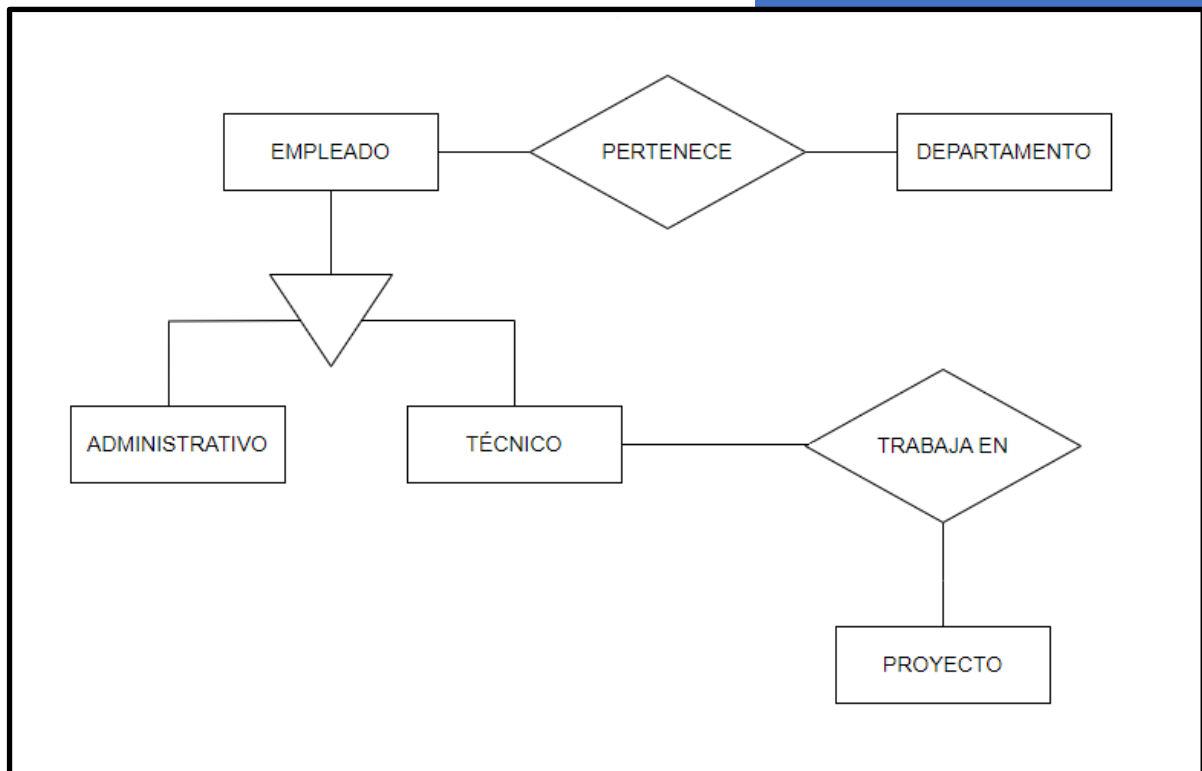
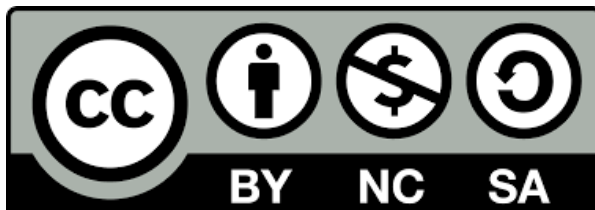


Tema 2

El diagrama entidad/relación (diagrama E/R)



Este documento ha sido elaborado para el alumnado del módulo “Bases de datos” de los CFGS DAM y DAW del IES Playamar.



Más información en <https://creativecommons.org/licenses/by-nc-sa/3.0/es/>

Contenido

1.	Introducción	1
2.	El diagrama entidad/relación (E/R).....	2
2.1.	Entidad	3
2.2.	Relación.....	4
2.3.	Atributos	9
2.4.	Otras notaciones del diagrama E/R	11
2.5.	El diagrama E/R extendido.....	12
3.	Del requerimiento al diagrama E/R.....	13
4.	Transformación del diagrama E/R a modelo lógico.....	17
4.1.	Transformación de entidades.....	18
4.2.	Transformación de relaciones.....	20
5.	Normalización.....	22
6.	Resumen.....	26

1. Introducción

Toda base de datos representa información contenida en algún dominio del mundo real. En cualquier proyecto informático, el diseño de la base de datos va a consistir en extraer todos los datos relevantes de un problema, y la información relevante va a depender de las características del problema a resolver.

Por ejemplo, las bases de datos de una biblioteca, de un banco o de un hospital guardan datos de personas, pero la información relevante en cada contexto varía:

- En una biblioteca se guardan datos de socios. Con almacenar su nombre, dirección postal, dirección de correo y un número de teléfono quizás es más que suficiente.
- En un banco se guardan datos de clientes. El banco necesita almacenar su situación laboral (desempleado, asalariado, autónomo), ingresos mensuales, cargas y deudas que tenga, etc.
- En un hospital se guardan datos de pacientes. El equipo médico necesita consultar la edad, altura y peso de un paciente, su historial médico, si es alérgico a un alimento y/o medicamento, antecedentes médicos familiares, etc.

Así pues, se debe analizar el dominio del problema para determinar los datos esenciales para la base de datos y descartar los que no son de utilidad. El análisis parte de la especificación de requisitos, documento acordado con el cliente en el que se indica (especifica) el comportamiento del sistema que se va a construir.

Tras ese análisis comienza el proceso de modelización de la base de datos. Es decir, construir un esquema que exprese con exactitud todos los datos que el problema requiere almacenar. El punto de partida en el diseño de una base de datos va a ser siempre el nivel conceptual de la arquitectura ANSI/X3/SPARC, ya que antes de pensar *cómo* guardar la información hay que tener claro *qué* hay que guardar.

Puesto que el informático es un experto en informática y no lo es en medicina, banca o industria, es importante acordar con el cliente la información relevante de su modelo de negocio. Es muy importante disponer de un esquema fácil de entender para que el cliente valide el modelo de datos. El esquema ya existe y se llama diagrama entidad/relación.

2. El diagrama entidad/relación (E/R)

El diagrama entidad/relación (o diagrama E/R) fue creado por Peter P. Chen a mediados de la década de 1970 para la representación conceptual de los datos y establecer qué relaciones existen entre ellos. Sus características son:

- La representación es gráfica.
- Es muy fácil de entender, no se requieren conocimientos técnicos para su comprensión (ideal para validar el modelo con el cliente).
- Es independiente de la base de datos¹.

CASO PRÁCTICO 1

La figura 1 muestra un ejemplo de diagrama E/R. Se puede entender sin mucho esfuerzo que un empleado pertenece a un departamento, que hay dos tipos de empleados: administrativos y técnicos, y que éstos últimos trabajan en proyectos.

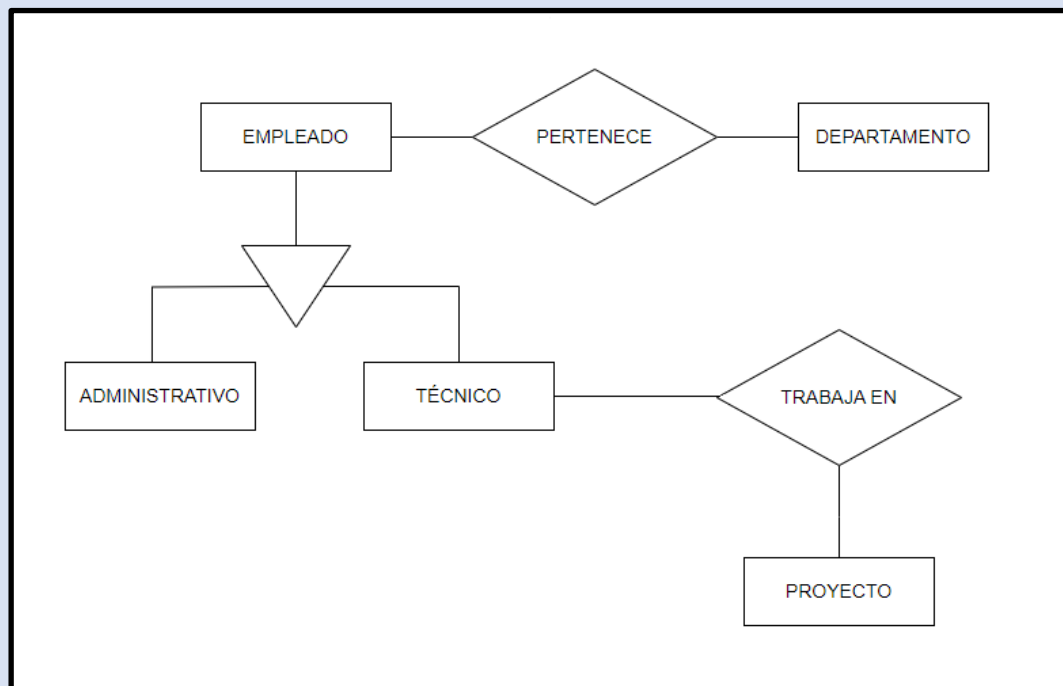


Figura 1. Ejemplo de diagrama E/R.

Como el diagrama se llama entidad/relación, lo primero que debe quedar claro es qué se entiende por entidad y qué se entiende por relación.

¹ Cuando el diagrama E/R fue creado, el modelo relacional acababa de aparecer y las bases de datos existentes principalmente eran jerárquicas o en red.

2.1. Entidad

Una entidad es cualquier objeto del se necesita recoger información. Se representan con un rectángulo y un sustantivo (preferentemente en singular) en su interior. En el ejemplo de la figura 1 se muestran cinco entidades: empleado, departamento, administrativo, técnico y proyecto.

Existen dos tipos de entidades: fuertes (o regulares) y débiles.

ENTIDAD FUERTE	ENTIDAD DÉBIL
Una entidad fuerte existe por méritos propios. La práctica totalidad de entidades de un diagrama son entidades fuertes.	Una entidad débil es aquélla en la que su existencia depende de otra entidad. Para diferenciarlas de las fuertes, las entidades débiles se representan con un doble rectángulo.

CASO PRÁCTICO 2

En una base de datos de un almacén se almacenan pedidos. Un pedido se compone de una cabecera (cliente, fecha de pedido, modo de pago, dirección de envío...) y una serie de líneas de pedido donde se detalla cada producto del pedido (descripción, precio, cantidad...). Mientras no haya un pedido no tiene sentido hablar de líneas de pedido.

Pedido

Línea de pedido

En una base de datos de un banco se almacenan cuentas de clientes. En una cuenta bancaria se realizan movimientos (ingresos, extracciones). Mientras un cliente no abra una cuenta bancaria no puede hacer movimientos en ella.

Cuenta bancaria

Movimiento

En una base de datos de una biblioteca se almacenan socios. Un socio solicita préstamos de libros. Mientras el socio no vaya a la biblioteca y retire el libro de la estantería, no se puede prestar el libro en préstamo.

Socio

Préstamo

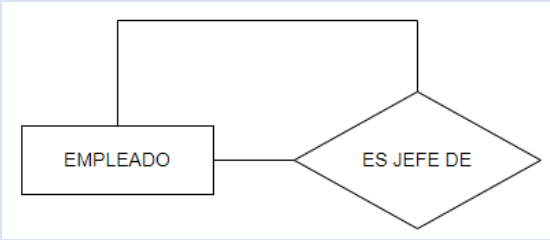

Una entidad representa un conjunto de valores. Cada elemento del conjunto es una instancia de la entidad. Por ejemplo, para la entidad Municipio, los valores “Fuengirola” o “Torremolinos” son instancias.

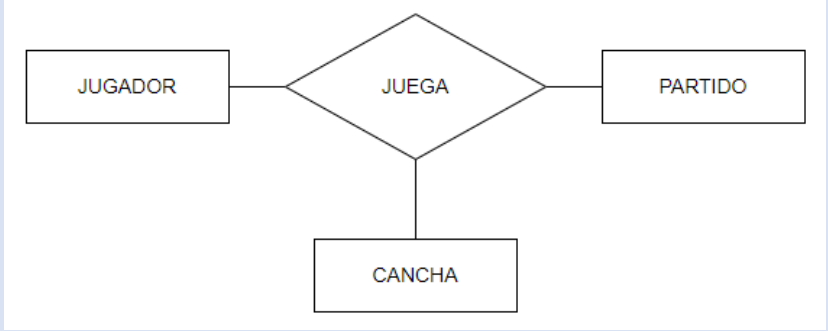
2.2. Relación

Una relación (o interrelación) es una correspondencia o asociación entre dos o más entidades. Se representan con un rombo y una expresión (un verbo o similar) en su interior. Las entidades que participan en la relación quedan unidas a la relación por medio de una línea. En el ejemplo de la figura 1 se muestran dos relaciones: pertenece (un empleado *pertenece* a un departamento) y trabaja (un técnico *trabaja en* un proyecto).

Las relaciones se pueden leer en ambos sentidos. Así, en el ejemplo de la figura 1 se puede leer “Un empleado pertenece a un departamento” o “A un departamento le pertenece un empleado”. El nombre de la relación debe ser lo más específico y descriptivo posible. En el ejemplo anterior, se podría decir también que “un departamento tiene empleados”, pero el verbo tener puede servir también para un “almacén tiene productos”.

Las relaciones se clasifican por su grado. El grado es el número de entidades que participan en la relación. Atendiendo a este criterio, las relaciones pueden ser:

GRADO	DESCRIPCIÓN
Grado 1 (reflexivas o unarias)	Interviene una única entidad que adopta varios papeles (o roles). 
Grado 2 (binarias)	Intervienen dos entidades (son las más comunes). 

GRADO	DESCRIPCIÓN
Grado 3 (ternarias)	<p>Intervienen tres relaciones.</p> 
Grado > 3 (n-arias)	<p>Intervienen más de tres relaciones.</p> <p>Es raro que aparezcan relaciones de este grado. No obstante, en el caso en que aparezcan, es preferible descomponerlas en relaciones binarias o ternarias.</p>

Cada entidad de una relación tiene asociada una **participación**. La participación indica el número mínimo y máximo de veces que puede aparecer una instancia de la entidad en la relación. Se representa con un par de valores en la forma (mínimo, máximo) en la línea que une la entidad con la relación. Las participaciones de las entidades quedan definidas por las reglas de negocio y deben estar recogidas en la especificación de requisitos del problema. Los posibles valores son:

PARTICIPACIÓN	DESCRIPCIÓN
(0,1)	Mínimo 0, máximo 1 (opcional)
(1,1)	Mínimo 1, máximo 1 (requerido)
(0, n)	Mínimo 0, máximo muchos.
(1, n)	Mínimo 1, máximo muchos.

CASO PRÁCTICO 3

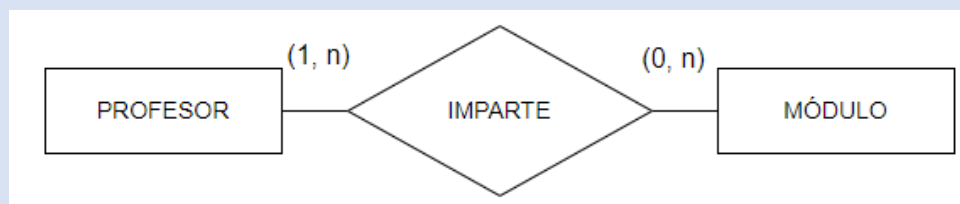
En un instituto que imparte Ciclos Formativos hay una serie de profesores que imparten módulos. Un módulo debe ser impartido por un profesor, pudiendo haber varios profesores que impartan el mismo módulo. Del mismo modo, un profesor puede

impartir uno o varios módulos, y podría darse el caso en que no imparta docencia (por tener otras ocupaciones: dirección, reducción de jornada...)

En la especificación se detectan dos entidades (Profesor y Módulo) y una relación (Imparte). Respecto a la participación:

- Por cada instancia de la entidad Módulo deben participar una instancia de la entidad Profesor (porque todo módulo es impartido por un profesor) o varias (porque puede haber varios profesores impartiendo el mismo módulo). Por tanto, la participación de la entidad Profesor es (1, n).
- Por cada instancia de la entidad Profesor deben participar una o varias instancias de la entidad Módulo (porque puede impartir varios módulos), pero puede darse el caso en que el profesor no imparta docencia. Por tanto, la participación de la entidad Módulo es (0, n).

Tras el análisis realizado, el diagrama E/R resultante es:



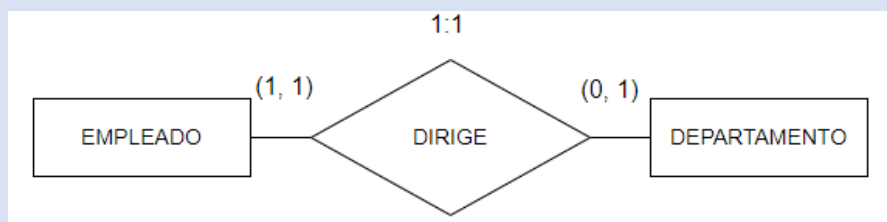
Toda relación tiene asociada una **cardinalidad**. La cardinalidad delimita los límites superiores de la relación y se calcula tomando los valores máximos de las participaciones de las entidades que intervienen en la relación. La cardinalidad de una relación puede ser:

CARDINALIDAD	DESCRIPCIÓN
1:1 (Uno a uno)	Una instancia de una entidad A se vincula con una única instancia de una entidad B, y viceversa.
1:N (Uno a muchos)	Una instancia de una entidad A se vincula con muchas instancias de una entidad B, pero una instancia de B se vincula con una única de A.

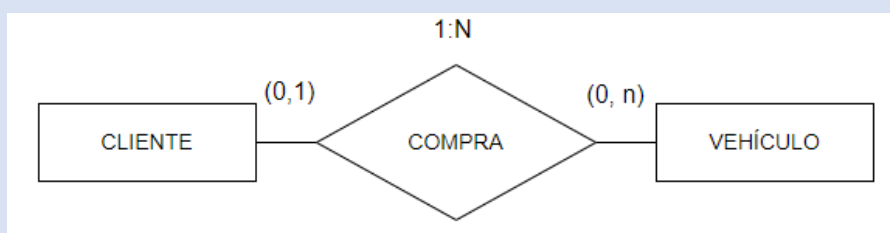
CARDINALIDAD	DESCRIPCIÓN
	Como las relaciones se pueden leer en ambos sentidos “Muchos a uno” es lo mismo que “Uno a muchos” por lo que lo normal es usar siempre la expresión 1:N.
N:M (Muchos a muchos)	Una instancia de una entidad A se vincula con muchas instancias de una entidad B, y una instancia de B se vincula a muchas de A. La cardinalidad 1:N se produce en los dos sentidos.

CASO PRACTICO 4

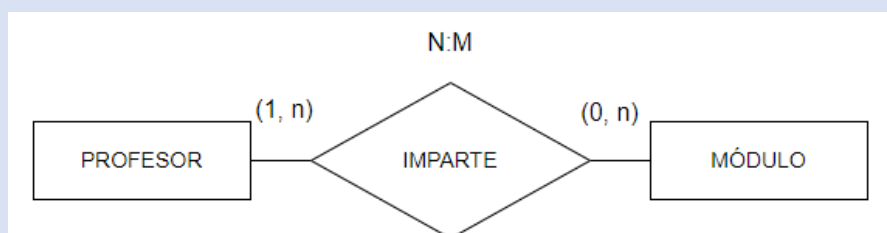
En una empresa hay departamentos que deben ser dirigidos por un empleado. Sólo puede haber un jefe por departamento y un empleado no puede ser jefe de más de un departamento.



En un concesionario hay clientes que compran vehículos. Cada cliente puede comprar un número indeterminado de vehículos, y sólo se permite un comprador por vehículo.



Un módulo debe ser impartido por un profesor, pudiendo haber varios profesores que impartan el mismo módulo. Del mismo modo, un profesor puede impartir uno o varios módulos, y podría darse el caso en que no imparta docencia.



En las relaciones ternarias, la participación correspondiente a cada entidad indica cómo se relacionan las instancias de dicha entidad con una vinculación de instancias del resto de las entidades participantes en la relación. Una vez obtenidas las participaciones individuales de cada entidad, la cardinalidad de la relación es la combinación de los valores máximos de las participaciones.

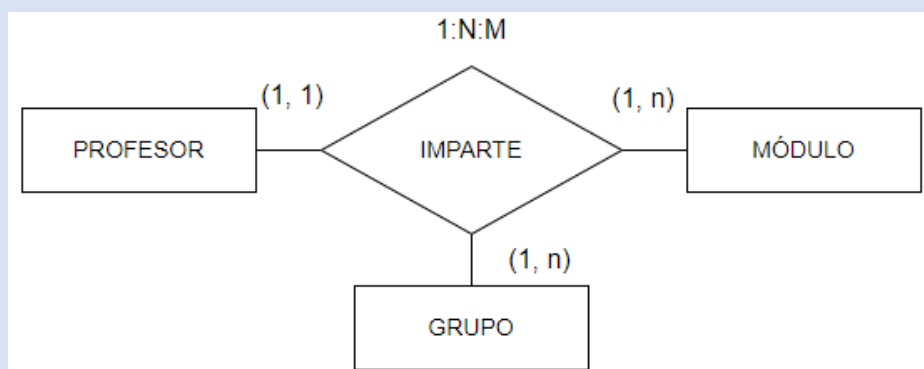
CASO PRÁCTICO 5

En el caso práctico 3 (los profesores y los módulos) se añade ahora el grupo para el que se imparte la docencia, considerando que un profesor relacionado con un grupo debe impartir al menos un módulo a dicho grupo, el profesor de un módulo puede ser docente en varios grupos y el módulo impartido a un grupo corresponde a un único profesor.

En este caso la relación Imparte relaciona tres entidades: Profesor, Módulo y Grupo. La participación de cada entidad se estudia tomando una combinación de las otras dos entidades vinculadas a la relación. Así, se tiene:

- Profesor: para la combinación módulo-grupo hay un profesor (algún docente debe impartir el módulo) y sólo uno (el módulo impartido a un grupo corresponde a un único profesor). Por tanto, la participación del profesor es (1, 1).
- Módulo: para la combinación profesor-grupo hay un módulo (un profesor relacionado con un grupo debe impartir al menos un módulo) o más de uno. Por tanto, la participación es (1, n).
- Grupo: por cada combinación profesor-módulo debe haber un grupo (si un profesor tiene asignado un módulo se lo impartirá a algún grupo) o más de uno. Por tanto, la participación es (1, n).

Por tanto, la relación es 1 a muchos y muchos y el diagrama E/R resultante es:



2.3. Atributos

Los atributos de una entidad son las características o propiedades que caracterizan a la entidad. Se representan mediante elipses conectadas directamente a la entidad, como se ilustra en la figura 2.

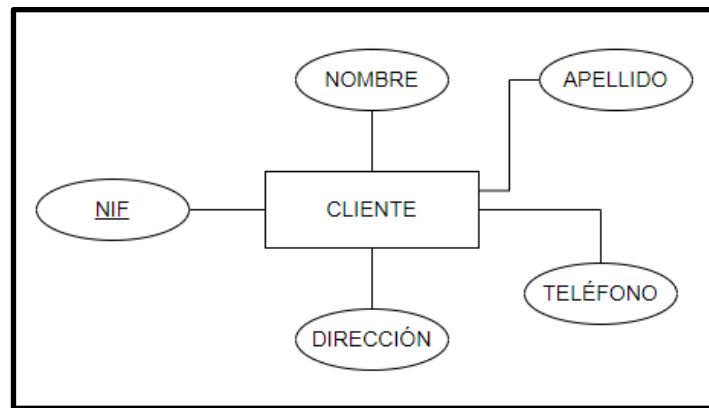


Figura 2. Una entidad con atributos.

Se llama **atributo clave** a todo atributo, o conjunto de atributos, que identifica a las instancias de la entidad. En el ejemplo, un cliente queda identificado con su NIF. Los atributos claves se representan con el nombre subrayado.

Las relaciones también pueden tener atributos. Un **atributo de relación** es un atributo propio de una relación y no puede ser cedido a una de las entidades que participan en la relación. Por ejemplo, se desea almacenar la fecha en la que un mecánico repara un vehículo. Esa fecha no pertenece al mecánico ni al vehículo sino a la reparación, por lo que debe asociarse a la relación Repara (figura 3).

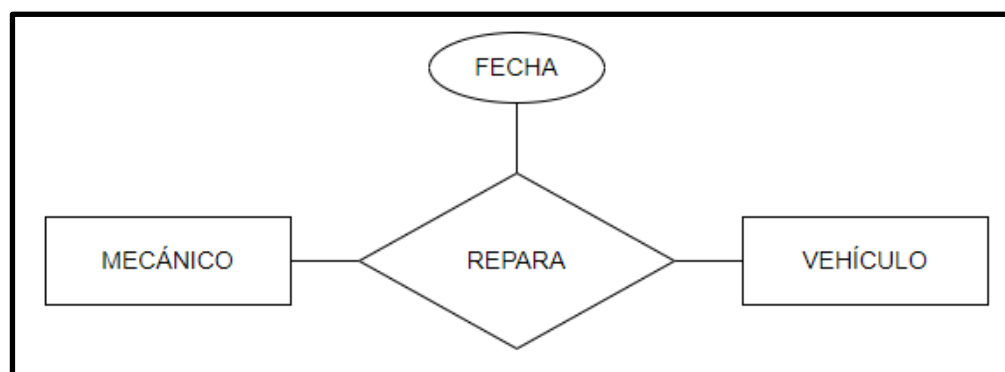


Figura 3. Atributo de relación.

Todo atributo tiene asociado un dominio, es decir, el tipo de datos que puede contener. También, en función de las características y restricciones que se obtengan de las reglas de negocio, los atributos se pueden clasificar según se resume en la siguiente tabla:

TIPO DE ATRIBUTO	DESCRIPCIÓN
Obligatorio (o requerido)	Atributo que debe tomar un valor obligatoriamente.
Opcional	Atributo que no es necesario informar y se puede dejar vacío (valor nulo).
Simple	Atributo que no es compuesto.
Compuesto	Atributo que se puede descomponer en otros más sencillos. Por ejemplo, la dirección (tipo de calle, nombre de calle, número...)
Univaluado	Atributo que toma un único valor.
Multivaluado	Atributo que puede tomar varios valores. Por ejemplo, el teléfono (fijo, móvil, personal, del trabajo...)
Derivado	Atributo cuyo valor se puede calcular a partir de otros atributos. Por ejemplo, la edad (se calcula a partir de la fecha de nacimiento y el día presente)

CONSEJO

¿Es necesario mostrar los atributos en el diagrama E/R? En la figura 2 se muestran cinco atributos de la entidad Cliente, pero realmente puede tener muchos más. Si a eso se le suman los atributos del resto de entidades, el diagrama *va a parecer un árbol de navidad con tanta bolita*.

Quizás es mejor no mostrar los atributos en el diagrama E/R (o sólo los más relevantes) y dejar el detalle de los atributos para el modelo lógico de la base de datos (que se verá más adelante).

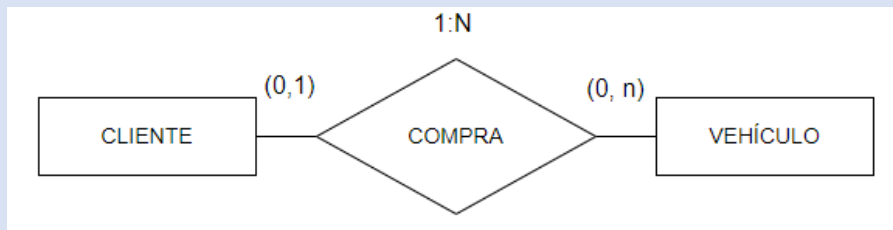
Es buena idea que en los diseños que realices, y en general en cualquier desarrollo de software, apliques el principio **KISS (Keep It Simple, Stupid!)**. Cuanto más sencillo es un diseño, más fácil es de entender y mantener.

2.4. Otras notaciones del diagrama E/R

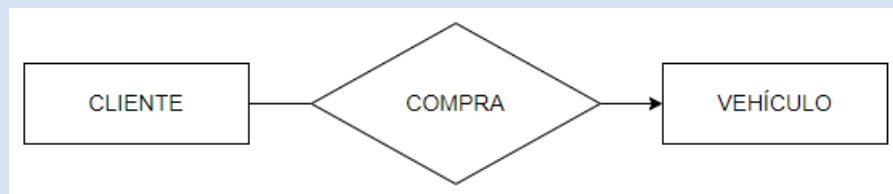
La notación explicada no es la única que existe para representar el diagrama E/R del modelo conceptual de la base de datos, por lo que es posible que encuentres diferencias si consultas otros libros sobre bases de datos o cambias de herramienta de diseño.

CASO PRÁCTICO 6

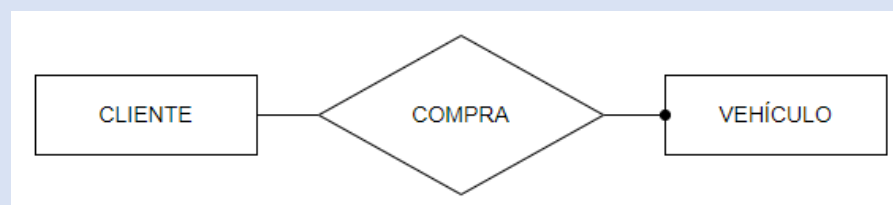
Por ejemplo, la relación “Cliente compra vehículo”, que hemos representado así:



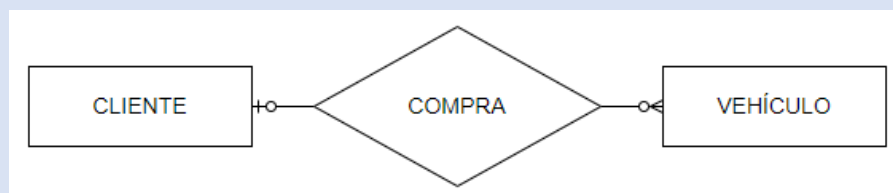
Podría representarse también de esta manera, donde una línea indica cardinalidad 1 y una flecha cardinalidad N:



O de esta otra, adaptando notación utilizada en UML:



También de esta otra, donde se indica la participación. El símbolo + indica participación 1, el círculo indica 0 (opcional) y la *pata de gallo* indica n:



2.5. El diagrama E/R extendido

El diagrama E/R explicado hasta ahora es más que suficiente para la mayoría de situaciones que se presentan al modelar una base de datos, pero no permite representar situaciones como la siguiente:

En una universidad los profesores son docentes o investigadores. Sólo los docentes imparten asignaturas, los investigadores participan en proyectos de investigación.

El diagrama E/R extendido incluye un nuevo elemento, la relación **es un**. Se representa con un triángulo (generalmente invertido) y asocia entidades en forma de jerarquía para representar el concepto de especialización o generalización. La figura 4 ilustra el concepto:

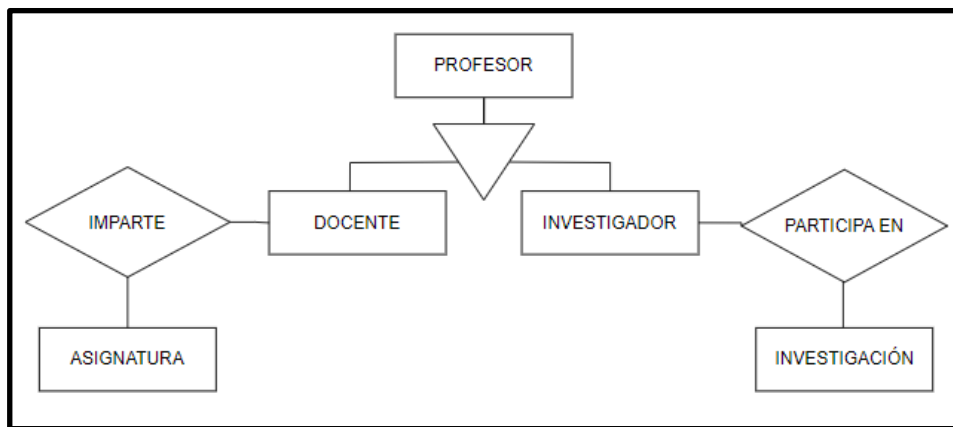


Figura 4. Diagrama E/R extendido con relación es un.

El triángulo representa que un profesor *es un* docente o *es un* investigador. Cuando actúa como docente, imparte materias. Cuando actúa como investigador, participa en proyectos de investigación. Tanto docente como investigador heredan todos los atributos de un profesor.

La relación *es un* se puede leer en los dos sentidos:

- Si se lee de arriba abajo, docente e investigador son especializaciones de un profesor y cada una de estas entidades puede añadir atributos propios.
- Si se lee de abajo arriba, docente e investigador son entidades que contienen atributos comunes que se pueden generalizar en la entidad profesor.

La relación *es un* puede ser inclusiva o exclusiva.

ES UN	DESCRIPCIÓN
Inclusiva	Cualquier instancia de la entidad genérica puede participar en cualquiera de las especializaciones. En el ejemplo de la figura 4, un profesor puede ser al mismo tiempo docente e investigador.
Exclusiva	Cualquier instancia de la entidad genérica sólo puede participar en una de las especializaciones. Si asumimos que un profesor sólo puede ser docente o investigador (y no puede ser ambas cosas a la vez), se representa con un arco como se muestra en la figura 5.



Figura 5. Relación de especialización exclusiva.

3. Del requerimiento al diagrama E/R

Esta es una guía de los pasos que deben darse para crear un diagrama E/R a partir de un requerimiento:

1. Lee el requerimiento cuantas veces sean necesarias para comprenderlo bien.
2. Obtiene una lista inicial de candidatos a entidades y relaciones, así como de sus correspondientes atributos. Para ello sigue estos consejos:
 - a. Identifica las entidades. Son nombres que son importantes para el desarrollo del problema. Recuerda que el universo de un problema no es una entidad.

- b. Extrae los atributos de cada entidad, identificando los que pueden ser clave. Los atributos suelen ser adjetivos asociados a un nombre identificado como entidad. Por ejemplo, color puede ir asociado a la entidad vehículo. Además, se debe establecer el tipo de datos que almacena y el tipo de atributo, seleccionando si es opcional, obligatorio, multivaluado, compuesto o derivado. Si es compuesto se indica su composición, y si es derivado, cómo se calcula.
 - c. Si hay atributos aplicables a más de una entidad tal vez sea posible establecer una generalización o especialización. Si es el caso, indica cuál es la entidad general y cuáles son las especializaciones.
 - d. Es posible que los nombres comunes contengan muy poca información y no sea posible incluirlas como entidades. En este caso, se pueden seleccionar como atributos de otra entidad. Por ejemplo, el autor de un libro puede ser una entidad, pero si solo se necesita del nombre del autor no tiene sentido tratarlo como una entidad. Mejor se añade como atributo en la entidad Libro.
 - e. Identifica las relaciones. Se pueden ver extrayendo los verbos del texto del problema. Las entidades relacionadas serán el sujeto y el predicado unidos por el verbo que hace de relación. Por ejemplo, agente inmobiliario vende edificio. El agente inmobiliario y el edificio representan entidades y *vende* es la relación.
 - f. Una vez identificadas las relaciones, piensa si entre las entidades relacionadas en la relación existe alguna entidad débil. Si la instancia de una entidad no tiene sentido al desaparecer la instancia de otra entidad, entonces se trata de una entidad débil.
 - g. Identifica los atributos de cada relación. Se suelen distinguir del mismo modo que los de entidad, por ser adjetivos. Recuerda que solo son aplicables a la relación y no a ninguna de las de las entidades relacionadas.
- 3. Averigua las participaciones y cardinalidades a partir de la información recopilada en el requerimiento.
 - 4. Refina el diagrama hasta eliminar todas las dudas e incoherencias. Vuelve a los pasos previos si es necesario. Incluso es posible que tenga que revisarse el requerimiento porque está incompleto.

CASO PRÁCTICO 7

Una empresa necesita una base de datos para almacenar los datos de su negocio. La empresa vende productos a sus clientes por medio de pedidos que éstos realizan. Se necesita conocer de cada cliente su nombre completo, NIF, dirección, número de teléfono y correo electrónico. De cada pedido se necesita la fecha de realización, la dirección de envío y la forma de pago. Cada producto tiene un nombre y un código, así como un precio unitario, y puede añadirse una cantidad variable de producto en cada pedido. La empresa vende productos de alimentación y de droguería. De los productos de alimentación se necesita conocer su fecha de caducidad y a qué temperatura se deben guardar. De los productos de droguería se necesita conocer si son peligrosos. Cada producto se guarda en un único almacén de la empresa. Los almacenes quedan identificados por un código. Los productos son suministrados por diferentes proveedores. Se debe tener en cuenta que un producto sólo puede ser suministrado por un proveedor, pero que un proveedor puede suministrar diferentes productos. Se desea conocer de cada proveedor su NIF, nombre, dirección, correo electrónico y número de teléfono.

Tras analizar el requerimiento, se identifican las siguientes entidades con sus correspondientes atributos:

ENTIDAD	ATRIBUTOS
Cliente	NIF, nombre completo, dirección, teléfono, email
Pedido	Fecha de realización, dirección de envío, forma de pago
Producto	Código, nombre, precio unitario.
Producto de alimentación	Código, nombre, precio unitario, temperatura, fecha de caducidad.
Producto de droguería	Código, nombre, precio unitario, ¿es peligroso?
Almacén	Código de almacén
Proveedor	NIF, nombre, dirección, teléfono, email

Y las siguientes relaciones:

- Realiza (un cliente realiza muchos pedidos. Un pedido es realizado por un cliente).
- Contiene (un pedido contiene muchos productos, y viceversa). Se identifica un atributo de relación: la cantidad del producto en el pedido.
- Suministra (un proveedor suministra muchos productos. Un producto es suministrado por un único proveedor)

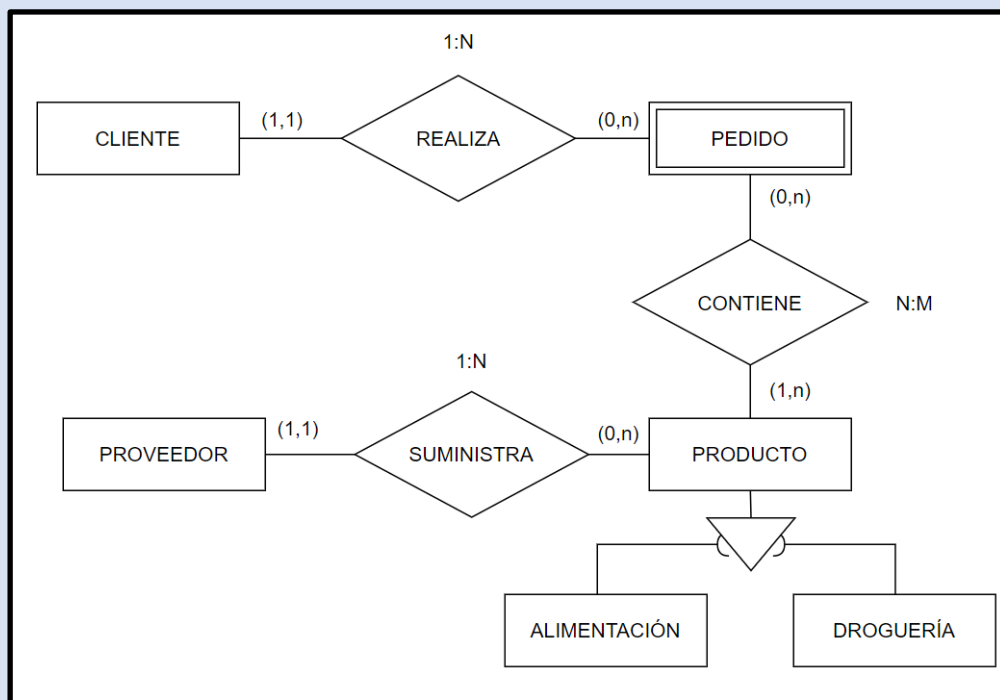
Examinando con más detalle las entidades, se descubre que:

- El atributo nombre completo del cliente es realmente un atributo compuesto de nombre y apellidos.
- Pedido es una entidad débil. Su existencia sólo tiene sentido si existe el cliente.
- Producto de alimentación y producto de droguería son especializaciones de la entidad genérica producto. La especialización es exclusiva.
- Aunque cliente y proveedor comparten atributos, no se crea una entidad genérica para guardar esos atributos ya que no tiene sentido mezclar clientes con proveedores.
- Del almacén sólo se dice que guarda productos y está identificado por un código. Quizás no sea necesario crear una entidad para el almacén sino guardar ese dato como atributo del producto.

Realizadas estas observaciones, las entidades y sus atributos son los siguientes (se subraya su atributo clave):

ENTIDAD	ATRIBUTOS
Cliente	<u>NIF</u> , nombre, apellidos, dirección, teléfono, email
Pedido	<u>Fecha de realización</u> , dirección de envío, forma de pago NOTA: se necesita también la clave del cliente para identificar de forma completa el pedido.
Producto	<u>Código</u> , nombre, precio unitario, almacén
Producto de alimentación	Atributos de producto más temperatura, fecha de caducidad.
Producto de droguería	Atributos de producto más ¿es peligroso?
Proveedor	<u>NIF</u> , nombre, dirección, teléfono, email

Por lo que el diagrama E/R queda del siguiente modo:



4. Transformación del diagrama E/R a modelo lógico

Una vez obtenido el diagrama E/R, el siguiente paso es conseguir el **modelo lógico** de los datos. Consiste en la representación del diagrama E/R en forma de relaciones o tablas. Como se va a trabajar con bases de datos relacionales, también suele llamarse modelo relacional.

RECUERDA

Una base de datos relacional organiza los datos en forma de relaciones o tablas. Las tablas están formadas por tuplas (filas) y campos (columnas). Además, cada tabla debe incluir uno o varios campos clave que permitan identificar las tuplas de la tabla.

Cada entidad incluida en el diagrama E/R debe tener un atributo clave, que puede ser un simple atributo o la combinación de varios atributos. Por ejemplo, para identificar a una persona basta su atributo NIF, pero para identificar un aula de un instituto con dos edificios y donde cada edificio tiene su propia numeración, se necesitan dos atributos: edificio y número de aula.

El atributo clave de la entidad será **clave principal** en la base de datos. En el caso en que la entidad tenga más de un atributo clave, cada uno de ellos es una **clave candidata** y es necesario elegir de entre todas ellas una para que se convierta en clave primaria. Por ejemplo, un alumno puede ser identificado por su NIF o por su número de expediente. Tiene por tanto dos claves candidatas y en el diseño de la base de datos se debe decidir cuál de las dos claves se convierte en la clave primaria.

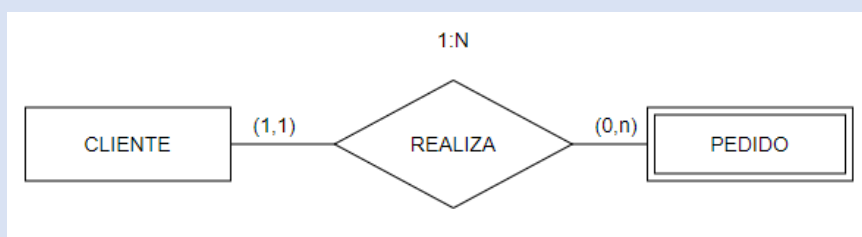
En el diseño de la base de datos puede ser necesario que en una tabla se haga referencia a los datos de otra tabla. Cuando eso sea necesario, se debe añadir en la tabla que referencia la clave primaria de la tabla referenciada, convirtiéndose así en **clave foránea**. Por ejemplo, un cliente realiza pedidos y en el pedido es necesario hacer referencia al cliente. Por tanto, en la base de datos los datos de pedido incluirán como clave foránea la clave primaria del cliente que realiza el pedido.

4.1. Transformación de entidades

Toda entidad se transforma en una tabla cuyos campos son los atributos de la entidad. La clave primaria es el atributo clave de la entidad o una de sus claves candidatas. Si la entidad es débil, además debe añadirse como clave foránea la clave primaria de la entidad fuerte de la que depende.

CASO PRÁCTICO 8

A partir del siguiente diagrama E/R:



Del cliente se necesita su NIF, nombre y apellidos, dirección, email y teléfono. Del pedido la fecha de realización, dirección de envío del pedido y forma de pago. La entidad Pedido es débil y debe hacer referencia al cliente que realiza el pedido. El modelo lógico resultante (centrándose únicamente en las entidades) es el siguiente:

CLIENTE(NIF, nombre, apellidos, dirección, email, teléfono)

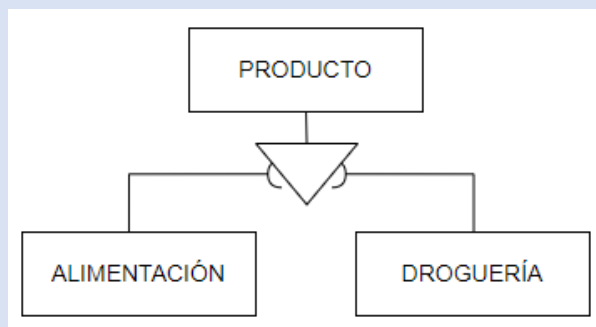
PEDIDO(NIFCliente, fecha, dirección de envío, forma de pago)

En el caso de generalizaciones/especializaciones, se puede transformar de cuatro formas distintas:

1. Se crea una tabla para la entidad general, que incluye todos los atributos comunes, y una tabla para cada especialización, que incluye la clave primaria de la entidad general y los atributos específicos de la especialización.
2. Se crea una tabla para cada especialización con todos los atributos comunes más los específicos, y no se crea tabla para la entidad general. No es adecuada para especializaciones inclusivas.
3. Se crea una tabla donde se incluyen los atributos generales y todos los atributos específicos de cada especialización. Además, un campo para indicar el tipo de especialización que representa los datos. Esta solución solo es viable para especializaciones exclusivas.
4. Se crea una única tabla como en la solución anterior, pero en lugar de añadir un campo tipo se añaden tantos como especializaciones haya de tal modo que cada uno de esos campos indica si los datos cumplen con la especialización asociada. Esta opción soporta especializaciones inclusivas.

CASO PRÁCTICO 9

A partir del siguiente diagrama E/R:



Del producto se necesita su código, nombre, precio unitario y almacén. Del producto de alimentación además temperatura de conservación y fecha de caducidad. Del producto de droguería si es peligroso. El modelo lógico resultante es el siguiente:

Según la opción 1

PRODUCTO(código, nombre, precio, almacén)

ALIMENTACIÓN(código, temperatura, fecha caducidad)

DROGUERÍA(código, esPeligroso)

Según la opción 2

ALIMENTACIÓN(código, nombre, precio, almacén, temperatura, fecha caducidad)

DROGUERÍA(código, nombre, precio, almacén, esPeligroso)

Según la opción 3

PRODUCTO(código, nombre, precio, almacén, temperatura, fecha caducidad,
esPeligroso, tipoProducto)

Según la opción 4

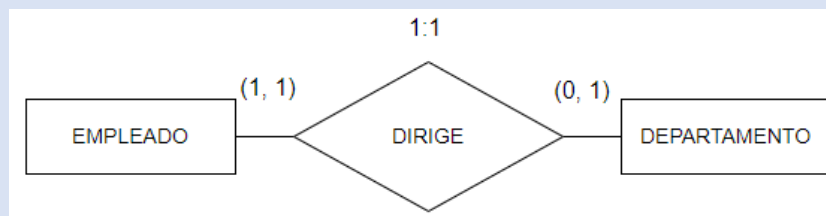
PRODUCTO(código, nombre, precio, almacén, temperatura, fecha caducidad,
esPeligroso, esAlimentación, esDroguería)

4.2. Transformación de relaciones

Todas las relaciones se pueden transformar, de forma general, en una tabla que contenga todas las claves primarias de las entidades que intervienen en la relación más los atributos propios de la relación. Para las relaciones N:M o de grado igual o superior a 3 es el único modo posible para transformar la relación en tabla. Sin embargo, para las relaciones 1:1 o 1:N existe otra opción:

- Para las relaciones 1:1, se puede trasladar la clave primaria de una entidad y los atributos de la relación a la otra entidad. No obstante, si la participación es (0,1) para ambas entidades, se puede optar por la solución general para evitar la propagación de muchos campos con valor nulo.
- Para las relaciones 1:N, se puede trasladar la clave primaria de la entidad con cardinalidad N y los atributos de la relación a la entidad con cardinalidad 1.

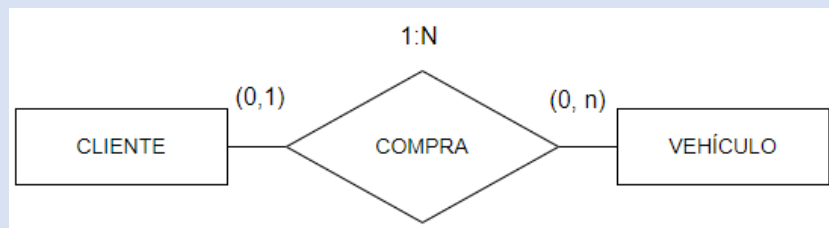
CASO PRÁCTICO 10



Relación 1:1, se puede trasladar a tablas así (en negrita, campos incorporados):

EMPLEADO(NIF, nombre, apellidos, email)

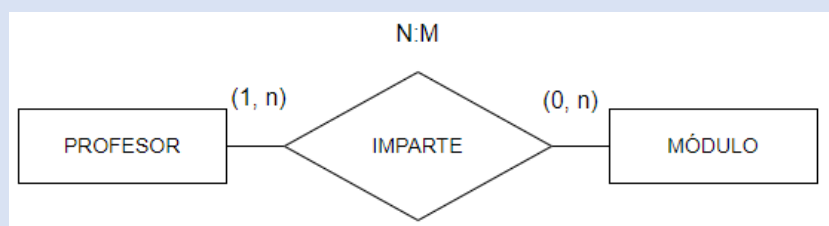
DEPARTAMENTO(código, despacho, **NIF**)



Relación 1:N, se puede trasladar así:

CLIENTE(NIF, nombre, apellidos, teléfono)

VEHÍCULO(matrícula, potencia, combustible, color, **NIF**, **fecha de compra**)

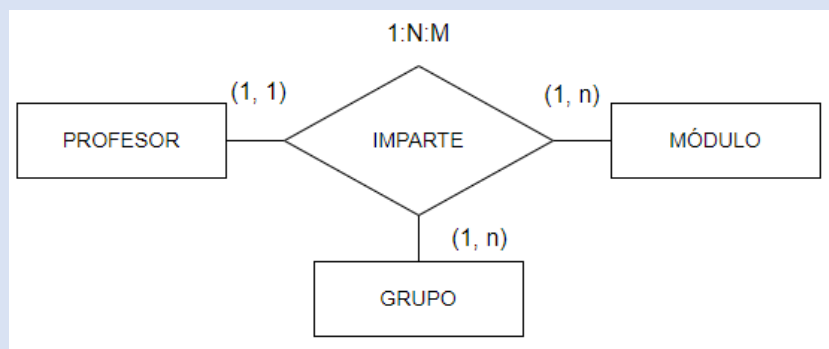


Relación N:M, es necesario crear una nueva tabla:

PROFESOR(NIF, nombre, apellidos, teléfono, email)

MÓDULO(código, descripción, horas a la semana)

IMPARTE(NIF, código, aula)



Relación ternaria, es necesario crear una nueva tabla:

PROFESOR(NIF, nombre, apellidos, teléfono, email)

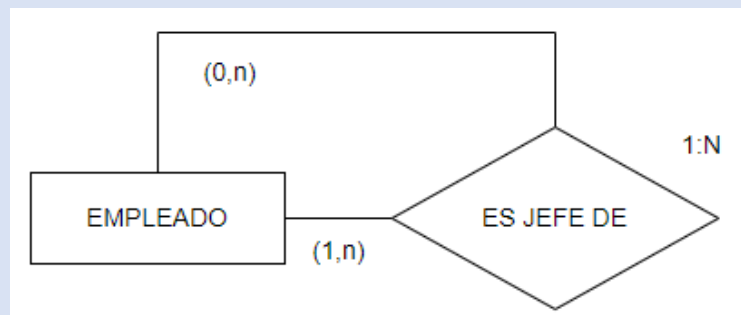
MÓDULO(código, descripción, horas a la semana)

GRUPO(código, nombre, curso)

IMPARTE(NIF, código módulo, código grupo, aula)

En los ejemplos previos en los campos que son clave foránea se ha mantenido el nombre de la clave primaria para facilitar la comprensión, pero no es necesario y se puede

cambiar el nombre. En este último ejemplo es necesario hacerlo para poder distinguir el código de módulo del código de grupo.



Relación 1:N reflexiva, aquí la clave foránea debe cambiar de nombre para distinguirla de la clave primaria:

EMPLEADO(NIE, nombre, apellidos, email, jefe)

5. Normalización

Lo más común es que el diseño de una base de datos finalice en el paso del modelo E/R al modelo lógico. No obstante, si el análisis conceptual no es bueno (o durante el diseño de la base de datos se obtiene el modelo lógico directamente sin realizar el análisis conceptual), es muy probable que la calidad del modelo lógico resultante sea baja.

Siempre que se diseña un sistema informático se ha de medir su calidad. Si el diseño no cumple determinados criterios de calidad conviene realizar, de forma iterativa, sucesivos refinamientos en el diseño hasta alcanzar la calidad deseada. En el diseño de una base de datos es importante alcanzar un diseño de calidad que permita aprovechar las ventajas que aporta una base de datos y evitar problemas de redundancia, inconsistencia o anomalías en las consultas.

CASO PRÁCTICO 11

Imagina que después de un mal diseño se obtiene una tabla para guardar productos de un pedido como la siguiente:

PRODUCTO(cód pedido, NIF, nombre, fecha, cód producto, descripción, precio, cantidad)

Este diseño plantea los siguientes problemas:

- Para cada producto de un pedido se repiten los datos de cliente (NIF y nombre), del pedido (fecha). Esta información repetida desperdicia espacio en la base de datos.
- La descripción del producto está también incluida en la tabla. Si este atributo del producto cambia, sería necesario actualizar todas las filas de pedido donde aparezca el producto, complicando así las actualizaciones de datos.
- En la actualización de datos repetidos, basta olvidar la actualización de una de las redundancias para introducir inconsistencias en la base de datos.

El proceso de refinamiento del modelo lógico se llama normalización. La **normalización** consiste en la aplicación de una serie de reglas que deben cumplir todas las tablas del modelo lógico con el objetivo de:

1. Almacenar cada dato solo una vez evitando la redundancia de datos.
2. Almacenar datos distintos en sitios distintos evitando anomalías a la hora de operar con los datos.

Toda tabla que no cumpla una regla de normalización debe ser revisada. El proceso de revisión consiste en fragmentar la tabla, es decir, separar los campos en distintas tablas hasta que se cumpla la regla. Existen seis reglas de normalización o **formas normales**, cada forma es más restrictiva que la anterior, y son las siguientes:

- Primera forma normal o 1FN.
- Segunda forma normal o 2FN
- Tercera forma normal o 3FN.
- Forma normal de Boyce-Codd o FNBC.
- Cuarta forma normal o 4FN.
- Quinta forma normal o 5FN.

OBSERVACIÓN

En este documento sólo se explican las tres primeras formas normales por tres razones:

1. En la práctica, es suficiente con normalizar una base de datos hasta 3FN o FNBC.
2. En la mayoría de casos una tabla en 3FN también lo está en FNBC.
3. Aplicar las reglas para normalizar la base de datos hasta 4FN o 5FN puede suponer fragmentar tanto la información que penaliza el rendimiento de las consultas.

Las reglas de normalización se basan en la **dependencia funcional** de los campos de la tabla. Si dice que un campo B depende funcionalmente de otro campo A, y se representa $A \rightarrow B$ (A implica B), si cualquier valor de A tiene asociado un único valor de B. Por ejemplo, la descripción de un producto depende funcionalmente de su código (el código implica la descripción) y se representa $\text{CódigoProducto} \rightarrow \text{DescripciónProducto}$.

La descripción de las formas normales se resume en la siguiente tabla:

FORMA NORMAL	DESCRIPCIÓN
1FN	Una tabla está en 1FN si cada campo almacena un único valor de su dominio.
2FN	Una tabla está en 2FN si está en 1FN y además cada campo que no pertenece a la clave primaria tiene dependencia funcional con toda la clave primaria y no con parte de ella.
3FN	Una tabla está en 3FN si está en 2FN y además no hay dependencias funcionales transitivas. Es decir, no hay campos con dependencias funcionales con un campo distinto de la clave primaria.

CASO PRÁCTICO 12

Se va a normalizar hasta 3FN el diseño del caso práctico 11:

PRODUCTO(cód pedido, NIF, nombre, fecha, cód producto, descripción, precio, cantidad)

1FN

Imagina que los campos de producto almacenan más de un valor (ya que un pedido puede incluir más de un producto). Un pedido podría almacenarse del siguiente modo (centrándonos sólo en los valores clave):

<u>cód pedido</u>	<u>cód producto</u>
1	10
	2
	4

La tabla no está en 1FN porque hay campos que almacenan más de un valor. Hay que cambiar la tabla para que almacene los datos del siguiente modo:

<u>cód pedido</u>	<u>cód producto</u>
1	10
1	2
1	4

2FN

Una vez puesta en 1FN hay que comprobar si está en 2FN. Para ello los campos que no son clave deben depender de toda la clave y no sólo de parte de ella:

PRODUCTO(cód pedido, NIF, nombre, fecha, cód producto, descripción, precio, cantidad)

Se detectan tres dependencias funcionales con parte de la clave:

cód producto → descripción

cód producto → precio

cód pedido → fecha

Por tanto, la tabla no está en 2FN y hay que fragmentar la tabla. En este caso la información se divide en tres tablas que sí cumplen 2FN.

PRODUCTO(cód producto, descripción, precio)

PEDIDO(cód pedido, NIF, fecha)

DETALLE_PEDIDO(cód pedido, NIF, nombre, cód producto, cantidad)

3FN

Una vez puesta en 2FN hay que comprobar si está en 3FN. Para ello no puede haber campos que dependan funcionalmente de un campo que no es clave. Las tablas PRODUCTO y PEDIDO ya están en 3FN, pero DETALLE_PEDIDO no porque NIF → nombre. Por tanto, la tabla DETALLE_PEDIDO debe fragmentarse y el resultado en 3FN es el siguiente:

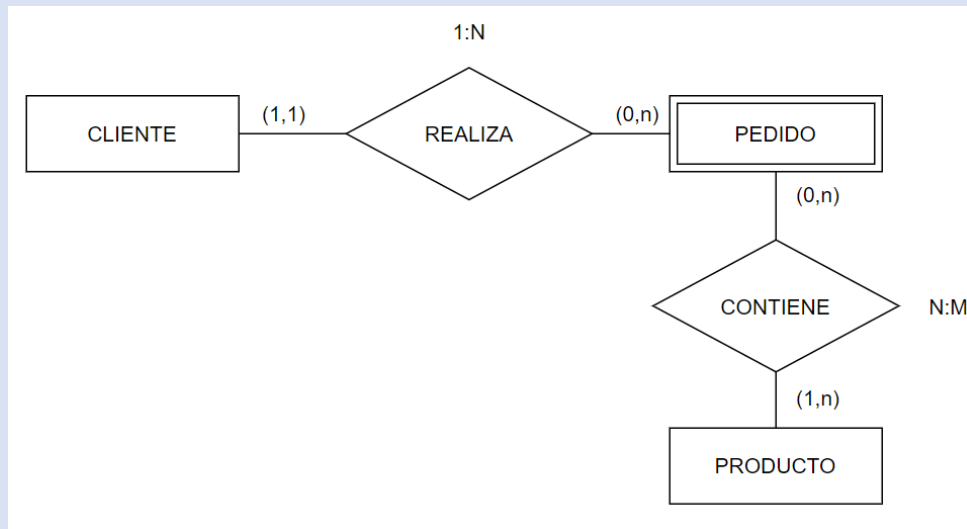
PRODUCTO(cód producto, descripción, precio)

PEDIDO(cód pedido, NIF, fecha)

CLIENTE(NIF, nombre)

DETALLE_PEDIDO(cód pedido, cód producto, cantidad)

Que es precisamente el resultado que se obtiene al aplicar las reglas de transformación del diagrama E/R a modelo lógico.



6. Resumen

Todo desarrollo de software parte de un análisis estructurado y una parte importante de este proceso es la descripción de los datos. Aunque el diccionario de datos incluido en la base de datos ofrece toda la información sobre el contenido almacenado, el primer paso de diseño de dicha base de datos es la elaboración del diagrama E/R.

El diagrama E/R describe el universo de datos que se va a informatizar como un conjunto de entidades relacionadas entre sí. Las distintas instancias de entidades interactúan con otras entidades a partir de las relaciones que las conectan, y la cantidad de instancias implicadas vendrá determinada por las participaciones y cardinalidades impuestas por las reglas de negocio.

Entidades y relaciones quedan caracterizadas por un conjunto de atributos, es el lugar donde se almacenan los datos. Las propiedades de cada atributo determinan el dominio de datos que puede almacenar.

A partir del modelo conceptual representado en el diagrama E/R se puede obtener, aplicando unas sencillas reglas de transformación, el modelo lógico de la base de datos. Se trata de una representación más cercana a la base de datos donde se reflejan las tablas a crear con sus correspondientes campos.

Si el diagrama E/R está bien diseñado, la calidad del modelo lógico resultante será buena y por tanto la base de datos que se construya a partir del modelo. En cualquier caso, conviene aplicar reglas de normalización para garantizar la calidad del diseño resultante. Una base de datos mal diseñada conlleva problemas de redundancia, inconsistencias, lentitud de consultas o dificultades para añadir o modificar datos.