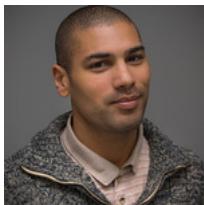


Diseño lógico de Bases de Datos.

Caso práctico



Alain Bachelier (CC BY-NC-SA)

En la unidad anterior habíamos visto como **Juan**, el director del departamento de informática de **BK Sistemas Informáticos**, encargaba a **Vindio** y **Noiba** de la informatización de un taller mecánico, de manera que habían estado estudiando distintas opciones posibles de almacenamiento de la información, antes de decidirse por implantar un S.G.B.D.

Antes de elegir uno de los SGBD que existen en el mercado tiene claro que debe diseñar un modelo flexible que dé respuesta a las necesidades actuales de gestión del taller, pero tiene que considerar también las posibilidades de crecimiento a medio y largo plazo.

Vindio, que lleva ya varios años trabajando en la empresa y conoce perfectamente el funcionamiento de los sistemas, recuerda a **Noiba** que este proceso requiere un diseño lógico, previo al diseño físico y a la implantación definitiva, y que debe hacerse teniendo en cuenta todas las especificaciones, características e información sobre el funcionamiento del taller para que pueda dar respuesta a las necesidades actuales y futuras. En definitiva vamos a ayudar a **Vindio** y **Noiba** a construir un MODELO DE DATOS que represente la realidad de ese Taller mecánico, y aprenderemos con ellos a realizar el diseño lógico para, en el tema siguiente, referirnos al diseño físico.

El modelado de datos es el primer paso para diseñar las bases de datos, y sirve de puente entre el mundo real y el modelo de base de datos que reside en el ordenador. Desempeña un papel fundamental ya que reduce las complejidades del mundo real y las convierte en abstracciones más fáciles de entender.

Un modelo de datos es la representación relativamente simple, generalmente gráfica, de estructuras de datos complejas del mundo real. Con relación a las bases de datos un modelo representa estructuras de datos y sus características, relaciones, restricciones y transformaciones.

Un buen diseño de la base de datos es el fundamento de buenas aplicaciones, y un buen diseño se inicia con la construcción de un buen modelo.

1.- Modelos de datos.

Caso práctico



[Juan Goldstein \(CC BY\)](#)

Para empezar a diseñar la base de datos del taller mecánico, **Juan** recuerda a **Vindio** y **Noiba** que se debe empezar por elegir un modelo que permita representar la realidad del taller: los elementos que intervienen y las relaciones entre ellos.

Como hemos visto, en la unidad anterior, existen distintos modelos tanto a nivel lógico como conceptual. A continuación veremos cuál es la elección más apropiada.

En el capítulo anterior hemos visto que una base de datos almacena datos mientras que el SGBD administra y controla el acceso a los datos y a la base de datos.

Al estar el SGBD entre la aplicación y la base de datos se logra eliminar la mayoría de las limitaciones del sistema a cambio de tener una estructura física más compleja, sobre todo en **los modelos jerárquicos y en red**, que llegaron a tener unas estructuras físicas tan complicadas que restaban importancia al diseño eficaz de la base de datos.

El **modelo de base de datos relacional** cambia esto, permitiendo al diseñador centrarse en la representación lógica de los datos y sus relaciones, en vez de ocuparnos sólo del almacenamiento físico.

Por ejemplo: Refiriéndonos a un ejemplo de automóviles es como si la base de datos relacional utilizara transmisión automática y nos libera de tener que utilizar la palanca de cambios mientras pisamos el embrague.

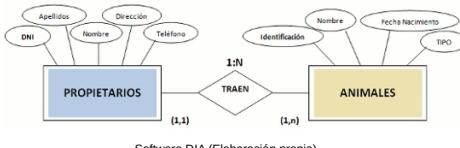
Aunque el modelo relacional supuso una considerable mejora sobre los modelos anteriores carecía de las características necesarias para ser una herramienta eficaz para diseñar bases de datos. Es más fácil examinar las estructuras gráficamente que describirlas como texto, por tanto se buscaba un modelo de datos conceptual de carácter gráfico.

El modelo Entidad/Relación (E/R, E-R, o ER) es el modelo conceptual, de carácter gráfico, más aceptado y que mejor se adapta al diseño de las bases de datos relacionales.

Este modelo no tiene en cuenta la implementación física de los datos, solo interesa el nivel conceptual, en cuanto a su aplicación práctica; el diseño de la base de datos se hace empleando este modelo y para implementarlo en un SGBDR se traduce al modelo relacional como modelo de diseño a nivel lógico.

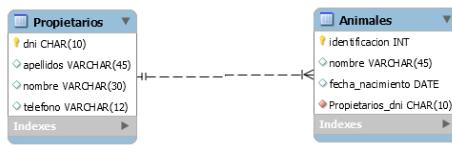
Veamos un **ejemplo del modelo E/R** y el **modelo relacional** correspondiente a una representación de la relación que existe entre los propietarios que traen a sus animales a una clínica veterinaria.

Modelo E/R Veterinaria



Software DIA (Elaboración propia)

Modelo Relacional de Veterinaria



Workbench (Elaboración propia)

Esta representación gráfica fue la que popularizó el uso de los diagramas E/R como herramienta de modelado de datos a nivel conceptual y complementó los conceptos de modelo de datos relacional, con lo que se establecieron las bases de un diseño de bases de datos bien estructuradas que garanticen el diseño apropiado de datos relacionales.

Autoevaluación

Rellena en el párrafo siguiente los espacios en blanco con las palabras que consideres apropiadas:

Para diseñar una base de datos bien estructurada actualmente se aplica el modelo como modelo de datos a nivel . Una vez obtenida la representación gráfica, se traduce al modelo que es el modelo que implementan la mayoría de los SGBD como modelo a nivel .

El modelo entidad/relación representa la realidad de una forma gráfica, mientras que el modelo relacional está más próximo al SGBD.

2.- Modelo Entidad-Relación.

Caso práctico

Noiba y Vindio comienzan a estudiar y analizar algunos de los objetos que intervienen en la actividad cotidiana del Taller mecánico, para considerar y decidir de cuáles de ellos interesa guardar información, como por ejemplo pueden ser:

- los clientes del taller.
- los vehículos que los clientes nos traen a reparar.
- la orden de reparación que se rellena antes de enviar el vehículo al taller.
- las facturas que se emiten con posterioridad a la reparación, etc.

Como veremos en esta unidad cada uno estos elementos se recogerán en nuestro modelo como una entidad y estudiaremos las relaciones o inter-relaciones que se dan entre estos objetos.



Alain Bachelier (CC BY-NC-SA)

El **modelo Entidad/Relación** lo introdujo Peter Chen en 1976 y se trata de un modelo conceptual que permite representar el mundo real mediante una serie de símbolos y expresiones determinados. El modelo de datos **Entidad/Relación (E/R , ER ó E-R)** está basado en una percepción consistente en objetos básicos llamados **entidades** y las **relaciones** entre estos objetos. Con el paso del tiempo, este modelo ha sufrido modificaciones y mejoras. Actualmente, el modelo **Entidad/Relación Extendido (ERE)** es el más aceptado, aunque existen variaciones que hacen que este modelo no sea totalmente un estándar. Ambos modelos serán estudiados a lo largo de esta unidad.

El modelo Entidad/Relación es una herramienta de referencia para la representación conceptual de problemas del mundo real. Su objetivo principal es facilitar el diseño de 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.

Se pretende 'visualizar' los objetos que pertenecen a la Base de Datos como **entidades** las cuales tienen unos **atributos** y se vinculan mediante **relaciones**.

Para saber más

Una de las cosas más importantes a la hora de establecer el modelo E/R que mejor se adapta a una situación del mundo real es establecer correctamente cada uno de sus elementos: las entidades, sus atributos, las relaciones que existen entre las entidades, la clave principal, las restricciones, etc. Aunque no existe un método propiamente dicho podemos seguir una serie de pasos que nos ayudarán a establecer nuestro diagrama E/R.

[Descargar Pasos para establecer un modelo E/R\[pdf - 0.12 Mb\]](#)

Autoevaluación

Sobre el modelo Entidad/Relación, selecciona las opciones que consideres correctas:

- En este modelo cada objeto del que queremos guardar información será una entidad y las asociaciones entre esos objetos serán las relaciones.

- Cuando en una entidad no exista una clave primaria elegiremos uno cualquiera de sus atributos.

- Se utiliza en una fase del diseño anterior al modelo relacional porque ambos se integran con facilidad.

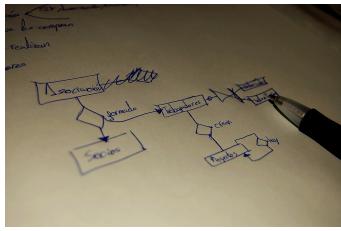
- Está muy aceptado, tanto por la facilidad de comprensión que ofrece su representación visual, como por los comandos para la manipulación de los datos.
-
- Antes de elegir el modelo E/R para nuestra base de datos necesitamos saber en qué sistema lo vamos a implementar y cuál será el soporte de almacenamiento.
-

[Mostrar retroalimentación](#)

Solución

1. Correcto
2. Incorrecto
3. Correcto
4. Incorrecto
5. Incorrecto

2.1.- Entidades.



ElFisogn (CC BY-NC-ND)

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 en nuestra base de datos. Cada entidad debe poseer alguna característica, o conjunto de ellas, que la haga única frente al resto de objetos. Por ejemplo, podemos establecer una entidad llamada **CLIENTE** que tendrá una serie de características, como número de identificación fiscal, nombre, apellidos dirección, etc. Cada cliente podría ser distinguido mediante su documento nacional de identidad (<abbr title="Documento Nacional de Identidad">DNI</abbr>), por ejemplo.

Entidad: objeto real o abstracto, con características diferenciadoras capaces de hacerse distinguir de otros objetos.

Un **conjunto de entidades** serán un grupo de entidades que poseen las mismas características o propiedades. Por ejemplo, al conjunto de clientes que realizan reservas o citas para un taller mecánico, se les puede definir como el conjunto de entidades **CLIENTE**. El conjunto de entidades **ALUMNO**, podría representar a todo el alumnado existente en un determinado Instituto.

Por lo general, se suele utilizar el término **entidad** para identificar conjuntos de entidades, y a cada elemento del conjunto de entidades se le denomina **ocurrencia, ejemplar o instancia** de esa entidad. Así por ejemplo, ocurrencias de la entidad **CLIENTE** pueden ser Pedro Salas Lara con <abbr title="documento nacional de identidad">dni</abbr> 27445334, Alberto Liz Rubira con dni 34567123, Rosa Relaño Sanz con dni 35456123, etc.

¿Y cómo se representa de forma gráfica una entidad?

Una **entidad se representa** por medio de un rectángulo con el nombre dentro del recuadro. El nombre suele ser un sustantivo en mayúscula.

Por ejemplo la entidad **CLIENTE** se representaría de la siguiente forma:



Existen dos **clases o tipos de entidades**:

- Entidades Fuertes o Regulares.
- Entidades Débiles.

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 de un Centro Educativo, la existencia de instancias concretas de la entidad **ALUMNO** no depende de la existencia de instancias u objetos de la entidad **CURSO**. 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.

Por ejemplo: La entidad **CLIENTE** es una entidad fuerte pues no depende de otras entidades para existir.

Entidades Débiles.

Son aquellas cuya existencia depende de la existencia de otras instancias de entidad. Una entidad puede ser fuerte o débil respecto de otras.

Por ejemplo: La entidad **FACTURA** será débil respecto a la entidad **CLIENTE** porque no existen facturas que no correspondan a un cliente, pero será fuerte respecto a la entidad **LINEA** refiriéndonos a las líneas de una factura.

Las entidades débiles se representan con un doble rectángulo:



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.

Un ejemplo de Entidad Débil en identificación sería la entidad **EJEMPLAR**, si consideremos las entidades **LIBRO** y **EJEMPLAR** en el siguiente contexto: Supongamos que los ejemplares de cualquier libro se numeran siempre con números del 1 en adelante (1,2,3...). La numeración de cada ejemplar no identificará completamente cada uno de ellos. Para poder identificar completamente un ejemplar es necesario saber también a qué libro pertenece. 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.

Reflexiona

Cada uno de los elementos que incluye la entidad se denomina ocurrencia de entidad o instancia de entidad. Un conjunto de entidades semejantes se conoce como conjunto de entidades. Por ejemplo el conjunto de los 5 empleados del taller mecánico constituyen el conjunto de entidades CLIENTE. En realidad se debería hablar de conjunto de entidades, pero en la práctica se utiliza la palabra entidad para referirnos al conjunto de entidades aunque no sea muy correcto, por tanto hablaremos de la entidad CLIENTE en lugar del conjunto de entidades CLIENTE. Del mismo modo el nombre de la entidad se debería escribir en mayúsculas y en singular, pero es frecuente utilizar el plural en muchos ejemplos

CLIENTE

Libre Office (Elaboración propia.)

Autoevaluación

Solo una de las siguientes opciones muestra una entidad débil respecto a otra. Marca la opción que consideres acertada.

- Relación PROVEEDORES suministran ARTICULOS
- Relación ALUMNOS tienen CALIFICACIONES
- La relación ALUMNOS se matriculan en ASIGNATURAS

Un artículo tiene existencia por sí mismo, independientemente del proveedor que nos lo suministra en cada momento.

No existirá una calificación de un alumno que no esté matriculado. La entidad calificaciones necesita de la entidad alumnos para existir.

La existencia de cada asignatura es independiente de que haya o no alumnos matriculados.

Solución

1. Incorrecto
2. Opción correcta
3. Incorrecto

2.2.- Atributos.



¿Cómo guardamos la 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.

By Everaldo Coelho and YellowIcon (GNU/GPL)

Atributo: es cada una de las propiedades o características que tiene un tipo de entidad o un tipo de relación. Los atributos toman valores de un dominio..

Por tanto, un atributo se utilizará para guardar información sobre alguna característica o propiedad de una entidad o relación.

Por ejemplo: La entidad CLIENTES tendrá atributos como: Código de Cliente, DNI, Apellidos, Nombre, Dirección, Teléfono.

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 CLIENTE.



DIA (Elaboración propia)

Dominio de un atributo : Es el conjunto de los posibles valores que ese atributo puede poseer. 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, apellidos y dirección de la entidad CLIENTE, están definidos dentro del dominio de cadenas de caracteres de una determinada longitud. El dominio del atributo teléfono estaría dentro del conjunto de combinaciones de 9 cifras que sean números de teléfonos válidos o una cadena de 9 caracteres con las combinaciones de dígitos válidas. El dominio del atributo sexo de un cliente sólo tiene dos valores posibles M o F,etc.

Podemos diferenciar distintos **tipos de atributos** en base a diferentes criterios.

Atributosopcionales y obligatorios.

Atributo obligatorio: es aquél que ha de estar siempre definido para una entidad o relación. Por ejemplo, para la entidad CLIENTE 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. Por ejemplo, el atributo teléfono puede ser opcional.

Atributos compuestos y simples.

Un **atributo compuesto** es un atributo que se puede subdividir en otros. Por ejemplo, el atributo dirección se puede considerar compuesto si diferenciamos las distintas partes en la que puede dividirse: calle, número, localidad, provincia y código postal.

Un **atributo simple** no se puede subdividir. Se representa tal y como hemos visto hasta ahora. Por ejemplo, un atributo simple puede ser el teléfono.

La forma de representar los atributos compuestos varía en función de los autores.

Un ejemplo de representación gráfica de un atributo compuesto, como por ejemplo dirección, sería:



DIA (Elaboración propia)

Atributos monovaluados y multivaluados.

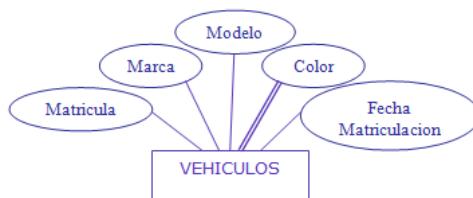
Atributos monovaluados o de un solo valor son los que pueden tener un único valor,

Por ejemplo: El DNI, el número de la seguridad social, el nombre de un cliente, etc. Eso no significa que sea un atributo simple, por ejemplo el número de habitación de un hotel se puede dividir: el primer dígito es la planta, los dos siguientes corresponden a la habitación dentro de la planta.

Atributos multivaluados o de valores múltiples son aquellos que pueden tener muchos valores.

Por ejemplo: El teléfono de un cliente (puede tener uno o más números de teléfono: varios fijos, varios móviles, etc.), el color de un coche a veces puede tomar distintos valores: color de la carrocería, de las molduras y del techo, etc.

Ejemplo de representación de un atributo de valores múltiples, como color:



DIA (Elaboración propia)

A menudo nos podemos encontrar con esta otra forma de representar atributos múltiples:



DIA (Elaboración propia)

Aunque en el modelo E/R se pueden manejar atributos con valores múltiples, esto no pueden ejecutarse en un SGBD relacional. Por tanto habrá que adoptar una de estas soluciones:

Crear atributos nuevos para cada uno de los componentes de atributo múltiple.

Crear una entidad nueva que contenga los componentes del atributo de valores múltiples.

Atributo derivado.

Un **atributo derivado** es aquel que se puede deducir u obtener a partir de otro u otros atributos. Puesto que se puede calcular, no se guarda en la base de datos.

Por ejemplo: La edad de una persona, que se puede calcular a partir de la fecha de nacimiento y la fecha del sistema. El importe de un pedido que se puede obtener multiplicando precio de cada unidad * unidades pedidas.

Una de las formas de representar a un atributo derivado es mediante una elipse punteada:



DIA (Elaboración propia)

Autoevaluación

Indica si la siguiente afirmación es verdadera o falsa.

Para una entidad como por ejemplo un ALUMNO, de quien se tiene como atributos su DNI, nombre, dirección, fecha de nacimiento, email y edad. El atributo edad es un atributo multivaluado.

- Verdadero Falso

Falso

El atributo edad se puede obtener a partir de su fecha de nacimiento, luego es un atributo derivado.

2.2.1.- Claves.



Everaldo Coelho and
YellowIcon (GNU/GPL)

Una de las restricciones del modelo E/R que comentábamos al principio de este apartado era que cada ocurrencia de entidad se debe poder identificar de forma única mediante un atributo identificador, **clave o llave**. Ahora es el momento de abordar con mayor detalle este concepto.

Está claro que es necesario identificar correctamente cada ocurrencia de entidad, 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. Pero, en algunas ocasiones, sabemos que puede no ser necesario utilizar todos, bastando con un subconjunto de ellos. Aunque puede ocurrir que ese subconjunto tenga idénticos valores para varias entidades, por lo que cualquier subconjunto no será válido.

Por tanto, los valores de los atributos de una entidad deben ser tales que permitan **identificar únicamente** a cada ocurrencia de 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 (Superkey): 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 identifican por si solas 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. Es cada una de las claves que está formada por el mínimo número de atributos posibles.

Clave primaria o principal (Primary Key): También llamada llave primaria o clave principal. De todas las claves candidatas, el diseñador de la base de datos ha de escoger una, que se denominará clave principal o clave primaria. La clave primaria es un atributo o conjunto de ellos, que toman valores únicos y distintos para cada ocurrencia de entidad, identificándola únicamente. Lo ideal es que la clave primaria esté formada por un solo atributo.

Debe cumplir:

- No puede contener valores nulos.
- Ha de ser sencilla de crear.

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

Por ejemplo: En la entidad CLIENTES podríamos encontrar:

Superclaves: el código de cliente + nombre, el conjunto apellidos + nombre+ el DNI, etc.

Claves candidatas: el código y el DNI del cliente.

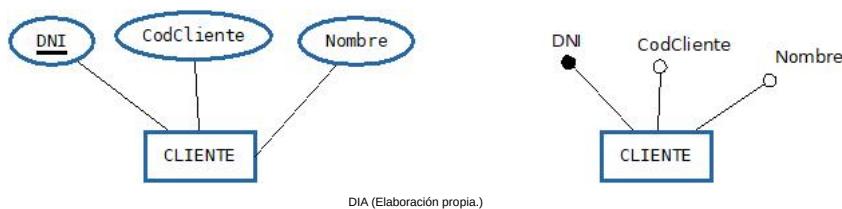
Clave principal: se puede elegir entre las candidatas, por ejemplo el DNI.

Claves alternativas: en este ejemplo sería el código del cliente, la candidata que no se ha elegido como clave principal.

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.



Clave artificial: es un atributo creado por el equipo de diseño de la base de datos, cuyo contenido es aleatorio o secuencial, no repetitivo. Al ser un atributo único, sirve para crear claves primarias más sencillas que una formada por la unión de varios atributos. En su creación se ha de tener en cuenta que ha de ser única e invariable. Suelen tener nombres como: código, identificador, número de....

Por ejemplo: Queremos recoger los datos de nuestra agenda de teléfonos y sabemos que puede haber teléfonos repetidos para miembros de la misma familia: Nombre, apellidos y teléfono. La clave principal podría estar formada por el conjunto de atributos: Apellidos y Nombre (podemos tener amigos con nombres que coincidan o hermanos cuyos apellidos sean iguales). Como se trata de una clave compleja en la que puede haber confusiones añadimos una clave artificial denominada CódigoAgenda que podría ser un número consecutivo.

Ejercicio resuelto

Queremos informatizar una pequeña clínica veterinaria de nuestro barrio donde los vecinos traen a sus mascotas. Cuando un vecino acude a la consulta por primera vez, se recoge la siguiente información:

Del propietario de la mascota: DNI, Apellidos, Nombre, Dirección, Teléfono. Hay que tener en cuenta que algunos vecinos llevan animales que están abandonados y no tienen, por tanto, ningún propietario.

Del animal: Nombre, Raza, Fecha de nacimiento, Peso, Altura, Vacunas, y otros datos recogidos como Descripción. (Por ejemplo una mancha en la pata, etc.)

Tanto la primera vez como cuando el propietario lleva de nuevo a su mascota para una consulta se necesita guardar:

Fecha de la consulta, motivo de la consulta, peso, diagnóstico y tratamiento

Se pide:

Obtener una lista con las entidades
Colocar los atributos de cada entidad
Elegir las claves principales

[Mostrar retroalimentación](#)

Entidades: PROPIETARIOS, ANIMALES, CONSULTAS

Para cada ENTIDAD, indicamos entre paréntesis la lista de atributos y subrayado el atributo principal o clave (simple o compuesto por varios atributos)

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

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

CONSULTAS (IdAnimal + FechaConsulta, Motivo, Peso, Diagnóstico, Tratamiento)

2.3.- 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 asociar datos entre sí.

En una relación se asocia un elemento de una entidad con otro de otra entidad.

Everaldo
Coelho/YellowIcon
(GNU/GPL)

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, que suele ser un verbo en activa o pasiva.. 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. Las relaciones siempre operan en los dos sentidos.

Por ejemplo: La relación entre CLIENTES y VEHICULOS se definen en dos direcciones:

Un CLIENTE es propietario de un VEHICULO.
Cada VEHICULO es propiedad de un CLIENTE.



DIA (Elaboración propia)



Y ¿puede tener atributos una relación entre entidades? La respuesta es si.

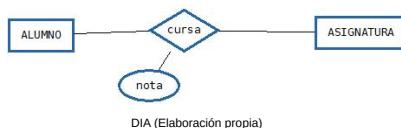
Relaciones con atributos.

Una relación puede tener atributos: normalmente las relaciones no tienen atributos, pero puede ocurrir que los tenga. 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 calificación que ha obtenido un alumno/a en una determinada asignatura.

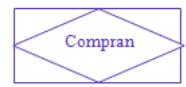
Otro ejemplo típico son las relaciones que representan **históricos**. Este tipo de relaciones suele constar de datos como fecha y hora. Cuando se emite una factura a un cliente o se le facilita un duplicado de la misma, es necesario registrar el momento en el que se ha realizado dicha acción. Para ello,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

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



A veces, cuando una relación tiene atributos, significa que la relación oculta una entidad que hasta ese momento no se ha definido, a esa entidad se le denomina entidad asociada. Se representará mediante un rombo inscrito en un rectángulo. Y se le denomina **Entidad asociada**.

Estas entidades darán lugar a una tabla que contenga esos atributos cuando se conviertan al modelo relacional.



DIA (Elaboración propia)

Descripción de las relaciones.

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

Grado de la relación.

Cardinalidad de la relación o correspondencia.

Cardinalidades de las entidades.

A continuación desarrollamos cada uno de ellos. Comenzamos con el grado de una relación y en los siguientes apartados veremos los otros conceptos

Grado de una relación: Se define como el número de entidades que participan en una relación.

Relación reflexiva o de grado 1: es aquella relación en la que solo participa una entidad.

Relación binaria de grado 2: es aquella relación en la que participan dos entidades.

Relación ternaria o de grado 3: es aquella relación en la que participan 3 entidades.

Relación n-aria: es aquella relación en la que participan más de 3 entidades.

Relación doble: ocurre cuando dos entidades están asociadas a través de dos relaciones diferentes.

Las relaciones pueden tener cualquier grado pero cuando se pueda se intentará que sean binarias.

Ejemplos de relaciones de grado 1, grado 2, grado 3 y doble:

Relación reflexiva o de grado 1

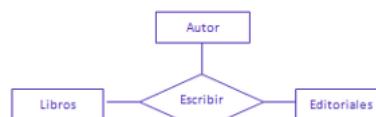


Relación binaria o de grado 2



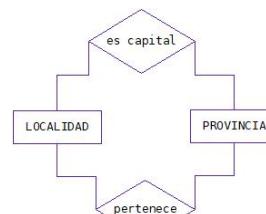
DIA (Elaboración propia)

Relación ternaria o de grado 3



DIA (Elaboración propia)

Relación doble



DIA (Elaboración propia)

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.

2.3.1.- Correspondencia y cardinalidad de entidades.

¿Y qué es eso de la correspondencia o la cardinalidad de relaciones? 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.

¿Es lo mismo la cardinalidad de las relaciones que la cardinalidad de las entidades? Veremos que no, la cardinalidad de las relaciones nos dirá el valor máximo de asociaciones que puede tener cada ocurrencia de una entidad con otra, en la relación que estemos estudiando; mientras que las cardinalidades de entidades nos expresan los valores máximo y mínimo de asociaciones que puede tener cada ocurrencia de las entidades relacionadas, según la relación de estudio.

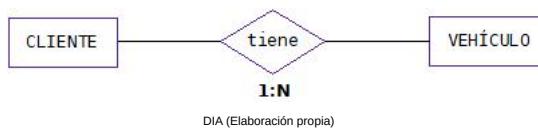
Vaya lio, verdad. Verás como con ejemplos se ve mucho más fácil.



Correspondencia o cardinalidad de relaciones.

Correspondencia o cardinalidad de una relación: es el número máximo de ocurrencias de cada entidad que pueden intervenir en una relación dada, o dicho de otra forma, el número máximo de entidades a las que puede asociarse otra entidad mediante una relación. 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 correspondencia o cardinalidad de la relación tiene es muchos a uno (expresada como 1:N) e indica el número de ocurrencias de la entidad CLIENTE que se relacionan con cada ocurrencia de la entidad VEHÍCULO y viceversa. Podríamos hacer la siguiente lectura: un cliente puede tener muchos vehículos y a un vehículo solo puede pertenecer a un único cliente.



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 estas correspondencias o cardinalidad de relaciones:

Relaciones 1:1 (uno a uno). Sean las entidades E1 y E2, una instancia u ocurrencia de la entidad E1 se relaciona únicamente con otra instancia de la entidad E2 y viceversa. Por ejemplo, para cada ocurrencia de la entidad EMPLEADO (e1) sólo habrá una ocurrencia relacionada de la entidad PUESTO (e2) y viceversa. O lo que es lo mismo, un empleado tiene un puesto asociado y un puesto sólo pertenece a un único empleado.

Relaciones 1:N (uno a muchos). Sean las entidades E1 y E2, una ocurrencia de la entidad E1 se relaciona con muchas o varias ocurrencias de la entidad E2 y una ocurrencia de la entidad E2 sólo estará relacionada con una única ocurrencia de la entidad E1. Por ejemplo, para cada ocurrencia de la entidad PROVEEDOR (e1) puede haber varias ocurrencias de la entidad PIEZA (e2) y para varias ocurrencias de la entidad PIEZA (E2) sólo habrá una ocurrencia relacionada de la entidad PROVEEDOR (e1) (si se establece que una pieza sólo puede ser suministrada por un único proveedor). O lo que es lo mismo, un proveedor puede suministrar varias piezas y una pieza sólo puede ser suministrada por un único proveedor.

Relaciones N:1 (muchos a uno). Sean las entidades E1 y E2, una ocurrencia de la entidad E1 está asociada con una única ocurrencia de la entidad E2 y una ocurrencia de la entidad E2 está relacionada con muchas ocurrencias de la entidad E1. Observa que las relaciones N:1 son como las 1:N, en las que se ha cambiado el lugar de las entidades. Por ejemplo, una ocurrencia de la entidad PIEZA (e1) sólo puede ser suministrada por un único PROVEEDOR (e2) y un PROVEEDOR (e2) puede suministrar muchas piezas (e1).

Relaciones N:M (muchos a muchos). Sean las entidades E1 y E2, una ocurrencia de la entidad E1 está relacionado con muchas o varias ocurrencias de la entidad E2 y viceversa. Por ejemplo, una ocurrencia de la entidad MECANICO puede estar relacionada con varias ocurrencias de la entidad REPARACION y una ocurrencia de la entidad REPARACION puede estar relacionada con varias ocurrencias de la entidad MECANICO. O lo que es lo mismo, un mecánico puede realizar varias reparaciones y una reparación puede ser realizada por varios mecánicos.

Vamos a ver ejemplos de los diferentes tipos de correspondencia:

Relaciones 1:1 (uno a uno)

Por ejemplo: Cada empleado ocupa un único puesto de trabajo y cada puesto de trabajo es ocupado por un solo empleado.



Relaciones 1:N (uno a muchos)

Por ejemplo: Un proveedor suministra muchas piezas. Cada pieza solo nos la suministra un único proveedor.



Relaciones N:M (muchos a muchos)

Por ejemplo: Cada mecánico puede realizar varias reparaciones y una misma reparación la pueden realizar varios mecánicos.



La cardinalidad de las relaciones puede representarse de varias maneras en los diagramas del modelo Entidad/Relación. Nosotros seguiremos la notación indicada en los ejemplos, que es la notación de Chen.

Cardinalidad de entidades.

Pero hemos dicho que no solo las relaciones tienen cardinalidades, las entidades también tienen cardinalidad. ¿Y qué es la cardinalidad de una entidad?

La cardinalidad de una Entidad o la cardinalidad con la que una entidad participa en una relación indica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ocurrencia o ejemplar de dicha entidad, esto es, indica el número de relaciones en las que una entidad puede aparecer.

La cardinalidad de una entidad se representa con el número mínimo y máximo de correspondencias en las que puede tomar parte cada ocurrencia 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**):

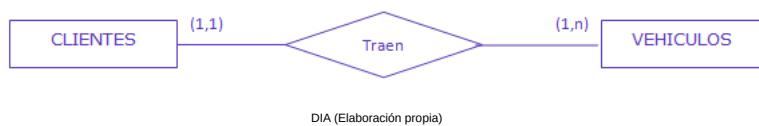
Cardinalidad mínima. Indica el número mínimo de asociaciones en las que aparecerá cada ocurrencia de la entidad (el valor que se anota es de cero o uno, aunque tenga una cardinalidad mínima de más de uno, se indica sólo un uno). El valor 0 se pondrá cuando la participación de la entidad sea opcional.

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).

Por ejemplo: En la relación CLIENTES **traen** VEHICULOS respecto a los clientes que traen a reparar sus vehículos a nuestro taller, las cardinalidades serían:

Un cliente trae a reparar como mínimo un vehículo y como máximo varios: la cardinalidad de CLIENTE es (1,n)
Cada vehículo es traído al taller por un cliente como mínimo y como máximo: la cardinalidad de VEHICULO es (1,1)

Observa que en la representación gráfica, la cardinalidad de la entidad CLIENTE que es (1,n), se pone en el lado opuesto, e igualmente se hace con la cardinalidad de VEHICULO que es (1,1).



Las cardinalidades de entidades son muy útiles a nivel de aplicación ya que permiten expresar de forma más detallada la manera en la que se relacionan las entidades.

Participación de una Entidad.

El concepto de participación está relacionado con las cardinalidades mínimas.

Si consideramos las entidades E1 y E2, se dice que la participación de la entidad E1 en una relación es **total (obligatoria)** si la existencia de cada una de sus ocurrencias necesita como mínimo de una ocurrencia de la entidad E2. En caso contrario, la participación es **parcial (opcional)**.

Por tanto:

La participación de cualquier entidad en una relación puede ser parcial o total. Si no es posible que una entidad exista a no ser que participe en la asociación, la participación es total; en caso contrario, la participación es parcial.

Para cada caso tendremos que:

Participación total u obligatoria: cardinalidad mínima 1

Participación parcial u opcional: cardinalidad mínima 0

Vamos a ver **algunos ejemplos**:

1.- Supongamos la relación PROFESOR - imparte - CLASE, cuando un PROFESOR necesita impartir al menos una CLASE, la entidad CLASE es **obligatoria**, si un PROFESOR puede estar registrado sin necesidad de impartir alguna CLASE, la entidad CLASE es **opcional**.

2.- Supongamos la relación anterior CLIENTE - trae - VEHICULO, si cada cliente que tenemos recogido en nuestra base de datos nos ha traído al taller al menos un vehículo, en ese caso la participación de la entidad CLIENTE en la relación es **obligatoria**. En el caso de que pudiéramos tener registrado algún cliente que no hubiera traído ningún coche al taller (clientes potenciales) la participación es **opcional**.

Reflexiona

Observa que a partir de las **cardinalidades máximas de cada entidad que participa en una relación**, podemos obtener la **correspondencia** o cardinalidad de esa relación.

En el siguiente ejemplo que ilustra el hecho de que un profesor imparte ninguna o muchas clases, y que una clase es impartida siempre por un único profesor, observa que la relación es de tipo 1:N, donde 1 viene de la cardinalidad máxima de CLASE y N viene de la cardinalidad máxima de PROFESOR.

Cardinalidades de la entidad PROFESOR es (0,n) - puede impartir ninguna o muchas clases

Cardinalidades de la entidad CLASE es (1,1) - siempre es impartida por un único profesor



DIA (Elaboración propia)

Por otra parte, observa que en este ejemplo, la **participación de PROFESOR es parcial u opcional**, pues tiene una cardinalidad mínima de 0, lo que significa que podemos tener registrados en nuestra base de datos a profesores que no imparten clases. En cambio, la **participación de CLASE es total u obligatoria**, pues tiene una cardinalidad mínima de 1, lo que significa que todas las clases que tengamos registradas en nuestra base de datos son impartidas por algún profesor..

2.3.2.- Relaciones fuertes y débiles.

En apartados anteriores hemos clasificado las entidades en entidades débiles y entidades fuertes, en función de que dependan o no de otra entidad para existir. Dado que las relaciones unen entidades, la existencia o no de dependencia entre éstas afecta también a las relaciones.



Everaldo Coelho (YellowIcon)
(GNU/GPL)

La existencia de entidades fuertes y débiles da origen a relaciones en las que se asocian ambos tipos de entidades, de modo que una entidad débil depende de otra fuerte para existir como las FACTURAS y los CLIENTES, ya que no hay facturas si no hay clientes. Un tipo diferente de dependencia de origen se da cuando la entidad débil es un subconjunto de la entidad fuerte, comopodría ser el caso de las entidades CLIENTE y PERSONA

La dependencia entre una entidad débil y otra fuerte puede ser de dos tipos:

Dependencia en existencia: (EX) Se dice que hay dependencia en existencia cuando las ocurrencias de la entidad débil no pueden existir si desaparece la ocurrencia de la entidad fuerte. En este tipo la entidad débil puede ser identificada sin necesidad de identificar la entidad fuerte de la que depende.

Por ejemplo, cuando un empleado de una empresa se da de baja será necesario dar de baja a todos sus familiares porque esa información ya no tendría sentido.

Dependencia en identificación: (ID) Cuando además de la condición anterior, la entidad débil no se puede identificar únicamente mediante los atributos propios de la misma, es necesario incluir la clave de la entidad fuerte de la que depende.

Por ejemplo, para identificar un ejemplar de cierto libro tomamos la identificación del libro y le añadimos la referente a ese ejemplar para diferenciarla de otros ejemplares del mismo libro.



DIA (Elaboración propia)

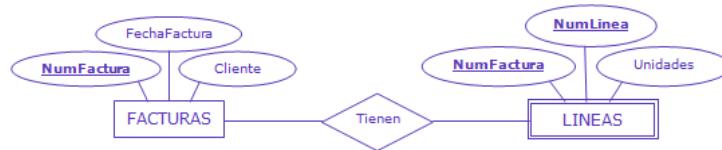
Relaciones débiles (relación no identificativa)

Si una entidad es independiente de la existencia de otra entidad, la relación entre ellas se describe como una relación débil o como una relación no identificativa.

Una relación débil existe cuando la clave primaria de la entidad relacionada (hijo) no contiene la clave primaria de la entidad padre.

Relaciones fuertes (relación identificativa) Una relación fuerte se da cuando las entidades que relaciona son dependientes en existencia y en identificación; es decir que la clave principal de la entidad relacionada contiene la clave principal de la entidad padre. En estos casos la clave principal de la entidad padre es clave ajena y parte de la clave principal en la entidad hijo.

Por ejemplo: En la relación FACTURAS tienen LINEAS, la clave principal de la entidad LINEAS se forma con la clave de la entidad FACTURAS junto con el número de línea.



DIA (Elaboración propia)

Ejercicio resuelto

Volviendo a la pequeña clínica veterinaria de nuestro barrio, en el apartado anterior establecimos las entidades, sus atributos y elegimos las claves principales.

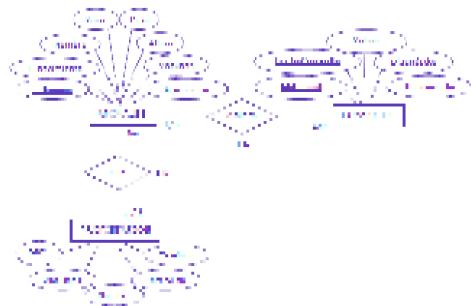
En estos momentos debemos establecer las relaciones entre esas entidades utilizando la representación gráfica del modelo entidad-relación. Para ello deberás determinar:

- Las relaciones que asocian a las entidades PROPIETARIOS, ANIMALES y CONSULTAS
- Las cardinalidades mínimas y máximas con las que participa cada entidad en la relación.
- Los tipos de correspondencia de las relaciones.

Se pide la representación gráfica del modelo Entidad-Relación, así como la respuesta a las siguientes cuestiones:

- 1.- ¿La participación de cada entidad es total o parcial? ¿Hay alguna entidad que participe en la relación de forma parcial?
- 2.- Explica si se trata de relaciones fuertes o débiles.

[Mostrar retroalimentación](#)



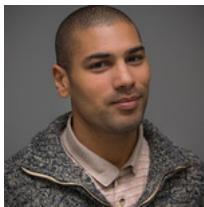
DIA (Elaboración propia)

1.- La participación de la entidad ANIMALES en la relación **Tienen** es **opcional o parcial** ya que algunos animales pueden no tener propietario (animales que son recogidos y están abandonados). El resto de entidades participan de forma **obligatoria o total**.

2.- La relación **Tienen** es una relación débil ya que cada ocurrencia de la entidad ANIMALES se identifica sin necesidad de incluir el DNI del propietario. En cambio la relación **Acuden** es una relación **fuerte** porque para identificar cada ocurrencia de la entidad CONSULTA es necesario incluir en su clave principal la clave de la entidad ANIMALES.

3.- Modelo Entidad-Relación ampliado.

Caso práctico



Alain Bachelier (CC BY-NC-SA)

Vindio y **Noiba** ya tienen casi listo el modelo Entidad/Relación que representa a nivel conceptual la base de datos que están diseñando para el Taller mecánico. Pero hay algunos aspectos que no terminan de reflejar de forma exacta. **Vindio** recuerda a **Noiba** que existe una ampliación del modelo E/R, el modelo Entidad/Relación Extendido o modelo ERE que amplía las posibilidades de diseño.

Para presentar los conceptos relacionados con este modelo Extendido vamos a partir de un ejemplo de entidad que se encuentra en la mayoría de empresas: la entidad empleados. En la mayoría de empresas existen distintos tipos de empleados, cuya descripción es difícil de recoger utilizando una misma entidad.

Por ejemplo en un centro de enseñanza nos podemos encontrar con profesores, personal de administración, personal de limpieza y mantenimiento, etc. De los profesores, por ejemplo, tendremos que recoger información que no se requiere para el resto de personal. Por ejemplo: el número de registro personal, la especialidad, el cuerpo al que pertenece, la fecha de alta en el cuerpo, etc.

Si recogemos a todo el personal en la misma entidad los que no sean profesores tendrían muchos valores nulos en esos atributos innecesarios.

Por otra parte existen muchos atributos comunes a todo el personal como: apellidos, nombre, DNI, dirección, fecha de contratación, etc.

Esto es lo que se pretende resolver empleando jerarquías de generalización.

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.

A continuación, nos vamos a centrar en una de esas nuevas características del modelo ERE que permite representar 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.

Generalización y especialización.

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

Por tanto, existirá la posibilidad de realizar una **especialización de un supertipo en subtipos**, y análogamente, establecer una **generalización de las subtipos en supertipos**. La **generalización** es la reunión en un supertipo de entidad de una serie de subtipos de entidades, que poseen características comunes. Los subtipos tendrán otras características que los diferenciarán entre ellos.

¿Cómo detectamos una generalización? Podremos identificar una generalización cuando encontramos una serie de atributos comunes a un conjunto de entidades, y otros atributos que sean específicos. Los atributos comunes conforman el supertipo y los atributos específicos el subtipo

La generalización y la especialización son técnicas de abstracción que permiten extraer de un conjunto de entidades una serie de atributos comunes y una serie de atributos específicos, de forma que los atributos comunes describen el **supertipo** y los atributos específicos los **subtipos**.

Una de las características más importantes de las jerarquías es la **herencia** por la que los subtipos heredan los atributos del **supertipo**.

De la misma forma si un supertipo participa en una relación, sus subtipos también.

La jerarquía de generalización o especialización recoge la relación entre entidades del tipo padre-hijo o supertipo-subtipo.

¿Cómo se representa una generalización o especialización? Existen varias notaciones, pero hemos de convenir que la relación que se establece entre un supertipo 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 supertipo y conectados a él a través de líneas rectas, las entidades subtipo.

El **supertipo** es la entidad de mayor nivel y contiene los atributos comunes.

El **subtipo** es la entidad de menor nivel que contiene los atributos específicos.

Observa el siguiente **ejemplo**:

La entidad supertipo es EMPLEADO.

Las entidades subtipos son PROFESOR, ADMINISTRATIVO Y MANTENIMIENTO.

Los atributos de EMPLEADO los heredan cada uno de los subtipos, entre ellos el atributo clave que es N_empleado

La forma de representar una jerarquía es mediante un tirángulo invertido, que indica que cada uno de los subtipos es un caso particular del supertipo.



En el siguiente apartado veremos que hay distintos tipos de jerarquías y las clasificaremos en función de diferentes criterios.

Autoevaluación

Indica si la siguiente afirmación es verdadera o falsa.

Si un supertipo de una jerarquía participa en una relación, sus subtipos también lo harán.

- Verdadero Falso

Verdadero

Al igual que los subtipos heredan los atributos del supertipo, heredan las relaciones en las que participa éste.

3.1.- Tipos de jerarquías.

Una generalización/especialización podrá tener diferentes restricciones semánticas como son:



Everaldo Coelho and
YellowIcon (GNU/GPL)

Totalidad: cuando todas las ocurrencias del supertipo pertenecen a alguno de los subtipos. Se dice que la jerarquía es total.

Por ejemplo, si todos los empleados del centro son o profesores o administrativos o de mantenimiento. Todos los tipos de empleado están incluidos en la clasificación.

Parcialidad: Cuando puede haber ocurrencias en el supertipo que no pertenezcan a ninguno de los subtipos. Se dice que la jerarquía es parcial.

Por ejemplo, si en nuestro centro existen empleados que no sean ni profesores, ni administrativos ni de mantenimiento.

Exclusividad: Cuando una ocurrencia del supertipo no puede estar a la vez incluida en más de un subtipo.

Por ejemplo, que un profesor no puede ser a la vez administrativo o de mantenimiento y viceversa. Se dice que la jerarquía es exclusiva.

Solapamiento: Cuando una ocurrencia del supertipo puede estar a la vez en varios subtipos.

Por ejemplo, si un empleado puede ser a la vez administrativo y de mantenimiento. Se dice que la jerarquía es con solapamiento.

Las cardinalidades de las jerarquías son:

(1,1) en el supertipo.

(0,1) en los subtipos, para las exclusivas.

(1,1) o (0,1) en los subtipos, para las solapadas.

¿Cómo se presenta gráficamente cada uno de esos tipos de jerarquías?

Si la jerarquía es total se pone un círculo sobre el triángulo invertido. Si es parcial no se pone.

Si la jerarquía es exclusiva se pone un arco sobre las líneas que enlazan al supertipo con los subtipos. Si es con solapamiento no se pone el arco.

Observa el siguiente ejemplo, en el que se muestra una representación gráfica de todas las posibles combinaciones.

Ejemplos de representación gráfica de los distintos tipos de jerarquía

DIA (Elaboración propia)

Ejercicio resuelto

Queremos recoger la información correspondiente a las aulas de un instituto de secundaria con la especialidad de informática:

De cada aula se quiere recoger la siguiente información: NºAula, Piso, Pasillo, Nº plazas. El número de aula es único y distinto para todas ellas.

Cuando las aulas son salas de ordenadores necesitamos saber: nº de ordenadores, escáner y nº de impresoras, así como otros equipamientos que pudieran instalarse.

Si se trata de laboratorios queremos saber el tipo (de ciencias, de idioma, etc.) y el equipamiento que tiene.

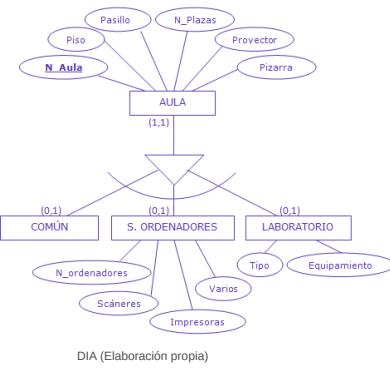
Todas las aulas, sean comunes, laboratorios o salas de ordenadores, pueden tener o no proyector y pizarra interactiva.

Existen otros tipos de aulas no recogidos en el modelo anterior como gimnasio, biblioteca, etc.

Cada aula se utiliza únicamente para las funciones que tiene asignadas (no se imparte una clase normal en un laboratorio, etc.)

Construye el diagrama E/R del enunciado anterior.

[Mostrar retroalimentación](#)



DIA (Elaboración propia)

4.- Represenatación gráfica del modelo Entidad-Relación.

Caso práctico



Noibia le comenta a Dílio, que ya están en condiciones para poder realizar el modelo Entidad/Relación de la base de datos, pero que no tiene claro si realizar su representación gráfica usando la simbología de Chen u otra diferente.

A continuación veremos que, con el paso del tiempo, se han ido desarrollando para este modelo distintas simbologías con la intención de adaptarse a las herramientas de diseño basadas en la utilización del ordenador, que paralelamente han ido surgiendo.

Afortunadamente encontramos en el mercado una gran variedad de herramientas, tanto de software libre como propietario, que nos van a ayudar a diseñar este modelo, corregir los posibles errores y en muchos casos nos facilitarán la generación del código.

Alain Bachelier (CC BY-NC-SA)

El modelo de datos E/R, a pesar de tener algunas desventajas, como herramienta de modelado cada vez está más extendida. De hecho los vendedores han agregado tantas extensiones a la presentación básica que sigue siendo la herramienta de diseño de bases de datos más utilizada.

En los ejemplos desarrollados a lo largo de la unidad hemos utilizado los símbolos del modelo Chen, sin embargo se han desarrollado otros estilos que se adaptan mejor a las herramientas de modelado de bases de datos basados en el ordenador.

Los más conocidos son:

Diagrama entidad-relación de Chen: es el que hemos visto hasta ahora.

Diagrama de pata de gallo.

Rein85.

IDEFX1.

El modelo de Chen, tal y como lo hemos estudiado en este tema, está basado en los trabajos de Chen (modelo básico) y en las mejoras incorporadas por Teorey, Yang y Fry (modelo ampliado). Ha sido el modelo en que se han basados las herramientas CASE dominante desde finales de los 80. Aunque actualmente no sea el modelo dominante como generador de modelos ER, todas las herramientas actuales tienen su origen en él.

El modelo pata de gallo, desarrollado por Bachman, es una herramienta de modelado muy extendida por usar una notación fácil de entender. Si en el modelo Chen utiliza la designación 1: N para el tipo de correspondencia entre relaciones y (0,1), (1,1), (0,n), (1,n) para las cardinalidades, el modelo pata de gallo reúne esta información en un solo símbolo. Sin embargo este modelo no puede representar cardinalidades distintas de 0, 1 o n como sí puede hacerse en el modelo Chen (por ejemplo (3,30) para designar que a una clase deben asistir como mínimo 3 y como máximo 30 alumnos). Algunas herramientas comerciales que utilizan este modelo evitan este problema añadiendo las estas cardinalidades en el diagrama usando texto y definiéndolas en el directorio de datos.

El modelo Rein85 fue desarrollado por Reiner en 1985. Basado en las mismas convenciones que el de pata de gallo, sin embargo sus símbolos son diferentes. Este modelo no reconoce cardinalidades, únicamente trata de conectividad. Su nivel de abstracción es mayor.

El modelo IDEFX1 fue desarrollado a finales de los 70 por la fuerza aérea de EE.UU mediante el programa ICAM (fabricación asistida por ordenador). Se trata de un programa que pretendía aumentar la productividad de la fabricación aplicando tecnología informática. Se desarrollaron unos estudios que obtuvieron métodos gráficos para definir funciones, estructuras de datos y dinámicas de empresas de manufacturas. Se convirtió en una herramienta de modelado de datos de manufactura en general, aunque utiliza menos símbolos que otros modelos y por tanto proporciona menos detalles.

Para saber más

Puedes ver una comparación de los distintos símbolos utilizados por cada uno de los modelos citados en el siguiente enlace:

[Comparativa de símbolos del modelo E-R](#) (pdf - 0,09 MB)

4.1.- Herramientas para construir diagramas E/R.



Actualmente hay muchas herramientas para construir diagramas E/R en el mercado; a lo largo de este curso necesitarás familiarizarte con alguna de ellas. A continuación te mostramos algunos ejemplos:

Everaldo Coelho and YellowIcon (GNU/GPL)

Recomendación

A continuación te mostramos diferentes sitios web donde puedes descargarte distinto software para realizar el diseño de bases de datos y/o construir diagramas E/R

Software Libre. Herramientas que puedan interpretar y generar modelos E/R, SQL y hacer análisis de base de datos MySQL. Por ejemplo: Workbench y DBDesigner.

[Página de descarga de Workbench](#)

[Página de descarga de DBdesigner](#)

Algunas herramientas de software libre se utilizan para generar únicamente el diagrama, sin tener conocimiento de lo que significan, ni generan SQL. Estos incluyen Kivio (incluida en la suite KDE Koffice) y el software Dia.

[Página de Dia](#)

[Página de Kivio](#)

Freeware. Una herramienta que puede generar la base de datos y la capa de código de la aplicación es el Editor de RISE.

[Página del editor de RISE](#)

Software propietario: Algunos de los diagramas ER de propiedad son:

herramientas ARIS, dbForge Studio para MySQL, Devgems Data Modeler, Oracle Designer, PowerDesigner, SQLyog, Toad Data Modeler, SQL Maestro y Microsoft Visio.

[Página de Toad Data Modeler](#)

Para saber más

Aunque las notaciones más extendidas son la de Chen y Pata de gallo, existen herramientas de modelado en el mercado que utilizan otras simbologías. Si quieres ampliar esta información puedes visitar los siguientes enlaces:

[Comparativa de todos los modelos](#)

[Definición en wikipedia del Modelo IDEF1X](#)

[Ejemplo de una base de datos de casas de colonias](#)

Sobre el modelo pata de gallo (o pata de cuervo, según otros autores) existe más documentación porque es un modelo muy utilizado por las herramientas de diseño de bases de datos o herramientas CASE.

Algunos ejemplos sobre el modelo pata de gallo son:

[Ejemplos sobre el modelo de pata de gallo](#)

Ejercicio resuelto

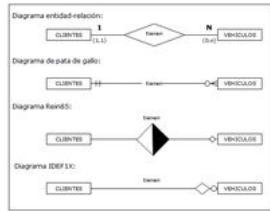
Tenemos la siguiente información:

Un cliente puede tener muchos vehículos.

Cada vehículo es de un cliente.
La participación de la entidad VEHICULOS es obligatoria.
Puede haber clientes registrados que no tengan ningún vehículo.

Construir el diagrama Entidad - Relación utilizando los modelos vistos anteriormente

[Mostrar retroalimentación](#)



DIA (Elaboración propia)

4.2.- Elaboración de diagramas Entidad/Relación



Ya hemos visto que existen diversas herramientas y notaciones para representar los diagramas o el modelo Entidad/Relación, pero ¿cómo creo un diagrama E/R? ¿Por dónde empiezo? ¿Y qué puedo hacer con todo lo visto?

En este apartado te vamos a dar algunas orientaciones para que puedas aplicar todos los conceptos aprendidos hasta ahora en la elaboración de diagramas Entidad/Relación.

Recuerda 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 modelo será como un plano que facilite la comprensión y solución del problema, de manera que el diagrama que construyamos va a ser una representación gráfica de los requisitos o condiciones que se derivan del problema a modelar.

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:

Identificación de entidades y relaciones.

Identificación de atributos, claves y jerarquías.

Identificación de entidades y relaciones.

a. Identificar entidades.

Para localizar aquellos elementos que serán las entidades de nuestro modelo, 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.

Otra forma de identificar entidades es localizando objetos o elementos que existen por sí mismos. Por ejemplo: CLIENTE, VEHICULO, etc. En otras ocasiones, la localización de varias características o propiedades puede dejar ver la existencia de una entidad.

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

Existencia propia.

Cada ocurrencia de un tipo de entidad debe poder ser diferenciada del resto de ocurrencias.

Todas las ocurrencias 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. Identificar relaciones.

Una vez localizadas las entidades, debemos establecer qué relación o asociación 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 reflexivas o relaciones de grado 1.

Cada una de las relaciones establecidas deberá tener asignado un nombre, **preferiblemente en minúsculas, representativo** de su significado o acción.

c. Establecer el tipo de correspondencia y las cardinalidades de las entidades.

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 reflexiva o unaria (de grado 1), hemos de representar en nuestro esquema los roles desempeñados por la entidad en dicha relación.

Identificación de atributos, claves y jerarquías.

No basta con la localización de entidades y relaciones. Hemos de completar el proceso realizando las siguientes tareas:

a. Identificación de atributos.

Revisaremos 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 COMPUSTO.

MÉTODO DE CÁLCULO, SI ES ATRIBUTO DERIVADO O CALCULADO.

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

b. Identificación de claves.

Del conjunto de atributos de una entidad se establecerán una o varias claves candidatas, escogiéndose una de ellas como clave o llave primaria de la entidad. Esta clave estará formada por uno o varios atributos que identificarán de manera única cada ocurrencia de entidad. El proceso de identificación de claves permitirá determinar la fortaleza (al menos una clave candidata) o debilidad (ninguna clave candidata) de las entidades encontradas.

Se representará la existencia de esta clave primaria mediante la notación elegida para la elaboración del diagrama E/R. Del mismo modo, se deberán representar adecuadamente las entidades fuertes o débiles.

c. Determinación de jerarquías.

Como ya hemos comentado, es probable que existan entidades con características comunes que puedan ser generalizadas en una entidad de nivel superior o supertipo (jerarquía de generalización). Pero también, puede ser necesario expresar en el esquema las particularidades de diferentes ocurrencias de un tipo de entidad, por lo que se crearán subtipos de un 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.

Debes conocer

Tutorial. Resumen del diseño de una base de datos relacional

Fases del diseño

En los siguientes videotutoriales puedes ver un resumen de todos los conceptos estudiados hasta ahora. Esto te servirá para entender mejor cómo realizar el diseño de una base de datos relacional. En este apartado nos vamos a centrar, mediante un ejemplo, en las 2 primeras fases de diseño.

Fases del diseño de una Base de Datos Relacional

<https://www.youtube.com/embed/ffKWjT253ms>

Isabel Cruz. [Descripción textual alternativa del video](#). (Dominio público)

Pasos para elaborar un diagrama ER. Ejemplo Veterinaria

En este videotutorial se especifican los pasos que vamos a seguir en el diseño conceptual o la creación del modelo E/R Y ERE.

El resultado final debe ser un diagrama Entidad/Relación **completo, correcto, legible y escalable**. Estas son las propiedades deseables que debe cumplir un diagrama Entidad/Relación bien construido.

Vemos como ejemplo el diseño conceptual de la base de datos VETERINARIA.

Pasos para crear el modelo Entidad/Relación. Ejemplo Veterinaria

<https://www.youtube.com/embed/BIN95AO1u-c>

Isabel Cruz. [Descripción textual alternativa del video](#). (Dominio público)

Uso del software DIA para elaborar el diagrama ER. Veterinaria.

En este videotutorial veremos el uso del Software DIA para confeccionar una diagrama ER y ERE, en concreto lo vamos a ver con el ejemplo de la base de datos Veterinaria.

Una vez descargado el software DIA de [software DIA](#) e instalado en tu equipo puedes ver cómo realizar el diseño del diagrama ER del ejemplo de Veterinaria.

Para utilizar los elementos típicos del modelo ERE es necesario instalar un plugin y añadirlo al software DIA instalado.

Su instalación consiste en:

1. - Descarga el [plugin](#) (zip - 7035 B)
2. - Copia el archivo descarga a la subcarpeta .dia que se habrá creado tras la primera ejecución de DIA en tu directorio de usuario.
3. - Se generará un nuevo archivo LICENSE y dos carpetas: shapes y sheets.
4. - Reinicia el programa Dia.
5. - Debajo de las herramientas, selecciona Otras hojas → ERE

Uso del software DIA. Creación del modelo ERE de veterinaria.

<https://www.youtube.com/embed/gKcCl9Mn75s>

Isabel Cruz. [Descripción textual alternativa del video](#) (Dominio público)

Tutorial. Creando el modelo Entidad/Relación paso a paso. Ejemplos

En este apartado vas a ver varios ejemplos resueltos sobre el diseño conceptual de bases de datos. Verás la creación de diagramas Entidad/Relación a partir de unas especificaciones concretas. Para ello:

1. - Se parte del enunciado del problema a modelar: el conjunto de requerimientos que debe cumplir la base de datos. (Documento de especificación de requerimientos)
2. - La herramienta utilizada para elaborar el diseño conceptual de la base de datos será el Software Dia. Obtendremos así el modelo conceptual o modelo Entidad/Relación de la base de datos.
3. - Los pasos que vamos a seguir en el diseño son los siguientes:
 - a. Identificar las entidades, atributos y relaciones.
 - b. Estudiar las cardinalidades de entidades y cardinalidades de relaciones.
 - c. Identificar las claves candidatas y determinar la clave primaria.
 - d. Estudiar otras características del modelo ER Extendido como las jerarquías.
 - e. Elaborar el diagrama E/R o ERE utilizando el software DIA.

Ejemplo 1. Diseño de la base de datos PELICULAS.

Enunciado.

Vamos a realizar el diseño de una base de datos relacional para guardar información sobre películas y los actores que actúan o participan en esas películas.

A continuación se indican las especificaciones y requerimientos que debe cumplir la base de datos. A partir de ellos debes realizar su diseño para obtener su modelo conceptual mediante un diagrama Entidad/Relación.

- 1.- Una película se caracteriza por: código, título, año de estreno, género.
- 2.- Una película pertenece obligatoriamente a un género y solo a uno. Pero de un determinado género puede haber varias películas o ninguna.
- 3.- El género de una película se caracteriza por un código, nombre y descripción.
- 4.- En una película participan uno o varios actores y cada actor puede participar en una o varias películas.
- 5.- De los actores interesa su código, DNI, nombre, fecha de nacimiento, edad, varios email.
- 6.- Hay que considerar dos posibles tipos de actores, aunque puede haber otros tipos:

Actores Oscarizados: de los que hay que almacenar además, el total de oscars y la fecha del primer oscar.

Actores Productores: de los que hay que almacenar además, el numero de películas producidas, aportación mínima, y primer año de producción.

Un actor oscarizado también puede ser productor.

Estudia y analiza esos requisitos y haz la representación gráfica de su modelo Entidad/Relación.

a. Entidades, relaciones y atributos.

Comenzamos identificando las entidades, relaciones entre entidades y atributos de cada entidad y/o relación.

En este ejemplo, vamos a utilizar diferentes acuerdos para resaltar cada uno de estos elementos. ENTIDADES en color azul y mayúsculas, ATRIBUTOS en color verde y entre paréntesis, y RELACIONES en formas verbales y color violeta.

a.1.- ENTIDADES. ¿Cómo localizar las entidades? Buscamos los sustantivos o nombres que representan objetos de los que interesa guardar información en la base de datos. Serían:

```
<strong><span style="color: #3366ff;">PELÍCULA,</span></strong> pues según el punto 1.- Una película se caracteriza por: código, título, año de estreno, género. (De PELÍCULA interesa guardar esos datos)
<strong><span style="color: #3366ff;">GÉNERO,</span></strong> pues según el punto 3.- El género de una película se caracteriza por un código, nombre y descripción. (De GÉNERO interesa guardar esos datos)
<strong><span style="color: #3366ff;">ACTOR,</span></strong> pues por el punto 5.- De los actores interesa su código , DNI, nombre, fecha de nacimiento, edad, varios email. (De ACTOR interesa guardar esos datos)
<strong><span style="color: #3366ff;">OSCARIZADO y PRODUCTOR, </span></strong> pues por el punto 6.- Hay que considerar dos posibles tipos de actores que son OSCARIZADO y PRODUCTOR (y nos dan otros datos específicos para esos tipos de actor)
```

a.2.- RELACIONES. ¿Cómo localizar las relaciones o inter-relaciones? Buscamos los verbos que conectan o asocian a las entidades encontradas. Serían:

```
<strong> <span style="color: #800080;">pertenece</span></strong> (pues según el punto 2.- Una película pertenece a un género , luego podemos establecer la relación: <span style="color: #3366ff;">PELÍCULA</span> - <span style="color: #800080;"><strong>pertenece</strong></span> - <span style="color: #3366ff;">GENERO</span>
<span style="color: #800080;"><strong>participa</strong></span> (pues según el punto 4.- En una película participan uno o varios actores, luego podemos establecer la relación: <span style="color: #3366ff;">PELÍCULA</span> - <span style="color: #800080;"><strong>participa</strong></span> - <span style="color: #3366ff;">ACTOR</span>
```

a.3.- ATRIBUTOS. ¿Cómo localizar atributos? Buscamos las características comunes de cada entidad y propiedades de las relaciones, si las hubiera. Encontramos las siguientes propiedades para cada entidad:

De PELÍCULA los atributos son: (código, título, año_estreno. Pues según el punto 1.- Una película se caracteriza por: código, título, año de estreno, género
De GÉNERO los atributos son: (código, nombre y descripción). Pues según el punto 3.- El género de una película se caracteriza por un código, nombre y descripción
De ACTOR los atributos son: (código, DNI, nombre, fecha_nacimiento, edad, email). Pues por el punto 5.- De los actores interesa su código , DNI, nombre, fecha de nacimiento, edad, varios email.
email es un atributo multivalorado, pues puede tomar más de un valor, nos dicen que puede haber 'varios email'.
edad es un atributo derivado o calculado, pues se puede obtener a partir del atributo 'fecha de nacimiento' del actor.

De momento, dejamos el punto 6 para resolverlo después, ya que nos habla de subtipos de actores, subtipos de la entidad ACTOR. El tipo de jerarquía lo resolvemos en el apartado relativo a las características del modelo ERE). En este punto podemos indicar los atributos de cada uno de esos subtipos:

De OSCARIZADO; los atributos son: (total_oscar, fecha_primer_oscar). Pues según el punto 6.- Actores oscarizados: de los que hay que almacenar además, el total de oscars y la fecha primer oscar.
De PRODUCTOR los atributos son: (num_pelis_producidas, importe_min y año_ultimo_producción). Pues según el punto 6.- Actores productores: de los que hay que almacenar además, el numero de películas producidas, aportación mínima, y primer año de producción.

Si resaltamos esos elementos en los requerimientos o especificaciones del enunciado, éste quedaría de la siguiente forma:

- 1.- Una PELÍCULA se caracteriza por: (código, título, año de estreno, género).
- 2.- Una película pertenece obligatoriamente a un género y solo a uno. Pero de un determinado género
puede haber varias películas o ninguna.
- 3.- El GÉNERO de una película se caracteriza por un (código, nombre y descripción).
- 4.- En una película participan uno o varios actores y cada actor puede participar en una o varias películas.
- 5.- De los ACTORES interesa su (código, DNI, nombre, fecha de nacimiento, edad, varios email).
- 6.- Hay que considerar dos posibles tipos de actores, aunque puede haber otros tipos:

Actor OSCARIZADO; de los que hay que almacenar además, el total de oscars y la fecha primer oscar. Actor PRODUCTOR; de los que hay que almacenar además, el numero de películas producidas, aportación mínima, y **primer año de producción**.

b. Cardinalidades.

Estudiamos la cardinalidad o tipo de correspondencia de la relaciones y la cardinalidad de las entidades que participan en esa relación.

¿Por dónde empezamos? Vamos a ir indicando cada una de las relaciones encontradas junto con las entidades implicadas en esa relación.

Recuerda que para obtener la cardinalidad de las relaciones debemos justificar o argumentar el requerimiento o especificación del enunciado que nos proporciona esa información. De igual forma debemos proceder para obtener la cardinalidad de las entidades que participan en esa relación, justificando el requerimiento del enunciado en el que nos basamos, o en su defecto deducir la cardinalidad por el propio contexto realista del enunciado.

b.1.- **Relación** **pertenece ** en la asociación: **PELICULA – pertenece – GENERO.**

Comenzamos estudiando la cardinalidad de las entidades que participan en esa relación, ya que la cardinalidad de la relación se obtendrá a partir de la cardinalidad máxima de cada una de las entidades que participan en ella. Observa que siempre hay que ver la entidad en el contexto de la relación en la que participa.****

Cardinalidad de la entidad PELÍCULA es (1,1).

¿Cómo la hemos obtenido? Viendo la relación de izquierda a derecha (relación directa) obtendremos la cardinalidad mínima y máxima de PELÍCULA. Para ello nos preguntamos: ¿Una película, a cuantos géneros pertenece, como mínimo y como máximo?

Cardinalidad **mínima es 1**. Una película ¿a cuantos géneros pertenece como mínimo?: a **1**, pues según el punto 2: Una película pertenece **obligatoriamente a un género**.

Cardinalidad **máxima es 1**. Una película ¿a cuantos géneros pertenece como máximo? a **1**, pues según el punto 2: Una película pertenece **un género y solo a uno**.)

Por lo tanto, la **cardinalidad de PELÍCULA es (1,1)** (Recuerda que se expresa como (cardinalidad mínima, cardinalidad máxima))

Cardinalidad de la entidad GÉNERO.

Viendo la relación de derecha a izquierda (relación inversa), nos preguntamos: ¿A un género, cuantas películas pertenecen como mínimo y como máximo?

Cardinalidad **mínima es 0**. A un género, ¿cuántas películas pertenecen como mínimo?: **0** , pues según punto 2: De un determinado género puede haber **ninguna** película.

Cardinalidad **máxima es n**. A un género ¿cuántas películas pueden pertenecer como máximo? **n**, pues según punto 2: De un determinado género puede haber **varias** películas.

Por lo tanto, la **cardinalidad de GÉNERO es (0,n)**.

Cardinalidad o correspondencia de la relación pertenece. (PELICULA – pertenece – GENERO.) es N:1

La cardinalidad de la relación **pertenec** se obtiene uniendo las cardinalidades máximas de las entidades que asocia y separándolas por dos puntos (escribiémos la letra **n** en mayúsculas).

En este caso sería **N:1**, obtenida de la siguiente forma: **cardinalidad máxima de GENERO : cardinalidad máxima de PELÍCULA.** (Una película pertenece como máximo a 1 género y a un género pertenecen como máximo N películas).

b.2.- **Relación** **participa** en la asociación: **PELÍCULA – participa – ACTOR.**

Cardinalidad de la entidad PELÍCULA. es (1,m).

¿Cómo la hemos obtenido? Viendo la relación de izquierda a derecha (relación directa) obtendremos la cardinalidad mínima y máxima de PELÍCULA. Para ello nos preguntamos: ¿En una película, cuantos actores participan, como mínimo y como máximo?

Cardinalidad **mínima es 1**. En una película, como mínimo, ¿cuántos actores participan? **1** , según punto 4: En una película participan **uno o ...**.

Cardinalidad **máxima es m**. En una película, como máximo, ¿cuántos actores participan? **m**, según punto 4: En una película participan **... o varios actores**.

Cardinalidad de la entidad ACTOR. es (1,n).

Viendo la relación de derecha a izquierda (relación inversa), nos preguntamos: ¿Un actor, en cuantas películas participa como mínimo y como máximo?

Cardinalidad **mínima es 1**. Un actor, como mínimo, ¿en cuantas películas puede participar? en **1**, según punto 4: Cada actor puede participar **en una o ...**.

Cardinalidad **máxima es n**. Un actor, como máximo, ¿en cuantas películas puede participar? en **n**, según punto 4: Cada actor puede participar **en ... o varias películas**.

Cardinalidad de la relación participa. (PELÍCULA – participa – ACTOR)es N:M

Se obtiene uniendo las cardinalidades máximas de las entidades que participan, separándolas mediante dos puntos, **esto es N:M** (ponemos las letras en mayúscula)

Es **N:M**, obtenida como **cardinalidad máxima de ACTOR : cardinalidad máxima de PELÍCULA** (En una película participan como máximo a M actores y un actor participa como máximo en N películas).

c. Claves.

En este apartado debemos indicar las claves candidatas de cada entidad, elegir la clave principal e indicar las claves alternativas que pueda tener cada entidad.

Recuerda que una **clave candidata** es una clave mínima (conjunto de atributos mínimo) que permite identificar de forma única cada ocurrencia, instancia o ejemplar de una entidad. Y la **clave principal** es la clave candidata que seleccionamos para identificar a cada instancia de una entidad. Seleccionada la clave principal, si había varias candidatas, el resto pasan a denominarse **claves alternativas**.

Entidad **PELICULA**.

Claves **candidatas**: **codigo**. Lo denominaré **codigo_p**.

Clave principal: codigo_p. Solo hay una candidata, luego será también la clave principal o primaria. Por lo tanto, no hay claves alternativas.

Entidad GENERO.

Claves candidatas: **codigo**. Lo denominaré **codigo_g**.

Clave principal: **codigo_g**. Solo hay una candidata, luego será también la clave principal o primaria. Por lo tanto no hay claves alternativas.

Entidad ACTOR.

Claves candidatas: codigo y DNI.

Clave principal: será el **codigo** que denominaré codigo_a.

Claves alternativas: **DNI**. (la candidata que no he elegido como clave principal)

d. Características ERE (Jerarquías).

Se observa, por el punto 6.- que hay que diferenciar a dos tipos de actores, los Oscarizados y los productores, que tienen atributos particulares según su tipo, además de los atributos generales. Por lo que podemos hablar de una **jerarquía**, una especialización de actores. Por lo tanto el supertipo sería ACTOR y los subtipos son **OSCARIZADO** y **PRODUCTOR**, pues en el punto 6 nos dicen:

6.- Hay que considerar dos posibles tipos de actores, aunque puede haber otros tipos:

Actores Oscarizados: de los que hay que almacenar además, el total de oscars y la fecha primer oscar.

Actores productores: de los que hay que almacenar además, el número de películas producidas, aportación mínima, y primer año de producción.

Un actor Oscarizado también puede ser productor.

¿de qué tipo es la jerarquía?

La **Jerarquía es de tipo Parcial y Solapada**:

Parcial: porque nos dicen que puede haber otros actores, además de los tipos Oscarizados y productores.

Solapada: porque nos indican que puede haber actores que a la vez sean Oscarizados y productores (como en la vida real).

Por tanto tendremos el supertipo **ACTOR** y los dos subtipos de **ACTOR** son:

OSCARIZADO con atributos particulares (total oscars, fecha primer oscar)
PRODUCTOR con atributos particulares (numero películas producidas, aportación mínima, primer año producción)

e. Diagrama Entidad/Relación.

Teniendo en cuenta el análisis realizado construimos el diagrama ERE con el software DIA, obteniendo el siguiente resultado.

A tener en cuenta:

Atributos: Cada atributo lo representamos en una elipse. Si es atributo clave principal lo subrayamos y si hay atributos clave alternativas, lo subrayamos y entonces al lado del principal ponemos **IP** (de Identificador Principal). Los atributos derivados se representan con elipse punteada y los atributos multivaluados con doble elipse.

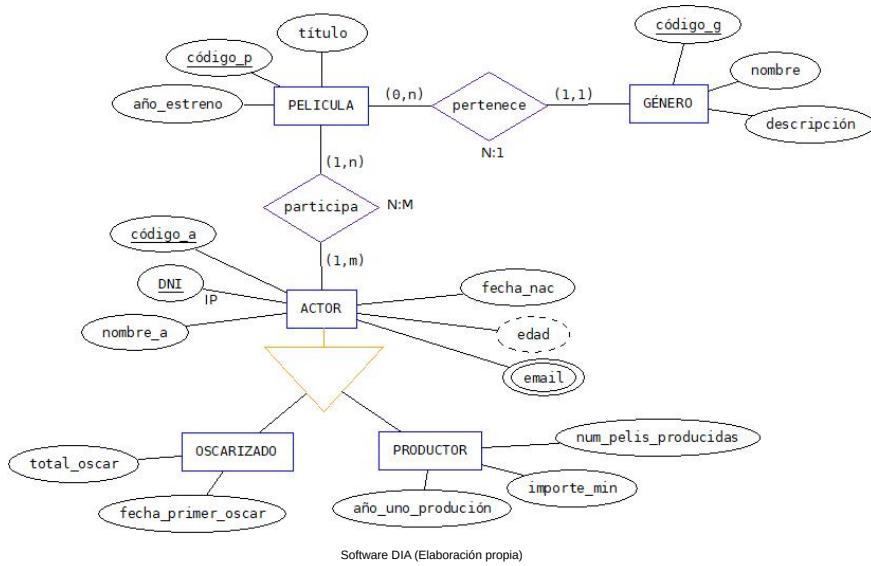
Entidades: Cada entidad fuerte o regular se escribe en un rectángulo y su cardinalidad se pone en el extremo opuesto de la relación. Cada entidad se denomina con un nombre diferente.

La **cardinalidad de cada entidad** se pone en el lado opuesto de la relación en la que participa.

Relaciones: Cada relación se escribe en un rombo y su **cardinalidad** se pone debajo, al lado o encima. Cada relación tiene un nombre verbal diferente.

Si enlazamos dos entidades con una relación, **en vertical**, la cardinalidad de la relación la escribimos de arriba a abajo.

Jerarquía: La representamos con un triángulo invertido y por ser parcial no lleva círculo en la base del triángulo y por ser solapada no lleva arco entre los subtipos.



Ejemplo 2. Diseño de la base de datos VIDEOTECA.

Enunciado

Un Videoclub Vintage dispone de copias de películas en formato Blu-ray que ofrece en alquiler. Nos piden realizar el diseño de una base de datos relacional para guardar información sobre las películas que se alquilan, sus actores y los clientes que las alquilan.

A continuación se indican las especificaciones y requerimientos que debe cumplir la base de datos. A partir de ellos debes realizar su diseño para obtener su modelo conceptual mediante un diagrama Entidad/Relación.

- 1.- Una película se caracteriza por un código, título, año de estreno y género.
- 2.- Una película pertenece obligatoriamente a un género y solo a uno. Pero de un determinado género puede haber varias películas o ninguna.
- 3.- El género de una película se caracteriza por un código, nombre (que es único) y su descripción.
- 4.- Todas las películas tienen una o varias copias y pueden ser alquiladas. Una copia es de una y solo una película. Si se descataloga la película no tiene sentido mantener la copia.
- 5.- Las copias de cada película se identifican con un número sucesivo (1, 2, 3, 4 ...) y de ellas se guarda la fecha de compra y estado de conservación.
- 6.- Las copias de las películas se alquilan a clientes. De los clientes interesa su DNI, nombre, fecha_nacimiento, edad y varios email.
- 7.- Los clientes puedes ser de dos tipos:
 - 7.1.- Socios: de ellos se guarda además un código, fecha de alta, descuento y cuota.
 - 7.2.- Eventuales: de estos clientes se guarda la fecha de su primer alquiler.
- 8.- Tanto los clientes socios como los eventuales han alquilado al menos una copia, pudiendo alquilar varias, y cada copia puede haber sido alquilada varias veces o ninguna.
- 9.- Al alquilar una copia interesa saber la fecha de alquiler y precio de alquiler.
- 10.- Los clientes eventuales pueden haber sido invitados por un único cliente eventual, y un cliente eventual puede invitar a varios cliente eventuales o a ninguno.

Estudia y analiza esos requisitos y haz la representación gráfica de su modelo Entidad/Relación.

a. Entidades, relaciones, atributos

Comenzamos identificando las entidades, relaciones entre entidades y atributos de cada entidad y/o relación.

En este ejemplo, vamos a utilizar diferentes acuerdos para resaltar cada uno de estos elementos. ENTIDADES en color azul y mayúsculas, ATRIBUTOS en color verde y entre paréntesis y RELACIONES en formas verbales y color violeta.

a.1.- **ENTIDADES.** ¿Cómo localizar las entidades? Serán aquellos nombres o sustantivos que representan objetos de los que interesa guardar datos. Serán:

PELÍCULA, según nos dicen en el punto 1.- de una película se desea guardar un código, título, año de estreno y género
GÉNERO, según nos dicen en el punto 3.- del género de una película se desea guardar un código, nombre (que es único) y su descripción.
COPIA, según nos dicen en el punto 5.- las copias de cada película se identifican con un número sucesivo (1, 2, 3, 4 ...) y de ellas se guarda la fecha de compra y estado de conservación.
y COPIA es débil en existencia de PELÍCULA, según el punto 4.- Si se descataloga la película no tiene sentido mantener la copia, luego

hay dependencia en existencia de copia respecto a película.

y también es débil en **identificación**, ya que las copias se numeran como 1, 2, 3, ..., número que no permite identificar de forma única a una copia, ya que para películas diferentes, si cada una tiene 3 copias, tendríamos repetidos los números 1, 2, 3.

CLIENTE, según nos dicen en el punto 6.- de los clientes interesa guardar su DNI, nombre, fecha_nacimiento, edad y varios email.. Además tenemos dos subtipos de esta entidad:

SOCIO y **EVENTUAL**, son subtipos de CLIENTE según nos dicen en el punto 7.- Los clientes puedes ser de dos tipos: socio y eventual.

a.2.- **RELACIONES**. ¿Cómo localizar las relaciones o inter-relaciones? Buscamos los verbos que conectan o asocian a las entidades encontradas. Serían:

pertenece, (según el punto 2.- Una película pertenece obligatoriamente a un género) luego la relación es: PELÍCULA – pertenece – GENERO.

tiene, (según el punto 4.- Las películas tienen una o varias copias) luego la relación es: PELICULA – tiene – COPIA.

se_alquila, (según el punto 6.- Las copias de la películas se alquilan a clientes), luego la relación es: COPIA – se_alquila – CLIENTE.

invita, es una relación reflexiva (según el punto 10.-Los clientes eventuales pueden haber sido ser invitados por un único cliente eventual), luego la relación es: EVENTUAL – invita – EVENTUAL.

a.3.- **ATRIBUTOS**. ¿Cómo localizar atributos? Buscamos las características comunes de cada entidad y propiedades de las relaciones, si las hubiera. Encontramos las siguientes propiedades o características para cada entidad:

De **PELICULA** los atributos son (**código**, **título**, **año_estreno**, **género**), según indica el punto 1.- Una película se caracteriza por un código, título, año de estreno y género

De **GENERO** los atributos son (código, nombre, descripción), según indica el punto 3.- El género de una película se caracteriza por un código, nombre (que es único) y su descripción

De **COPIA** los atributos son (**numero_copia**, **fecha_compra**, **estado**) según el punto 5.- Las copias de cada película se identifican con un número sucesivo (1, 2, 3, 4 ...) y de ellas se guarda la fecha de compra y estado de conservación.

De **CLIENTE** los atributos son (<abbr title="Documento Nacional de Identidad">DNI</abbr>, nombre, fecha_nacimiento, edad) (que es derivado pues se puede calcular a partir de la fecha de nacimiento), email (que es multivalorado) según el punto 6.- De los clientes interesa su **DNI**, nombre, fecha_nacimiento, edad y varios email.

Del subtipo **SOCIO** los atributos son (**código**, **fecha_alta**, **descuento**, **cuota**) según el punto 7.1.- De los socios se guarda además un código, fecha de alta, descuento y cuota.

Del subtipo **EVENTUAL** el atributo es (**fecha_primer_alquiler**) según el punto 7.2.- De los eventuales se guarda además la fecha de su primer alquiler

La relación **se_alquila**, COPIA – se_alquila – CLIENTE tiene atributos:

los atributos son: (**fecha_alquiler**, **precio_alquiler**) según el punto 9.- Al alquilar una copia interesa saber la fecha de alquiler y precio de alquiler.

b. Cardinalidades

Estudiamos la cardinalidad o tipo de correspondencia de la relaciones y la cardinalidad de las entidades que participan en esa relación.

¿Por dónde empezamos? Vamos a ir indicando cada una de las relaciones encontradas junto con las entidades implicadas en esa relación. Primero obtenemos la cardinalidades de cada entidad que participa en esa relación y después, uniendo por dos puntos las máximas de las entidades de esa relación, obtenemos la cardinalidad de la relación.

b.1.- **Cardinalidad relación pertenece**, PELICULA – pertenece – GENERO, es: **N:1** puesto que:

Cardinalidad de la entidad PELICULA es **(1,1)**.

¿Cómo la hemos obtenido? Viendo la relación de izquierda a derecha (relación directa) obtendremos la cardinalidad mínima y máxima de **PELICULA**. Para ello nos preguntamos: ¿Una película, a cuantos géneros pertenece, como mínimo y como máximo?

Cardinalidad **mínima** de **PELICULA** es **1**, pues por el punto 2: Una película pertenece obligatoriamente a un género. (Como mínimo una película pertenece a un género, hay obligatoriedad)

Cardinalidad **máxima** de **PELICULA** es **1**, pues por el punto 2: Una película pertenece .. a uno y solo un género. (Como máximo una película pertenece a un género)

Cardinalidad de la entidad GENERO es **(0,n)**.

¿Cómo la hemos obtenido? Viendo la relación de derecha a izquierda (relación inversa) obtendremos la cardinalidad mínima y máxima de **GÉNERO** Para ello nos preguntamos: ¿A un género, cuantas películas pueden pertenecer, como mínimo y como máximo?

Cardinalidad **mínima** de **GÉNERO** es **0**, pues por el punto 2: De un determinado género puede haber ninguna película. (Como mínimo de un género puede haber **0** películas)

Cardinalidad **máxima** de **GÉNERO** es **n**, pues por punto 2: De un determinado género puede haber varias películas. (Como máximo de un género puede haber **n** películas)

Luego la **cardinalidad** de la relación **pertenece**, en PELICULA – pertenece – GENERO es: **máxima de GÉNERO : máxima de PELÍCULA**, esto es: **N:1**.

b.2.- **Cardinalidad relación tiene**, PELICULA – tiene – COPIA, es: **1:N** puesto que:

Cardinalidad de la entidad PELICULA es **(1,n)**.

Viendo la relación de izquierda a derecha (relación directa) obtendremos la cardinalidad mínima y máxima de **PELÍCULA**. Para ello nos preguntamos: ¿Una película, cuantas copias puede tener, como mínimo y como máximo? Pues por el punto

4: Todas las películas tienen varias copias. Luego:

Como **mínimo** una película tiene una copia, **1**, (ya que todas tienen) y como **máximo** tiene varias copias, **n**.

Cardinalidad de la entidad **COPIA** es **(1,1)**.

Viendo la relación de derecha a izquierda (relación inversa) obtendremos la cardinalidad mínima y máxima de **COPIA**.

Para ello nos preguntamos: ¿Una copia, de cuantas películas puede ser, como mínimo y como máximo? Pues por el punto 4: Una copia es de una y solo una película. Luego:

(Como **mínimo** cada copia es de **1** película y como **máximo** cada copia es de **1** película.

Luego la **cardinalidad** de la relación **tiene**, en PELÍCULA - tiene - COPIA es: **máxima de COPIA : máxima de PELÍCULA**, esto es: **1:N**.

b.3.- **Cardinalidad relación se_alquila**, COPIA – se_alquila - CLIENTE, es: **M:N**.

Cardinalidad de la entidad **COPIA** es **(0,n)**.

Pues por el punto 8.- Cada copia puede haber sido alquilada varias veces o ninguna. (Como **mínimo** una copia puede no haber sido alquilada, **0**, y como **máximo** puede haber sido alquilada varias veces, **n**).

Cardinalidad de la entidad **CLIENTE** es **(1,m)**.

Pues por el punto 8.- Tanto los clientes socios como los eventuales han alquilado al menos una copia, pudiendo alquilar varias. (Como **mínimo** un cliente ha alquilado **1** copia y como **máximo** puede haber alquilado varias, **m**)

Luego **cardinalidad** de la relación **se_alquila** en COPIA -se_alquila - CLIENTE es: **máxima de CLIENTE : máxima de COPIA**, esto es: **M:N**.

b.4.- **Cardinalidad relación invita**, EVENTUAL – invita - EVENTUAL, es una **relación reflexiva**, de cardinalidad: **1:N**.

Cardinalidad de **EVENTUAL** con el rol, el que invita a **(0,n)**

Pues por el punto 10.- un cliente eventual puede invitar a varios clientes eventuales o a ninguno. (Como **mínimo** puede no haber invitado a nadie, **0**, y como **máximo** puede invitar a varios, **n**)

Cardinalidad de **EVENTUAL** con el rol, ha sido invitado es: **(0,1)**

Pues por punto 10.- Los clientes eventuales pueden haber sido invitados por un único cliente eventual. (Como **mínimo 0**, pues puede no haber sido invitado, y como **máximo** solo puede haber sido invitado por **1** cliente eventual).

Luego **cardinalidad** de EVENTUAL - invita - EVENTUAL es: **máxima de EVENTUAL rol 'ha sido invitado' : máxima de EVENTUAL rol 'el que invita'**, esto es: **1:N**

c. Claves

En este apartado debemos indicar las claves candidatas de cada entidad, elegir la clave principal e indicar las claves alternativas que pueda tener cada entidad.

Recuerda que una **clave candidata** es una clave mínima (conjunto de atributos mínimo) que permite identificar de forma única cada ocurrencia, instancia o ejemplar de una entidad. Y la **clave principal** es la clave candidata que seleccionamos para identificar a cada instancia de una entidad. Seleccionada la clave principal, si había varias candidatas, el resto pasan a denominarse **claves alternativas**.

Cuando hay entidades débiles en identificación, significa que ninguno de los atributos propios de esa entidad débil puede identificar de forma única a cada ocurrencia, por lo que estas entidades lo que tienen es una identificador o clave débil.

Entidad **PELICULA**.

Claves **candidatas**: codigo. Lo denominaré codigo_p.

Clave **principal**: codigo_p. Solo hay una candidata, luego será también la clave principal o primaria. Por lo tanto, no hay claves alternativas.

Entidad GENERO.

Claves **candidatas**: **codigo, nombre**, (pues el valor de nombre es único)

Clave **principal**: elegimos como clave principal o primaria el **codigo** lo denominaré **codigo_g**.

Clave **alternativa**: **nombre**.

Entidad **COPIA**.

Al ser una entidad débil de **PELICULA** en identificación, necesita de la clave primaria de **PELICULA** para poder identificar cada copia.

Lo que tiene es una **clave DÉBIL num_copia**.

Entidad **CLIENTE**.

Claves **candidatas**: **DNI**.

Clave **principal**: solo hay una candidata, luego el **DNI** será también la clave principal o primaria. Por lo tanto, no hay claves alternativas.

Entidad **SOCIO** (subtipo de **CLIENTE**).

Claves **candidatas**: **DNI** (que hereda de **SOCIO**) y **codigo**.

Clave **principal**: **DNI**

Clave **alternativa**: **codigo**

Entidad **EVENTUAL** (subtipo de **CLIENTE**).

Hereda la clave principal **DNI** de **CLIENTE**.

d. Características ERE (Jerarquías)

Observamos que hay que diferenciar a dos tipos de clientes, los socios y los eventuales, que tienen atributos particulares según su tipo, además de los atributos generales. Por lo que podemos hablar de una **jerarquía**, una especialización de clientes. Por

lo tanto el supertipo sería **CLIENTE** y los subtipos son **SOCIO** y **EVENTUAL**, pues en el punto 7.- nos dicen:

7.- Los clientes puedes ser de dos tipos:

7.1.- Socios: de ellos se guarda además un código, fecha de alta, descuento y cuota.

7.2.- Eventuales: de estos clientes se guarda la fecha de su primer alquiler.

¿de qué tipo es la jerarquía?

La **Jerarquía es de tipo Total y Exclusiva**:

Total: porque nos dicen que los clientes pueden ser dos tipos, sin indicación de que pudiera haber algún otro tipo de cliente.

Exclusiva: porque un cliente no puede ser a la vez socio y eventual.

Por tanto tendremos el supertipo **CLIENTE** y los dos subtipos de **CLIENTE** son:

SOCIO con atributos particulares (codigo, fecha_alta, descuento, cuota)

EVENTUAL con atributos particulares (**fecha_primer_alquiler**)

e. Diagrama Entidad/Relación

Teniendo en cuenta el análisis realizado construimos el diagrama ERE con el software DIA, obteniendo el siguiente resultado.

A tener en cuenta:

Atributos: Cada atributo lo representamos en una elipse. Si es atributo clave principal lo subrayamos y si hay atributos clave alternativas, lo subrayamos y entonces al lado del principal ponemos **IP** (de Identificador Principal). Los atributos derivados se representan con elipse punteada y los atributos multivaluados con doble elipse. Los atributos que son clave débil (en entidades débiles) los representamos en elipse con subrayado discontinuo.

Entidades: Cada entidad fuerte o regular se escribe en un rectángulo y su cardinalidad se pone en el extremo opuesto de la relación. Cada entidad se denomina con un nombre diferente.

Las entidades débiles las denotamos con doble recuadro. Y si es débil en identificación, se puede poner <abbr title="Identificación">ID</abbr> dentro del rombo que representa la relación.

La **cardinalidad de cada entidad** se pone en el lado opuesto de la relación en la que participa.

Relaciones: Cada relación se escribe dentro de un rombo y su **cardinalidad** se pone debajo, al lado o encima. Cada relación tiene un nombre verbal diferente.

Si enlazamos dos entidades con una relación, en vertical, la cardinalidad de la relación la escribimos de arriba a abajo.

Jerarquía: La representamos con un triángulo invertido. Por ser total lleva círculo en la base del triángulo y por ser exclusiva o excluyente lleva un arco entre los subtipos.

Software DIA (Elaboración propia)

Ejemplo 3. Diseño de la base de datos CAMPEONATO.

Enunciado

La Asociación Andaluza de Videojuegos o e-Sports necesita gestionar diferentes campeonatos de juegos online mediante una base de datos que almacene información sobre los concursantes, sus equipos y los diferentes juegos en los que participan los concursantes.

Se debe realizar el diseño de una base de datos relacional a partir de las siguientes especificaciones o requisitos que debe cumplir:

- 1.- De los concursantes se desea almacenar un código, su nombre, sexo, fecha de inscripción y cuota de inscripción.

- 2.- Cada concursante pertenece obligatoriamente a un equipo y solo a uno. Pero en un equipo puede haber ninguno o varios concursantes.
- 3.- Cada equipo se caracteriza por un código, nombre, comunidad y año de fundación.
- 4.- Los equipos, no todos, pueden organizar varios juegos. Cada juego es organizado por uno y solo un equipo.
- 5.- Los juegos se caracterizan por su código, nombre, nivel de dificultad y número de megusta.
- 6.- Los concursantes pueden participar en ninguno o varios juegos, independientemente del equipo al que pertenezcan.
- 7.- En un juego pueden participar ninguno o varios concursantes.
- 8.- Cuando un concursante participa en un juego, se anota la fecha de comienzo en la que inició ese juego y se guardan también los puntos que va acumulando en ese juego.
- 9.- Los concursantes pueden designar a otro concursante como su ídolo, de manera que un concursante puede ser ídolo de ninguno o varios concursantes, pero como ídolo solo puede tener a un único concursante o a ninguno.

Estudia y analiza esos requisitos y haz la representación gráfica de su modelo Entidad/Relación.

a. Entidades, relaciones, atributos

Comenzamos identificando las entidades, relaciones entre entidades y atributos de cada entidad y/o relación.

En este ejemplo, vamos a utilizar diferentes acuerdos para resaltar cada uno de estos elementos. ENTIDADES en color azul y mayúsculas, ATRIBUTOS en color verde y entre paréntesis y RELACIONES en formas verbales y color violeta.

a.1.- **ENTIDADES.** ¿Cómo localizar las entidades? Serán aquellos nombres o sustantivos que representan objetos de los que interesa guardar datos. Serán:

CONCURSANTE, según nos dicen en el punto 1.- De los concursantes se desea almacenar determinada información.

EQUIPO, según nos dicen en el punto 3.- Cada equipo se caracteriza por un código, nombre, comunidad y año de fundación.

JUEGO, según nos dicen en el punto 5.- Los juegos se caracterizan por su código, nombre, nivel de dificultad y número de megusta..

a.2.- **RELACIONES.** ¿Cómo localizar las relaciones o inter-relaciones? Buscamos los verbos que conectan o asocian a las entidades encontradas. Serán:

pertenece, (según el punto 2.- Cada concursante pertenece obligatoriamente a un equipo y solo a uno) luego la relación es: CONCURSANTE – pertenece – EQUIPO.

organiza, (según el punto 4.- Los equipos, no todos, pueden organizar varios juegos) luego la relación es: EQUIPO – organiza – JUEGO.

participa, (según el punto 6.- Los concursantes pueden participar en ninguno o varios juegos), luego la relación es: CONCURSANTE – participa – JUEGO.

es ídolo, es una relación reflexiva (según el punto 9.- Los concursantes pueden designar a otro concursante como su ídolo), luego la relación es: CONCURSANTE – es ídolo – CONCRUSANTE.

a.3.- **ATRIBUTOS.** ¿Cómo localizar atributos? Buscamos las características comunes de cada entidad y propiedades de las relaciones, si las hubiera. Encontramos las siguientes propiedades o características para cada entidad:

De **CONCURSANTE** los atributos son (**código, nombre, sexo, fecha_incri, cuota_inscri**), según indica el punto 1.- De los concursantes se desea almacenar un código, su nombre, sexo, fecha de inscripción y cuota de inscripción.

De **EQUIPO** los atributos son (código, nombre, comunidad, año_funda), según indica el punto 3.- Cada equipo se caracteriza por un código, nombre, comunidad y año de fundación.

De **JUEGO** los atributos son (**código, nombre, dificultad, megusta**) según el punto 5.- Los juegos se caracterizan por su código, nombre, nivel de dificultad y número de megusta.

La relación **participa**, CONCURSANTE – participa – JUEGO tiene atributos:

los atributos son: (**fecha_comienzo, puntos**) según el punto 8.- Cuando un concursante participa en un juego, se anota la fecha de comienzo en la que inició ese juego y se guardan también los puntos que va acumulando en ese juego.

b. Cardinalidades

Estudiamos la cardinalidad o tipo de correspondencia de las relaciones y la cardinalidad de las entidades que participan en esa relación.

¿Por dónde empezamos? Vamos a ir indicando cada una de las relaciones encontradas junto con las entidades implicadas en esa relación. Primero obtenemos la cardinalidades de cada entidad que participa en esa relación y después, uniendo por dos puntos las máximas de las entidades de esa relación, obtenemos la cardinalidad de la relación.

b.1.- **Cardinalidad relación pertenece**, en CONCURSANTE– pertenece – EQUIPO, es: **N:1** puesto que:

Cardinalidad de CONCURSANTE es (1,1).

¿Cómo la hemos obtenido? Viendo la relación de izquierda a derecha (relación directa) obtendremos la cardinalidad mínima y máxima de **CONCURSANTE**. Para ello nos preguntamos: ¿Un concursante, a cuantos equipos pertenece, como mínimo y como máximo?

Cardinalidad **mínima** de CONCURSANTE es **1**, pues por el punto 2: Cada concursante pertenece obligatoriamente a un equipo. (Como mínimo un concursante pertenece a un equipo, hay obligatoriedad)

Cardinalidad **máxima** de CONCURSANTE es **1**, pues por el punto 2: Cada concursante pertenece obligatoriamente a un equipo y solo a uno. (Como máximo un concursante pertenece a un equipo)
Cardinalidad de EQUIPO es (0,n).

¿Cómo la hemos obtenido? Viendo la relación de derecha a izquierda (relación inversa) obtendremos la cardinalidad mínima y máxima de **EQUIPO**. Para ello nos preguntamos: ¿A un equipo, cuantos concursantes pueden pertenecer, como mínimo y como máximo?

Cardinalidad **mínima** de **EQUIPO** es **0**, pues por el punto 2: Pero en un equipo puede haber ningún concursante. (Como mínimo en un equipo puede haber 0 concursantes)

Cardinalidad **máxima** de **EQUIPO** es **n**, pues por punto 2: Pero en un equipo puede haber varios concursantes. (Como máximo en un equipo puede haber n concursantes)

Luego la **cardinalidad** de CONCURSANTE- pertenece - EQUIPO es: **máxima de EQUIPO: máxima de CONCURSANTE**, esto es: **N:1**.

b.2.- **Cardinalidad relación organiza**, en EQUIPO - organiza - JUEGO, es: **1:N** puesto que:

Cardinalidad de EQUIPO es (0,n).

Viendo la relación de izquierda a derecha (relación directa) obtendremos la cardinalidad mínima y máxima de **EQUIPO**. Para ello nos preguntamos: ¿Un equipo, cuantos juegos puede organizar como mínimo y como máximo?

Pues por el punto 4: Los equipos, no todos, pueden organizar varios juegos. (Como **mínimo** un equipo puede no organizar juegos, luego **0**, ya que no todos organizan) y como **máximo** puede organizar varios juegos, **n**)

Cardinalidad de JUEGO es (1,1).

Viendo la relación de derecha a izquierda (relación inversa) obtendremos la cardinalidad mínima y máxima de **JUEGO**.

Para ello nos preguntamos: ¿Un juego, cuantos equipos lo pueden organizar, como mínimo y como máximo?

Pues por el punto 4: Cada juego es organizado por uno y solo un equipo.. (Como **mínimo** cada juego es organizado por **1** equipo y como **máximo** también)

Luego **cardinalidad** de EQUIPO - organiza - JUEGO es: **máxima de JUEGO : máxima de EQUIPO**, esto es: **1:N**.

b.3.- **Cardinalidad relación participa**, en CONCURSANTE - participa - JUEGO, es: **M:N**.

Cardinalidad de CONCURSANTE es (0,n).

Pues por el punto 6.- Los concursantes pueden participar en ninguno o varios juegos. (Como **mínimo** un concursante participa en ningún juego, **0**, y como **máximo** puede participar en varios, **n**)

Cardinalidad de JUEGO es (0,m).

Pues por el punto 7.- En un juego pueden participar ninguno o varios concursantes. (Como **mínimo** en un juego participa ningún concursante, **0**, y como **máximo** puede participar en varios, **m**)

Luego **cardinalidad** de CONCURSANTE -participa - JUEGO es: **máxima de JUEGO : máxima de CONCURSANTE**, esto es: **M:N**.

b.4.- **Cardinalidad relación es_idolo**, en CONCURSANTE - es_idolo - CONCURSANTE, es una **relación reflexiva**, de cardinalidad: **1:N**.

Cardinalidad de CONCURSANTE con el rol, es el ídolo, es (0,n)

Pues por el punto 9.- Un concursante puede ser ídolo de ninguno o varios concursantes. (Como **mínimo** puede ser ídolo de ninguno, **0**, y como **máximo** puede ser ídolo de varios, **n**)

Cardinalidad de CONCURSANTE con el rol, tiene por ídolo es: (0,1)

Pues por punto 9.- Un concursante como ídolo solo puede tener a un único concursante o a ninguno. (Como **mínimo 0**, pues puede no tener ídolo, y como **máximo** solo puede tener a uno,1).

Luego **cardinalidad** de CONCURSANTE - es ídolo - CONCURSANTE es: **máxima de CONCURSANTE rol 'tiene por ídolo' : máxima de CONCURSANTE rol 'es el ídolo'**, esto es: **1:N**

c. Claves

En este apartado debemos indicar las claves candidatas de cada entidad, elegir la clave principal e indicar las claves alternativas que pueda tener cada entidad.

Recuerda que una **clave candidata** es una clave mínima (conjunto de atributos mínimo) que permite identificar de forma única cada ocurrencia, instancia o ejemplar de una entidad. Y la **clave principal** es la clave candidata que seleccionamos para identificar a cada instancia de una entidad. Seleccionada la clave principal, si había varias candidatas, el resto pasan a denominarse **claves alternativas**.

Entidad **CONCURSANTE**.

Claves **candidatas**: codigo. Lo denominaré codigo_c.

Clave **principal**: **codigo_c**. Solo hay una candidata, luego será también la clave principal o primaria. Por lo tanto, no hay claves alternativas.

Entidad **EQUIPO**.

Claves **candidatas**: **codigo**. Lo denominaré **codigo_e**.

Clave **principal**: **codigo_e**. Solo hay una candidata, luego será también la clave principal o primaria. Por lo tanto, no hay claves alternativas.

Entidad **JUEGO**.

Claves **candidatas**: **codigo**. Lo denominaré **codigo_j**.

Clave **principal**: **codigo_j**. Solo hay una candidata, luego será también la clave principal o primaria. Por lo tanto, no hay claves alternativas.

d. Características ERE

No se aprecia ninguna Jerarquía ni ninguna otra característica especial del modelo Extendido.

e. Diagrama Entidad/Relación

Teniendo en cuenta el análisis realizado construimos el diagrama ERE con el software DIA, obteniendo el siguiente resultado.

A tener en cuenta:

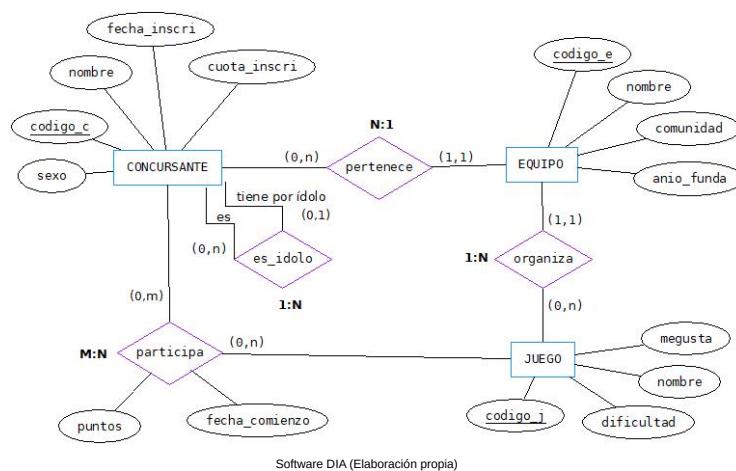
Atributos: Cada atributo lo representamos en una elipse. Si es atributo clave principal lo subrayamos y si hay atributos clave alternativas, lo subrayamos y entonces al lado del principal ponemos **IP** (de Identificador Principal).

Entidades: Cada entidad fuerte o regular se escribe en un rectángulo y su cardinalidad se pone en el extremo opuesto de la relación. Cada entidad se denomina con un nombre diferente.

La **cardinalidad de cada entidad** se pone en el lado opuesto de la relación en la que participa.

Relaciones: Cada relación se escribe dentro de un rombo y su **cardinalidad** se pone debajo, al lado o encima. Cada relación tiene un nombre verbal diferente.

Si enlazamos dos entidades con una relación, en vertical, la cardinalidad de la relación la escribimos de arriba a abajo.



Recomendación

En el siguiente enlace puedes acceder a un videotutorial para repasar todos los conceptos vistos hasta ahora sobre el modelo ER y ERE, así como el desarrollo de algunos ejemplos.

[Resumen y ejemplos del modelo ER y ERE en Youtube](#)

En el siguiente video puedes ver la resolución, paso a paso, del modelo o diagrama E/R para cierta base de datos, a partir de unas especificaciones y utilizando como herramienta de modelado el software DIA. La notación que se sigue para elaborar el diagrama E/R es la notación de Chen.

[Ejemplo de modelo E/R en Youtube](#)

5.- Modelo Relacional.

Caso práctico

Noiba y Vindio le presentan a Juan el diagrama Entidad/Relación que acaban de elaborar y que recoge todas las condiciones establecidas en el documento de especificación de requerimientos.

—¿Y ahora cómo se pasa este diagrama a una base de datos real? —pregunta Noiba.

—Aún hay que obtener el "paso a tablas" de lo representado en el diagrama. En cuanto realicemos esa transformación tendremos los elementos necesarios para implementar nuestra base de datos en cualquier SGBD relacional —le aclara Vindio.

Por lo tanto, empezaremos viendo en primer lugar las características del **Modelo Relacional**, que es modelo lógico de bases de datos al que vamos a transformar nuestro modelo E/R:



Alain Bachelier (CC BY-NC-SA)

E.F.Codd es el padre del modelo relacional, que definió en un artículo publicado en 1970. Este modelo está basado en dos teorías matemáticas: la teoría de conjuntos y la lógica de predicados de primer orden, que nos proporcionan los elementos necesarios para crear una base de datos relacional; pero no es necesario aprender estas teorías para poder utilizar el modelo relacional.

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

Independencia física de los datos: el modo de almacenamiento de los datos no debe influir en su manipulación lógica.

Independencia lógica de los datos: los cambios que se realicen en los objetos de la base de datos no deben repercutir en los programas y usuarios que accedan a ella.

Flexibilidad: presentar los datos a los usuarios de la forma más adecuada a la aplicación que utilicen.

Uniformidad: presentación de las estructuras lógicas en forma de tablas, fáciles de comprender y manipular por los usuarios.

Sencillez: las características anteriores, junto con los lenguajes de usuario hacen que este modelo sea fácil de entender y utilizar por el usuario.

Para Codd los datos se estructuran en forma de relaciones; entendiendo la relación como un elemento de la teoría matemática de las relaciones y que se corresponde con la idea de tabla o entidad. Los conceptos básicos de este modelo son:

Interrelación es la asociación entre dos tablas.

El conjunto de columnas que representan las propiedades de la tabla y que se denominan atributos.

El conjunto de filas que se denominan tuplas y que contienen los valores que toman cada uno de los atributos para cada elemento de la relación.

Para Codd lo importante es el diseño lógico, un modelo abstracto, por lo que no indica nada acerca de la implementación de este modelo en SGBD comerciales, sin embargo los trabajos de Codd y otros investigadores dieron lugar a diversas bases de datos comerciales, DB2 de IBM, Oracle, etc.; y es el modelo lógico en el que se basan la mayoría de los SGBD existentes actualmente en el mercado.

Un SGBD sólo necesita que el usuario pueda percibir la base de datos como un conjunto de tablas. Esta percepción sólo se aplica a la estructura lógica de la base de datos (en el nivel externo y conceptual de la arquitectura de tres niveles ANSI-SPARC). No se aplica a la estructura física de la base de datos, que se puede implementar con distintas estructuras de almacenamiento.

El modelo relacional, como todo modelo de datos tiene una parte estática constituida por los objetos permitidos y las restricciones, y una parte dinámica o conjunto de operaciones definidas sobre la estructura, que permite la transición entre estados de la base de datos. La parte dinámica está constituida por el álgebra relacional y el cálculo relacional es un conjunto de operaciones que describen paso a paso cómo computar una respuesta sobre las relaciones, tal y como éstas son definidas en el modelo relacional

En el modelo relacional 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.

Reflexiona

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

Autoevaluación

En el siguiente párrafo rellena los espacios en blanco con las palabras que consideres acertadas:

En el modelo relacional, el modelo el almacenamiento de los datos no influye en su manipulación. A esto se le denomina independencia [redacted]. En este modelo los cambios en los objetos no repercuten en los programas. A esto se denomina independencia [redacted]. Para el usuario la base de datos es un conjunto de tablas constituidas por columnas o [redacted] y por filas o [redacted].

La independencia del modelo con relación a como estén almacenados los datos y a los programas que utilizan esos datos son dos objetivos del modelo relacional. Un concepto básico de este modelo son las tablas o relaciones.

5.1.- Estructura del modelo relacional.

Las relaciones se utilizan para almacenar información sobre los objetos que se representan en la base de datos.

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

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



Una **relación** es una tabla con columnas y filas. El usuario percibe la base de datos como un conjunto de tablas. Una tabla sirve para almacenar los datos de una entidad.

Un **dominio** es el conjunto de valores que puede tomar un atributo. Cada atributo se define sobre un dominio. Se trata de un conjunto finito de valores homogéneos y atómicos.

Los valores de cada columna pertenecen a un dominio que se define previamente. Todos los dominios tienen un nombre y un tipo de datos asociado. Existen dos tipos de dominios:

Dominios generales: son aquéllos cuyos valores están comprendidos entre un máximo y un mínimo. Ejemplo el código postal formado por todos los números positivos de cinco cifras.

Dominios restringidos: son los que pertenecen a un conjunto de valores específico. Ejemplo el sexo que puede tomar los valores H o M.

El concepto de dominio es muy importante a la hora de definir operaciones que pueden tener sentido o no (por ejemplo no tiene sentido comparar una ciudad con un número de teléfono), pero su implementación en un **SGBD** es muy compleja.

Un **atributo** es el nombre de una columna de una relación. En otros términos un atributo es el papel o rol que desempeña un dominio en una relación. Representa el uso del dominio para una determinada relación. Aporta un significado semántico a un dominio. Ejemplo: atributo nombre definido sobre el dominio: conjunto de 15 caracteres.

Una **tupla** es cada fila de una relación (o tabla). El orden de las tuplas no es relevante.

El **grado** de una relación es el número de atributos que contiene, es decir el número de columnas de la tabla.

La **cardinalidad** de una relación es el número de tuplas que contiene.

El **valor** es la intersección entre una fila y una columna.

Veamos el siguiente ejemplo: Relación o tabla EMPLEADOS



Autoevaluación

Indica si la siguiente afirmación es verdadera o falsa.

En el modelo relacional de bases de datos podemos concebir como una relación a una tabla formada por filas y columnas, donde las filas son las tuplas y las columnas los atributos de la relación.

- Verdadero Falso

Verdadero

La forma en la que podemos materializar de manera simple una relación del modelo relacional es como una tabla formada por filas y columnas.

5.1.1.- Características de una relación.

La visualización desde un punto de vista lógico de una base de datos relacional se corresponde con un conjunto de tablas relacionadas.

Para el usuario una relación o tabla es un conjunto de entidades (simplificando, sustituimos el término conjunto de entidades por el de entidad)

Una **relación** es una tabla que tiene una serie de características:

Una tabla se percibe como una estructura bidimensional compuesta de filas y columnas

Cada tabla o relación tiene un nombre que la diferencia de las demás.

Everaldo Coelho and
YellowIcon (GNU GPL)

Cada fila de la tabla se denomina tupla y representa la ocurrencia de una entidad.

No admiten filas duplicadas

El orden de las filas no es significativo.

Cada columna representa un atributo, y cada columna tiene un nombre distinto.

El orden de las columnas es irrelevante. No están ordenadas.

La tabla es plana, es decir, en la intersección fila/columna solo puede haber un valor, no se admiten atributos multivaluados.

Cada tabla debe tener un atributo o conjunto de atributos que identifique a cada fila de forma única.

Cada columna tiene un mismo tipo de datos y un intervalo de valores específico: dominio de un atributo.

Tipos de relaciones o tablas.

En un SGBD relacional pueden existir varios tipos de relaciones o tablas, aunque no todos manejan todos los tipos. Unas se almacenan en la base de datos y otras son resultados de consultas.

Relaciones base. Son relaciones reales que tienen nombre y forman parte directa de la base de datos almacenada (son autónomas).

Vistas. También denominadas relaciones virtuales, son relaciones con nombre y derivadas a partir de una consulta: se representan mediante la definición de la consulta en términos de otras relaciones con nombre, no poseen datos almacenados propios.

Instantáneas. Son relaciones con nombre y derivadas. Pero a diferencia de las vistas, son reales, no virtuales: están representadas no sólo por su definición en términos de otras relaciones con nombre, sino también por sus propios datos almacenados. Son relaciones de sólo de lectura y se refrescan periódicamente por el sistema.

Resultados de consultas. Son las relaciones resultantes de alguna consulta especificada. Pueden o no tener nombre y no persisten en la base de datos.

Resultados intermedios. Son las relaciones que contienen los resultados de las subconsultas. Normalmente no tienen nombre y tampoco persisten en la base de datos.

Resultados temporales. Son relaciones con nombre, similares a las relaciones base o a las instantáneas, pero la diferencia es que se destruyen automáticamente en algún momento apropiado.

Reflexiona

En la siguiente imagen puedes ver el esquema o modelo de una base de datos relacional, donde se pueden ver varias tablas relacionadas entre sí, con relaciones 1 a muchos y dónde la columna o atributo marcado con una llave es la clave principal de la tabla.

Esquema o modelo de una base de datos relacional

Open Office Base (Elaboración propia)

Autoevaluación

Indica si la siguiente afirmación es verdadera o falsa

Entre las características que deben cumplir las tablas del modelo relacional son las siguientes:

No puede haber filas repetidas.

No importa el orden de las filas.

No importa el orden de las columnas.

En la intersección de una fila/columna puede haber más de una valor.

- Verdadero Falso

Falso

Una de las características del modelo relacional es que los atributos son atómitos, esto es, en la intersección de una fila y columna solo puede haber un valor indivisible o atómico.

5.1.2.- Claves primarias y ajenas.

A continuación vamos a ver los diferentes tipos de claves o llaves que podemos diferenciar:

Clave candidata: atributo o conjunto de atributos que identifican únicamente 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 o primaria: clave candidata elegida para identificar cada una de las tuplas. Una tabla solo puede tener una clave principal. Esa clave puede estar formada por un solo atributo o por varios.

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.

Everaldo Coelho and
YellowIcon (GNUGPL)

A continuación te detallamos cada uno de estos conceptos y vemos sus características.

Cuando un atributo o combinación de atributos de una tabla poseen un conjunto de valores que identifican de forma única a cada fila de la tabla o relación, se le denomina **clave primaria o principal**.

La clave primaria identifica inequívocamente a cada fila de una tabla. Sus características son:

Es única y mínima (a veces la fila se puede identificar por un único atributo, pero otras veces es necesario recurrir a más de uno). Existe y es conocida y pública.

Es NO NULA.

No es ambigua.

No se permiten claves primarias redundantes (repetidas). (no se puede descartar ningún atributo de la clave para identificar la fila)

Cuando una clave se compone de más de un atributo se denomina **clave compuesta**.

Existen tablas con uno o más atributos o grupos de atributos que poseen características de clave primaria. A ese atributo o grupo de atributos que pueden originar una clave primaria de su tabla se les denomina **claves candidatas**.

Características de las claves candidatas:

Al menos hay una por tabla y está formada por el conjunto de todos los atributos (puesto que no puede haber dos filas iguales). Ha de tener el mínimo número posible de atributos. Entre uno y el total de campos de la tabla.

No han de poseer campos superfluos.

La clave candidata seleccionada se convierte en clave primaria. El resto de claves candidatas se denominan claves alternativas.

Cuando un atributo o combinación de atributos de una relación (tabla) poseen un conjunto de valores que no forman la clave primaria en la tabla, pero son el conjunto de atributos que forman la clave primaria en otra tabla se les denomina **clave ajena, clave foránea o clave externa**.

Las claves ajenas, al compartir atributos, permiten relacionar tablas, por lo que facilitan la relación entre las mismas, para originar nuevas tablas o vistas.

Características de las claves ajenas:

Su dominio ha de ser el mismo que el dominio de la clave primaria de la tabla a la que haga referencia.

Se permite que el nombre de los atributos sea diferente de la clave ajena y que el nombre del atributo de la clave primaria y de la tabla asociada sean diferentes.

Su valor puede estar duplicado o ser nulo

Los atributos que la forman pueden formar parte de la clave primaria de la tabla a la que pertenecen.

Una tabla que no se relacione con otras no tiene claves ajenas.

Autoevaluación

Rellena los espacios en blanco con la palabra correcta teniendo en cuenta el siguiente esquema relacional y la siguiente restricción: el nombre de una sección es único.

Tabla EMPLEADOS:

CLAVES CANDIDATAS: IDEMPLEADO, <label for="clozeBlank54_45.0" class="sr-av">Rellenar huecos (1):</label> <input class="autocomplete-off" type="text" value="" id="clozeBlank54_45.0" style="width:3em" /> JXUwMDE2JXUwMDA3JXUwMDBm CLAVE PRINCIPAL: <label for="clozeBlank54_45.1" class="sr-av">Rellenar huecos (2):</label> <input class="autocomplete-off" type="text" value="" id="clozeBlank54_45.1" style="width:10em" /> JXUwMDE2JXUwMDA3JXUwMDBm CLAVES ALTERNATIVAS: <label for="clozeBlank54_45.2" class="sr-av">Rellenar huecos (3):</label> <input class="autocomplete-off" type="text" value="" id="clozeBlank54_45.2" style="width:3em" /> JXUwMDE2JXUwMDA3JXUwMDBm CLAVE AJENA: <label for="clozeBlank54_45.3" class="sr-av">Rellenar huecos (4):</label> <input class="autocomplete-off" type="text" value="" id="clozeBlank54_45.3" style="width:9em" /> JXUwMDE2JXUwMDA3JXUwMDBm

Tabla SECCIONES:

CLAVES CANDIDATAS: IDSECCION, CLAVE PRINCIPAL: CLAVE ALTERNATIVA:

Revisa detenidamente los tipos de claves.

5.2.- Restricciones.

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

Los tipos de restricciones en el modelado relacional pueden ser:

- Restricciones inherentes al modelo
- Restricciones de usuario o semánticas.

Restricciones inherentes al modelo.

Dake. (CC BY-SA)

Son restricciones propias del modelo e indican las características propias de una relación, por tanto **han de cumplirse obligatoriamente**.

- Ausencia de tuplas o filas repetidas.
- Irrelevancia del orden de las tuplas.
- Irrelevancia del orden de los atributos.
- Cada atributo solo puede tomar un único valor del dominio al que pertenece.

Otras restricciones inherentes son:

La **restricción de clave primaria (PRIMARY KEY)**, permite declarar uno o varios atributos como clave primaria de una relación.

En nuestro ejemplo la clave principal de la relación CLIENTES sería el CodCliente.

Integridad de clave: es una restricción que exige que todos los atributos de la clave primaria (PRIMARY KEY), han de contener valores no nulos (NOT NULL).

En el ejemplo de la relación FACTURAS- LINEAS, la clave principal de la tabla LINEAS es una clave compuesta de los campos NumFactura+NumLinea. Esta restricción implica que ambos atributos deben contener valores que no sean nulos.

Integridad referencial (FOREIGN KEY): Consiste en que no puede haber un valor en una clave ajena de una tabla, si antes no existe en la tabla de la que ese campo o conjunto de campos formen la clave primaria.

Para explicar este concepto denominamos tabla hija a la tabla que contiene la clave ajena y tabla padre a la tabla que contiene la clave principal.

Se permiten valores nulos en las claves ajenas; es decir que una clave ajena debe coincidir con un valor de clave primaria de la relación a la que apunta o tener valor nulo.

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

Restricciones de usuario.

Son restricciones impuestas para cada problema concreto, desde la definición de los dominios de un campo a condiciones impuestas a un campo de acuerdo con el valor de otros. Para establecer estas condiciones disponemos de las siguientes restricciones:

Restricciones de verificación: (CHECK) comprueban en una actualización si se cumplen las condiciones exigidas en la restricción o no, antes de ejecutarla.

Por ejemplo: Que la edad de un trabajador sea mayor o igual a 16

Restricción de unicidad: (UNIQUE) permite definir claves alternativas. Los valores de los atributos no pueden repetirse.

Por ejemplo: En la tabla CLIENTES la clave principal seleccionada es el CodCliente, pero el atributo DNI, que es clave alternativa en dicha tabla, tampoco admite valores duplicados. Para ello establecemos una restricción de tipo UNIQUE para este campo.

Restricción de obligatoriedad: (NOT NULL) permite declarar si uno o varios atributos pueden tomar valores nulos. Por definición una clave principal no puede contener valores nulos en ninguno de los atributos que la componen (integridad de clave) por tanto se definirán como NOT NULL.

Por ejemplo: La restricción NOT NULL puede ser necesaria también con otro tipo de atributos como el DNI citado en el ejemplo anterior, el NOMBRE_SECCION,etc.

Los disparadores (TRIGGER) son acciones, métodos según la terminología orientada a objetos, que ejecuta un objeto, como respuesta a un evento o acción que se realiza sobre él, cuando se cumple una determinada condición.

Por ejemplo si tenemos dos tablas:

una con los datos de los vehículos de nuestros clientes (matrícula y resto de datos que identifican al vehículo) y otra que recoge cada una de las entradas en el taller de esos vehículos (matrícula, fecha y hora), podemos crear un trigger que cada vez que se inserte un vehículo en la tabla de vehículos añada una fila en la tabla entradas en taller con los datos anteriores.

Autoevaluación

Indica si la siguiente afirmación es verdadera o falsa.

Una clave ajena o foránea nunca puede tomar el valor nulo.

- Verdadero Falso

Falso

Una clave ajena podrá tomar el valor nulo cuando esa columna no tenga la restricción NOT NULL y el significado del valor nulo será que esa fila no está relacionada con ninguna fila de la tabla padre.

5.2.1.- Restricciones y operaciones relacionales.

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

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

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

No pueden modificarse los atributos que forman parte de la clave primaria para dar origen a registros con valores de clave primaria redundantes.

Tizio (Dominio público)

La modificación de atributos que forman la clave principal no puede dar lugar a que aparezcan en esos atributos valores nulos.

Para aceptar los cambios en un atributo, se han de cumplir las restricciones de clave, referenciales y de usuario que hayan sido definidas para ese atributo en la tabla a la que pertenezca.

Si se modifica el valor de un atributo que forme parte de la clave primaria, habrá que sustituir los valores de los atributos que forman claves ajenas, en los que aparece el valor antes de modificar el contenido del atributo, por su nuevo valor. Es lo que se denomina actualización en cascada.

Respecto de la de **inserción de filas o registros**:

No pueden añadirse registros si alguno de los atributos que forman parte de la clave primaria posee un valor nulo.

No pueden añadirse registros si el conjunto de valores de los atributos que forman parte de la clave primaria origina una clave primaria redundante

Como consecuencia de la actualización en cascada, para añadir un valor a un atributo que forma parte de la clave ajena, ese valor ha de existir en la clave primaria a la que hace referencia la clave ajena.

Para aceptar los cambios en un atributo se han de cumplir las restricciones de clave, referenciales y de usuario que hayan sido definidas para ese atributo en la tabla a la que pertenezca.

Respecto del **borrado de filas o registros**:

Al borrar una fila que tenga asociados, mediante la clave primaria, otras filas en otras tablas, se pierde la consistencia de la tabla. Para solucionarlo existen dos posibilidades:

Borrado en cascada: consiste en borrar todas las filas de las tablas en los que aparezca como clave ajena el conjunto de atributos que forman la clave primaria del registro a borrar.

Para borrar un registro cuya clave primaria es clave ajena en otras tablas, primero hay que borrar los registros con la clave primaria asociada, para después borrar el registro deseado.

Teniendo en cuenta las consideraciones anteriores las posibilidades existente **para mantener la integridad referencial** son las siguientes:

Borrado y/o modificación en cascada (**CASCADE**). El borrado o modificación de una fila en la relación padre (relación con la clave primaria) ocasiona un borrado o modificación de las filas relacionadas en la relación hija (relación que contiene la clave ajena).

Borrado y/o modificación restringido (**RESTRICT**). En este caso no es posible realizar el borrado o la modificación de las filas de la relación padre si existen filas relacionadas en la relación hija.

Borrado y/o modificación con puesta a nulos (**SET NULL**). Esta restricción permite poner la clave ajena en la tabla referenciada a NULL, si se produce el borrado o modificación en la tabla primaria o padre.

Borrado y/o modificación con puesta a valor por defecto (**SET DEFAULT**). El valor que se pone en las claves ajenas de la tabla referenciada es un valor por defecto que se habrá especificado en la creación de la tabla.

Reflexiona

Imagínate a nuestra base de datos del taller mecánico, en la que tendremos una tabla CLIENTE con los datos de los clientes y la tabla VEHICULO almacenando los datos de los vehículos de nuestros clientes, y que estas tablas las relacionamos mediante el DNI del cliente, de manera que la columna DNI será la clave primaria en la tabla CLIENTE y en la tabla VEHICULO habrá una columna denominada, por ejemplo, DNI_cli que será clave ajena y que es la que permite relacionar ambas tablas.

- 1.- ¿Que debería pasar si eliminamos a un cliente que posee uno o más vehículos guardados en la tabla VEHICULOS?
- 2.- ¿Se podría insertar una fila en la tabla VEHICULO cuyo valor DNI_cli fuera un número de DNI que no existe en la tabla CLIENTE?

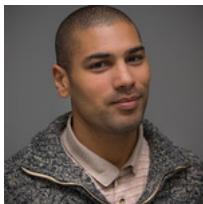
Imagino que habrás llegado a las siguientes conclusiones:

- 1.- Al eliminar a ese cliente, para mantener la integridad referencial, o bien se eliminan en cascada todos sus vehículos, o no deja eliminarlo si tiene vehículos, o bien pone a NULL o a un valor por defecto la columna que es clave ajena, DNI_cli en la tabla VEHICULO, lo que significaría que esos vehículos no están relacionados con ningún cliente.
- 2.- No se podría insertar ese vehículo, pues se estaría infringiendo la integridad referencial.



6.- Paso del diagrama Entidad-Relación al modelo relacional.

Caso práctico



Alain Bachelier (CC BY-NC-SA)

Una vez conseguido el esquema conceptual de datos mediante el modelo E/R, para su implementación es necesario transformarlo a esquema lógico mediante el modelo relacional. Ayudaremos a **Vindio** y **Noiba** en este proceso aplicando una serie de reglas de transformación que se describen a continuación.

En la fase del **diseño conceptual** has obtenido el diagrama E/R, un modelo conceptual que permite una gran independencia respecto del modelo físico de la base de datos, es decir todos los aspectos que tienen que ver con la implementación física de la base datos o **diseño físico**. Entre estas fases de diseño debemos abordar el **diseño lógico** para obtener un esquema basado en un modelo de base de datos concreto (relacional, jerárquico, de objetos, etc.)

En nuestro caso, en la fase de **diseño lógico** vamos a transformar el diagrama E/R obtenido a un esquema lógico basado en el **modelo de datos relacional**, que es en el que se basa el SGBD con el que haremos la implementación física.

Los principios de transformación del diagrama E/R al modelo de datos Relacional se basan en:

- 1.- **Transformación de entidades.** De forma general cada entidad se transforma en una tabla, y debemos tener en cuenta los diferentes casos en los que aparecen entidades: entidades fuertes, entidades débiles y jerarquías.
- 2.- **Transformación de relaciones.** Básicamente hay dos posibilidades: la relación o inter-relación se transforma en una nueva tabla, o bien se realiza 'propagación de clave' que consiste en exportar la clave primaria de una tabla a la otra tabla en la que será clave ajena, permitiendo así relacionar ambas tablas.
- 3.- **Transformación de atributos.** De forma general los atributos de una entidad se convierten en columnas de la tabla resultante, y tendremos que prestar atención a ciertos atributos (compuestos, multivaluados y derivados) para resolverlos adecuadamente. Los atributos de las relaciones o inter-relaciones pasarán a formar parte de la nueva tabla que genera la relación, o bien se exportarán junto a la clave, si se realiza propagación de clave; dependerá de cómo se haya transformado la relación entre entidades.

Software Libre Office. Fases diseño Base de Datos (Elaboración propia)

Antes de dar por finalizado el diseño lógico es necesario validar, y modificar si procede, el modelo o esquema lógico obtenido comprobando que no hay irregularidades que ocasionen redundancias e inconsistencias en las tablas obtenidas. Este proceso recibe el nombre de **normalización**, y consiste en aplicar una serie de reglas que validan esquemas lógicos basados en el modelo relacional. Una vez normalizado, el esquema lógico relacional podrá ser implementado en cualquier SGBD relacional.

En los siguientes apartados verás de forma detallada cómo realizar estas transformaciones, pero antes te indicamos mediante un ejemplo la notación utilizada para representar el modelo relacional obtenido.

Debes conocer

La manera de **representar las tablas obtenidas** en esta transformación y sus atributos puede ser básicamente de dos formas:

1.- **Forma textual.** De manera textual se escribe en una misma línea el nombre de la tabla en mayúsculas y entre paréntesis las columnas de esa tabla, columnas que provienen de los atributos de las entidades y/o transformación de relaciones entre entidades. Para denotar la **clave primaria** usamos un **subrayado continuo** y para denotar una **clave ajena** o foránea usamos un **subrayado discontinuo**.

2.- **Forma tabular.** Se dibuja la tabla y dentro de ella se escribe el nombre de las columnas que la componen. Al lado de las columnas que constituyen la clave primaria se pone la abreviatura PK y al lado de las columnas que son clave ajena o foránea se escribe la abreviatura FK. Esta forma textual se puede elaborar mediante el Software DIA. De igual forma, si utilizamos una herramienta Software de modelado de datos, como el Data Modeler de Workbench la representación el esquema lógico de la base de datos será de forma tabular.

De forma generaliza, **en esta unidad, usaremos la Forma textual en todos los ejemplos**. Además, indicaremos la manera en la que se relacionan las tablas dibujando una flecha que siempre saldrá desde una clave ajena o foránea hasta la clave primaria o tabla a la que referencia.

En la siguiente imagen puedes ver ambas formas de representar el modelo relacional.

Software Libre Office. *Paso ER al relacional* (Elaboración propia)

6.1.- Transformación de entidades.

¿En qué se transforma cada entidad del modelo E/R?

La norma general es que **cada entidad, fuerte o débil, se transforma en una tabla** que se denominará igual que la entidad de la que proviene.

Entidades fuertes

Por ejemplo, la transformación de la entidad PROPIERTARIO que es una entidad fuerte, se transformaría en una tabla del mismo nombre y con columnas los atributos de la entidad. Como clave primaria el atributo clave principal dni.

Libre Office y software DIA. Transformación Entidad Fuerte (Elaboración propia)

Entidades débiles

En el caso de **entidades débiles**, hay que tener en cuenta el tipo de dependencia:

Dependencia en existencia. En este caso, cuando solamente hay dependencia en existencia, se transforma de igual manera que una entidad fuerte. La tabla resultante tendrá como clave primaria la misma que la entidad débil y la clave primaria de la entidad fuerte se exportará a la tabla resultante de transformar la entidad débil, donde será clave ajena que referenciará a la tabla resultante de transformar la entidad fuerte

En la siguiente imagen puedes ver un ejemplo. La transformación de la entidad débil FAMILIAR que depende en existencia de ALUMNO, se transformaría en una tabla del mismo nombre y con columnas los atributos de la entidad. Como clave primaria su atributo clave principal **nif**, y se exporta la clave primaria nmatri de la tabla ALUMNO como clave ajena o foránea referenciando a la tabla ALUMNO.

Libre Office y software DIA. Transformación Entidad Débil Existencia (Elaboración propia)

Dependencia en identificación.- En este caso, además de la dependencia en existencia, la entidad débil depende en identificación de una entidad fuerte, por lo que 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.

En la siguiente imagen puedes ver un ejemplo. La transformación de la entidad EJEMPLAR que depende en existencia y en identificación de LIBRO (observa que EJEMPLAR tiene una clave débil numero que no permite identificar de forma única cada ocurrencia o instancia de

EJEMPLAR, pues si el libro de código A y el libro de código B tienen ambos 2 ejemplares, estarían numerados como 1, 2 en ambos casos, sin poder diferenciar si el ejemplar es del libro de código A o B), se transformaría en una tabla del mismo nombre y con columnas los atributos de la entidad. La clave primaria de la entidad fuerte LIBRO se exporta a EJEMPLAR, donde formará parte de su clave primaria que será codigo+numero, siendo a su vez código clave ajena que referenciará a la tabla LIBRO.

Libre Office y software DIA. Transformación Entidad Débil Identificación a Relacional (Elaboración propia)

Autoevaluación

Indica si la siguiente afirmación es verdadera o falsa.

Al pasar del modelo Entidad/Relación al modelo Relacional, las entidades débiles nunca pasan a tablas.

- Verdadero Falso

Falso

Todas las entidades, fuertes y débiles pasan a tablas.

6.2.- Transformación de atributos.

Los atributos generalmente los encontraremos en las entidades, pero también puede haber atributos en las relaciones. Veamos cómo transformar los atributos en ambos casos.

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. Tendremos que distinguir entre los diferentes tipos de atributos:

Atributo o atributos que conforman la clave principal. Pasarán a ser la clave primaria de la tabla y los denotaremos con subrayado continuo.

Atributos claves alternativas. Pasan a ser columnas de la tabla, claves alternativas. No los denotamos de forma especial.

Atributos descriptores. Pasan a ser columnas de la tabla.

Atributos compuestos. Cada uno de los atributos elementales que lo componen pasan a ser columnas de la tabla y desaparece el atributo compuesto.

Atributos derivados o calculados. Se deben eliminar de la tabla, o razonar el motivo por el que interesa mantenerlos.

Atributos multivaluados.- Un atributo multivaluado se puede resolver de las dos siguientes formas:

Crear un nueva tabla con el atributo multivaluado y la clave primaria de la entidad. La clave primaria de esta nueva tabla estará compuesta por el atributo multivaluado y la clave primaria de la entidad que también será clave ajena. Esta es la forma general de resolver un atributo multivaluado.

Crear tantas columnas para el atributo multivaluado como valores pueda tomar, en el caso de que el número de valores posibles se sepa de antemano y sea un número no muy grande, por ejemplo 2 o 3 valores. En este caso existe la posibilidad de valores nulos en esas columnas cuando no se almacenen todos los posibles valores del multivaluado .

DIA y LibreOffice. Trasnformación atributos entidad (Elaboración propia)

En la imagen del ejemplo puedes ver como:

La clave principal, nmat, se transforma en la clave primaria de la tabla ALUMNO.

La clave alternativa, dni, se trasforma en una columna.

Los atributos descriptores, nombre, fenac, se transforman en columnas.

El atributo compuesto, dirección, se elimina y sus partes elementales, calle, num, pobla, van a columnas.

El atributo derivado, edad, se elimina.

El atributo multivaluado, habilidad, da lugar a la nueva tabla HABILIDAD con clave primaria nmat+habilidad y clave ajena nmat.

El atributo multivaluado, telef, da lugar a dos columnas, tel1, tel2, pues en el modelo E/R se indica que hay un máximo de 2 teléfonos.

Atributos de las relaciones

En el siguiente apartado vas a ver en detalle como se transforma una relación o inter-relación, pero como esas relaciones pueden tener atributos, en este punto veremos un ejemplo para que sepas cómo se transforman los atributos que aparecen en las relaciones.

Cuando una relación tiene atributos nos podemos encontrar dos casos:

La relación se transforma en tabla. En este caso todos sus atributos pasan a ser columnas de esa tabla.

La relación se transforma mediante propagación de clave y desaparece la relación. En este caso sus atributos migran junto a la clave, a la tabla destino que corresponda.

DIA y Libre Office. Transformación atributos relaciones (Elaboración propia)

De este ejemplo, lo que interesa es ver cómo se transforman los atributos que aparecen en las relaciones ingresa y cursa, esto es, como se transforman los atributos: nota, **incidencias** y fechai.

En la imagen del ejemplo puedes ver como:

La relación cursa se transforma en tabla, y sus atributos, nota e **incidencias** pasan a ser columnas de esa nueva tabla.

La relación ingresa se transforma mediante propagación de clave y desaparece. En este caso el atributo fechai migra o se exporta junto a la clave, a la tabla ALUMNO.

No te preocupes ahora por saber el motivo por el que una relación se ha transformado en tabla y la otra no. Eso lo vamos a ver en detalle y con ejemplos en el siguiente apartado.

Autoevaluación

Indica si la siguiente afirmación es verdadera o falsa.

Al pasar del modelo Entidad/Relación al modelo Relacional los atributos multivaluados, de forma general, se transforman en una nueva tabla.

Verdadero Falso

Verdadero

De forma general el atributo multivaluado da lugar a una nueva tabla, salvo excepciones en las que se sepa que ese atributo como mucho tomará 2 o 3 valores, por ejemplo, y entonces se crean 2 o 3 columnas en la tabla de la entidad para poder almacenarlos. En este caso hay posibilidad de que aparezcan valores nulos.

6.3.- Transformación de relaciones o inter-relaciones.

¿Cómo se transforma una relación del diagrama E/R al modelo Relacional? La transformación va a depender de la cardinalidad de la relación.

Cardinalidad N:M

Una relación con **cardinalidad N:M siempre se transformará en una tabla nueva** que tendrá una clave primaria compuesta, formada por 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 también clave ajena que referenciará a la tabla donde este atributo es clave primaria.

Si la relación contiene atributos, éstos pasarán a formar parte de la nueva tabla, y en este caso debemos comprobar si la clave primaria compuesta de la nueva tabla es suficiente, pues **a veces es necesario añadir a esa clave primaria algún atributo procedente de la relación**, especialmente si es de tipo fecha.

EJEMPLO 1.-

En el siguiente ejemplo, observa que la relación cursa se ha transformado en una tabla cuya clave primaria es compuesta, formada por la concatenación de las claves primarias de las tablas que relaciona (nmat+codigo). Y además, cada una de ellas es clave ajena para referenciar a las tablas de procedencia.

DIA y Libre Office. Transformación Relación N:M (Elaboración propia)

EJEMPLO 2.-

Veamos ahora un ejemplo en el que la relación cursa tiene además el atributo eval, lo que denota que interesa tener la nota e **incidencias** separadas por evaluación. En este caso, observa que se hace necesario añadir el atributo eval como parte de la clave primaria de la nueva tabla cursa, para poder identificar las tres evaluaciones del mismo alumno en la misma asignatura.

DIA y Libre Office. Transformación relaciones N:M caso 2 (Elaboración propia)

Cardinalidad 1:N o N:1

Una **relación con cardinalidad 1:N o N:1 se transforma** según alguna de las siguientes dos soluciones:

- 1) **Propagando la clave** principal de la entidad de cardinalidad máxima N, como clave ajena, a la tabla que se genera de transformar la entidad de cardinalidad máxima 1. Si existen atributos propios de la interrelación o relación, éstos también se propagan a la tabla a la que se ha propagado la clave, como clave ajena.

2) Transformar la relación en una nueva tabla, donde la clave primaria de la nueva tabla, será solo la clave primaria de la entidad de cardinalidad máxima 1. Y serán claves ajenas las claves primarias de las entidades que relaciona. La nueva tabla acogerá además los posibles atributos de la relación.

Para **decidir entre una u otra solución** debemos tener en cuenta la cardinalidad mínima de la entidad que tiene cardinalidad máxima 1 y ver si es 1 o 0, esto es:

Si esa **cardinalidad es (1,1)** optaremos por la **solución 1**) propagar la clave principal de la entidad de cardinalidad máxima N a la otra entidad, donde será clave ajena. Los posibles atributos de la relación se pasarán a la entidad receptora de la clave ajena. O dicho de otra forma: '*se exporta la clave primaria de la entidad al lado de la cardinalidad (1,1) a la tabla generada por la otra entidad, donde será clave ajena para referenciar a su tabla de procedencia. Se exportan en el mismo sentido los posibles atributos de la relación*'.

DIA y Libre Office. Transformación relaciones N:1 (Elaboración propia)

Observa que en este ejemplo, la entidad ALUMNO tiene una cardinalidad (1,1), lo que significa que una ocurrencia de alumno está relacionada como mínimo y como máximo con un curso, por lo que se propaga la clave principal, código, de la entidad de cardinalidad máxima N que es CURSO, como clave ajena a la tabla ALUMNO. Como hay obligatoriedad, esa clave ajena nunca podrá tomar valor NULL. También se propaga el atributo fecha_matri de la relación a la tabla ALUMNO.

Si esa **cardinalidad es (0,1)** optaremos por la **solución 2**) crear una nueva tabla que tendrá como clave primaria la clave primaria de la entidad con cardinalidad máxima 1, y como claves ajenas las claves primarias de las entidades que relaciona. La nueva tabla acogerá los posibles atributos de la relación. La justificación de crear en este caso una nueva tabla, en vez de propagar clave, es porque con la creación de esta nueva tabla eliminamos los valores NULL en la clave ajena que puedan aparecer, ya que no todas las ocurrencias de las dos entidades estarán relacionadas, ya que hay participación opcional en la entidad de cardinalidad (0,1).

DIA y Libre Office. Transformación N:1 caso 2 (Elaboración propia)

Observa que en este caso, la entidad VEHICULO tiene una cardinalidad (0,1), (participación opcional o parcial en esa relación), lo que significa que no todos los vehículos han sido vendidos por un empleado. En este caso, optamos por transformar la relación vende en una nueva tabla cuya clave primaria será la clave principal de la entidad VEHICULO, (entidad de cardinalidad máxima 1), ya que un vehículo solo puede ser vendido por un empleado y por lo tanto el valor de código no se puede repetir en esa tabla, y como claves ajenas tendrá a la clave principal de VEHICULO y la clave principal de EMPLEADO para poder referenciar a esas tablas. Como columnas adicionales en la tabla vende estará el atributo fecha_venta de la relación vende.

Tabla resumen paso relaciones 1:N o N:1 al modelo relacional

Cardinalidades de entidad	¿Cómo se pasa al modelo relacional?
(1,n) o (0,n) y (0,1)	CREAR nueva tabla con columnas las claves primarias de ambas entidades que serán claves ajenas para referenciar a esas entidades, y los propios atributos de la relación. La clave primaria

	de esta nueva tabla será la que corresponde a la entidad de cardinalidad máxima 1.
(1,n) o (0,n) y (1,1)	PROPAGAR la clave primaria de la entidad al lado de la cardinalidad (1,1) a la tabla de la otra entidad, siendo en la tabla destino clave ajena que referenciará a su tabla de procedencia. Los atributos de la relación también se propagan en el mismo sentido que la clave.

Una vez que has visto las dos posibles soluciones, hay que resaltar que en la mayoría de los casos cuando nos encontramos con una relación **N:1** o **1:N** (uno a muchos) en los que se podría resolver con la creación de una nueva tabla, también se utiliza a veces la **propagación de clave** de la entidad con cardinalidad máxima N hacia la entidad con cardinalidad máxima 1, sobre todo cuando se sabe que la mayoría de las ocurrencias de ambas entidades estarán relacionadas y no se generarán muchos valores NULL. En este caso, se debe argumentar ese hecho, como el motivo por el que se propaga clave.

Observa que si en el ejemplo anterior, optásemos por propagar la clave desde EMPLEADO a VEHICULO nos encontraríamos con que todos los vehículos no vendidos tendrían nulo o NULL el valor del NIF. Por ello en el caso de que tengamos una participación opcional (0,1) en la entidad de cardinalidad máxima 1, si optáramos por propagar clave (puede haber ocurrencias de la entidad VEHICULO sin relacionar (0,1)), se deberá permitir valores nulos en la clave ajena NIF. Además, se debe justificar que se ha propagado clave en vez de crear una tabla intermedia, debido por ejemplo a que se sabe que serán pocos los valores NULL que se generan en ese diseño según contexto de trabajo.

Cardinalidad 1:1

Una **relación con cardinalidad 1:1 se transformará** según alguna de las siguientes dos soluciones:

- 1) **Propagando la clave primaria** de una de las entidades, como clave ajena, a la tabla resultante de la otra entidad.
- 2) **Creando una nueva tabla.**

Para optar por una u otra opción debemos tener en cuenta las cardinalidades de las dos entidades relacionadas:

La **opción 1** se elegirá si al menos una cardinalidad de alguna entidad es (1,1), esto es, la participación de alguna de las entidades es total u obligatoria. Podemos tener dos casos:

- a. Si las cardinalidades de las entidades son (1,1) y (0,1) entonces la propagación se hará de la entidad al lado de la cardinalidad (1,1) a la entidad al lado de la cardinalidad (0,1).
- b. 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. La propagación es indiferente, y se hará atendiendo a los criterios de frecuencia de acceso (consulta, modificación, inserción, etc..) a cada una de las tablas en cuestión.

DIA y Libre Office. Transformación relaciones 1:1 (Elaboración propia)

Si observas las cardinalidades de las entidades, éstas indican que toda TESIS la dirige un PROFESOR, pero no todo profesor dirige alguna TESIS. Por lo tanto, lo correcto es exportar o propagar la clave primaria de PROFESOR a TESIS, donde será clave ajena, evitando con ello los valores nulos que se producirían en caso de realizar la propagación al contrario.

La **opción 2**) es recomendable si las cardinalidades de las entidades son (0,1) y (0,1), esto es, cuando la participación de las dos entidades en esa relación es opcional. En este caso crearemos una nueva tabla, con el mismo tratamiento visto para el caso de 1:N, pues en ambas entidades habrá ocurrencias no relacionadas con las de la otra entidad.

Tabla resumen paso relaciones 1:1 al modelo relacional

Cardinalidades de entidad	¿Cómo se pasa al modelo relacional?
(0,1) y (0,1)	CREAR nueva tabla con columnas las claves primarias de las tablas relacionadas, que serán claves ajenas para relacionar las tablas, y los atributos que tenga la relación.. La clave primaria de

	la nueva tabla será una de las claves primarias de alguna de las entidades que relaciona. Añadir los atributos propios de la relación.
(1,1) y (1,1)	PROPAGAR la clave primaria desde una entidad cualquiera hacia la otra convirtiéndose en clave ajena. Si hay atributos en la relación también se propagan o exportan en el mismo sentido.
(0,1) y (1,1)	PROPAGAR la clave primaria de la entidad al lado de la cardinalidad (1,1) a la tabla de la otra entidad, siendo allí clave ajena. Si hay atributos en la relación también se propagan o exportan en el mismo sentido.

Relaciones con grado diferente a 2.

Las relaciones **REFLEXIVAS** o de grado 1 se transforman siguiendo las reglas indicadas anteriormente para cada uno de los casos: N:M, 1:N y 1:1, con la salvedad de que en vez de dos entidades distintas se trata de la misma entidad relacionada con ella misma.

Vemos un ejemplo de cada caso.

EJEMPLO Reflexiva N:M.

En este ejemplo vemos una relación reflexiva de la entidad PIEZA con ella misma: una pieza está compuesta por 1 o varias piezas (1,n); y una pieza compone a ninguna o a varias piezas. (0,m). La cardinalidad de la relación es por tanto N:M y su transformación al modelo relacional implica crear una nueva tabla para la relación compone. En esta nueva tabla, la clave primaria está formada por la concatenación de la clave primaria codigo de la tabla PIEZA vista con los dos roles (pieza compuesta por otras piezas y pieza que compone a otras piezas). Como se trata de la misma columna y el nombre de una columna no se puede repetir en la tabla, le damos un nombre diferente y descriptivo del rol que representa, por ejemplo codigo_comp (código de la pieza que compone).

DIA y Libre Office. Transformación Reflexiva N:M (Elaboración propia)

EJEMPLO Reflexiva 1:N.-

En este ejemplo vemos una relación reflexiva de la entidad EMPLEADO con ella misma: una ocurrencia de empleado puede ser jefe de ninguno o varios empleados (0,n), y una ocurrencia de empleado tiene por jefe a ninguno o a uno (0,1). La cardinalidad de la relación reflexiva es por lo tanto 1:N y como la participación es opcional (0,1), para evitar nulos se optaría por la **solución 1**) Crear una nueva tabla, como ves en la imagen. Pero si en nuestro contexto se sabe que no se generan muchos nulos si se propaga clave, optaríamos por la **solución 2**) y así nos ahorraremos una tabla.

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

Vemos cómo se transformaría, por ejemplo, una relación ternaria en una nueva tabla. La nueva tabla estará formada por los atributos propios de la relación, y las tres claves primarias de las entidades relacionadas que serán cada una de ellas clave ajena que referencian a la tabla de origen correspondiente. La elección de la clave primaria se hará dependiendo de la cardinalidad de las relaciones.

Elección de la clave primaria en una relación ternaria o de grado 3

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.
M:1:1	Compuesta por la clave primaria de la entidad etiquetada con cardinalidad M 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.

EJEMPLO.

Vemos el ejemplo de una ternaria de cardinalidad **M:N:P**, donde vendedores venden productos en diferentes zonas.

Autoevaluación

Indica si la siguiente afirmación es verdadera o falsa.

En las relaciones de cardinalidad 1:N o N:1 siempre se propaga clave.

- Verdadero Falso

Falso

No siempre se propaga clave, ya que se debería transformar en una tabla en el caso de que una de las entidades tenga como cardinalidad (0,1) y se requiera no almacenar nulos en la correspondiente clave foránea o ajena.

Recomendación

En el siguiente enlace puedes acceder a un videotutorial para repasar mediante ejemplos el paso del modelo Entidad/Relación al modelo Relacional.

[Ejemplos de paso del modelo Entidad/Relación al modelo Relacional en Youtube](#)

6.4.- Transformación de jerarquías.

Las relaciones de herencia o ISA que se representan como jerarquías por generalizaciones o especializaciones, en las que aparecen entidades que son supertipos otras son subtipos tienen varias posibilidades para su transformación del modelo ERE al modelo Relacional, dependiendo del tipo de Jerarquía del que se trate.

En general hay tres formas de resolverlas:

Opción A).- Crear una tabla para el supertipo y una para cada uno de los subtipos, cada una con sus correspondientes atributos. La tabla supertipo contendrá los atributos comunes y como clave primaria la clave o identificador de la superclase. Las tablas correspondientes a los subtipos contendrán los atributos específicos y heredan como clave primaria la clave primaria de la superclase, que a su vez es clave foránea en la tabla del subtipo referenciando a la tabla del supertipo. Esta es la mejor opción desde el punto de vista semántico. Es válida para cualquier tipo de jerarquía (total o parcial, disjunta o solapada), pero no siempre la más eficiente, pues obliga a enlazar o reunir varias tablas.

Opción B).- Crear una tabla para cada subtipo que, además de sus atributos específicos, contendrá los atributos comunes del supertipo, y con clave primaria la del supertipo. Sólo es aconsejable cuando la jerarquía es total (cada ocurrencia del supertipo debe pertenecer al menos a uno de los subtipos) y sin solapamiento (excluyente), y suele ser bastante eficiente. En caso de solapamiento daría lugar a redundancias. Las relaciones que mantuviese el supertipo se heredan en los subtipos, además de las relaciones propias que pueda tener cada subtipo.

Opción C).- Crear una única tabla para el supertipo y sus subtipos que recogería todos los atributos. En este caso se debe añadir un atributo nuevo que indique el subtipo al que se refiere cada ocurrencia o fila de la tabla. Puede ser conveniente en el caso de que haya pocos atributos diferentes entre los subtipos (en otro caso ocasionaría demasiados valores nulos) y si todos los subtipos participan en las mismas interrelaciones, y los subtipos son sin solapamiento. En este caso, pueden ofrecer una implementación más eficaz que las anteriores..

EJEMPLO.

Como ejemplo para ilustrar las tres opciones para transformar una jerarquía supongamos el siguiente: Una jerarquía total y exclusiva formada por la superentidad **EMPLEADO** y los subtipos **AUXILIAR** y **COMERCIAL**. En este caso, por el tipo de jerarquía que es (total y exclusiva) y sin considerar las posibles relaciones que pudiera haber con otras entidades, la transformación según las tres opciones vistas anteriormente quedaría de esta forma que se muestra en la siguiente imagen.

DIA y Libre Office. Transformación Jerarquías (Elaboración propia)

Autoevaluación

Indica si la siguiente afirmación es verdadera o falsa.

Una jerarquía parcial y sin solapamiento es correcto trasnformarla al Modelo Relacional en una tabla para cada uno de los subtipos y desaparece el supertipo. Por ejemplo, el supertipo **AULA** y los subtipos **AulaTIC** y **AulaLAB**, sabiendo que puede haber otros tipos de Aulas.

Sugerencia

- Verdadero Falso

Falso

Al ser la jerarquía parcial, significa que hay instancias del supertipo que no se corresponden con ningún subtipo, por lo que si eliminamos el supertipo no se podrían representar esas instancias. Por ejemplo una jerarquía parcial donde el supertipo esa **AULA** y los subtipos **AulaTIC**, **AulaLAB**; si eliminamos **AULA**, no se podría representar por ejemplo las Aulas normales u otro tipo de Aula que no fuera ni **AulaTIC** ni **AulaLAB**.

Debes conocer

Para ampliar la información sobre las reglas de transformación vistas en los apartados anteriores pulsa en el siguiente enlace.

[Ampliación reglas de transformación del modelo E/R al modelo relacional](#) (pdf - 0,26 MB)

6.5.- Ejemplos resueltos: Paso del modelo E/R al modelo Relacional.

Una vez que ya sabes cómo se transforman las entidades, atributos, relaciones y jerarquías al modelo Relacional, puedes afrontar la fase de diseño lógico de una base de datos completa, transformando su modelo ER a su modelo relacional o MR.

En la **fase del diseño lógico**, aplicaremos las reglas de transformación que has estudiado en los apartados anteriores. Los pasos que te recomendamos que sigas para **pasar del modelo Entidad/Relación al modelo Relacional** son los siguientes:

1.- Paso a tablas de las entidades y sus atributos, con especial atención a los atributos especiales y los casos de dependencia en identificación.

DIA y Libre Office. *ModeloER-ModeloREL* (Elaboración propia)

2.- Paso a tablas de las Jerarquías, si las hay.

3.- Resolver las relaciones entre entidades pasando a tablas o propagando clave, y sus posibles atributos.

4.- Representar el conjunto de tablas finales identificando las claves primarias de las tablas (subrayado continuo) y las claves ajena (subrayado discontinuo). Se deben unir las tablas con flechas que siempre salen de una clave ajena o foránea hacia la tabla que referencian o bien su clave primaria. Esto ayuda a un mejor entendimiento del modelo obtenido y tener claro la forma en la que se relacionan las tablas entre sí.

Siguiendo los pasos indicados anteriormente te mostramos con un **videotutorial** el paso a tablas del modelo **Entidad/Relación** de la base de datos VETERINARIA.

Paso de modelo ER a Relacional - VETERINARIA

Recuerda que en el apartado 4.2 de esta unidad obtuvimos el modelo **ER** para la base de datos VETERINARIA:

- 1.- A partir de los requerimientos del problema identificamos las entidades, atributos y relaciones entre entidades.
- 2.- Identificamos las claves candidatas y principal de cada entidad, y realizamos el estudio de cardinalidades de las entidades y las relaciones.
- 3.- Una vez realizado ese estudio o análisis representamos el diagrama **ER** de la base de datos mediante el software DIA, cuyo modelo final era el siguiente:

Modelo Entidad/Relación

Software DIA. *Modelo ER base de datos VETERINARIA* (Elaboración propia)

Ahora vamos a obtener el modelo relacional de la base de datos pasando a tablas las entidades, los atributos a columnas de las tablas y las relaciones a tablas o bien propagando clave.

Modelo Relacional

<https://www.youtube.com/embed/QPfsZrrchJw>

Isabel Cruz. *Descripción textual alternativa del video*. (Dominio público)

A continuación te facilitamos varios ejemplos resueltos, cuyo modelo Entidad/Relación lo obtuvimos en el punto 4.2: las bases de datos: PELICULAS, VIDEOTECA y CAMPEONATO.

Recuerda que su **modelo Entidad/Relación** lo obtuvimos tras realizar el **diseño conceptual** a partir de los siguientes pasos:

- 1.- Identificación de las entidades (fuertes y débiles), las relaciones entre las entidades y los atributos (de entidades y posiblemente de relaciones), especificando los atributos especiales: atributos calculados o derivados, compuestos y multivaluados.
- 2.- Estudio e identificación de las cardinalidades de las entidades y después las cardinalidades de las relaciones.
- 3.- Identificación de las claves candidatas y determinar la clave primaria de cada entidad.
- 4.- Estudio de otras características del modelo Entidad/Relación Extendido, como las Jerarquías, así como de otras características que no es posible representar con el modelo ERE.
- 5.- Elaborar el diagrama E/R o ERE utilizando el software DIA.

La propuesta que te hacemos es que intentes resolver cada uno de los supuestos por ti mismo y después compares tu solución con la solución propuesta.

Películas

El diseño conceptual de ese ejercicio, paso a paso, hasta obtener el modelo Entidad/Relación lo tienes en el apartado 4.2 de esta unidad.

Vamos a realizar el **diseño de una base de datos relacional para guardar información sobre películas y los actores que actúan o participan en esas películas**.

A continuación se indican las especificaciones y requerimientos que debe cumplir la base de datos. A partir de ellos debes realizar su diseño conceptual y su diseño lógico.

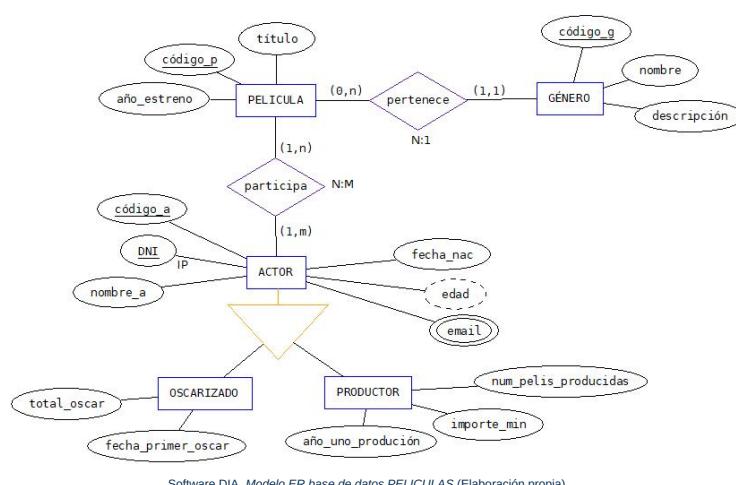
- 1.- Una película se caracteriza por: código, título, año de estreno, género.
- 2.- Una película pertenece obligatoriamente a un género y solo a uno. Pero de un determinado género puede haber varias películas o ninguna.
- 3.- El género de una película se caracteriza por un código, nombre y descripción.
- 4.- En una película participan uno o varios actores y cada actor puede participar en una o varias películas.
- 5.- De los actores interesa su código, DNI, nombre, fecha de nacimiento, edad, varios email.
- 6.- Hay que considerar dos posibles tipos de actores, aunque puede haber otros tipos:

Actores oscarizados: de los que hay que almacenar además, el total de oscar y la fecha del primer oscar.

Actores productores: de los que hay que almacenar además, el numero de películas producidas, aportación mínima, y primer año de producción.

Un actor oscarizado también puede ser productor.

Realizado el estudio y análisis de estos requerimientos durante el diseño conceptual obtuvimos el siguiente modelo Entidad / Relación.



Software DIA. Modelo ER base de datos PELÍCULAS (Elaboración propia)

Ahora debes realizar el diseño lógico y obtener el modelo Relacional.

[Mostrar retroalimentación](#)

[Solución base de datos PELÍCULAS](#) (pdf - 0,25 MB)

Videoteca

El diseño conceptual de ese ejercicio, paso a paso, hasta obtener el modelo Entidad/Relación lo tienes en el apartado 4.2 de esta unidad.

Un Videoclub Vintage dispone de copias de películas en formato Blu-ray que ofrece en alquiler. Nos piden **realizar el diseño de una base de datos relacional para guardar información sobre las películas que se alquilan**, sus actores y los clientes que las alquilan.

A continuación se indican las especificaciones y requerimientos que debe cumplir la base de datos. A partir de ellos debes realizar su diseño para obtener su modelo conceptual mediante un diagrama Entidad/Relación y su modelo lógico.

- 1.- Una película se caracteriza por un código, título, año de estreno y género.
- 2.- Una película pertenece obligatoriamente a un género y solo a uno. Pero de un determinado género puede haber varias películas o ninguna.
- 3.- El género de una película se caracteriza por un código, nombre (que es único) y su descripción.
- 4.- Todas las películas tienen una o varias copias y pueden ser alquiladas. Una copia es de una y solo una película. Si se descataloga la película no tiene sentido mantener la copia.
- 5.- Las copias de cada película se identifican con un número sucesivo (1, 2, 3, 4 ...) y de ellas se guarda la fecha de compra y estado de conservación.
- 6.- Las copias de la películas se alquilan a clientes. De los clientes interesa su DNI, nombre, fecha_nacimiento, edad y varios email.
- 7.- Los clientes puedes ser de dos tipos:
 - 7.1.- Socios: de ellos se guarda además un código, fecha de alta, descuento y cuota.
 - 7.2.- Eventuales: de estos clientes se guarda la fecha de su primer alquiler.
- 8.- Tanto los clientes socios como los eventuales han alquilado al menos una copia, pudiendo alquilar varias, y cada copia puede haber sido alquilada varias veces o ninguna.
- 9.- Al alquilar una copia interesa saber la fecha de alquiler y precio de alquiler.
- 10.- Los clientes eventuales pueden haber sido invitados por un único cliente eventual, y un cliente eventual puede invitar a varios cliente eventuales o a ninguno.

Realizado el estudio y análisis de estos requerimientos durante el diseño conceptual obtuvimos el siguiente modelo Entidad /Relación.

Software DIA. Modelo ER base de datos VIDEOTECA (Elaboración propia)

Ahora debes realizar el diseño lógico y obtener el modelo Relacional.

[Mostrar retroalimentación](#)

[Solución base de datos VIDEOTECA](#) (pdf - 0,29 MB)

Campeonato

El diseño conceptual de ese ejercicio, paso a paso, hasta obtener el modelo Entidad/Relación lo tienes en el apartado 4.2 de esta unidad.

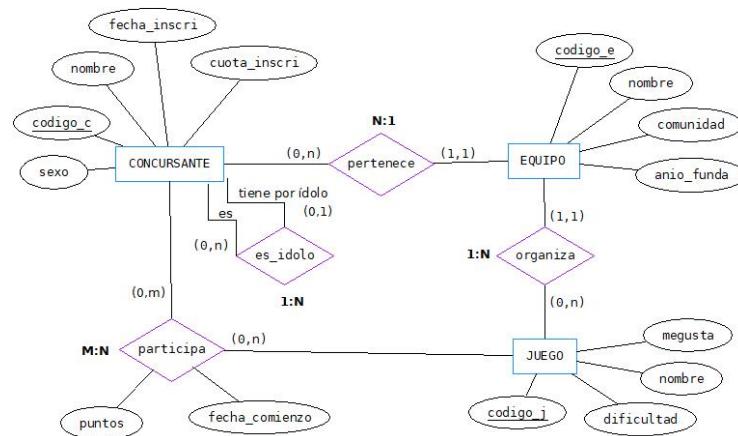
La Asociación Andaluza de Videojuegos o e-Sports necesita gestionar diferentes campeonatos de juegos online mediante una base de datos que almacene información sobre los concursantes, sus equipos y los diferentes juegos en los que participan los concursantes.

Se debe realizar el diseño de una base de datos relacional a partir de las siguientes especificaciones o requisitos que debe cumplir:

- 1.- De los concursantes se desea almacenar un código, su nombre, sexo, fecha de inscripción y cuota de inscripción.
- 2.- Cada concursante pertenece obligatoriamente a un equipo y solo a uno. Pero en un equipo puede haber ninguno o varios concursantes.
- 3.- Cada equipo se caracteriza por un código, nombre, comunidad y año de fundación.
- 4.- Los equipos, no todos, pueden organizar varios juegos. Cada juego es organizado por uno y solo un equipo.

- 5.- Los juegos se caracterizan por su código, nombre, nivel de dificultad y número de me gusta.
- 6.- Los concursantes pueden participar en ninguno o varios juegos, independientemente del equipo al que pertenezcan.
- 7.- En un juego pueden participar ninguno o varios concursantes.
- 8.- Cuando un concursante participa en un juego, se anota la fecha de comienzo en la que inició ese juego y se guardan también los puntos que va acumulando en ese juego.
- 9.- Los concursantes pueden designar a otro concursante como su ídolo, de manera que un concursante puede ser ídolo de ninguno o varios concursantes, pero como ídolo solo puede tener a un único concursante o a ninguno.

Realizado el estudio y análisis de estos requerimientos durante el diseño conceptual obtuvimos el siguiente modelo Entidad /Relación.



Software DIA. Modelo ER base de datos CAMPEONATO (Elaboración propia)

Ahora debes realizar el diseño lógico y obtener el modelo Relacional.

[Mostrar retroalimentación](#)

[Solución base de datos CAMPEONATO](#) (pdf - 0,34 MB)

7.- Normalización.

Caso práctico



[Jonny Goldstein \(CC BY\)](#)

Juan recuerda a **Vindio** y **Noiba**, que una vez que hayan conseguido establecer un modelo lógico para la base de datos que recoja toda la información necesaria sobre el funcionamiento del taller deben plantearse el siguiente paso que es el diseño físico de la base de datos, pero aunque el paso del modelo entidad-relación al modelo relacional se haya hecho aplicando ciertas reglas, las tablas obtenidas pueden presentar problemas que se derivan de las restricciones que pueden existir entre atributos de distintas tablas o entre atributos de la misma tabla. A esas restricciones se les denomina **dependencias**.

Así pues tendremos que dar un paso más en nuestro diseño lógico y aplicar una técnica denominada **normalización** que consiste en verificar el modelo para eliminar las inconsistencias que pudiera tener.

Vamos a estudiar las dependencias existentes en los atributos para conseguir que nuestras tablas se encuentren en un nivel de normalización aceptable.

¿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 y no se van a producir inconsistencias ni anomalías al manipular los datos. Esta comprobación la realizaremos mediante el proceso de normalización.

¿Y en qué consiste el proceso de normalización?

El proceso de normalización.

Si en nuestro diseño de base de datos hemos seguido el modelo Entidad/Relación parece que deberíamos obtener tablas bien estructuradas pero es posible que un buen diseño produzca tablas defectuosas. Un buen diseño de base de datos debe ser acorde con las buenas estructuras de datos. Diseñar tablas bien estructuradas nos ayudará a controlar la redundancia de los datos y evitar la inconsistencia. Estos problemas, cuando existan, pueden ser eliminados mediante el proceso de normalización.

La Normalización es un proceso que consiste en asignar atributos a las entidades verificando que se cumplan ciertas reglas. Reduce las redundancias de datos y ayuda a eliminar las anomalías que se derivan de las redundancias.

La normalización se basa en lograr la independencia de los datos con respecto a las aplicaciones que los usan, obteniendo unas tablas con una estructura óptima y eficaz, de manera que se optimizan los procesos de inserción, modificación y borrado en la base de datos.

Veremos que 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 los veremos en el siguiente apartado.

¿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.

Para referirnos a algunas de estas formas normales se suele usar la abreviatura siguiente: **1FN, 2FN, 3FN, BCFN, 4FN, 5FN**

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 algunas aplicaciones, sin embargo, llegaremos hasta la 4FN, y en aplicaciones muy especializadas de investigación estadística será necesario llegar más allá.

Este proceso se basa en la descomposición sin pérdida de las tablas que están en una forma normal inferior, obteniendo una forma normal superior. Se descompone la tabla en otras con menor cantidad de atributos, sin que haya pérdida de información.

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

7.1.- Dependencias funcionales y formas normales.

Para aplicar correctamente la normalización es necesario partir del concepto de dependencia.

La dependencia es un conjunto de restricciones que se imponen a determinados atributos de las tablas.

[Edrujil \(CC BY-SA\)](#)

Se denominan **dependencias** a las relaciones que existen entre los atributos en el mundo real y que son recogidas en el modelo lógico de la base de datos.

Vamos a estudiar los diferentes **tipos de dependencias**:

Dependencias funcional.

Dependencias funcional completa.

Dependencia funcional elemental.

Dependencia funcional trivial.

Dependencia transitiva.

Dependencia multivaluada.

Dependencia de unión.

Dependencia funcional.

Las **dependencias funcionales** se producen cuando tenemos una tabla con una serie de atributos. Se dice que un atributo tiene dependencia funcional de otro cuando a cada valor del primero le corresponde un solo valor del segundo.

Para una definición más formal diremos que: Si X e Y son subconjuntos de atributos de una tabla, diremos que X tiene dependencia funcional de Y (o también que X determina a Y) si cada valor de X tiene asociado siempre un único valor de Y.

$$X \rightarrow Y \text{ (se lee: X determina a Y)}$$

Las dependencias funcionales pueden ser:

Dependencia funcional completa: cuando un atributo depende de otro que es un atributo compuesto (un conjunto de atributos), se dice que la dependencia funcional es completa si el atributo dependiente no depende de ningún subconjunto del atributo compuesto.

Es decir; en una dependencia funcional $X \rightarrow Y$, cuando X es un conjunto de atributos, decimos que la dependencia funcional es **completa** si solo depende de X, no de ningún subconjunto de X.

$$X \rightarrow Y$$

Dependencia funcional elemental: Una dependencia funcional es elemental si Y posee un único atributo.

Si tenemos una dependencia completa $X -> Y$ diremos que es una dependencia funcional elemental si Y es un atributo, y no un conjunto de atributos.

Dependencia funcional trivial. Es la dependencia que tiene un atributo con relación a otro compuesto cuando forma parte de él. Se produce cuando Y es un subconjunto de X

Una dependencia funcional $X \rightarrow Y$ es trivial cuando Y es parte de X. Esto ocurre cuando X es un conjunto de atributos y Y es un subconjunto de X.

Dependencia transitiva.

Es una dependencia de un atributo no principal de otro no principal.

Cuando se tienen tres atributos X, Y y Z se cumple que $X \rightarrow Y$, $Y \rightarrow Z$ pero no $Y \rightarrow X$ (Y no determina X). Por tanto Z tiene dependencia transitiva con respecto a X, a través de Y.

Dependencia multivaluada.

Se producen cuando un atributo puede tomar múltiples valores en la misma tabla, independientemente de los valores que tomen el resto de los atributos.

Existe dependencia funcional multivaluada si dados 3 atributos de una tabla, para cada valor del primer atributo existen múltiples valores del segundo y no existen ninguna relación entre el tercer atributo y el primero, a no ser a través del segundo atributo.

Se representa como: $X \rightarrow Y$ y se lee X multidetermina Y.

Este tipo de atributos implica redundancia ya que el resto de los atributos se repiten tantas veces como valores diferentes tenga el atributo multivalorado.

Dependencia en unión.

Para comprender este tipo de dependencias es necesario explicar previamente los conceptos:

Proyección: Creación de una tabla cuyos elementos forman un subconjunto de una tabla dada. Se incluyen todas las filas y algunas columnas.

Unión: Formar, a partir de dos tablas, una nueva con todos los campos de una de ellas y los registros de ambas, excepto los repetidos. Ambas tablas han de tener el mismo grado y las mismas columnas.

Se dice que hay dependencia de unión o producto si una tabla tiene dependencia de unión con varias de sus proyecciones y se puede obtener la tabla por medio de la unión de dichas proyecciones.

Autoevaluación

Partiendo de la siguiente relación, seleccionar la alternativa que refleje las dependencias existentes entre sus atributos:

MODULOS (ClaveModulo, NombreModulo, Horas, Aula, ClaveCiclo, NombreCiclo)

- ClaveMódulo → NombreMódulo, Horas, Aula, ClaveCiclo, NombreCiclo ClaveCiclo → NombreCiclo
- ClaveMódulo → NombreMódulo, Horas, Aula, ClaveCiclo, NombreCiclo
- ClaveMódulo → NombreMódulo, Horas, Aula ClaveCiclo → NombreCiclo
- ClaveCiclo → NombreCiclo, ClaveMódulo, NombreMódulo, Aula, Horas ClaveMódulo → NombreMódulo

Has seleccionado correctamente las dependencias funcionales y transitivas

Estudia mejor la respuesta.

Comprueba de nuevo si están incluidas todas las dependencias.

Revisa de nuevo las dependencias.

Solución

1. Opción correcta
2. Incorrecto
3. Incorrecto
4. Incorrecto

7.1.1.- Primera Forma Normal o 1FN.

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.

Una tabla está en **Primera Forma Normal (1FN)** sí, y sólo sí, todos los atributos de la misma contienen valores atómicos, es decir, que cada atributo debe contener un único valor del dominio.

En consecuencia los atributos en cada tabla de una base de datos no podrán tener una lista de valores (ni del mismo dominio ni de otro) y, por tanto, cada atributo tiene que tener un nombre único.

Las restricciones de la primera forma normal coinciden con las condiciones de una tabla en el modelo relacional por lo tanto siempre es obligatorio aplicar la 1FN.

Para aplicar esta forma normal nos podemos encontrar con los siguientes casos:

Un atributo compuesto.

Por ejemplo: El nombre del cliente: Juan Martínez Arce. Podemos considerar el nombre como un dato atómico Nombre o, si nos interesa, separarlo en Apellido1, Apellido2, Nombre, y quedaría:

CLIENTE(CodCliente

Libre Office (Elaboración propia)

Un atributo multivaluado.

Por ejemplo: CodCliente,, teléfono. En el caso de que quisieramos recoger distintos números de teléfono para cada cliente tendríamos que crear una nueva tabla que recoja los números de clientes y los CodCliente para relacionar con la anterior, y quedaría: (observa que la clave primaria de la tabla TELEFONOS es compuesta)

CLIENTE(CodCliente, Nombre, Apellido1, Apellido2) TELEFONOS(CodCliente(fk), Teléfono

O bien crear tantas columnas para guardar el teléfono como nos interesaran, por ejemplo para 3 teléfonos quedaría:

CLIENTE(CodCliente

Otro ejemplo:

CLIENTES(<u>CodCliente</u>, Nombre, Apellido1, Apellido2, Telef1, Telef2, Telef3).

De la tabla de CLIENTES de nuestro taller tomamos como ejemplo los atributos CodCliente y Nombre. Podemos determinar que a cada código de cliente le corresponde un único nombre de cliente. Por tanto hay una dependencia funcional entre ellos: el código de cliente determina el nombre de ese cliente (no al revés). Eso no quiere decir que si nos sabemos el Código de un cliente sepamos también el nombre, sino que para cada código de cliente solo puede haber un nombre.

CodCliente → Nombre

Como vemos en la mayoría de los casos no será necesario aplicar este proceso ya que nuestras tablas ya se encontrarán en 1FN si hemos hecho un buen diseño conceptual y lógico para la base de datos.

Autoevaluación

¿Está la siguiente tabla en 1FN? Selecciona la respuesta correcta.

CodigoAlumno	Apellido1	Apellido2	Nombre	Asignaturas
Asir_M_01	Álvarez	Salces	Carmen	Redes, Base de datos
Asir_M_02	Calle	Martín	Javier	Redes, Base de datos, Sistemas Operativos

- Sí, porque el Nombre del alumno que es un atributo compuesto se ha dividido en datos atómicos.
- No, porque no hay dependencia entre la clave CodigoAlumno y las Asignaturas.
- No, porque el atributo Asignaturas es un atributo multivaluado.
- Sí, porque todos los atributos tienen dependencia respecto de la clave CodigoAlumno.

Esa división es correcta pero debes revisar la tabla de nuevo..

La 1 FN no analiza las dependencias entre los atributos.

Así es, para que una tabla esté en 1FN a cada atributo le debe corresponder un único valor.

La 1 FN no se basa en la existencia o no de dependencias sino en que los atributos sean atómicos.

Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta
4. Incorrecto

7.1.2.- Segunda Forma Normal o 2FN.

Una tabla está en **segunda forma normal** o 2FN cuando está en 1FN y además todos los atributos que no forman parte de la clave principal tienen dependencia funcional completa de la clave y no de parte de ella.

Es obvio que una tabla que esté en 1FN y cuya clave esté compuesta por un único atributo, estará en 2FN.

Significa, por tanto, que en una relación sólo se debe almacenar información sobre un tipo de entidad, y que se traduce en que los atributos que no aporten información directa sobre la clave principal deben almacenarse en una relación separada.

Ejemplo.

[Everaldo Coelho \(GNU/GPL\)](#)

En la tabla PRODUCTIVIDAD recogemos los tiempos que utiliza cada empleado en realizar determinadas reparaciones con relación a los tiempos previstos, para calcular la productividad de ese empleado.

PRODUCTIVIDAD (<u>NumReparacion, CodEmpleado, FechaActuación</u> NombreEmpleado, PuestoTrabajo, SalarioBase, NumHoras, DescriReparación, TiempoPrevisto, ImporteReparación, TiempoEmpleado)

Estudiamos las Dependencias:

<u>NumReparacion, CodEmpleado, FechaActuación</u> → TiempoEmpleado
CodEmpleado → NombreEmpleado, PuestoTrabajo, SalarioBase
NumReparación → DescriReparación, TiempoPrevisto, ImporteReparación

Tenemos dependencias parciales, es decir, dependencias basadas en una parte de la clave primaria que producirán redundancias al insertar nuevas filas en la tabla. Para eliminar esas dependencias crearemos nuevas tablas que recojan las dependencias:

PRODUCTIVIDAD (<u>NumReparacion, CodEmpleado FechaActuación</u>, TiempoEmpleado).
EMPLEADOS(<u>CodEmpleado</u>, NombreEmpleado, PuestoTrabajo, SalarioBase).
REPARACIONES(<u>NumReparacion</u>, DescriReparación, TiempoPrevisto, ImporteReparación).

Estas tablas están en 2FN ya que:

Están en 1FN

Ningún atributo depende solamente de una parte de la clave primaria.

Aunque nuestras tablas se encuentren en 2FN, todavía presentan anomalías; ya que el Salario Base de todos los empleados que tengan el mismo puesto de trabajo coincide, por tanto, si queremos hacer algún cambio en el salario base asociado a un puesto de trabajo determinado, debemos hacer el cambio en todos los empleados que tengan ese puesto o generaremos inconsistencias. Esto lo resolveremos con la Tercera Forma Normal.

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

Autoevaluación

La siguiente relación o tabla no está en 2 FN. De las opciones planteadas selecciona la que cumpla las condiciones necesarias.

CONSULTA (CodPaciente,CodMedico, FechaConsulta, NombrePaciente, Diagnóstico, Tratamiento, Especialidad)

- CONSULTA (**CodPaciente**, NombrePaciente, Diagnóstico, Tratamiento, FechaConsulta) MEDICO (**CodMedico**, Especialidad)
- CONSULTA (**CodPaciente**, **CodMedico**, **FechaConsulta**, Diagnóstico, Tratamiento) MEDICO (**CodMedico**, Especialidad) PACIENTE (**CodPaciente**, NombrePaciente)
- MEDICO (**CodMedico**, Especialidad) PACIENTE (**CodPaciente**, NombrePaciente) CONSULTA (**FechaConsulta**, Diagnóstico, Tratamiento)

Revisa de nuevo las tabla CONSULTA para ver si todos los atributos dependen de la clave

Eso es, ningún atributo depende de una parte de la clave primaria, sino de la clave completa.

Estudia de nuevo las dependencias en la tabla CONSULTA.

Solución

1. Incorrecto
2. Opción correcta
3. Incorrecto

7.1.3.- Tercera Formal o 3FN.

Una tabla o relación está en **tercera forma normal o 3FN** si está en **2FN** y no existen atributos que no pertenezcan a la clave primaria que puedan ser conocidos mediante otro atributo que no forme parte de la clave primaria. Es decir, que no existan dependencias funcionales transitivas.

Consiste por tanto en eliminar las dependencias funcionales transitivas.

[Edruui \(CC BY-SA\)](#)

Ejemplo.

Siguiendo con el **ejemplo** anterior vemos que existen dependencias transitivas: ya que PuestoTrabajo determina el SalarioBase y el PuestoTrabajo no determina a CodEmpleado.

EMPLEADOCodEmpleado, NombreEmpleado, PuestoTrabajo, SalarioBase)
PuestoTrabajo → SalarioBase

Para eliminar esta dependencia guardamos esos atributos en una nueva tabla. Sin embargo dejamos el determinante (PuestoTrabajo) en la tabla original como clave ajena para poder relacionar ambas tablas.

De esta forma nuestro ejemplo quedaría en 3FN con las siguientes tablas:

PRODUCTIVIDAD(<u>NumReparacion, CodEmpleado, FechaActuación</u>, TiempoEmpleado).
REPARACIONES(<u>NumReparacion</u>, Descripción, TiempoPrevisto, ImporteReparación).
EMPLEADOS(<u>CodEmpleado</u><u>, </u>NombreEmpleado, PuestoTrabajo(fk))
PUESTOS(<u>PuestoTrabajo</u><u>, </u>SalarioBase)

Estas tablas están en 3FN ya que:

Están en 2FN
No existen dependencias transitivas

Observaciones.

Aunque las tablas anteriores están en 3FN y no presentan dependencias parciales ni transitivas aún podemos mejorar el diseño teniendo en cuenta los siguientes aspectos:

La clave principal de la tabla PUESTOS puede dar lugar a errores si se cometen imprecisiones al llenar los datos. Sería conveniente añadir un CódigoPuesto que evite problemas de pérdida de integridad, pero se crearán dependencias:

CodPuesto → PuestoTrabajo, SalarioBase

PuestoTrabajo → SalarioBase

No se trata de una dependencia transitiva ya que el PuestoTrabajo determina el SalarioBase sin ser clave principal, pero resulta que PuestoTrabajo determina también a CodPuesto (son equivalentes)

El campo NombreEmpleado de la tabla EMPLEADOS, deberíamos descomponerlo en EmpApellido1, EmpApellido2 y EmpNombre para cumplir con el requerimiento de atomicidad y facilitar algunas operaciones como listas ordenadas por apellidos, etc.

En la tabla EMPLEADOS faltan campos como FechaContratación, CuotaSegSocial, etc.

En la tabla PRODUCTIVIDAD se supone que un empleado solo puede introducirse una actuación por cada empleado y cada tipo de reparación. Si el empleado realiza dos actuaciones correspondientes a un mismo tipo de reparación en el mismo día no podría registrarse. Es conveniente añadir un CódigoActuación que sería la clave principal y mantendríamos NumReparación y CodEmpleado como claves ajenas.

Autoevaluación

Rellena los espacios en blanco con las palabras correctas:

En la relación ENFERMEDADES (IdEnfermedad, Descripción, Síntomas, Medicamento, EfectosSecundarios), podemos decir que todos los dependen funcionalmente de la por eso la relación está en pero no está en ya que existe una dependencia funcional . El atributo puede ser conocido a través de valor del atributo .

Para que una tabla esté en 3FN, además de estar en 2FN, ningún atributo que no forme parte de la clave puede determinar a otro que tampoco sea clave.

Recomendación

En el siguiente enlace puedes ver un videotutorial en Youtube sobre las formas normales 1FN, 2FN y 3FN y aplicarlas a ejemplos concretos.

[Repaso de las formas normales 1FN, 2FN y 3FN mediante ejemplos](#)

7.1.4.- Forma Normal de Boyce-Codd o FNBC.

Una tabla está en **Forma Normal de Boyce-Codd** o **FNBC** si todo determinante en ella es una clave candidata. Dicho de otra forma: Una tabla está en FNBC si cualquier atributo sólo facilita información sobre claves candidatas, y no sobre atributos que no formen parte de ninguna clave candidata.

La FNBC es un caso especial de la 3FN, por tanto en la mayoría de los casos al aplicar la 3FN se cumplen los requerimientos de FNBC, pero no siempre.

Realmente se trata de ver, si existen atributos no clave que determinen a una parte de la clave.

Una tabla está en 3FN y no está en FNBC cuando un atributo que no es clave determina a un atributo de la clave

Ejemplo 1.

Si X, Y, Z y F son atributos de una tabla, y suponiendo que la clave primaria es X+Y. TABLA(X,Y,Z,F)

Everaldo Coelho (Dominio público).

Si ocurriera los siguientes:

$$X+Y \rightarrow Z, F$$

$$Z \rightarrow Y$$

Esta tabla estaría en 3FN porque no tiene dependencias ni parciales, ni transitivas, pero no está en FNBC.

Para corregir esto es necesario:

Cambiar la clave primaria de X+Y a x+z porque Z resulta ser determinante de Y.

Eliminar las dependencias parciales que se han creado ya que Y depende de Z:

$$X+Z \rightarrow F$$

$$Z \rightarrow Y$$

Ejemplo 2.

Tenemos una tabla con los atributos:

CALIFICACIONES(<u>CodAlumno, CodProfesor</u>, CodClase, Calificación)

Y las siguientes dependencias:

CodAlumno + CodProfesor → CodClase, Calificación

CodClase → CodProfesor.

El resultado para que las tablas estén en FNBC será:

CALIFICACIONES (<u>CodAlumno, CodClase</u>, Calificación)

CLASES (<u>CodClase</u>, CodProfesor).

Si una tabla tiene una sola clave candidata la 3FN y la FNBC son equivalentes.

Si la clave primaria está formada por un solo atributo y está en 3FN, ya está en FNBC

Autoevaluación

Dada la siguiente relación:

FACTURA (NumFactura, FechaFactura, CódigoCliente, ImporteTotal)

¿Podemos afirmar que cumple la FNBC?. Selecciona las respuestas acertadas

- No, porque no cumple la 3FN.

- Sí, porque cumple la 3FN y solo hay una clave candidata.

- Sí, porque la clave principal está formada por un único atributo y está en 3FN.

- No, porque un atributo que no es clave determina a un atributo de la clave.

- Sí, porque ningún atributo de la clave determina a un atributo que no es clave

[Mostrar retroalimentación](#)

Solución

1. Incorrecto
2. Correcto
3. Correcto
4. Incorrecto
5. Incorrecto

7.1.5.- Cuarta Forma Normal o 4FN.

Una tabla está en Cuarta Forma Normal o 4FN si está en FNBC y las únicas dependencias funcionales multivaluadas que existen son las dependencias funcionales de la clave con los atributos que no formen parte de la clave. Es decir que toda dependencia funcional multivaluada se determina por una clave candidata (se trata de dependencias triviales)

Es posible que nos encontremos con bases de datos mal diseñadas o creadas a partir de hojas de cálculo, en el que existan atributos con valores múltiples.

Las **dependencias multivaluadas** son una consecuencia de la 1FN, que prohíbe que un atributo de una fila tenga un conjunto de valores.

Si tenemos dos o más atributos multivaluados independientes en la misma tabla, tendremos que repetir todos los valores de uno de los atributos con cada valor del otro atributo para que las filas de la tabla sean consistentes.

Para evitar esto se descompone la tabla en otras dos, manteniéndose en cada una de ellas una dependencia múltiple, siendo la clave de las nuevas tablas la combinación de todos los campos. Veamos esto con un ejemplo.

Ejemplo.

Consideramos la tabla MECÁNICOS supongamos que se dan las siguientes alternativas:

Un mecánico puede trabajar en una avería con un ayudante.

Cada mecánico puede trabajar en varias averías. En este caso el mecánico ME-001 trabaja en las averías AV-555 y AV-552

Cada mecánico puede tener varios ayudantes. El mecánico ME-001 trabaja con los ayudantes AYU-23 y AYU-20.

Las averías y los ayudantes del mecánico no tienen relación directa.

MECANICO	AVERIA	MECANICO
ME-001	AV-555	AYU-23
ME-001	AV-552	AYU-20
ME-002	AV-551	AYU-18

[Everaldo Coelho, \(GNU/GPL\)](#)

En este caso la clave principal estaría formada por los tres atributos.

Para evitar las múltiples redundancias y los problemas de actualización de esta tabla que tiene dependencias multivaluadas será necesario dividir la tabla en dos. La clave de cada tabla estará formada por los dos atributos.

MECANICO	AVERIA	MECANICO	AYUDANTE
ME-001	AV-555	ME-001	AYU-23
ME-001	AV-552	ME-001	AYU-20
ME-002	AV-551	ME-002	AYU-18

7.1.6.- Quinta Forma Normal o 5FN.

La quinta forma normal se refiere a dependencias que son extrañas, las **dependencias de unión**. Tiene que ver con tablas que pueden dividirse en subtablas, pero que no pueden reconstruirse.

Se emplea cuando en una misma tabla existe información redundante, con pocos atributos o cuando una tabla posee una gran cantidad de atributos y se hace por ello inmanejable.

Se dice que una tabla está en **Quinta Forma Normal o 5FN** si está en 4FN y las únicas dependencias que existen son las dependencias de unión de una tabla con sus proyecciones relacionándose entre sí mediante la clave primaria o cualquier clave candidata.

Para conseguir que una tabla que está en 4FN esté en 5FN, se divide la tabla en tantas como sea necesario. Estas tablas tendrán en común los campos que formaban la clave principal en la tabla original.

Para conseguir que una tabla que está en 4FN que tenga gran cantidad de atributos o pocos atributos y muchos registros, esté en 5FN, se divide la tabla en tantas como sea necesario. Estas tablas tendrán en común con una o más de las otras tablas la existencia de una o más claves ajenas.

Ejemplo 1.

Everaldo Coelho and YellowIcon (GNU/GPL)

Tenemos una tabla EMPLEADOS que contiene la información sobre los empleados de nuestra empresa.

EMPLEADOS (CodEmpleado, DNI, Apellido1, Apellido2, Nombre, Dirección, CP, Teléfono, FechaNacimiento, EstadoCivil, NumHijos, FechaAlta, Categoría, Puesto, NSegSocial, IRPF,)

Se trata de una tabla con muchos atributos, lo cual ralentiza las consultas. Podemos dividir esa tabla en:

EMPLEADOS_ Personal (CodEmpleado, DNI, Apellido1, Apellido2, Nombre, Dirección, CP,)

EMPLEADOS_Profesional (CodEmpleado, Categoría, Puesto, NSegSocial, IRPF,)

EMPLEADOS_ Familiar (CodEmpleado, EstadoCivil, NumHijos, ..)

Ejemplo 2.

Tenemos una tabla de ALQUILERES de vehículos que tiene pocos atributos con información redundante.

ALQUILERES (Matricula, CodCliente, Fecha)

Esta tabla contiene múltiples registros con los datos que se recogen a diario, de forma que si queremos consultar los vehículos alquilados por un determinado cliente la velocidad de respuesta será elevada. Sería conveniente partir esta tabla en otras:

VEHICULO-CLIENTE (Matricula, CodCliente)

VEHICULO-FECHA (Matricula, Fecha)

FECHA-CLIENTE (Fecha, CodCliente)

7.2.- Desnormalización.

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

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

Coelho and YellowIcon
(GNU/GPL)

Los diseñadores de una base de datos deben conciliar 3 requerimientos a menudo incompatibles entre sí

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

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

La combinación de normalización y modelado Entidad/Relación produce un modelo relacional con estructuras de tabla apropiadas

Autoevaluación

Indica si la siguiente afirmación es verdadera o falsa

A veces es aconsejable **desnormalizar** parte de una base de datos con el objetivo de dar una respuesta rápida a las demandas del usuario.

- Verdadero Falso

Verdadero

Cada vez que avanzamos en el nivel de normalización se necesitan más uniones entre las entidades y eso ralentiza la respuesta del sistema, por lo que a veces se hace necesario desnormalizar parte de la base de datos, en busca de una mayor rapidez de respuesta a las demandas del usuario.

Condiciones y términos de uso de los materiales

Materiales desarrollados inicialmente por el Ministerio de Educación, Cultura y Deporte y actualizados por el profesorado de la Junta de Andalucía bajo licencia Creative Commons BY-NC-SA.

Antes de cualquier uso leer detenidamente el siguiente [Aviso legal](#)

Historial de actualizaciones

Versión: 03.00.00	Fecha de actualización: 25/06/22	Autoría: Isabel Cruz Granados
-------------------	----------------------------------	-------------------------------

Ubicación: Apartado 6

Mejora (tipo 1): Mejora (tipo 3): Los siguientes puntos:

- 1.- Se ha modificado el contenido del apartado 6.-Paso del diagrama Entidad-Relación al modelo relacional, sustituyendo su contenido por una introducción a este apartado que posteriormente se va a desglosar de forma más detallada en diferentes subapartados, pues solo había un resumen del proceso y un par de ejemplos.
- 2.- Nuevo apartado 6.1.- Transformación de entidades, con diferentes ejemplos de entidades fuertes y débiles (en existencia y en identificación).
- 3.- Nuevo apartado 6.2.- Transformación de atributos, con ejemplos resolviendo atributos compuestos, derivados y multivaluados
- 4.- Nuevo apartado 6.3.- Transformación de relaciones o inter-relaciones, con diferentes ejemplos aclaratorios de la resolución de relaciones N:M, 1:N, 1:1, reflexivas y ternarias.
- 5.- Nuevo apartado 6.4.- Transformación de jerarquías, detallando las posibilidades de transformación y poniendo un ejemplo aclaratorio.
- 6.- Nuevo apartado 6.5.- Ejemplos resueltos: Paso del modelo E/R al modelo Relacional, incluyendo ejemplos mediante tutoriales y/o videotutoriales, en los que se realiza paso a paso, la transformación, de ejemplos completos, del modelo ER y ERE al modelo Relacional, (el paso a tablas). Esto se ha realizado con los mismos ejemplos (VETERINARIA, PELICULAS, VIDEOTECA y CAMPEONATO) cuyo modelo ER o ERE se ha resuelto en el apartado 4.2 de esta misma unidad.

De esta forma se tiene una continuidad en las fases del diseño de una base de datos relacional.

Ubicación: Apartado 6.

Mejora (tipo 3): 1.-Incluir ejemplos simples y aclaratorios de la resolución del tipo de relaciones N:M, N:1 y 1:1.

2.-Incluir ejemplos aclaratorios de la transformación de una Jerarquía del modelo ERE.

3.-Incluir ejemplos mediante tutoriales y/o videotutoriales, en los que se vea cómo realizar paso a paso, la transformación, de ejemplos completos, del modelo ER y ERE al modelo Relacional, (el paso a tablas).

Esto se hará con los mismos ejemplos cuyo modelo ER o ERE se ha resuelto en el apartado 4.2 de esta misma unidad.

De esta forma se tiene una continuidad en las fases del diseño de una base de datos relacional.

Ubicación: ASIR_GBD_Unidad2

Mejora (Mapa conceptual): Actualización de Material unidad 2

Se han añadido los siguientes ítems y nodos:

- Bajo el nodo Modelo Entidad-Relacion se ha añadido la rama se muestran ejemplos resueltos con el nodo Bases de datos: VETERINARIA, PELICULAS, VIDEOTECA, CAMPEONATO.
- Bajo el nodo `Paso del modelo ER al modelo relacional en la rama según las reglas siguientes se ha modificado el nodo con la siguiente información - Transformación de entidades. Dependencias./Transformación de atributos /Transformación de relaciones (N:M, 1:N, 1:1, reflexivas, exclusivas)/Transfromación de Jerarquías
- Bajo el nodo `Paso del modelo ER al modelo relacional se ha añadido la rama se muestran ejemplos resueltos con el nodo Bases de datos: VETERINARIA, PELICULAS, VIDEOTECA, CAMPEONATO.

Ubicación: ASIR_GBD_Unidad2

Mejora (Orientaciones del alumnado): Actualización de Material unidad 2

En el índice o tabla de contenidos de la unidad se han añadido los siguientes subapartados al apartado

6.-Paso del diagrama Entidad-Relación al modelo relacional.

6.1.- Transformación de entidades.

6.2.- Transformación de atributos.

6.3.- Transformación de relaciones o inter-relaciones.

6.4.- Transformación de jerarquías.

6.5.- Ejemplos resueltos: Paso del modelo E/R al modelo Relacional.

Versión: 02.00.00	Fecha de actualización: 24/05/21	Autoría: Isabel Cruz Granados
-------------------	----------------------------------	-------------------------------

Ubicación: En el apartado 4.2

Mejora (tipo 3): En el apartado 4.2 se pueden incluir varios ejemplos mediante tutoriales o videotutoriales paso a paso, en los que sea visto cómo ir elaborando el diagrama E/R mediante el programa DIA que cumpla los requerimientos de enunciados concretos. Se pueden incluir también en un nuevo apartado 4.3.

En estos ejemplos se verán diferentes casuísticas de forma práctica y realista.

hasta su representación con el programa DIA y su posterior paso a tablas o modelo relacional.

Ubicación: Ningún cambio

Mejora (Mapa conceptual): Ningún cambio

Ubicación: Ningún cambio

Mejora (Orientaciones del alumnado): Ningún cambio

Versión: 01.00.00	Fecha de actualización: 23/07/20
-------------------	----------------------------------

Versión inicial de los materiales.

