

Diseño Conceptual de Bases de Datos

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

Modelos de datos.

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.

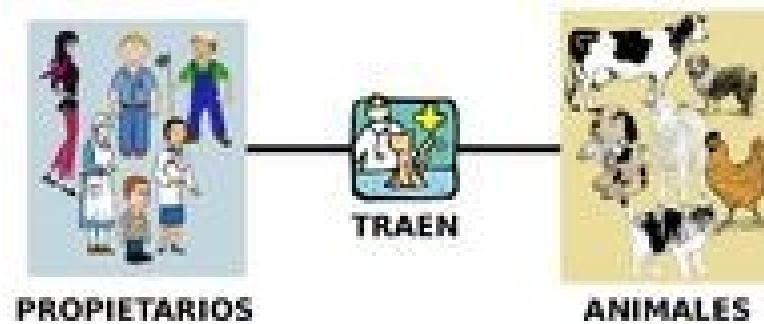
Aunque el modelo relacional supuso una considerable mejora sobre los modelos anteriores carecía de las características necesarias para ser una herramienta eficaz para diseñar bases de datos. Es más fácil examinar las estructuras gráficamente que describirlas como texto, por tanto se buscaba un modelo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

El modelo entidad/relación (E/R) es la herramienta gráfica más aceptada y que mejor se adapta a las bases de datos relacionales.

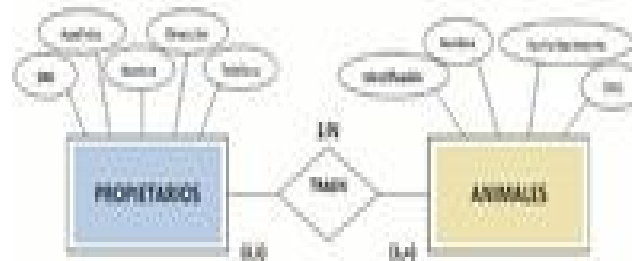
Este modelo no tiene en cuenta la implementación física de los datos, solo interesa el nivel conceptual, en cuanto a su aplicación práctica; el diseño de la base de datos se hace empleando este modelo y para implementarlo en un SGBD se traduce al modelo relacional como modelo de diseño a nivel lógico.

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

Relación veterinarios



Modelo ER - veterinarios



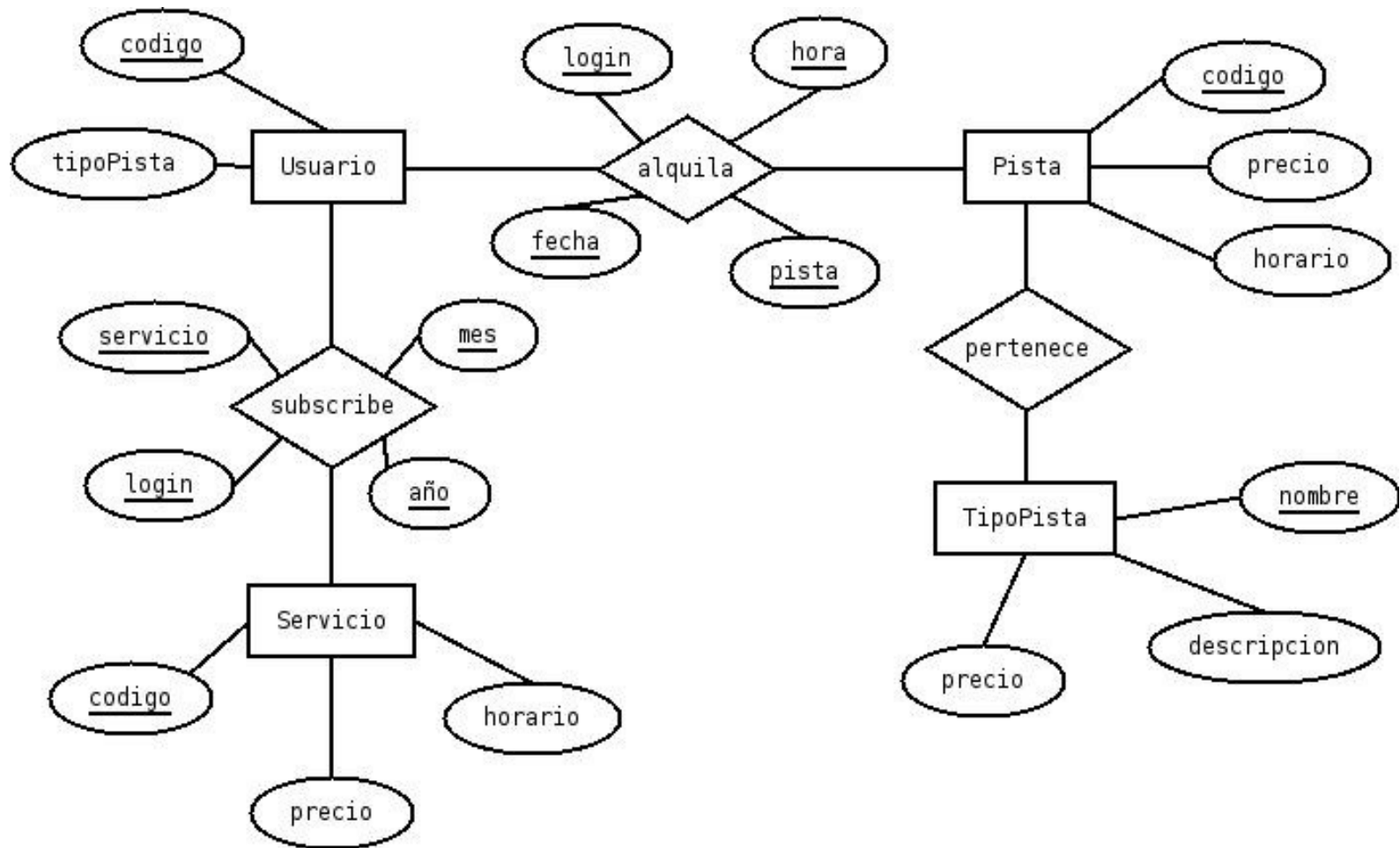
Modelo Relacional Veterinarios



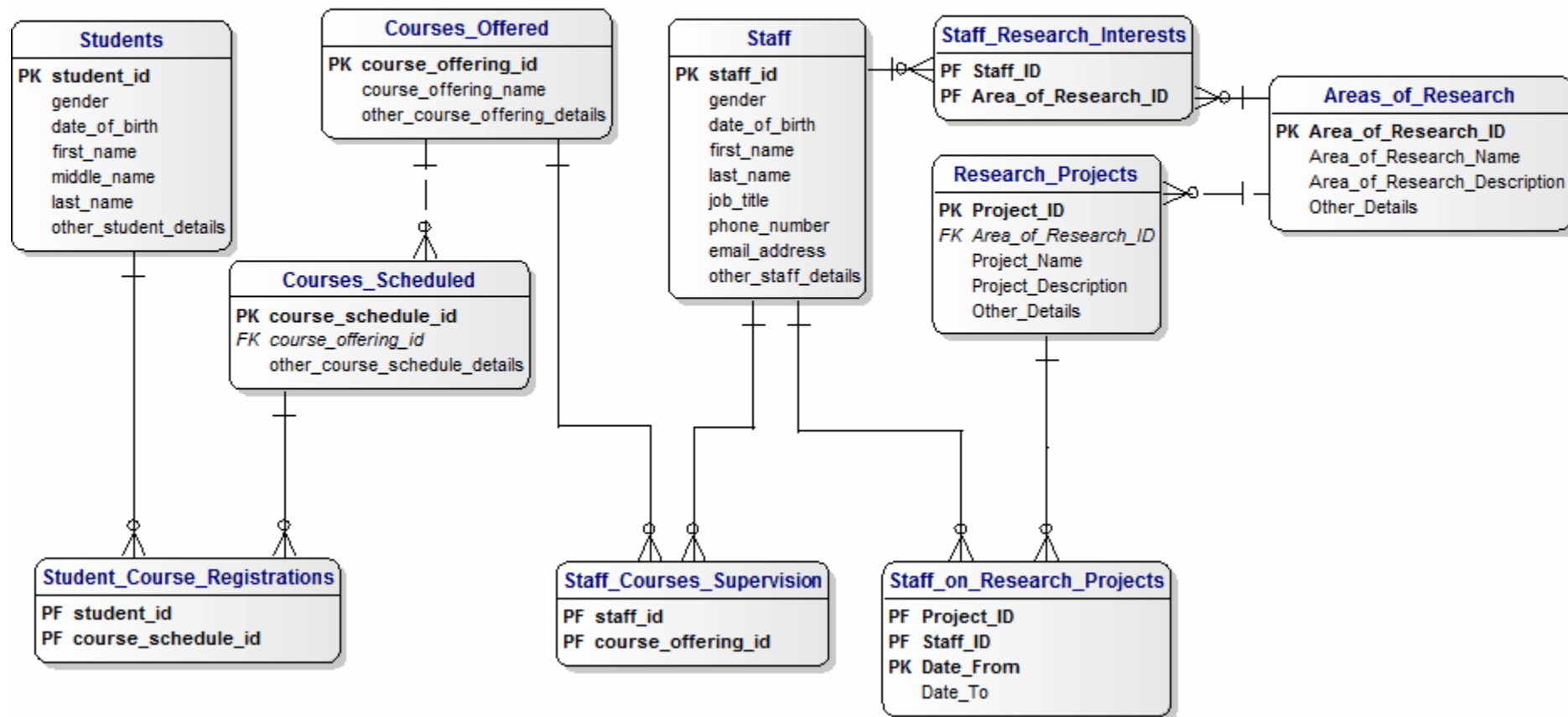
El **modelo Entidad-Relación (E/R)** lo introdujo Peter Chen en 1976 y produjo una representación gráfica de entidades y sus relaciones en una herramienta de modelado de datos a nivel conceptual.

Es un modelo muy extendido y potente para la representación de los datos. Se simboliza haciendo uso de grafos y de tablas.

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



Data Model for the UK's Open University
Barry Williams
May 5th.2012
DatabaseAnswers.org



Las **características** del modelo E/R son:

- Abstracción de alto nivel respecto del mundo real, creando unos elementos (entidades) que representan la realidad.
- Los registros, conjunto de datos que contiene cada entidad, su nivel de abstracción y de detalle son independientes de su uso posterior, de las limitaciones de almacenamiento y de la velocidad de proceso y del sistema en el que se vaya a implementar la base de datos.
- Esa independencia física del soporte de almacenamiento permite que el número de entidades pueda crecer y modificarse.

Las **exigencias o restricciones** de este modelo son:

- La existencia de la clave primaria.
- La obligatoriedad de que las entidades estén asociadas mediante una relación y la imposibilidad de asociar dos relaciones entre sí.

Ventajas:

- Simplicidad conceptual: los diseños de bases de datos complejas se crean y se manejan con mucha más facilidad que con cualquier otro sistema.
- Representación visual: permite a los diseñadores, programadores y usuarios finales una representación de los datos y sus relaciones fácil de entender.
- Herramientas de comunicación efectivas: permite que el diseñador integre las visualizaciones de los datos que tienen los distintos usuarios.
- Está muy bien integrado con el modelo relacional.

Desventajas:

- No puede representar algunas restricciones. Por ejemplo puede reflejar fácilmente que un profesor puede impartir entre 1 y 4 clases, pero no puede recoger que un profesor no pueda impartir más de 3 clases seguidas. Estas restricciones las tiene que manejar la aplicación.
- La representación de relaciones es limitada, recoge las relaciones entre las entidades pero no recoge, por ejemplo, las relaciones que puedan existir entre los atributos.
- En el modelo ER no hay comandos de manipulación de datos.
- Cuando se representan todos los atributos el modelo se congestiona, por tanto los diseñadores tienden a simplificarlo y con ello se pierde información.

El modelo E/R es un modelo que utiliza entidades y relaciones para representar la realidad de una forma gráfica, independiente del almacenamiento y del sistema, y que se integra fácilmente con el modelo relacional.

Entidades.

Es una persona, lugar o cosa sobre la cual se tienen que reunir y guardar datos. Una entidad se representa por medio de un rectángulo con el nombre dentro del recuadro. Suele tomarse como base de partida la lista de sustantivos de la descripción del problema. Por ejemplo la entidad CLIENTE.



Lista de categorías conceptos

Categoría del concepto	Ejemplo
Objetos físicos tangibles	Camión
Especificaciones o descripciones de cosas	Descripción del producto
Lugares	Tienda, Almacén, Delegación
Transacciones	Venta, Pago, Reserva
Línea o elemento de una transacción	Línea de una Venta
Papeles de las personas	Vendedor, Camionero

Lista de categorías conceptos

Categoría del concepto	Ejemplo
Contenedores de cosas	Tienda, Almacén
Cosas dentro del contenedor	Producto
Otros sistemas software externos	Sistema de autorización de tarjeta de crédito
Conceptos abstractos	Hambre
Organizaciones	Dpto de Ventas
Eventos	Pago, Anulación

Lista de categorías conceptos

Categoría del concepto	Ejemplo
Procesos	Venta de un producto
Reglas y políticas	Política de reembolso por anulación
Catálogos	Catálogo de productos
Documentos, libros	Manual de Personal, Ticket de compra
Instrumentos y servicios financieros	Existencias, Línea de crédito

Existen dos clases de entidades:

•**Fuertes**: son aquéllas que tienen existencia por sí mismas.

Por ejemplo: La entidad CLIENTE no depende de otras entidades para existir.

•**Débiles**: son aquéllas cuya existencia depende de otra u otros tipos 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:



Relación PROVEEDORES suministran ARTICULOS

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

La relación ALUMNOS se matriculan en ASIGNATURAS

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

Relación ALUMNOS obtienen CALIFICACIONES

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

Cada uno de los elementos que incluye la entidad se denomina ocurrencia de entidad o instancia de entidad.

Un conjunto de entidades semejantes (con la misma estructura de información) 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.

Atributos.

Cada entidad se describe como un conjunto de atributos. Un atributo describe una característica en particular de la entidad, es decir cada una de las propiedades que posee la entidad de la que se desea guardar información.

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

Los atributos se representan mediante una elipse horizontal con el nombre en su interior, unidos por una línea a la entidad a la que pertenecen. (En algunos casos se representa también el dominio: conjunto de datos posibles.)



Atributos compuestos y simples: Un atributo compuesto es un atributo que se puede subdividir en otros. Un atributo simple no se puede subdividir.

Un **ejemplo** claro es el atributo dirección que puede dividirse en: calle, número, localidad, provincia y código postal.

La forma de representar estos atributos varía en función de los autores y en muchos casos no se representan en el modelo E/R. Un ejemplo de representación gráfica sería:



Atributos de un solo valor y atributos de valores múltiples o multivaluados:

1. Atributos de un solo valor son los que pueden tener un único valor,

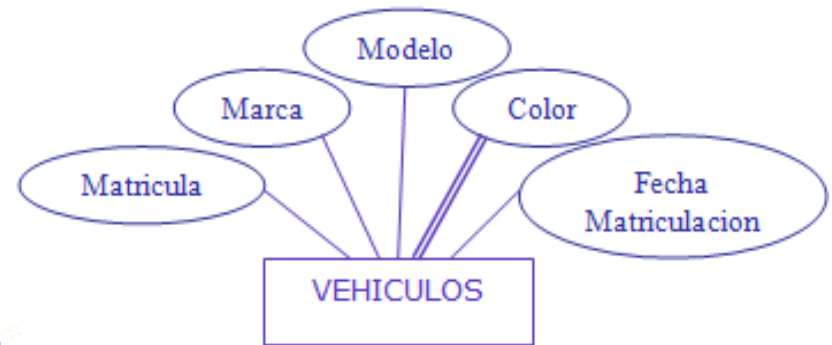
Por ejemplo: El DNI, el número de la seguridad social, etc. Eso no significa que sea un atributo simple, por ejemplo el número de habitación de un hotel se puede dividir: el primer dígito es la planta, los dos siguientes corresponden a la habitación dentro de la planta.

2. Atributos de valores múltiples son aquellos que pueden tener muchos valores.

Por ejemplo: El número de teléfono de un cliente (puede tener uno o más números 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:

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



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 de otro/s atributos/s mediante un algoritmo. 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.

Dominio de un atributo: Es el conjunto de los posibles valores que ese atributo puede poseer.

Por ejemplo: El dominio del atributo Luis se toma del dominio Nombre por ser un uno de los nombres posibles para los clientes. El dominio para el atributo nota media de un estudiante sería (0,10) porque el valor más bajo posible es el 0 y el mas alto el 10. El dominio del atributo teléfono estaría dentro del conjunto de combinaciones de 9 cifras que sean números de teléfonos válidos. El dominio del atributo sexo de un cliente sólo tiene dos valores posibles M o F, etc.

A menudo se confunden los dominios con los tipos de datos, aunque no son lo mismo. Tipo de datos es un concepto físico, mientras que dominio es un concepto lógico.



Claves.

Se denomina clave o clave primaria al conjunto de atributos que identifican de forma única a cada ejemplo de entidad.

Al menos hay una clave, la formada por todos los atributos.

Por ejemplo: En la entidad CLIENTES podríamos encontrar las siguientes claves: el código de cliente, el conjunto apellidos + nombre, el DNI, el nombre, apellidos y la dirección, etc.

Clave candidata: Es cada una de las claves que está formada por el mínimo número de campos posibles.

En el **ejemplo** anterior serían el DNI y el Código de cliente.

Clave primaria o principal: (primary key): Es la clave candidata seleccionada por el diseñador de la base de datos. Tiene que cumplir:

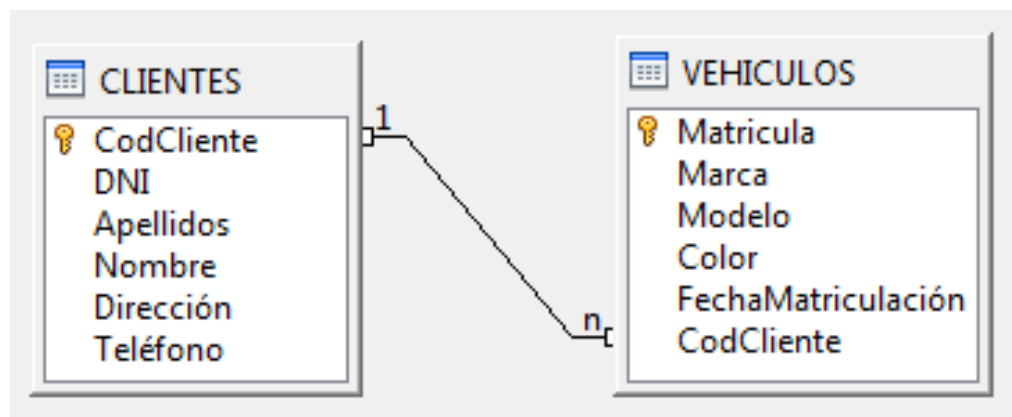
- No puede contener valores nulos.
- Ha de ser sencilla de crear.
- No puede cambiar con el tiempo.

El atributo o conjunto de atributos que formen esta clave se representan subrayados. Lo ideal es que la clave primaria esté formada por un solo atributo. A cada una de las claves candidatas no seleccionadas se le denomina clave alternativa.

En el **ejemplo** anterior podría ser el Código de cliente.

Clave ajena foránea (foreign key): Es el atributo o conjunto de atributos de una entidad que forman la clave primaria en otra entidad. Se utilizan para representar las relaciones entre las tablas. Es indiferente que los atributos que formen la clave ajena tengan distinto nombre, lo que interesa es que su contenido pertenezca al mismo dominio que la clave principal, aunque es preferible que tengan el mismo nombre para facilitar el trabajo.

Por ejemplo: La entidad CLIENTES con los atributos que habíamos expuesto: CodCliente (primary key), DNI, Apellidos, Nombre, Dirección y Teléfono; podría relacionarse con la entidad VEHÍCULOS cuyos atributos podrían ser: Matrícula (primary key), Marca, Modelo, Color y FechaMatriculación. El CodCliente de la entidad VEHÍCULOS sería una clave ajena (foreign key), porque es la clave principal en la entidad CLIENTES.



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....

Algunas pueden inventarse por el propio SGBD (identity, autonuméricos, UniqueIdentifier, autoincrement,...)

Por ejemplo: Queremos recoger los datos de nuestra agenda de teléfonos: 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 denominadaCodigoAgenda que podría ser un número consecutivo.

Ejemplo

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.
- 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

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

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

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

Relaciones.

Una relación es la asociación de dos o más entidades.

Generalmente se las identifica con un verbo (activo o pasivo) y se representan mediante un rombo. 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

Entidad asociada: Normalmente las relaciones no tienen atributos, pero puede ocurrir que los tenga. 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.



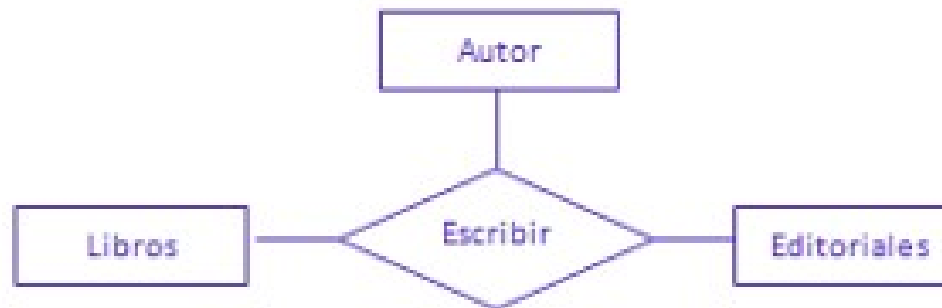
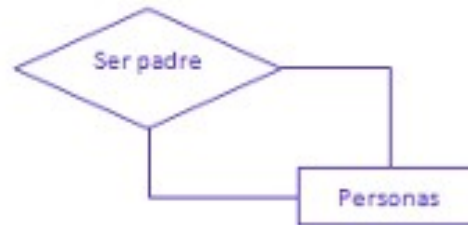
Estas entidades darán lugar a una tabla que contenga esos atributos cuando se conviertan al modelo relacional.

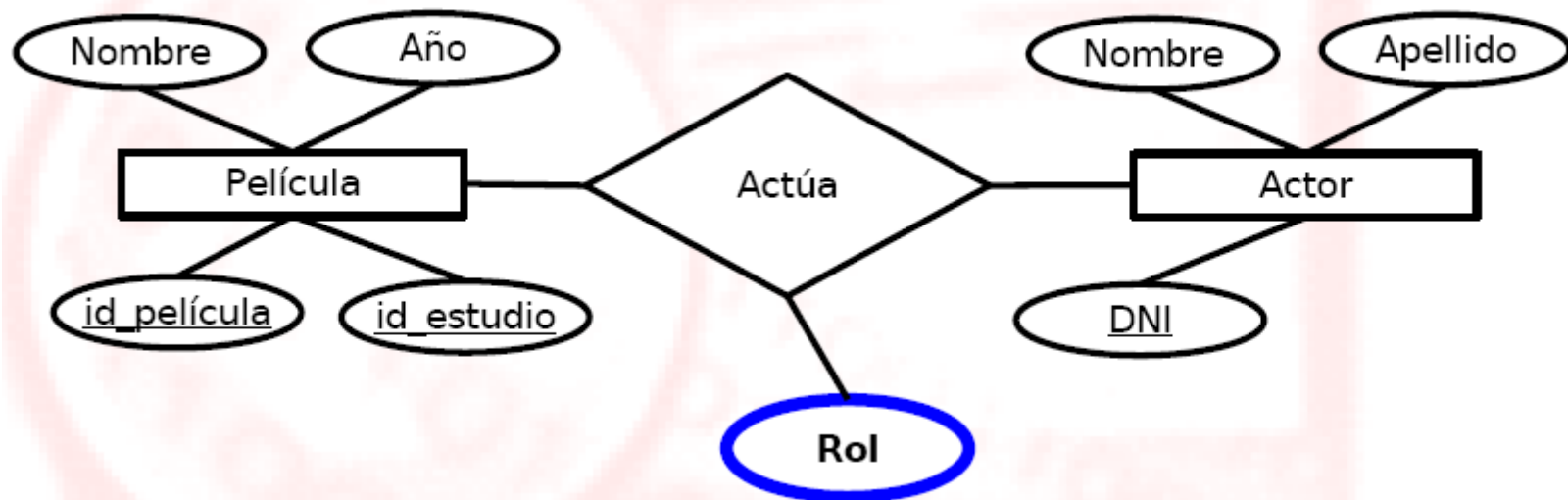
Grado de una relación: Se define como el número de entidades que participan en una relación.

- Cuando solo participa una entidad se denominan de grado uno o relaciones reflexivas.
- Cuando participan dos entidades en una relación se denominan binarias o de grado dos.
- Si participan 3 entidades se denominan de grado 3 o ternarias.
- Cuando participan más de 3 entidades se denominan n-arias.

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

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







Correspondencia y cardinalidad.

En el modelo E-R se representan ciertas restricciones a las que deben ajustarse los datos contenidos en una base de datos.

Tipo de correspondencia: expresa el número de entidades a las que puede asociarse otra entidad mediante una relación. Los tipos de correspondencia para relaciones binarias son:

- Relaciones **1:1**
- Relaciones **1:N**
- Relaciones **M:N**

A esta clasificación se la denomina también **conectividad**.

Relaciones 1:1 (uno a uno): a cada elemento de la primera entidad le corresponde sólo uno de la segunda entidad, y a la inversa.

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



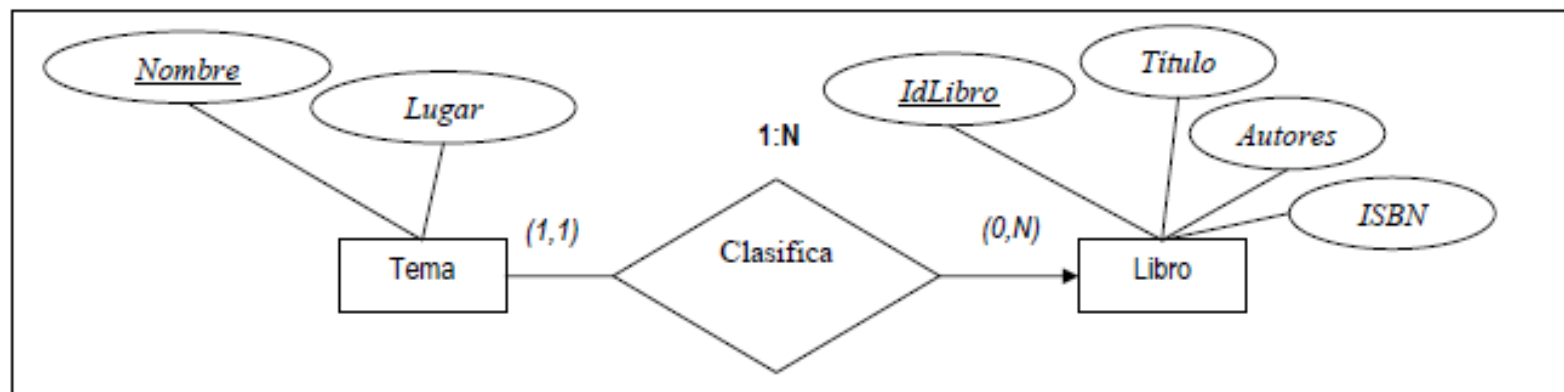




Relaciones 1: N (uno a muchos): a cada elemento de la primera entidad le corresponde uno o más elementos de la segunda entidad, y a cada elemento de la segunda entidad le corresponde uno sólo de la primera entidad.

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

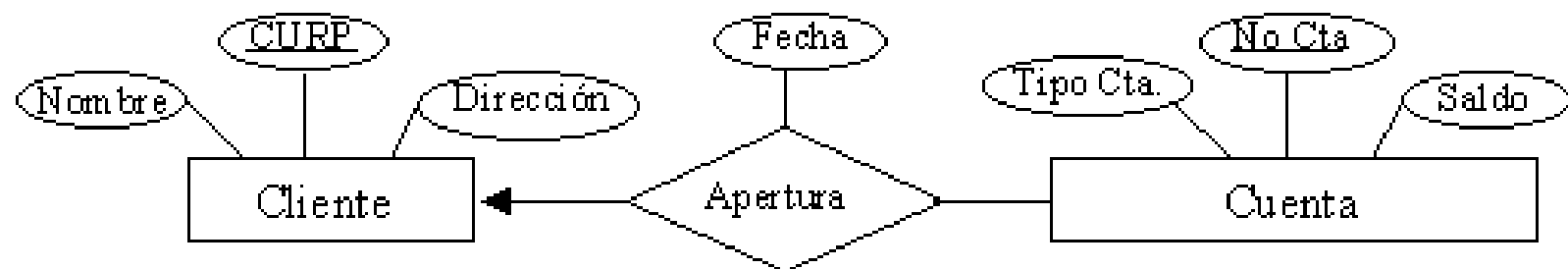




Relaciones N: M (muchos a muchos): a cada elemento de la primera entidad le corresponde uno o más elementos de la segunda entidad, y a cada elemento de la segunda entidad le corresponden uno o más elementos de la primera.

Por ejemplo: Cada mecánico puede intervenir varias reparaciones y una misma reparación la llevan a cabo varios mecánicos.





Cardinalidad de una relación: Expresa el número de ocurrencias de una entidad asociadas con una ocurrencia de la entidad relacionada. Se coloca al lado de las entidades y se representa con el formato:

(1,n) : el primer valor representa el valor mínimo y el segundo valor representa el máximo

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.
- Cada vehículo es traído al taller por un cliente como mínimo y como máximo.



Las cardinalidades son muy útiles a nivel de aplicación.

Participación:

El concepto de participación está relacionado con las cardinalidades mínimas.

- 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 definir el grado de participación se utilizan también los términos **opcional u obligatoria**. 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**.

Por ejemplo: En la relación anterior 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 CLIENTES en la relación es total. En el caso de que pudiéramos tener registrado algún cliente que no hubiera traído ningún coche al taller (clientes potenciales) la participación serial **parcial**.

Participación total u obligatoria: cardinalidad mínima 1

Participación parcial u opcional: cardinalidad mínima 0

Relaciones fuertes y débiles (identificadoras o no identificadoras):

En el punto anterior 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 estas afecta también a las relaciones.

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 dependa 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, como en el caso de las entidades cliente y persona

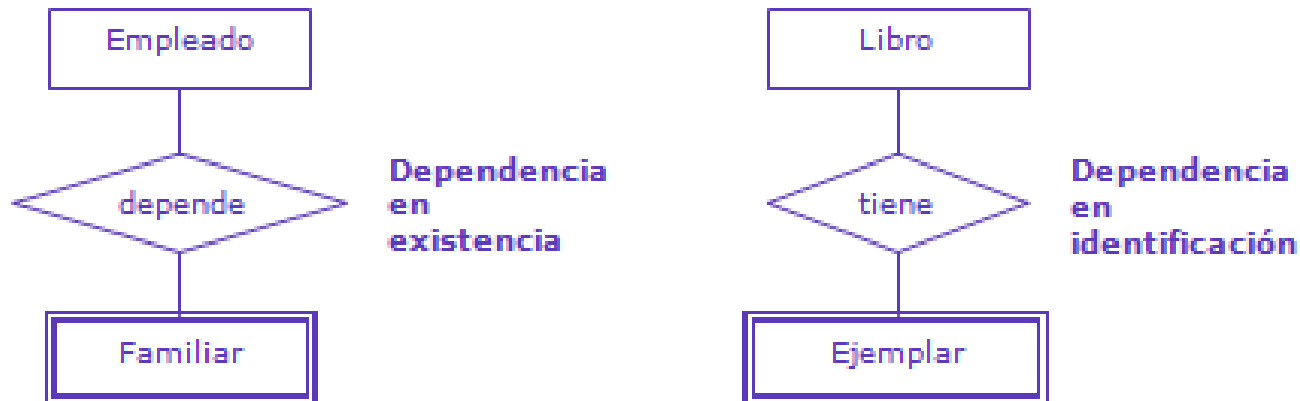
La dependencia entre una entidad débil y otra fuerte puede ser de dos tipos:

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

Cuando un empleado de nuestra 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.

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



Relaciones débiles (relación no identificativa):

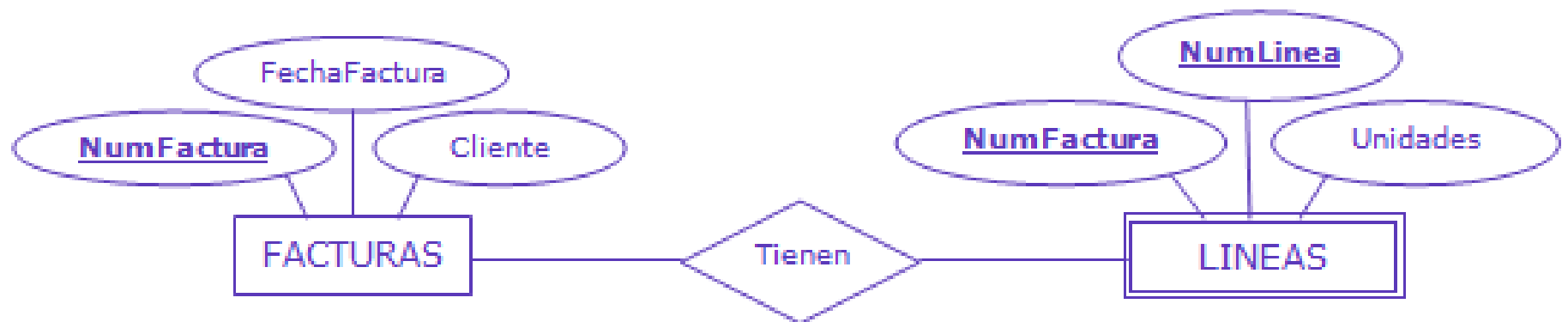
Si una entidad es independiente de la existencia de otra entidad, la relación entre ellas se describe como una relación débil o como una relación no identificativa.

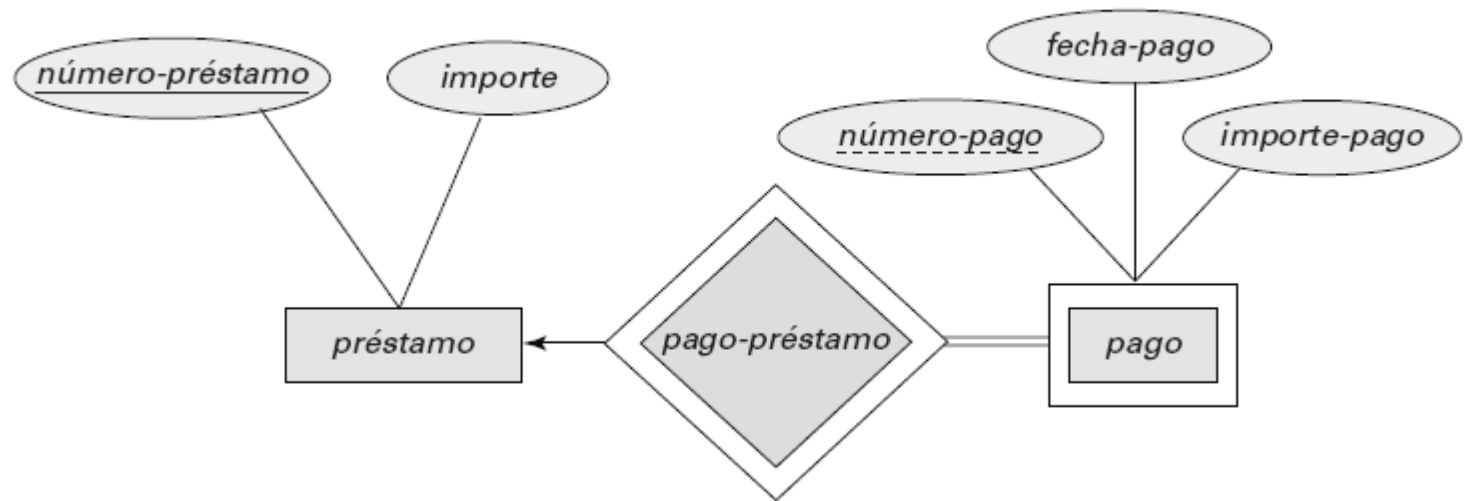
Una relación débil existe cuando la clave primaria de la entidad relacionada (hijo) no contiene la clave primaria de la entidad padre.

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.

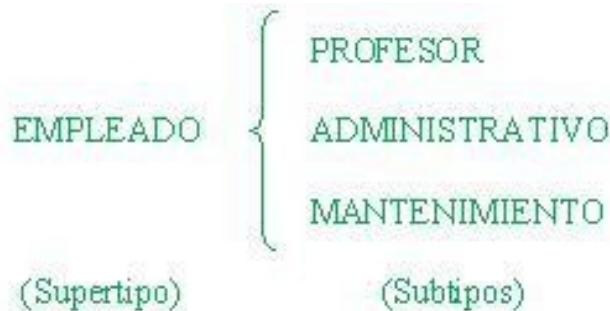




Modelo Entidad-Relación ampliado.

Para presentar los siguientes conceptos 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.**

Originalmente, el modelo entidad-relación sólo incluía los conceptos de entidad, relación, dominio y atributo. Más tarde, se añadieron otros conceptos, como los atributos compuestos y las jerarquías de generalización, en lo que se ha denominado modelo entidad-relación extendido, o ampliado.

Algunos aspectos correspondientes al modelo ampliado se han visto en los apartados anteriores vamos a referirnos pues a las jerarquías de generalización

Generalización y jerarquías de generalización.

La generalización es una técnica de abstracción que permite 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**.

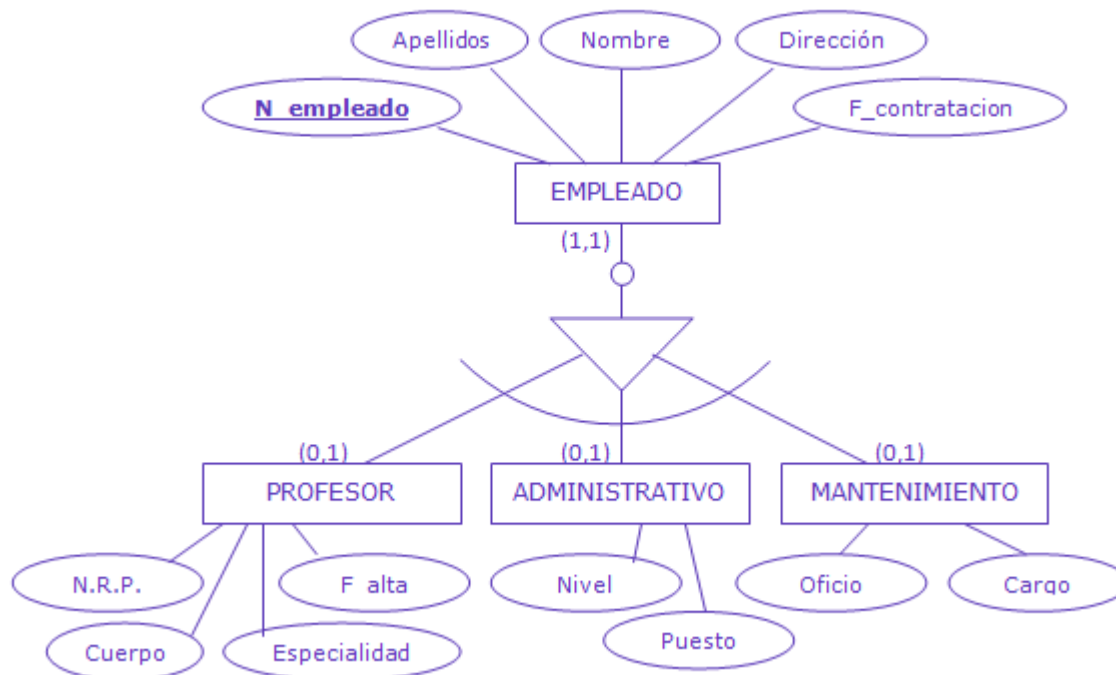
Una de las características más importantes de las jerarquías es la **herencia** por la que los subtipos heredan los atributos del **supertipo**.

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

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



Tipos de jerarquías.

La generalización puede ser:

- Total:** cuando todas las ocurrencias del supertipo pertenecen a alguno de los subtipos.

En nuestro ejemplo es total si todos los empleados del centro son o profesores o administrativos o de mantenimiento. Todos los tipos de empleado están incluidos en la clasificación.

- Parcial:** Cuando puede haber ocurrencias en el supertipo que no pertenezcan a ninguno de los subtipos.

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

- Exclusiva:** 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.

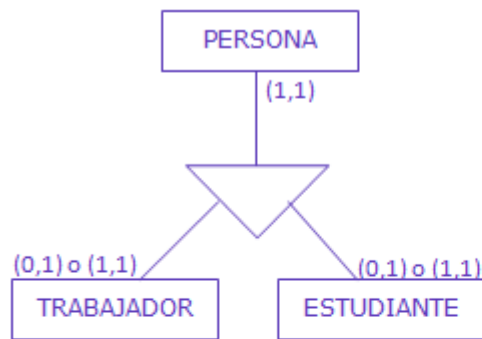
- Solapada:** Cuando una ocurrencia del supertipo puede estar a la vez en varios subtipos.

Por ejemplo: Si un empleado puede ser a la vez administrativo y de mantenimiento.

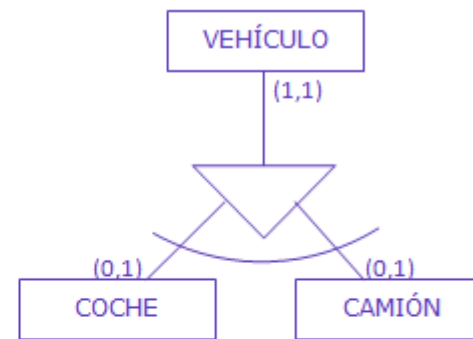
Las cardinalidades de las jerarquías son:

- **(1,1)** en el supertipo
- **(0,1)** en los subtipos, para las exclusivas
- **(1,1) o (0,1)** en los subtipos, para las solapadas

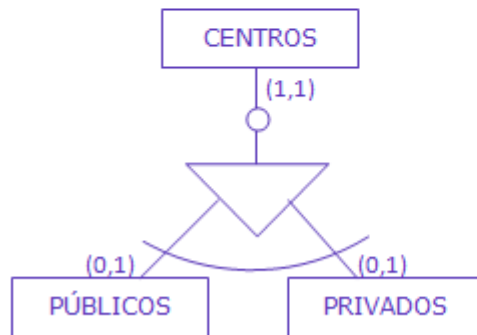
Ejemplos de representación gráfica de los distintos tipos de jerarquía



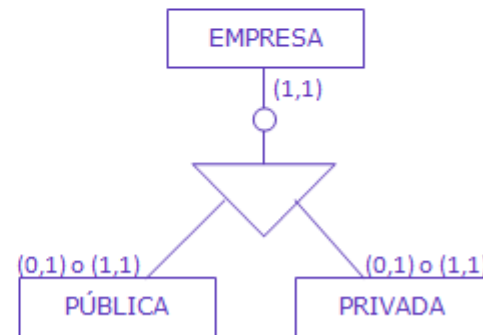
Parcial con solapamiento



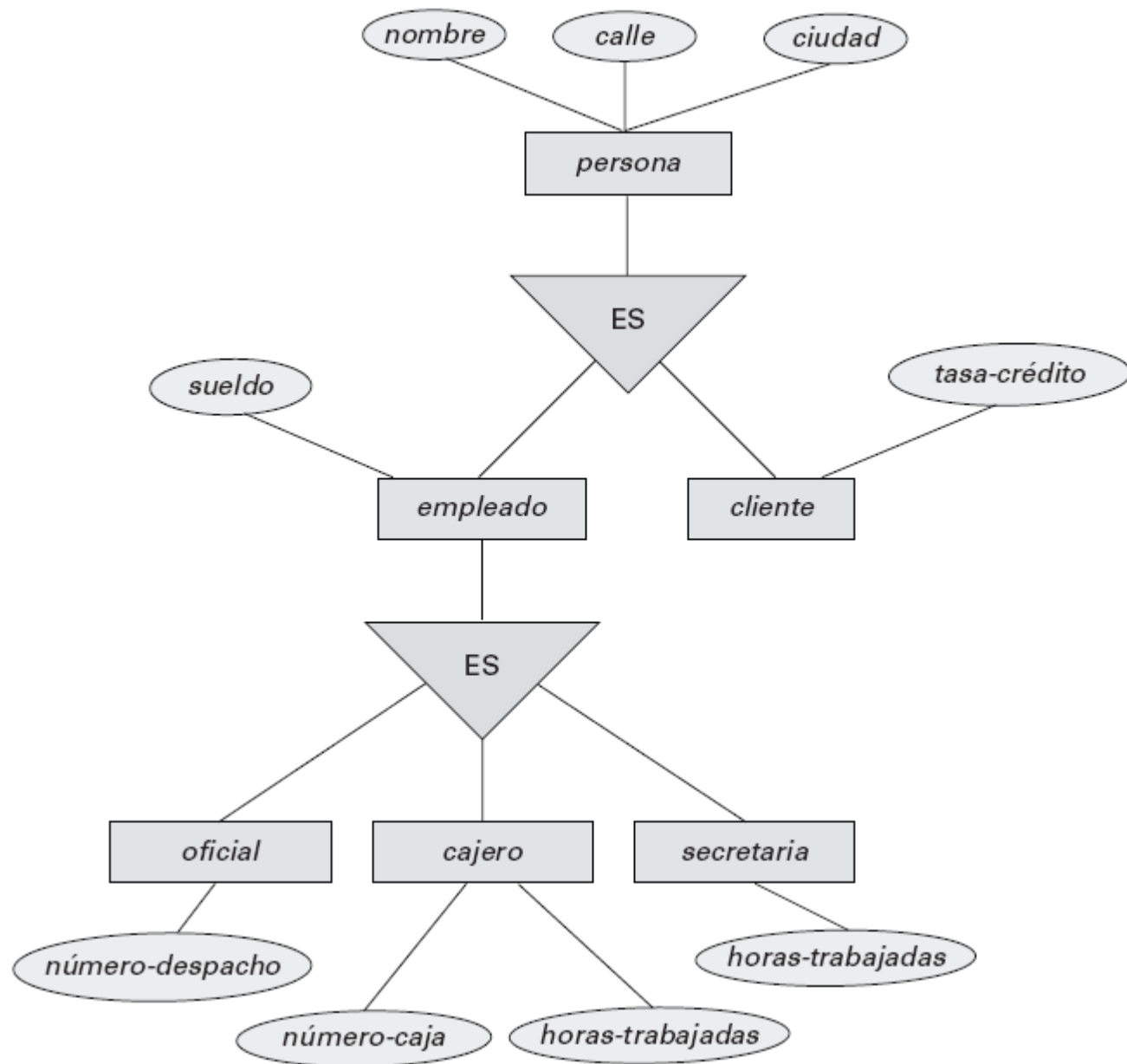
Parcial sin solapamiento



Total sin solapamiento



Total con solapamiento



Representación gráfica del modelo Entidad-Relación.

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.



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
- Diagrama de pata de gallo
- Rein85
- IDEF1X

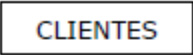
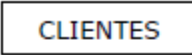
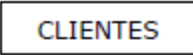
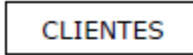
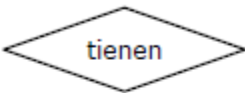



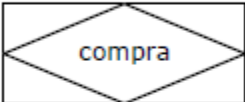
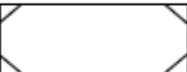


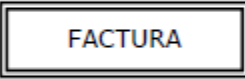




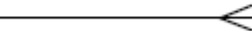








El modelo de Chen, tal y como lo hemos estudiado en este tema, está basado en los trabajos de Chen (modelo básico) y en las mejoras incorporadas por Teorey, Yang y Fry (modelo ampliado). Ha sido el modelo en que se han basados las herramientas CASE dominante desde finales de los 80. Aunque actualmente 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 IDEF1X 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.

Comparación de símbolos utilizados en el modelo E/R

	CHEN	PATA DE GALLO	IDEF1X	REIN85
ENTIDAD				
RELACION				
ENTIDAD ASOCIADA				
ENTIDAD DEBIL				
RELACION A 1	1			
RELACION A MUCHOS	N			
CARDINALIDAD MINIMA 1 (obligatoria)	(1,1) (1,n)	 		
CARDINALIDAD MINIMA 0 (opcional)	(0,1) (0,n)	 		

Pasos para diseñar un modelo entidad-relación

Dada una situación del mundo real para la cual queremos establecer un modelo Entidad/Relación que se ajuste a los requerimientos del problema, deberemos seguir los siguientes pasos:

1. Se parte de una descripción textual del problema o sistema de información a automatizar (los requisitos).
2. Se hace una lista de los sustantivos y verbos que aparecen.
3. Los sustantivos son posibles entidades o atributos.
4. Los verbos son posibles relaciones.
5. Analizando las frases se determina la cardinalidad de las relaciones y otros detalles.
6. Se elabora el diagrama (o diagramas) entidad-relación.
7. Se completa el modelo con listas de atributos y una descripción de otras restricciones que no se pueden reflejar en el diagrama.