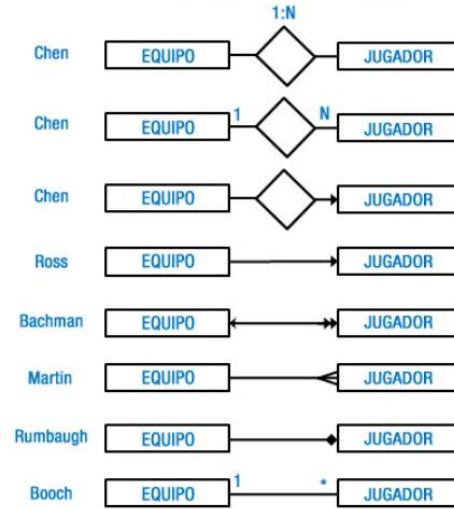
TEMA 2: BASE DE DATOS RELACIONALES.

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

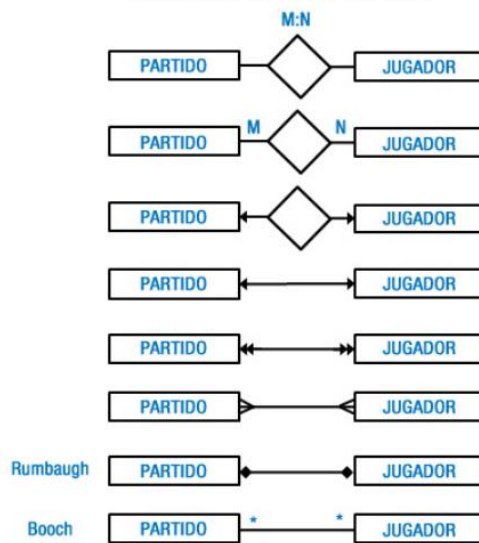
Relaciones uno a uno



Relaciones uno a muchos

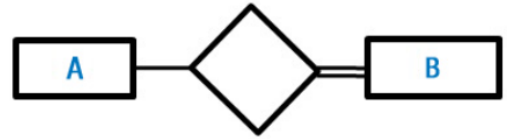


Relaciones muchos a muchos



5.3 CARDINALIDAD DE ENTIDADES.

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

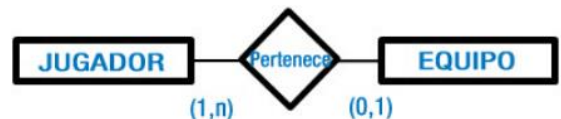


Sean las entidades A y B, la participación de la entidad A en una relación es **obligatoria (total)** si la existencia de cada una de sus ocurrencias necesita como mínimo de una ocurrencia de la entidad B (ver figura). En caso contrario, la participación es **opcional (parcial)**.

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

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

Veámoslo más claro a través del siguiente ejemplo: un **JUGADOR** pertenece como mínimo a ningún **EQUIPO** y como máximo a uno (0,1) y, por otra parte, a un **EQUIPO** pertenece como mínimo un **JUGADOR** y como máximo varios (1,n). Como puedes ver, la cardinalidad (0,1)



de **JUGADOR** se ha colocado junto a la entidad **EQUIPO** para representar que un jugador puede no pertenecer a ningún equipo o como máximo a uno. Para la cardinalidad de **EQUIPO** ocurre igual, se coloca su cardinalidad junto a la entidad **JUGADOR** para expresar que en un equipo hay mínimo un jugador y máximo varios.

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

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



6. SIMBOLOGÍA DEL MODELO E/R.

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

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

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

Notaciones del modelo Entidad/Relación		
	Atributo Derivado	
	Atributo opcional	
	Clave primaria	
	Clave alternativa	

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

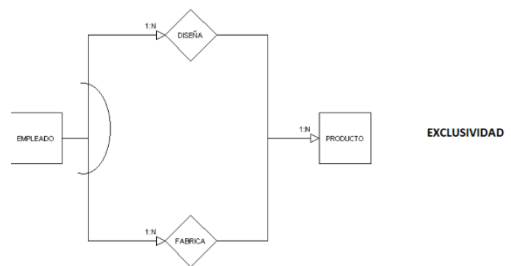
A continuación, se detallan estas nuevas características que convierten al modelo E/R tradicional en el **modelo Entidad/Relación Extendido**, como son: tipos de restricciones sobre las relaciones, especialización, generalización, conjuntos de entidades de nivel más alto y más bajo, herencia de atributos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

7.1 RESTRICCIONES EN LAS RELACIONES.

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

a. Restricción de exclusividad.

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



Por ejemplo, los empleados, en función de sus capacidades, o son diseñadores de productos o son operarios y los fabrican, no es posible que ningún empleado sea diseñador y fabricante, 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, un profesor no puede recibir e impartir el mismo curso, aunque al contrario que en la restricción anterior puede impartirlo o recibirlo.. Pero si un profesor imparte un determinado curso, no podrá estar recibéndolo 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

TEMA 2: BASE DE DATOS RELACIONALES.

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.**

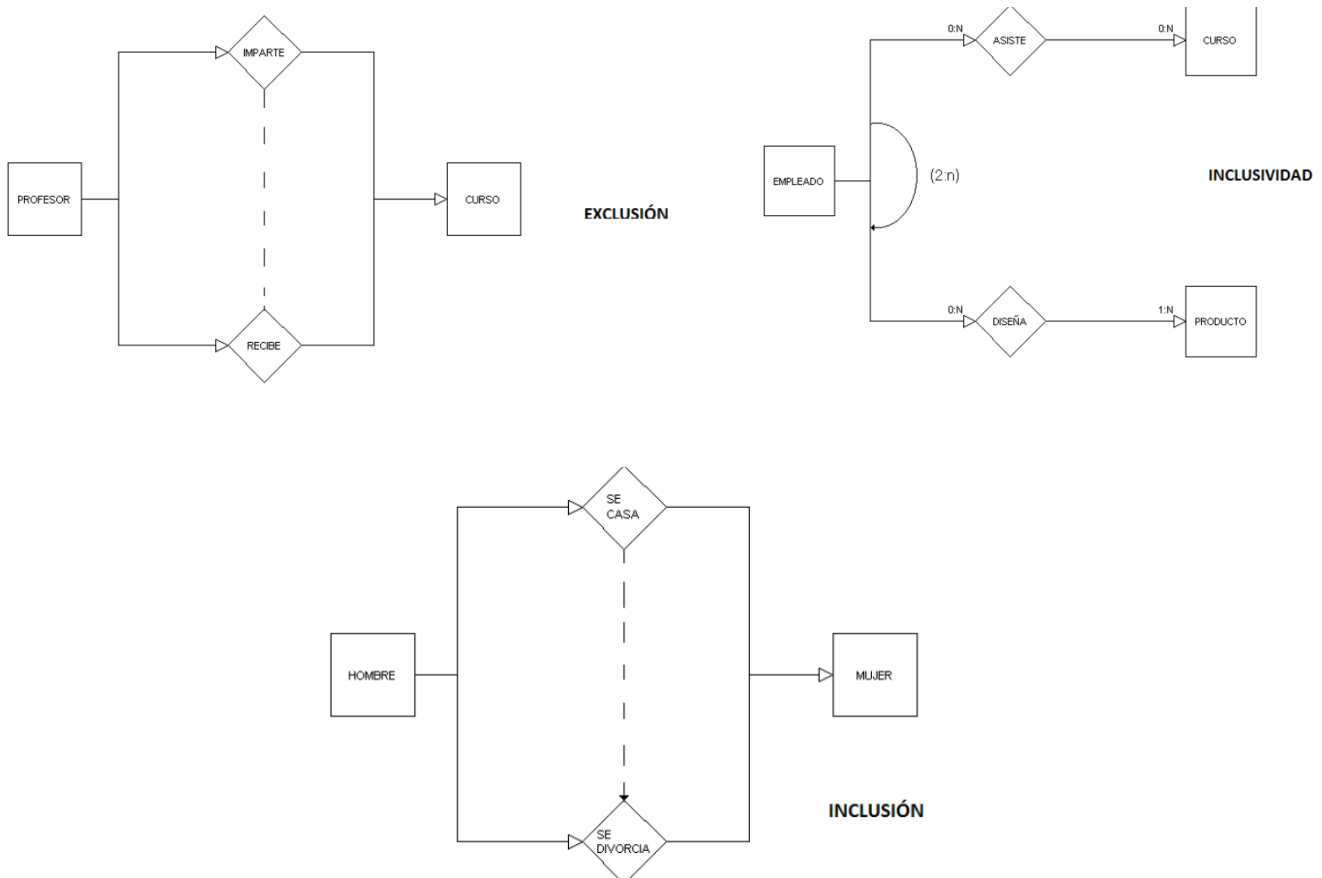
Por ejemplo, para que un empleado pueda trabajar como diseñador de productos deber haber asistido, al menos, a dos cursos.

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

d. Restricción de inclusión.

Se establece una restricción de inclusión cuando todo ejemplar de una entidad, **para participar en la asociación con otro elemento de otra entidad mediante una interrelación, es necesario que ambos elementos estén asociados por una segunda interrelación por naturaleza.**

Como se ve en el ejemplo, para que un hombre se divorcie de una mujer, previamente ha de haber otra interrelación, haberse casado con ella. Se representa mediante una **flecha discontinua entre las dos relaciones**, tal y como se muestra en el ejemplo.



7.2 GENERALIZACIÓN Y ESPECIALIZACIÓN.

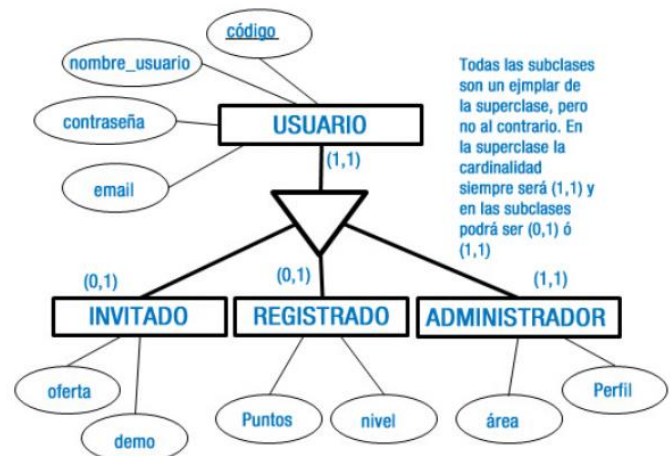
La segunda extensión incorporada en el modelo Entidad/Relación Extendido se centra en nuevos tipos de relaciones que van a permitir modelar la realidad de una manera más fiel. Estos nuevos tipos de relación reciben el nombre de **jerarquías** y se basan en los conceptos de generalización, especialización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

Cuando estamos diseñando una base de datos puede que nos encontremos con conjuntos de entidades que posean características comunes, lo que permitiría crear un tipo de entidad de nivel más alto que englobase 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.

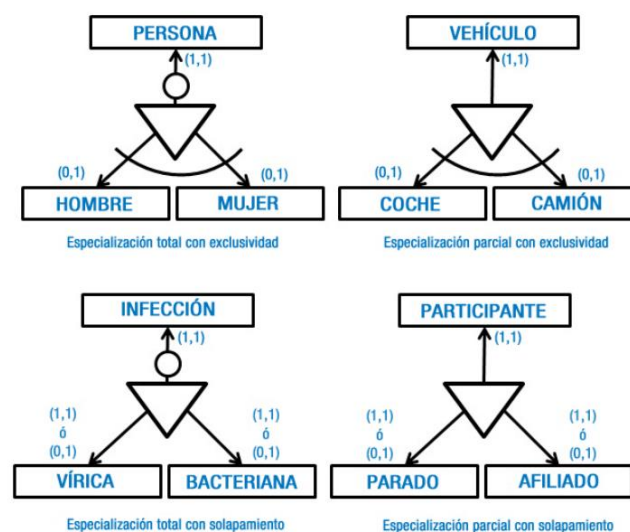
En el ejemplo de la imagen, las subclases **INVITADO**, **REGISTRADO** y **ADMINISTRADOR** constituyen subclases de la superclase **USUARIO**. Cada una de ellas aporta sus propias características y heredan las pertenecientes a su superclase.

TEMA 2: BASE DE DATOS RELACIONALES.

Una generalización/especialización podrá tener las siguientes restricciones semánticas:

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

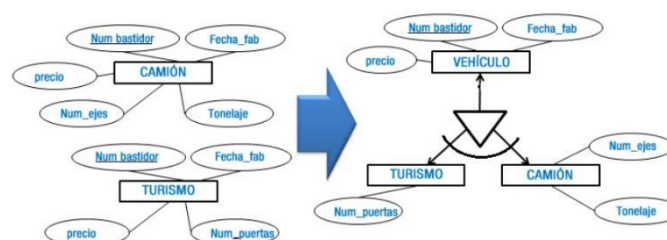
Las diferentes restricciones semánticas descritas tienen su representación gráfica, a través del gráfico que a continuación te mostramos podrás entender mejor su funcionamiento.



Ejercicio resuelto

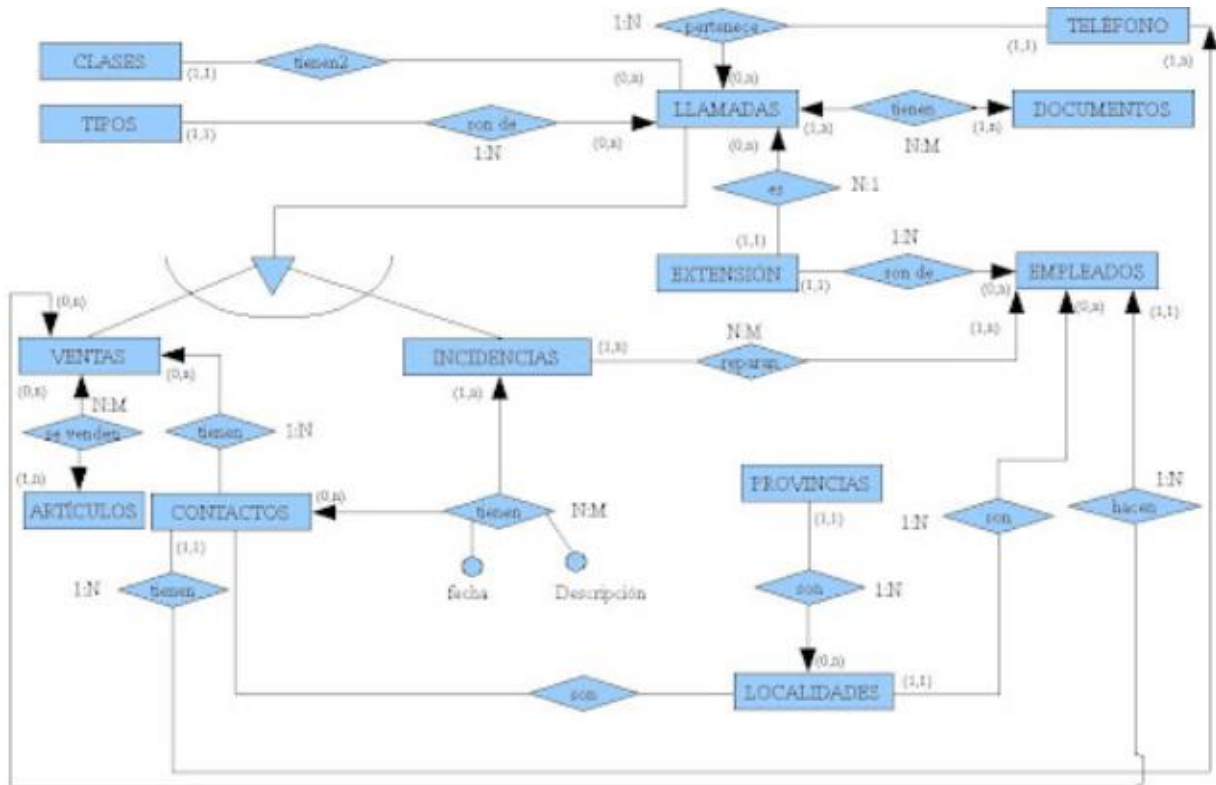
Supongamos la existencia de dos entidades **TURISMO** y **CAMION**. Los atributos de la entidad **TURISMO** son: **Num_bastidor**, **Fecha_fab**, **precio** y **Num_puertas**. Los atributos de la entidad **CAMION** son: **Num_bastidor**, **Fecha_fab**, **precio**, **Num_ejes** y **Tonelaje**.

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 **VEHICULO**. Las subclases **TURISMO** y **CAMION**, con sus atributos específicos, quedarán asociadas a la superclase **VEHICULO** mediante una jerarquía parcial con exclusividad. En el siguiente gráfico puedes apreciar la transformación.



TEMA 2: BASE DE DATOS RELACIONALES.

REPRESENTACIÓN DE JERARQUÍA EN UN ESQUEMA CONCEPTUAL.

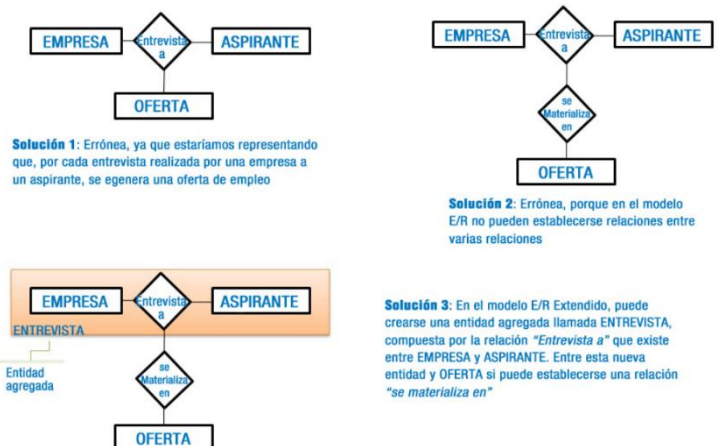


7.3 AGREGACIÓN.

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



TEMA 2: BASE DE DATOS RELACIONALES.

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.

En la siguiente figura puedes apreciar los tipos de agregación y su representación gráfica.



Para saber más

Con la agregación hemos terminado de detallar las extensiones más importantes del modelo Entidad/Relación Extendido. A lo largo de tu andadura por el mundo de las bases de datos y, en concreto, en todo lo relacionado con los esquemas conceptuales y diagramas Entidad/Relación, es probable que te encuentres con diferentes notaciones y simbologías. Algunas ya las hemos representado a lo largo de esta unidad y otras podrás encontrarlas en el enlace que te ofrecemos a continuación. Además, puedes utilizar la información que te proponemos para reforzar y ampliar todo lo visto.

[Modelo E/R Extendido.](#)

8. ELABORACIÓN DE DIAGRAMAS E/R.

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

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

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

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

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 base de datos 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:

- a. **Identificación de entidades:** Es un proceso bastante intuitivo. Para localizar aquellos elementos que serán las entidades de nuestro esquema, analizaremos la especificación de requerimientos en busca de nombres o sustantivos. Si estos nombres se refieren a objetos importantes dentro del problema probablemente serán entidades. 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.

TEMA 2: BASE DE DATOS RELACIONALES.

¿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.

- b. **Identificación de relaciones:** Localizadas las entidades, debemos establecer qué relación existe entre ellas. Para ello, analizaremos de nuevo el documento de especificación de requerimientos en busca de **verbos o expresiones verbales que conecten unas entidades con otras**. 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.

Reflexiona

En ocasiones, el identificador de una relación está compuesto por varias palabras, como por ejemplo: es supervisado, trabaja para, etc. Es recomendable que utilices 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 recursiva¹, reflexiva o unaria, hemos de representar en nuestro esquema los roles desempeñados por la entidad en dicha relación.

¹ Que puede repetirse indefinidamente. Un programa o función que se llama a sí mismo. La recursividad permite definir un objeto (problemas, estructuras de datos) en términos de sí mismo. Casos típicos de estructuras de datos definidas de manera recursiva son los árboles y las listas ligadas. Algunos ejemplos de problemas que se definen recursivamente son el factorial de un número, la serie de Fibonacci, etc

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 unívoca 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:

- a. **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.
- b. **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.
- c. **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.
- d. **Metodología Mixta:** Es empleada en problemas complejos. Se dividen los requerimientos en subconjuntos que serán analizados independientemente. Se crea un esquema que servirá como estructura en la que irán interconectando los conceptos importantes con el resultado del análisis de los subconjuntos creados. Esta metodología utiliza las técnicas ascendente y descendente. Se aplicará la técnica descendente para dividir los requerimientos y en cada subconjunto de ellos, se aplicará la técnica ascendente.

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

8.4 REDUNDANCIA EN DIAGRAMAS E/R.

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

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

La redundancia de datos puede provocar problemas como:

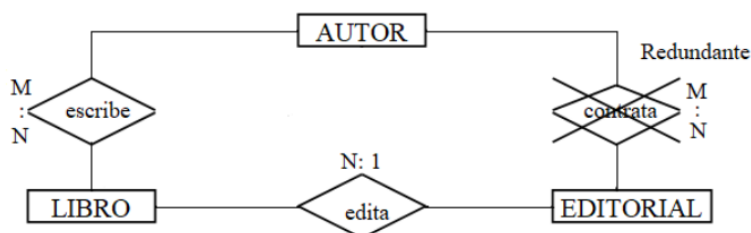
- **Aumento de la carga de trabajo:** al estar almacenado un dato en varios lugares, las operaciones de grabación o actualización de datos necesitan realizarse en varias ocasiones.
- **Gasto extra de espacio de almacenamiento:** al estar repetidos, los datos ocupan mayor cantidad de espacio en el medio 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 base de datos funcione óptimamente, hay que empezar realizando un buen diseño de ella. Es imprescindible que nuestros diagramas E/R controlen la redundancia y, para ello, debemos analizar el esquema y valorar qué elementos pueden estar incorporando redundancia a nuestra solución.

¿Dónde buscamos indicios de redundancia en nuestros esquemas? 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 o cíclica a través de varias relaciones, es lo que se conoce como **un ciclo**. En caso de existir un ciclo, deberemos tener en cuenta las siguientes condiciones, antes de poder eliminar dicha relación redundante:
 - ✓ Que el significado de las relaciones que componen el ciclo sea el mismo.
 - ✓ Que si eliminamos la relación redundante, el significado del resto de relaciones es el mismo.
 - ✓ Que si la relación eliminada tenía atributos asociados, éstos puedan ser asignados a alguna entidad participante en el esquema, sin que se pierda su significado.

La siguiente figura representa un ciclo entre las entidades AUTOR, LIBRO y EDITORIAL.



TEMA 2: BASE DE DATOS RELACIONALES.

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

Como norma general, puede decirse que **un ciclo no siempre implica la existencia de relaciones redundantes**. En este caso deberán estudiarse con detenimiento y analizar si al prescindir de una interrelación es siempre posible relacionar de forma indirecta las dos entidades unidas por la interrelación considerada redundante, comprobando igualmente que no se pierden atributos. Pero hay que tener en cuenta que **no siempre que exista un ciclo estaremos ante una 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 base de datos. Por ejemplo, si el método de cálculo del valor de un determinado atributo derivado es complejo (varias operaciones matemáticas o de cadenas de caracteres, varios atributos implicados, etc.) y ralentiza el funcionamiento de la base de datos, 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:

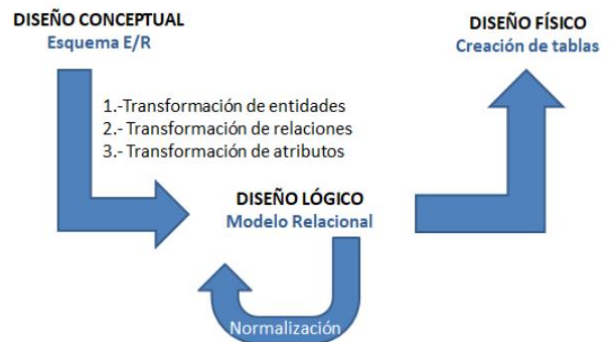
- ✓ **Compleitud:** Un diagrama E/R será completo si es posible verificar que cada uno de los requerimientos está representado en dicho diagrama y viceversa, cada representación del diagrama tiene su equivalente en los requerimientos.
- ✓ **Corrección:** Un diagrama E/R será correcto si emplea de manera adecuada todos los elementos del modelo Entidad/Relación. La corrección de un diagrama puede analizarse desde dos vertientes:
 - Corrección sintáctica: Se producirá cuando no se produzcan representaciones erróneas en el diagrama.
 - Corrección semántica: Se producirá cuando las representaciones signifiquen exactamente lo que está estipulado en los requerimientos. Posibles errores semánticos serían: la utilización de un atributo en lugar de una entidad, el uso de una entidad en lugar de una relación, utilizar el mismo identificador para dos entidades o dos relaciones, indicar erróneamente alguna cardinalidad u omitirla, etc.
- ✓ **Minimalidad:** Un diagrama E/R será mínimo si se puede verificar que al eliminar algún concepto presente en el diagrama, se pierde información. Si un diagrama es redundante, no será mínimo.

- ✓ **Sencillez:** Un diagrama E/R será sencillo si representa los requerimientos de manera fácil de comprender, sin artificios complejos.
- ✓ **Legibilidad:** Un diagrama E/R será legible si puede interpretarse fácilmente. La legibilidad de un diagrama dependerá en gran medida del modo en que se disponen los diferentes elementos e interconexiones. Esta propiedad tiene mucho que ver con aspectos estéticos del diagrama.
- ✓ **Escalabilidad:** Un diagrama E/R será escalable si es capaz de incorporar posibles cambios derivados de nuevos requerimientos.

9. TRANSFORMACIÓN DE DIAGRAMAS E/R AL MODELO RELACIONAL.

La fase de **diseño conceptual** desarrollada, y que se materializa en el diagrama E/R, permite una gran independencia respecto al **diseño físico**, es decir de las cuestiones relativas a la implementación física de la base de datos.

Pero entre el diseño conceptual y físico se interpone el **diseño lógico**. Una vez comprobado que nuestro esquema E/R cumple todos y cada uno de los requerimientos del problema a modelar, es hora de transformarlo al esquema lógico basado en un modelo de base de datos concreto, en nuestro caso al **modelo de datos relacional**, que es en el que se basa nuestro SGDB con el que implementaremos finalmente las tablas.



La transformación del diagrama E/R al modelo de datos Relacional debe abordarse en tres fases:

1.- Transformación de las entidades. Con carácter general cada entidad se transforma en una tabla, pero deberemos prestar especial atención a los casos de dependencias y jerarquías entre entidades.

2.-Transformación de las relaciones. Básicamente hay dos posibilidades: crear una nueva tabla con la relación, o exportar la clave primaria de una tabla a la otra tabla donde será clave ajena, lo permitirá relacionar ambas tablas.

3.-Transformación de los atributos. Generalmente los atributos de una entidad se convierten en columnas de la tabla resultante, y los atributos de las relaciones o formarán parte de la nueva tabla, o bien se exportarán junto a la clave, según se haya transformado su relación.

Es de resaltar que el diseño conceptual es relativamente subjetivo, y una vez transformado al diseño lógico según el modelo relacional, se hace necesario llevar a cabo un proceso que permitirá testar, y modificar en su caso, el esquema lógico obtenido de la transformación. Este proceso recibe el nombre de **normalización**, que consiste en un conjunto de técnicas que permiten validar esquemas lógicos basados en el modelo relacional. Una vez normalizado, el esquema lógico relacional podrá ser implementado en cualquier SGDB relacional.

9.1 EL MODELO RELACIONAL.

Antes de acometer la transformación del esquema conceptual (diagrama E/R) al esquema lógico, es necesario conocer mínimamente el modelo de datos en que se basará este último: el relacional.

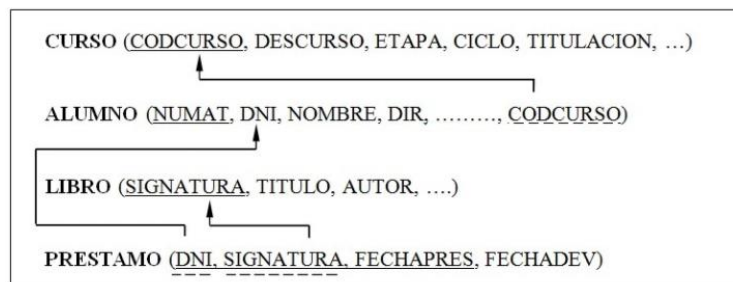
El modelo de datos relacional.

En este modelo los datos se estructuran lógicamente en forma de relaciones o tablas compuestas de atributos (columnas) y tuplas (filas). En las tablas no puede haber filas idénticas y el orden de sus filas y de sus columnas es irrelevante.

Aunque en la próxima unidad lo estudiaremos con detalle, hay que aclarar algunos conceptos necesarios para realizar la transformación del diagrama E/R a este modelo de datos.

- **Claves.**
 - **Clave candidata:** atributo o conjunto de atributos que identifican unívoca y mínimamente a cada una de las tuplas de la relación. Por la propia naturaleza de la tabla, siempre hay al menos una clave candidata formada por el conjunto de todos los atributos, que será la **Superclave**. Una tabla puede tener más de una clave candidata.
 - **Clave principal:** clave candidata elegida para identificar cada una de las tuplas. Una tabla solo puede tener una clave principal.
 - **Clave alternativa:** clave candidata no elegida como principal.
 - **Clave ajena:** clave candidata exportada a otra tabla, lo que permite relacionar una o varias tuplas de la tabla receptora con una tupla de la tabla exportadora.
- **Representación.**

En el esquema lógico del modelo Relacional, cada tabla se representa con su nombre y a continuación, y entre paréntesis, la lista de sus atributos. La clave principal de cada tabla se representa con subrayado continuo, la clave o claves ajenas con subrayado discontinuo y las claves alternativas no se representan.



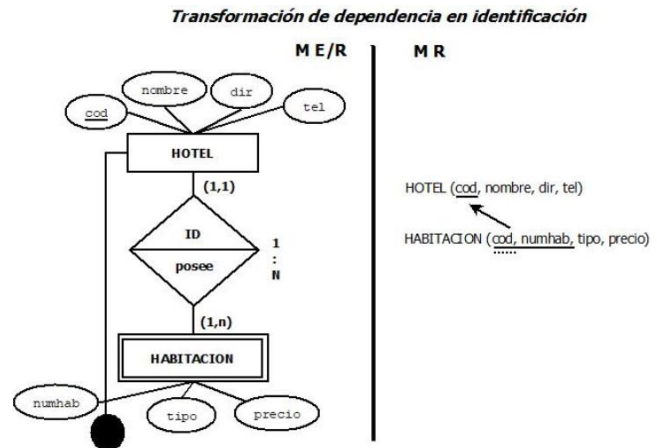
Donde en cada tabla se indica su clave primaria, que en el caso de PRESTAMO es compuesta de tres atributos. El atributo *COODCURSO* de la tabla ALUMNO es clave ajena que referencia a la tabla CURSO por su clave primaria, y los atributos *SIGNATURA* y *DNI* de la tabla PRESTAMO, además de formar parte de la clave primaria, son claves ajenas. *SIGNATURA* referencia a la tabla LIBRO por su clave primaria, y *DNI* referencia a la tabla ALUMNO por una clave alternativa.

Las flechas representan las referencias que las claves ajenas hacen hacia las claves candidatas, normalmente claves primarias y, aunque no forman parte de la simbología del esquema lógico, facilitan su comprensión.

9.2 TRANSFORMACIÓN DE ENTIDADES.

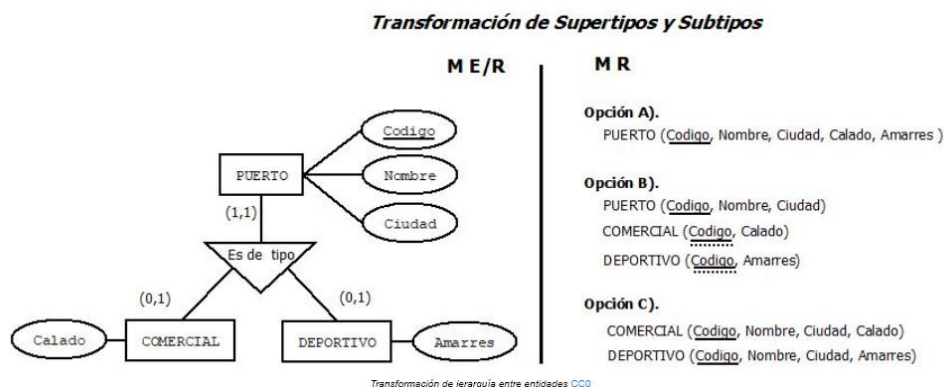
La norma general es que cada entidad se transforma en una tabla que se nombrará igual que la entidad de donde proviene. No obstante es conveniente prestar atención a los siguientes casos especiales:

- a. **Dependencia en identificación.**- En el caso de que una entidad débil dependa en identificación de una entidad fuerte, la clave primaria de la entidad fuerte se exportará a la tabla resultante de transformar la entidad débil, donde formará parte de su clave primaria, siendo a su vez clave ajena que referenciará a la tabla resultante de transformar la entidad fuerte.



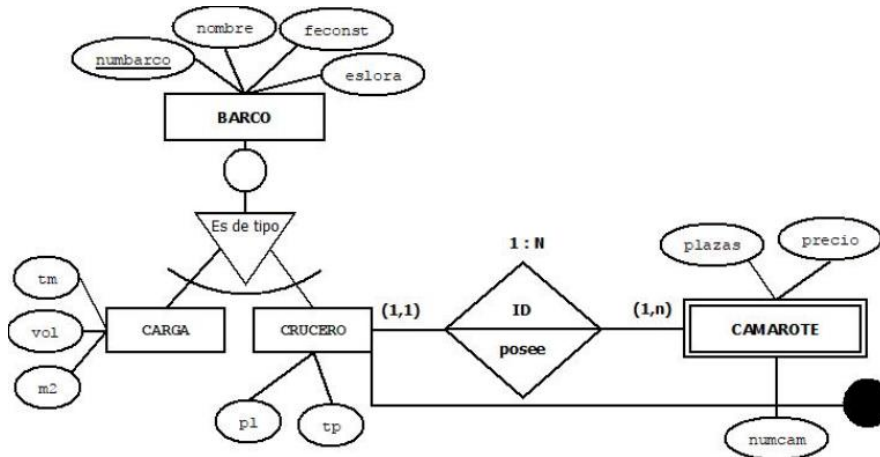
- b. **Generalización y especialización.**- En el caso de que el diagrama E/R presente jerarquías entre entidades, existen distintas posibilidades para transformar el diagrama E/R al modelo Relacional:

- ✓ Crear una única tabla para el supertipo y sus subtipos que recogería todos los atributos. Si los subtipos no tienen atributos diferenciadores habrá que añadir un atributo que indique el subtipo al que se refiere. Esta solución permite una mayor simplicidad, aunque puede provocar valores nulos en los atributos propios de cada subtipo, por lo que suele escogerse cuando los subtipos se diferencien en muy pocos atributos y las relaciones sean las mismas para todos los subtipos o no existen.
- ✓ Crear una tabla para el supertipo y una para cada uno de los subtipos, cada una con sus correspondientes atributos. Esta solución es aconsejable cuando los subtipos tienen atributos distintos y/o sus relaciones con otras entidades sean diferentes. Permite mantener agrupados en una tabla los atributos comunes.
- ✓ Crear una tabla para cada subtipo que, además de sus atributos específicos, contendrá los atributos comunes del supertipo. Suele aplicarse para jerarquías totales con exclusividad y se presentan las mismas condiciones que en el caso anterior: subtipos con atributos dispares y relaciones diferentes con otras entidades.



Ejercicio resuelto

A una naviera que solo posee dos tipos de barcos: de carga y crucero de pasajeros, le hemos diseñado el siguiente esquema conceptual según el modelo E/R. Se nos pide que lo transformemos al esquema lógico según el modelo relacional:



El esquema lógico constaría de las siguientes tablas:

CARGA (numbarco, nombre, feconst, eslora, tm, vol, m2)
CRUCERO (numbarco, nombre, feconst, eslora, p1, tp)
CAMAROTE (numbarco, numcam, plazas, precio)

Para transformar la jerarquía, hemos optado por la opción C), es decir, por crear una tabla para cada subtipo, ya que la jerarquía es total y exclusiva, los subtipos tienen atributos distintos y presentan distintas relaciones con otras entidades.

La dependencia en identificación indica que la clave disponible en la entidad débil es insuficiente para identificar a cada una de sus ocurrencias y debe apoyarse en la clave primaria de la entidad fuerte, por lo que su transformación se realiza exportando la clave primaria de la entidad fuerte a la débil, donde será clave ajena.

No obstante para transformar la jerarquía también se podría haber optado por la opción B), creando una tabla para el supertipo y una por cada uno de los subtipos, con lo que se mantendríamos agrupados en la primera todos los atributos comunes:

BARCO (numbarco, nombre, feconst, eslora)
CARGA (numbarco, tm, vol, m2)
PASAJEROS (numbarco, p1, tp)
CAMAROTE (numbarco, numcam, plazas, precio)

9.3 TRANSFORMACIÓN DE RELACIONES.

La forma en que se transformará una relación del diagrama E/R al modelo Relacional dependerá de su cardinalidad:

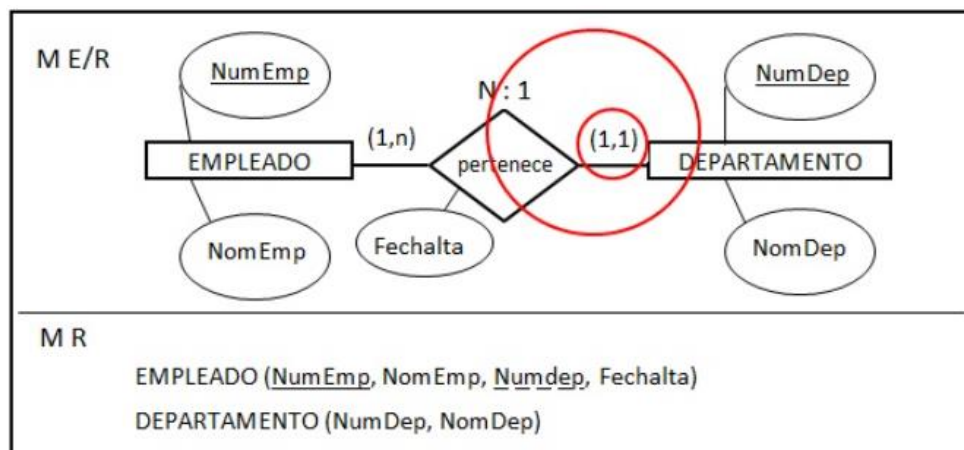
- ✓ **Cardinalidad N:M.-** Una relación con **cardinalidad N:M** siempre se transformará en una tabla nueva que tendrá como clave primaria compuesta la concatenación de las claves primarias de las entidades que relaciona, cada una de las cuales, además de ser parte de la clave primaria de la nueva tabla, será clave ajena que referenciará a la tabla de procedencia (a la tabla donde este atributo es clave primaria).

Si la relación contiene atributos, éstos pasarán a formar parte de la nueva tabla. En este caso es importante que comprobemos que la clave primaria compuesta de la nueva tabla es suficiente, siendo habitual que deba completarse con algún atributo procedente de la relación, especialmente si es de tipo fecha.

- ✓ **Cardinalidad 1:N o N:1.-** Para transformar una relación de uno a muchos o de muchos a uno hay dos soluciones:
 - a. Propagar la clave primaria de la entidad con cardinalidad 1 a la tabla resultante de transformar la entidad con cardinalidad N donde será una clave ajena que referenciará a la tabla donde es clave primaria.
 - b. Transformar la relación en una nueva tabla, como si se tratara de una relación N:M.

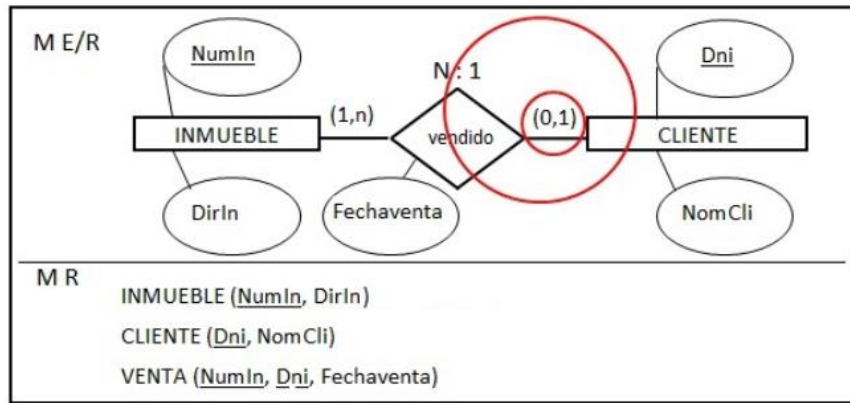
Para decidir sobre una u otra solución nos fijaremos en la cardinalidad de la entidad que tiene cardinalidad máxima 1:

Si esa cardinalidad es (1,1) optaremos por propagar la clave desde la entidad con cardinalidad máxima 1 a la entidad con cardinalidad máxima N, donde será clave ajena. Los posibles atributos de la relación se pasarán a la entidad con tipo N.



Si esa cardinalidad es (0,1) optaremos por crear una nueva tabla que tendrá como clave primaria la clave primaria de la entidad con cardinalidad N, y como clave ajena la clave primaria de la entidad con cardinalidad 1. La nueva tabla acogerá los posibles atributos de la relación.

TEMA 2: BASE DE DATOS RELACIONALES.



Nótese que si en este caso optásemos por propagar la clave nos encontraríamos con que todos los inmuebles no vendidos tendrían nulo el valor del DNI.

Resumen del paso al modelo relacional de la relación UNO a MUCHOS

Cardinalidades de entidad	¿Cómo pasamos al modelo relacional?
(1,n) y (0,1)	CREAR nueva tabla con las claves primarias de ambas entidades y los propios atributos de la relación. La clave primaria de esta nueva tabla será la que corresponde a la entidad de cardinalidad N. Las claves ajenas serán los atributos que hemos propagado de ambas entidades y referenciarán a ellas respectivamente.
(1,n) y (1,1)	PROPAGAR la clave primaria desde la entidad con cardinalidad máxima 1 hacia la entidad con cardinalidad máxima N siendo además clave ajena que referenciará a su tabla de procedencia. Los atributos de la relación también se propagarán a la entidad con cardinalidad N.

Después de observar los dos posibles casos, en la mayoría de los casos cuando nos encontramos con una relación UNO a MUCHOS en los que se podría emplear la creación de una nueva tabla, también se utiliza la PROPAGACIÓN de clave de la entidad con cardinalidad máxima 1 hacia la entidad con cardinalidad máxima N. La ventaja de crear una nueva tabla tan sólo es para evitar el almacenamiento de valores NULL.

✓ **Cardinalidad 1:1.-** La transformación de relaciones con cardinalidad de uno a uno presenta dos posibilidades:

- Crear una nueva tabla.
- Propagar la clave primaria de una de las entidades, como clave ajena, a la tabla resultante de la otra entidad.

Para elegir una u otra opción debemos prestar atención a las cardinalidades de las dos entidades relacionadas:

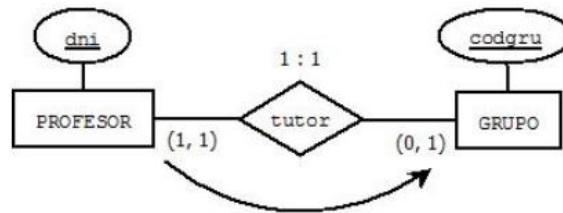
La opción A será recomendable si las cardinalidades de las entidades son (0,1) y (0,1). En este caso crearemos una nueva tabla, con el mismo tratamiento visto para el caso de M:N, pues en ambas entidades habrá ocurrencias no relacionadas con las de la otra entidad.

TEMA 2: BASE DE DATOS RELACIONALES.

La opción B se elegirá si al menos una cardinalidad de entidad es (1,1). Podemos tener dos casos:

1. Si las cardinalidades de las entidades son (0,1) y (1,1) entonces la propagación se hará desde la entidad con cardinalidad (1,1) hacia la que tiene cardinalidad (0,1).
2. Si las cardinalidades de las entidades son (1,1) y (1,1) entonces podemos elegir propagar la clave primaria de una de las tablas hacia la otra o viceversa.

Supongamos el siguiente esquema E/R:



Un análisis detallado de las cardinalidades de las entidades indica que todo grupo tiene un profesor tutor, pero no todo profesor es tutor de un grupo. Lo correcto será exportar la clave primaria de PROFESOR a GRUPO, donde será clave ajena, evitando con ello los valores nulos que se producirían en caso contrario.

Resumen del paso al modelo relacional de la relación UNO a UNO

Cardinalidades de entidad	¿Cómo pasamos al modelo relacional?
(0,1) y (0,1)	CREAR nueva tabla poniendo como clave primaria una de las claves primarias de alguna de las entidades que relaciona.
(1,1) y (1,1)	PROPAGAR la clave primaria desde una entidad cualquiera hacia la otra convirtiéndose en clave ajena.
(0,1) y (1,1)	PROPAGAR la clave primaria desde la entidad con cardinalidad (1,1) hacia la entidad con cardinalidad (0,1). Siendo allí clave ajena.

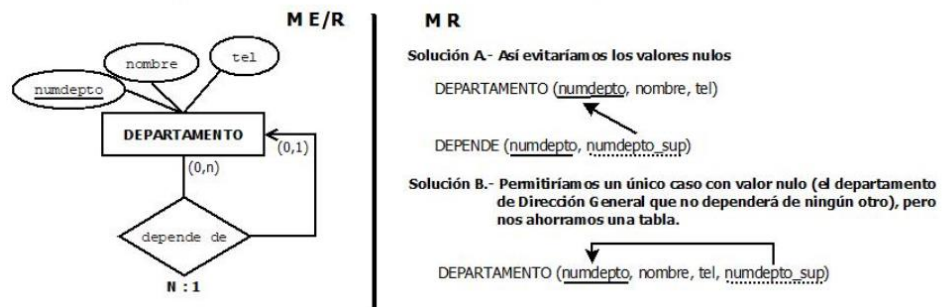
Si una cardinalidad de entidad es (1,1) siempre vamos a propagar la clave y no crearemos ninguna tabla nueva. Tendremos que fijarnos en la cardinalidad de la otra entidad para saber si podemos propagar la clave de la entidad que decidamos nosotros como diseñadores o bien debemos propagar la clave de una de ellas hacia la otra.

✓ Relaciones no binarias, con grado distinto a 2.

- Las relaciones de grado 1 o reflexivas se transforman siguiendo las pautas indicadas anteriormente, con la salvedad de que en vez de dos entidades distintas se trata de la misma entidad.

TEMA 2: BASE DE DATOS RELACIONALES.

Transformación de relación de grado 1 o reflexiva

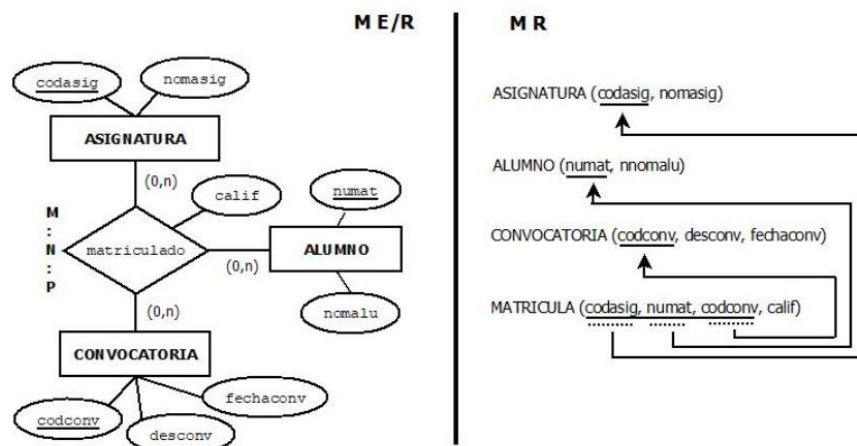


- Las relaciones de grado 3 o superior es recomendable descomponerlas en varias de grado 2, pero no siempre es posible sin que se pierda semántica asociada a la situación real. En este caso suelen transformarse creando una nueva tabla que contendrá como claves ajenas las tres claves primarias de las entidades que relaciona. La elección de la clave primaria dependerá de la cardinalidad:

Cómo elegir la clave primaria en las relaciones ternarias

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

Transformación de relación de grado 3 o ternaria



- ✓ **Relaciones con restricciones de exclusividad, inclusividad, exclusión e inclusión:** Al transformar el diagrama E/R al modelo Relacional no se tienen en cuenta las posibles restricciones de las relaciones. La implementación de esas restricciones se realizará con disparadores que se estudiarán más adelante.

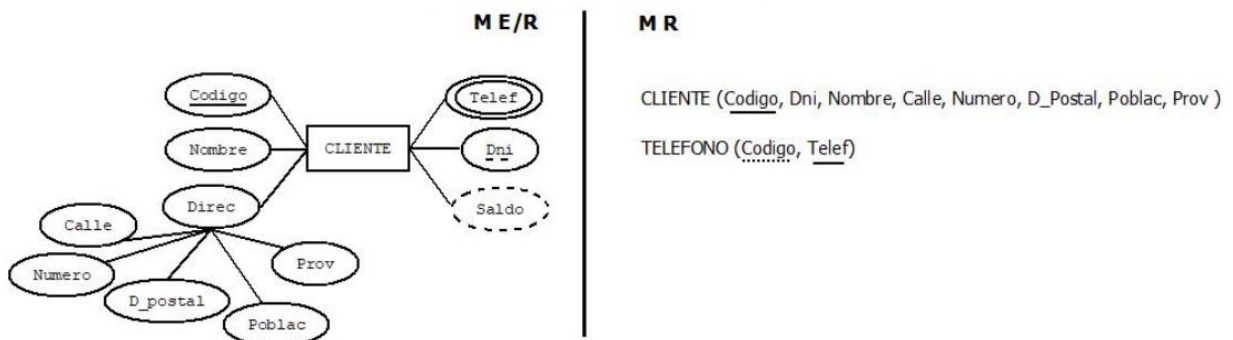
9.4 TRANSFORMACIÓN DE ATRIBUTOS.

✓ Atributos de las entidades.

Como norma general cada atributo de una entidad se transforma en una columna de la tabla a la que ha dado lugar la entidad. Distinguiremos los distintos tipos de atributos:

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

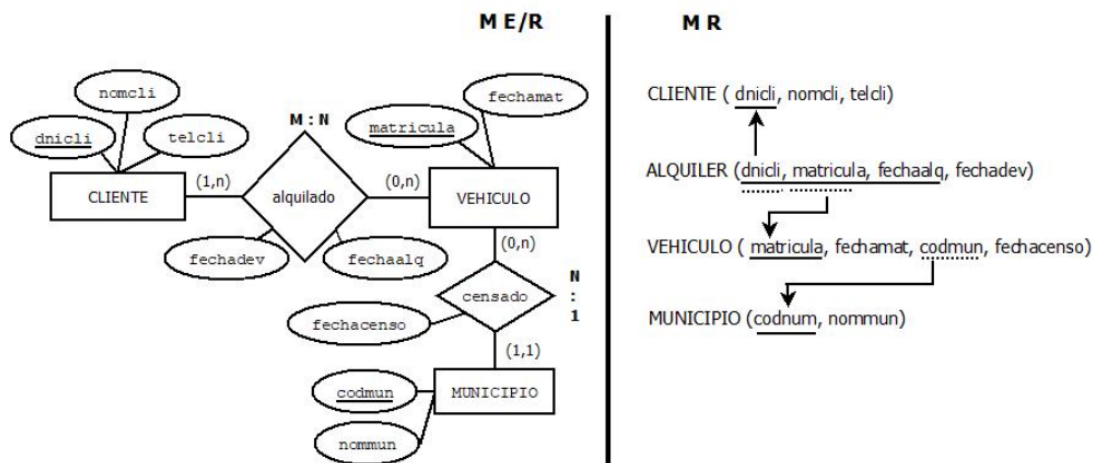
Transformación de Atributos de entidades



✓ Atributos de las relaciones:

Si una relación se transforma en tabla, todos sus atributos pasan a ser columnas de la tabla. Si por el contrario se transforma mediante propagación de clave, sus atributos migran, junto a la clave a la tabla que corresponda.

Transformación de atributos de relaciones



10. EJERCICIOS DE DISEÑO CONCEPTUAL Y LÓGICO DE BASES DE DATOS.

Antes de ponerse manos a la obra puedes ver un ejemplo más para elaborar un diagrama Entidad-Relación y representarlo con la [herramienta DIA](#). Además, también se realizará el paso al modelo relacional correspondiente siguiendo las reglas explicadas en el apartado anterior. Lo primero que hay que hacer es identificar los elementos del diagrama ER.

DESCRIPCIÓN EER: FASHION WEEK ALMERIA

La empresa que gestiona el evento de la semana de la moda ALMERIA nos ha pedido que diseñemos la base de datos con los siguientes requerimientos.

Los diseñadores que participan en dicho evento tendrán que registrarse con su DNI, nombre, apellidos y teléfono de contacto. Además se desea guardar datos sobre los diseños que cada uno de los diseñadores van a presentar. De esos diseños se desea guardar la referencia que es única, el nombre, una breve descripción y la fecha de creación. Un mismo diseñador podrá presentar varios diseños y cada uno de los diseños es único de cada diseñador.

Durante esa semana se realizarán varios desfiles de los cuales queremos saber el número identificador del desfile, la fecha y hora de comienzo y duración. Cada desfile está reservado para un único diseñador y un diseñador sólo podrá realizar un desfile durante esa semana.

En cada desfile podrán participar varios modelos de los que interesa saber el número personal que lo identifica, el nombre, la estatura, el peso y talla de pie.

En el siguiente videotutorial se hace un resumen de las entidades y relaciones identificadas junto a sus atributos. Se determina el atributo principal de cada entidad y por último se realiza el estudio de las cardinalidades de entidades y relaciones.

diseños y cada uno de los diseños es único de cada diseñador.

Durante esa semana se realizarán varios **desfiles** de los cuales queremos saber el **número identificador del desfile, la fecha y hora de comienzo y duración**. Cada desfile está **reservado** para un único diseñador y un diseñador sólo podrá realizar un desfile durante esa semana.

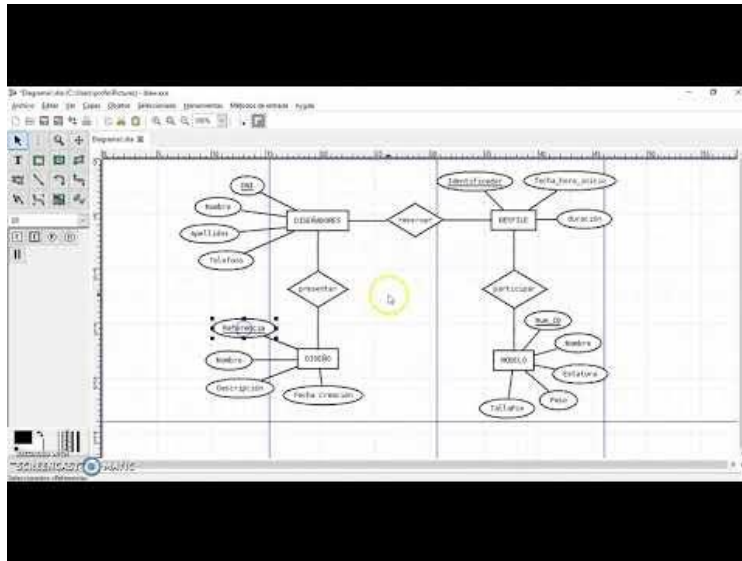
En cada desfile podrán **participar** varios **modelos** de los que interesa saber el **número personal que lo identifica, el nombre, la estatura, el peso y talla de pie**. Un modelo podrá participar en tantos desfiles como le requieran y un dato esencial es el **número de pie** en el que un modelo participa en cada desfile.

IDENTIFICACIÓN DE ENTIDADES Y ATRIBUTOS

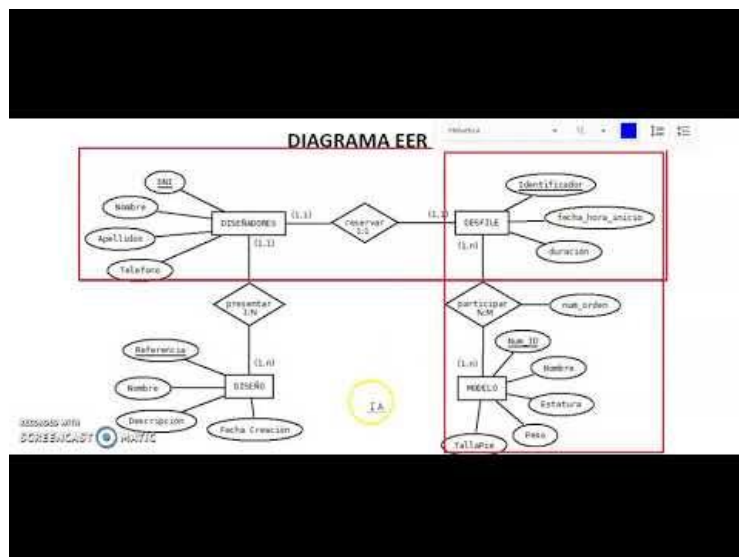
ENTIDAD	ATRIBUTOS
DISEÑADOR	DNI, nombre, apellidos, teléfono contacto.
DISEÑO	Referencia, nombre, descripción, fecha creación.
DESFILE	Identificador, fecha y hora comienzo, duración
MODELOS	Número identificador, nombre, estatura, peso, talla pie.

TEMA 2: BASE DE DATOS RELACIONALES.

Ahora ya se puede representar el diagrama Entidad Relación utilizando una herramienta gráfica como DIA que contiene la notación elegida para la representación.



Por último se realiza el paso al modelo relacional de este ejemplo, pasando las entidades a tablas, los atributos como campos y las relaciones según las cardinalidades propagando claves de unas tablas a otras o bien creando nuevas tablas resultantes.



TEMA 2: BASE DE DATOS RELACIONALES.

Ahora te toca practicar la representación de diagramas ER y pasar a su correspondiente MR. Para eso, te proponemos varios ejercicios en los cuales se plantea los requerimientos necesarios para que realices el diseño de la correspondiente base de datos. Es muy recomendable que **primero** intentes resolver por tí mismo cada uno de los supuestos, y **después** cotejes tu solución con la propuesta.

¡Ánimo, inténtalo! Verás que es más fácil de lo que piensas.

¿Qué debes realizar exactamente?

Pues lo que se pide es realizar el análisis y diseño de la base de datos. Para ello debes obtener el **diagrama E/R** (entidades, atributos, relaciones, claves, cardinalidades y otras características del modelo Entidad Relación Extendido) y su posterior transformación al **Modelo Relacional** (conjunto de tablas con sus claves correspondientes y sus relaciones)

Te aconsejamos que sigas estos pasos:

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

1. Identificar las entidades (fuertes y débiles), las relaciones entre las entidades y los atributos (tanto de entidades como posiblemente en algún caso de relaciones).
2. Identificar las cardinalidades de las entidades y después las cardinalidades de las relaciones.
3. Identificar las claves (candidatas y primarias) y los atributos calculados, compuestos y multivaluados.
4. Estudiar si existen otras características del modelo E/R que no se puedan representar con el modelo ER.

Para la **representación del Modelo Relacional**:

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

Centro de Estudios

Un centro de estudios desea diseñar una base de datos para llevar el control de los alumnos matriculados y los profesores que imparten clases en ese centro.

- De cada profesor y cada alumno se desea almacenar el nombre, apellidos, dirección, población, dni, fecha de nacimiento, código postal y teléfono. Un alumno puede tener hermanos en el centro, nos interesaría saber que alumnos son hermanos de otros.
- Los alumnos se pueden matricular en una o más asignaturas hasta un máximo de 8 y en una asignatura se pueden matricular hasta un máximo de 30 alumnos por cuestiones de disponibilidad en las aulas. De cada asignatura se desea almacenar el código de asignatura, nombre y número de horas que se imparten a la semana.
- Un profesor del centro puede impartir varias asignaturas, pero una asignatura sólo es impartida por un único profesor. De cada una de las asignaturas se desea almacenar también la nota que saca el alumno y las incidencias que puedan darse con él.
- Además, se desea llevar un control de los cursos que se imparten en el centro de enseñanza. De cada curso se guardará el código y el nombre. En un curso se imparten varias asignaturas, y una asignatura sólo puede ser impartida en un único curso.
- Las asignaturas se imparten en diferentes aulas del centro. En el centro hay varios tipos de aulas, entre ellas: Aulas TIC con ordenadores y acceso a Internet y aulas normales sin ordenadores. De cada aula se quiere almacenar el código y piso del centro en el que se encuentra. De las Aulas TIC también nos interesa saber el número de ordenadores que tiene y de las aulas normales el número de pupitres de que dispone.
- Una asignatura se puede dar en diferentes aulas, y en un aula se pueden impartir varias asignaturas. Se desea llevar un registro de las asignaturas que se imparten en cada aula. Para ello se anotará el mes, día y hora en el que se imparten cada una de las asignaturas en las distintas aulas.
- La dirección del centro también designa a varios profesores como tutores en cada uno de los cursos. Un profesor es tutor tan sólo de un curso. Un curso tiene un único tutor. Se habrá de tener en cuenta que puede que haya profesores que no sean tutores de ningún curso.

Solución Centro de Estudios

Hospital

Con el objeto de crear un software para la administración de un hospital, se pretende diseñar una base de datos. Tras un estudio de la información existente en dicho centro, se obtuvieron los siguientes datos:

- En el hospital se almacena información relativa a los enfermos: código de enfermo, nombre, dirección, DNI, fecha de nacimiento, teléfono de contacto, edad.
- También se almacena la información relativa a los trabajadores del hospital (en este caso sólo serán médicos y auxiliares), de los que se desea conocer: Nombre, dirección, DNI, sueldo, teléfono de contacto. En el caso de los médicos, además se desea conocer la especialidad.
- Los enfermos están alojados en habitaciones, siendo interesante almacenar la fecha de hospitalización del enfermo así como la fecha de alta cuando el enfermo abandona el hospital.
- En una misma habitación, puede haber más de un enfermo. Para cada habitación se almacena además de su número, la cantidad de camas que tiene y se anotan las observaciones. Las habitaciones pertenecen a las distintas plantas del hospital no pudiéndose repetir su numeración en distintas plantas.
- Para cada planta se almacena la cantidad de habitaciones que hay y la especialidad de la planta además de su número.
- Existe un conjunto de auxiliares de enfermería. Cada auxiliar, está asignado a una única planta.
- Cada enfermo tendrá un conjunto de diagnósticos que son emitidos por los médicos que los curan. Para cada diagnóstico del enfermo, se guarda la fecha del diagnóstico, y el informe, junto con el código correspondiente. Cada diagnóstico es emitido por un único médico.

Solución Hospital

Restaurante

Se desea crear una base de datos para un restaurante. Tras el estudio de la información existente se obtienen las siguientes especificaciones:

- Interesa conocer el nombre y apellidos de cada empleado, su DNI y su número de SS además del teléfono fijo y el móvil.
- Algunos empleados son cocineros. De los cocineros se quiere saber (además de los datos propios de cada empleado) su fecha de alta en la empresa.
- Hay empleados/as que son pinches. De los pinches también interesa saber su fecha de nacimiento.
- Interesa almacenar más trabajadores del restaurante que no son ni pinches ni cocineros.
- De cada plato nos interesa conocer su nombre como “pollo a la carloteña”, “bacalo al pil-pil”,... el precio del plato y los ingredientes que lleva. También si cada plato es un entrante, un primer plato, segundo plato o postre.
- De los ingredientes interesa saber el nombre, precio/kilo, la cantidad que necesitamos de él en cada plato y en qué almacén y estante del mismo los tenemos. Un ingrediente sólo puede estar en un estante pero en un estante sí podemos tener varios ingredientes.
- Nos interesa también conocer la cantidad de cada ingrediente que hay en los estantes.
- Cada almacén tiene un nombre (despensa principal, cámara frigorífica A, cámara frigorífica B...), un número de almacén y una descripción del mismo.
- Cada estante en el almacén se identifica con una letra además nos interesa conocer su tamaño en centímetros. Dos almacenes distintos pueden tener estantes con la misma letra.
- Necesitamos también saber qué cocineros son capaces de preparar cada plato teniendo en cuenta que cada cocinero está especializado en una serie de platos que sólo puede preparar él.
- Cada pinche está asignado a un único cocinero. Un cocinero sólo podrá disponer como máximo de un pinche.

Solución Restaurante

Seguros

Una compañía de seguros de automóviles quiere crear una base de datos que recoja el funcionamiento de su negocio y que se refleja en los siguientes supuestos semánticos:

- El elemento fundamental de información es la póliza, la cuál se identifica mediante un número único, tiene un tipo de seguro (a todo riesgo, a terceros, etc.), un importe de cobertura máxima y un estado (alta, baja, suspensión, etc.). La póliza pertenece a un único cliente (un cliente puede tener más de una póliza diferente) y referencia a un único vehículo. Cada vehículo sólo puede tener una póliza.
- Los clientes se identifican mediante su NIF, además se quiere guardar su nombre y apellidos, su teléfono, fecha de nacimiento, fecha de obtención del permiso de conducir y su dirección completa (calle, número, ciudad, código postal y provincia).
- De los vehículos es importante conocer su número de chasis, su matrícula, la marca, el modelo, la potencia, el año de fabricación y el color. Además un vehículo puede tener una serie de extras (alarma, radio, etc.) de los que guardaremos un código identificador y el nombre del extra.
- Una póliza puede tener una serie de personas autorizadas, de las cuales se quiere tener su NIF, nombre y apellidos, fecha de nacimiento y relación con el cliente. Un autorizado sólo tendrá relación con un único cliente.
- Cuando se produce un siniestro, se crea un parte de accidente (identificado por un número de siniestro) donde se recoge la información del siniestro: datos de la póliza del cliente, datos del conductor (sólo puede ser el cliente, o alguien autorizado), fecha del siniestro y datos del taller donde se va a reparar el vehículo. Además es importante guardar la fecha e importe de la reparación teniendo en cuenta que un siniestro podría ser reparado en uno o varios talleres. Si el accidente es contra otro vehículo no se guardan ninguna información del vehículo contrario, si es de la misma compañía el cliente ya dará su propio parte de accidente.
- Los datos del taller que se almacenan serán, el nombre, la dirección y el teléfono además de un código identificador.

Solución Seguros

11. NORMALIZACIÓN DE MODELOS RELACIONALES.

¿Crees que tu base de datos ya podría implementarse directamente sobre el SGBD relacional elegido?. La respuesta podría ser afirmativa, pero si queremos que nuestra base esté correctamente diseñada y en el futuro funcione con plena fiabilidad, es necesario que testemos el esquema lógico obtenido para comprobar que sus tablas estén normalizadas. Esta comprobación la realizaremos mediante el proceso de normalización.

¿Y en qué consiste 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.

Edgar Frank Codd fue el que ideó estos conceptos.

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

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

Como norma general, para garantizar que no existan problemas en la actualización de datos, **es recomendable aplicar el proceso de normalización hasta Tercera Forma Normal o incluso hasta Forma Normal de Boyce-Codd**. En los siguientes epígrafes se describen las características y requisitos de cada una de las formas normales.

11.1 TIPOS DE DEPENDENCIAS.

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, ...Ak y B, se dice que B depende funcionalmente de forma completa de A1, A2, ...Ak, si y solo si B depende funcionalmente del conjunto de atributos A1, A2, ...Ak, pero no de ninguno de sus posibles subconjuntos.
- ✓ **Dependencia Transitiva:** Dados tres atributos A, B y C, se dice que existe una dependencia transitiva entre A y C, si B depende funcionalmente de A y C depende funcionalmente de B. A, B y C podrían ser un solo atributo o un conjunto de ellos.

Para ilustrar los tipos de dependencias descritas, analiza el siguiente ejercicio resuelto.

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:

1. Indica qué atributos presentan una dependencia funcional de la clave primaria de la tabla **EMPLEADO**.
2. Indica qué atributos presentan una dependencia funcional completa en la tabla **LIBRO**.
3. Indica qué atributos presentan una dependencia transitiva en la tabla **EMPLEADO**.

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. (Consideraremos para este ejemplo que todos los empleados registrados en esta base de datos tienen nombres distintos). Expresemos estas dependencias funcionales mediante su notación: $DNI \rightarrow Nombre$ $DNI \rightarrow Dirección$

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 **Título_libro**, por lo que no presenta una dependencia funcional completa de los atributos que forman la clave.

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 afirmar que existe una dependencia transitiva entre **Localidad** y **DNI**. Si lo representamos con la notación asociada a las dependencias funcionales, quedaría: $DNI \rightarrow Cod_Localidad \rightarrow Localidad$.

11.2 FORMAS NORMALES.

Una vez conocidos los conceptos sobre los que se basa el proceso de normalización, se han de llevar a cabo una serie de etapas consecutivas en las que se aplicarán las propiedades de cada una de las formas normales definidas por Codd. A continuación se exponen los requisitos a cumplir por las tablas de nuestra base de datos según la forma normal que apliquemos.

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

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

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

1. 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.
2. Con los atributos restantes, se crea otra tabla que tendrá por clave el subconjunto de atributos de la clave inicial de los que dependen de forma completa. Se comprueba si esta tabla está en 2FN. Si es así, la tabla inicial ya está normalizada y el proceso termina. Si no está en 2FN, tomamos esta segunda tabla como tabla inicial y repetiremos el proceso.

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

- a. 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.
- b. Con los atributos restantes, se crea otra tabla con los otros 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.

Forma Normal de Boyce Codd: Una tabla está en Forma Normal de Boyce-Codd (FNBC o BCFN) sí, y sólo sí, está en 3FN y todo determinante es una clave candidata. Un determinante será todo atributo simple o compuesto del que depende funcionalmente de forma completa algún otro atributo de la tabla. Aquellas tablas en la que todos sus atributos forman parte de la clave primaria, estarán en FNBC. Por tanto, si encontramos

TEMA 2: BASE DE DATOS RELACIONALES.

un determinante que no es clave candidata, la tabla no estará en FNBC. Esta redundancia suele ocurrir por una mala elección de la clave. Para normalizar a FNBC tendremos que descomponer la tabla inicial en dos, siendo cuidadosos para evitar la pérdida de información en dicha descomposición.

Otras formas normales

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

Ejercicio resuelto

Sea la siguiente tabla: **COMPRAS** (cod_compra, cod_prod, nomb_prod, fecha, cantidad, precio, fecha_rec, cod_prov, nomb_prov, tfno).

Se pide normalizarla hasta FNBC.

Comprobamos 1FN:

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

Comprobamos 2FN:

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

Veamos las dependencias: cod_prod → nomb_prod, y cod_prod es parte de la clave primaria.

Al no estar en 2FN, hemos de descomponer la tabla **COMPRAS** en: **COMPRA1** (cod_compra, cod_prod, fecha, cantidad, precio, fecha_rec, cod_prov, nomb_prov, tfno). **PRODUCTO** (cod_prod, nomb_prod).

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

Comprobamos 3FN:

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

Veamos las dependencias: cod_prov → nomb_prov cod_prov → tfno (siendo cod_prov el código del proveedor y nomb_prov el nombre del proveedor)

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

COMPRA2 (cod_compra, cod_prod, fecha, cantidad, precio, fecha_rec, cod_prov) **PROVEEDOR** (cod_prov, nomb_prov, tfno)

TEMA 2: BASE DE DATOS RELACIONALES.

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)

Recomendación

Si aún tienes dudas sobre el proceso de normalización te recomendamos que visualices el siguiente videotutorial:

Formas Normales

Segunda Forma Normal

Una relación está en Segunda Forma Normal (2NF) si y sólo si está en 1NF y cada atributo no-clave de la relación es total y funcionalmente dependiente de la clave primaria.

Diagrama de dependencias:

Producto (Clave Primaria) → Valor (Atributo no-clave)

Tabla de datos:

Nº Factura	Fecha	Nº Cliente	Nombre Cliente	Cantidad
000127	01/07/2008	1234567	Juan Perez	Valor
000128	01/07/2008	1234568	Ana Palom	Segs.

Tabla de datos:

Producto	Valor
Lapiz Bto	250
Block	200
Lapiz Gordin	70
Cuadernos	350
Regla	20

Tabla de datos:

Nº Factura	Producto	Cantidad	Descuento
000127	Lapiz Bto	250	5 %
000127	Block	200	5 %
000127	Lapiz Gordin	70	0 %
000128	Lapiz Bto	400	6 %
000128	Cuadernos	350	6 %
000128	Regla	20	0 %

12. ENLACES DE REFUERZO Y AMPLIACIÓN.

- Herramienta ON-LINE para diagramas: <https://www.draw.io/>
- Programa para diseño de Diagramas Entidad-Relación "Día": <http://dia-installer.de/index.html.es>
- Ejemplo de uso de "Día" para la elaboración de un Diagramas Entidad-Relación sencillo : <https://www.youtube.com/watch?v=kixAsBgB6ms>
- Aclaración conceptos básicos E-R: <http://es.slideshare.net/DaniSantia/t1-modelo-entidadrelacin>
- Como pasar del modelo E-R al modelo Relacional todos los elementos: http://es.slideshare.net/buster_blader99/conversin-diagrama-er-a-modelo-relacional?related=1
- Como pasar del modelo E-R al modelo Relacional exclusivamente relaciones de grado 3 (paso a tablas de relaciones ternarias): <http://www.dataprix.com/34-transformacion-interrelaciones-ternarias>
- Definiciones y ejemplos sobre Normalización sobre el Modelo Relacional: <http://mysql.conclase.net/curso/index.php?cap=004#>